



دانشگاه صنعتی شریف
دانشکده مهندسی برق

رساله‌ی دکتری
گرایش مخابرات سیستم

بازیابی تنک و یادگیری دیکشنری بر مبنای روش‌های پراکسیمال در بهینه‌سازی

نگارنده

مصطفی صادقی

استاد راهنما

دکتر مسعود بابائی‌زاده

اردیبهشت‌ماه ۱۳۹۷

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

توجه

این پروژه براساس قرارداد شماره ی ۹۶۰۰۰۷۸۰ از حمایت مالی صندوق حمایت از پژوهشگران و فناوران کشور برخوردار شده است.

بسمه تعالی

دانشگاه صنعتی شریف
دانشکده مهندسی برق

رساله‌ی دکتری

عنوان: بازیابی تَنک و یادگیری دیکشنری بر مبنای روش‌های پراکسیمال در بهینه‌سازی

نگارش: «مصطفی صادقی»

اعضاء هیأت داوران:

امضاء:..... دکتر مسعود بابائی‌زاده

امضاء:..... دکتر فرخ مروستی

امضاء:..... دکتر محمدباقر شمس‌الهی

امضاء:..... دکتر حمید شیخ‌زاده

امضاء:..... دکتر حمیدرضا امین‌داور

تاریخ: ۲۴ اردیبهشت‌ماه ۱۳۹۷.

قدردانی

در ابتدا خداوند متعال را شاکرم که به من قدرت پیمودن مسیر کسب علم و دانش را عطا فرمود.

از استاد گرانقدرم، آقای دکتر بابائی زاده، که طی دوران تحصیلات کارشناسی ارشد و دکتری افتخار شاگردی ایشان را داشتم و در این مدت همواره مرا مورد حمایت و تشویق خود قرار داده‌اند، کمال قدردانی را دارم. همچنین، در طول این مدت از راهنمایی‌های ارزشمند و دلسوزانه‌ی پروفسور Christian Jutten بهره برده‌ام که در اینجا بطور ویژه از ایشان تشکر می‌کنم.

از اساتید محترم، آقایان دکتر امین‌داور، دکتر شمس‌الهی، دکتر شیخ‌زاده و دکتر مروستی که زحمت داوری این رساله را بر عهده گرفته و با نظرات سازنده‌ی خود موجب بهبود کیفیت این رساله گردیدند صمیمانه سپاسگزارم. در طول دوره‌ی فرصت مطالعاتی خود در دانشگاه KTH سوئد، افتخار همکاری و استفاده از تجربیات و راهنمایی‌های دکتر Saikat Chatterjee و پروفسور Mikael Skoglund را داشتم که بدین وسیله از زحمات ایشان قدردانی می‌نمایم.

از دوستان عزیزم در دانشگاه صنعتی شریف و دانشگاه KTH، آقایان دکتر محمدرضا ملک‌محمدی، سجاد امینی، محسن جنیدی، مهدی بلورساز مشهدی، جواد پارسا، حسام عراقی و علیرضا مهدوی جاوید، که به شیوه‌های مختلف بنده را یاری رساندند، سپاسگزارم.

در انتها، از همسر عزیزم، فاطمه، که بی‌شک با نظرات، کمک‌ها و دلسوزی‌های بی‌دریغ خود باعث انگیزه‌ی روزافزون من شده و نقش مهمی در پیشرفت‌های علمی من در طول دوره‌ی دکتری داشت، بطور خاص قدردانی می‌کنم.

تقدیم به ...

همسر عزیز و مهربانم، فاطمه؛

که حمایت، تشویق و فداکاری بیش در طول انجام این رساله باعث دلگرمی و انگیزه‌ی روزافزون من شده، و هر لحظه در کنارم بوده است.

چکیده:

نمایش تُنک طی دهه‌ی اخیر مورد توجه گسترده‌ای قرار گرفته است. ایده‌ی اصلی نمایش تُنک این است که سیگنال‌های طبیعی، برخلاف بُعد ظاهری بالای آنها، محتوای اطلاعاتی به مراتب کم‌تری دارند و بنابراین، برحسب تعداد کمی سیگنال پایه (که اتم نامیده می‌شوند) قابل نمایش هستند. در اینجا دو سوال اساسی مطرح می‌شود. اول این که مجموعه‌ی سیگنال‌های پایه را چگونه طراحی کنیم و سوال دوم این که با فرض در دسترس بودن سیگنال‌های پایه، اتم‌های مناسب برای یک سیگنال داده شده را چگونه از آن بین انتخاب کنیم. جواب مناسب برای سوال اول این است که مجموعه‌ی اتم‌ها با استفاده از یک دسته داده‌ی آموزشی یاد گرفته شود. این کار، یادگیری دیکشنری نام دارد که برای آن روش‌های مختلفی تاکنون پیشنهاد شده است. برای بدست آوردن نمایش تُنک، با یک دیکشنری داده شده، نیز الگوریتم‌های متنوعی ارائه شده است. با این حال، امروزه با توجه به ابعاد بالای داده‌ها، نیاز به الگوریتم‌های کارآمدتری احساس می‌شود. همچنین، کمینه‌سازی توابع ناهموار و نامحدب تشویق‌کننده‌ی تُنکی و نیز تابع همبستگی متقابل برای یادگیری دیکشنری‌های با میزان شباهت کم بین اتم‌ها، به دلیل چالش‌های زیاد، کمتر مورد توجه قرار گرفته است. بعلاوه، در حوزه‌ی یادگیری توزیع‌شده‌ی دیکشنری با استفاده از تعدادی پردازنده‌ی در حال تعامل با یکدیگر، الگوریتم‌های موجود در مورد پردازنده‌های با محدودیت قدرت پردازشی و حجم انتقال داده کارآمد نیستند.

در این رساله، الگوریتم‌هایی جدید برای بازیابی نمایش تُنک و یادگیری دیکشنری ارائه می‌شود. ویژگی اصلی الگوریتم‌های پیشنهادی، ساختار ساده ولی در عین حال عملکرد بالای آنها است. این الگوریتم‌ها عمدتاً مبتنی بر دسته‌ی مهمی از الگوریتم‌های بهینه‌سازی، به نام الگوریتم‌های پراکسیمال (proximal)، هستند. الگوریتم‌های پراکسیمال روش‌هایی مرتبه‌ی اول برای حل مجموعه‌ی وسیعی از مسائل بهینه‌سازی موجود در پردازش سیگنال و یادگیری ماشین هستند. بطور خاص، با الهام از الگوریتم نُرم صفر هموار شده (SLO)، الگوریتم‌های جدیدی برای بازیابی نمایش تُنک ارائه می‌شوند که نسبت به الگوریتم SLO سرعت همگرایی بیشتری داشته و نسبت به نویز مقاوم‌تر هستند. اثبات همگرایی این الگوریتم‌ها نیز ارائه می‌شود. همچنین، الگوریتم‌هایی برای یادگیری دیکشنری‌های با همبستگی متقابل کم ارائه می‌شود که برخلاف روش‌های قبلی، بصورت مستقیم از تابع همبستگی متقابل در مسأله‌ی یادگیری دیکشنری استفاده کرده و نسبت به الگوریتم‌های موجود پیچیدگی محاسباتی کمتری دارند. بعلاوه، با کمینه‌سازی همبستگی متقابل با استفاده از روش‌های پراکسیمال، الگوریتم جدیدی برای طراحی فریم‌های ناهمبسته (incoherent frames)، که در مخابرات و حسگری فشرده کاربرد دارند، معرفی می‌شود. علاوه بر این‌ها، برای یادگیری دیکشنری از روی داده‌های با ابعاد بالا، الگوریتم جدیدی معرفی می‌شود که مبتنی بر کاهش بُعد داده‌ها و توزیع آنها بین چند پردازنده است. برای کاهش میزان اطلاعات مبادله شده بین پردازنده‌ها و مرکز پردازش، یک ساختار تُنک برای دیکشنری فرض می‌شود. شبیه‌سازی‌های انجام شده روی داده‌های مصنوعی و نیز تصاویر طبیعی حاکی از عملکرد امیدبخش روش‌های پیشنهادی است.

کلمات کلیدی:

- 1- نمایش تُنک Sparse Representation
- 2- یادگیری دیکشنری Dictionary Learning
- 3- الگوریتم‌های پراکسیمال Proximal Algorithms
- 4- داده‌های بُعد بالا High-dimensional Data

فهرست مطالب

۱	پیشگفتار	۱
۶	نمایش تُنک سیگنال	۲
۶	۱-۲ مقدمه	۲
۶	۲-۲ معرفی مسأله	۲
۸	۳-۲ یکتایی تُنک‌ترین جواب	۲
۱۰	۴-۲ الگوریتم‌های بازیابی نمایش تُنک	۲
۱۰	۱-۴-۲ روش‌های حریص	۲
۱۰	۱-۱-۴-۲ الگوریتم MP	۲
۱۱	۲-۱-۴-۲ الگوریتم OMP	۲
۱۲	۳-۱-۴-۲ الگوریتم CoSaMP	۲
۱۳	۲-۴-۲ روش‌های آستانه‌گذاری	۲
۱۳	۱-۲-۴-۲ الگوریتم IDE	۲
۱۴	۲-۲-۴-۲ الگوریتم‌های IST	۲
۱۵	۳-۲-۴-۲ الگوریتم IMAT	۲
۱۶	۳-۴-۲ روش‌های تقریب نُرم صفر	۲
۱۷	۵-۲ مقدمه‌ای بر حسگری فشرده	۲
۱۹	۶-۲ جمع‌بندی	۲
۲۰	۳ یادگیری دیکشنری و پردازش داده‌های بُعد بالا	۲

دو		
۲۰	۱-۳ مقدمه
۲۰	۲-۳ یادگیری دیکشنری
۲۴	۳-۳ الگوریتم‌ها
۲۵	۱-۳-۳ الگوریتم MOD
۲۵	۲-۳-۳ الگوریتم KSVD
۲۷	۴-۳ روش‌های پردازش داده‌های با ابعاد بالا
۲۸	۱-۴-۳ روش‌های مبتنی بر گرادیان تصادفی
		۲۹
۳۱	۳-۴-۳ روش‌های مبتنی بر Data Sketching
۳۴	۵-۳ جمع‌بندی
۳۵		۴ مروری بر الگوریتم‌های پراکسیمال برای کمینه‌سازی غیرمحدب
۳۵	۱-۴ مقدمه
۳۶	۲-۴ عملگر پراکسیمال
۳۹	۳-۴ کمینه‌سازی توابع ترکیبی (هموار+ ناهموار)
۴۱	۴-۴ کمینه‌سازی توابع ترکیبی (ناهموار+ ناهموار)
۴۲	۵-۴ تعمیم به بیش از یک بلوک
۴۳	۶-۴ مبانی اثبات همگرایی
۴۴	۱-۶-۴ تعاریف و لم‌های مورد نیاز
۴۵	۲-۶-۴ ویژگی‌های همگرایی پایه
۴۶	۳-۶-۴ همگرایی زبردنباله به نقطه‌ی بحرانی
۴۸	۴-۶-۴ همگرایی کلی به نقطه‌ی بحرانی
۵۰	۵-۶-۴ نرخ همگرایی
۵۱	۷-۴ جمع‌بندی
۵۲		۵ الگوریتم‌های پیشنهادی برای کدینگ تُنگ
۵۲	۱-۵ مقدمه

۵۳	الگوریتم تُنک‌سازی-تصویرسازی پیاپی (ISP)	۲-۵
۵۳	بازبینی الگوریتم نُرم صفر هموار شده (SL0)	۱-۲-۵
۵۷	الگوریتم پیشنهادی	۲-۲-۵
۶۳	آنالیز همگرایی	۳-۲-۵
۶۳	شبیه‌سازی	۴-۲-۵
۷۳	الگوریتم نُرم صفر-آستانه‌گذاری نُرم (L0Soft)	۳-۵
۷۳	فرمول‌بندی مسأله	۱-۳-۵
۷۷	شبیه‌سازی	۲-۳-۵
۸۰	الگوریتم پراکسیمال-تصویرسازی پیاپی (IPP)	۴-۵
۸۰	فرمول‌بندی مسأله	۱-۴-۵
۸۲	ارتباط با الگوریتم‌های دیگر	۲-۴-۵
۸۳	توابع ناهموار مختلف	۳-۴-۵
۸۴	آنالیز همگرایی	۴-۴-۵
۸۸	شبیه‌سازی	۵-۴-۵
۹۲	جمع‌بندی	۵-۵

۶ الگوریتم‌های پیشنهادی برای یادگیری دیکشنری و طراحی ماتریس حسگر

۹۴	مقدمه	۱-۶
۹۵	یادگیری دیکشنری برای داده‌های بُعد بالا	۲-۶
۹۵	الگوریتم پیشنهادی	۱-۲-۶
۹۹	نتایج شبیه‌سازی	۲-۲-۶
۱۰۰	یادگیری دیکشنری‌های با همبستگی متقابل کم	۳-۶
۱۰۳	الگوریتم‌های پیشنهادی	۴-۶
۱۰۵	حالت نامقید	۱-۴-۶
۱۰۸	حالت مقید	۲-۴-۶
۱۱۴	شبیه‌سازی	۳-۴-۶
۱۱۷	طراحی ماتریس حسگر	۵-۶
۱۱۹	الگوریتم پیشنهادی	۱-۵-۶
۱۲۲	شبیه‌سازی	۲-۵-۶
۱۲۴	جمع‌بندی	۶-۶

۷ جمع‌بندی و پیشنهادات

۱۲۶

۱-۷ جمع‌بندی ۱۲۶

۲-۷ پیشنهاداتی برای کارهای آینده ۱۲۸

فهرست جداول

۶۶	۱-۵	مقادیر نهایی NMSE برای الگوریتم‌های مختلف.
	۲-۵	زمان متوسط (برحسب ثانیه) الگوریتم‌های مختلف برای سطوح تُنکی گوناگون و در حالت بدون نویز. هر خانه از این جدول زمان متوسطی را نشان می‌دهد که هر الگوریتم صرف می‌کند تا به یک سطح خاص از MSE برسد. علامت خط تیره نشان می‌دهد که الگوریتم متناظر قادر نبوده به آن سطح از MSE برسد. بهترین نتیجه با فونت برجسته مشخص شده است.
۹۱	۳-۵	مشابه جدول ۲-۵ ولی برای حالت نویزی.
۹۱	۴-۵	مقایسه‌ی عملکرد الگوریتم‌های مختلف در بازیابی تصاویر از روی نسخه‌های فشرده‌ی بلوک‌های آنها و برای مقادیر مختلف نسبت‌های نمونه‌برداری که با δ نشان داده می‌شود. مقادیر گزارش شده، PSNR برحسب دی بی هستند. بهترین نتیجه با فونت برجسته مشخص شده است.
۹۲	۱-۶	پیچیدگی‌های محاسباتی برای الگوریتم‌های مختلف. ابعاد دیکشنری $n \times N$ و تعداد داده‌های آموزشی برابر با M است.
۱۱۴	۲-۶	مقادیر همبستگی متقابل برای الگوریتم‌های SIDCO [۱۰۳]، LZYCB [۷۸] و IFD-AMPM و فریم‌های $\mathbf{F} \in \mathbb{R}^{n \times 120}$ با مقادیر مختلف برای n
۱۲۳	۳-۶	مقادیر همبستگی متقابل برای الگوریتم‌های SIDCO [۱۰۳]، LZYCB [۷۸] و IFD-AMPM و فریم‌های $\mathbf{F} \in \mathbb{R}^{r \times m}$ با مقادیر مختلف برای m . همچنین، مقادیر بدست آمده از طریق عددی، که در [۳۸] گزارش شده است نیز در این جدول ذکر شده است.
۱۲۵		

فهرست اشکال

۱۱	الگوریتم جستجوی تطابق (MP) [۸۶]	۱-۲
	الگوریتم CoSaMP [۹۳]. منظور از s ، انتخاب s تا از مؤلفه‌ها با بزرگترین قدرمطلق و	۲-۲
۱۲	صفر کردن بقیه است.	
	نمودار بلوکی الگوریتم IMAT [۸۸]. بلوک‌های DT و IDT، به ترتیب، نشان‌دهنده‌ی یک	۳-۲
۱۶	تبدیل و عکس آن هستند.	
	نمایش مفهوم عملگر پروکسیمال. در این شکل، منحنی تیره نشان‌دهنده‌ی دامنه تابع بوده و	۱-۴
۳۷	منحنی‌های کمرنگ تعدادی از سطوح تراز آن را نشان می‌دهند [۹۶].	
۳۸	عملگر پراکسیمال مربوط به دو تابع معروف.	۲-۴
۴۰	نمایش تابع هموار f و تقریب مرتبه‌ی دوم آن $\tilde{f}(x, y)$ در نقطه‌ی y	۳-۴
	تکرارهای الگوریتم FBS که در معادله‌ی (۴-۱۵) آمده است در این شکل نشان داده شده	۴-۴
	است. در این شکل، نقطه‌ی z_k عبارت است از نتیجه‌ی اعمال یک گام گرادیان روی تابع f	
۴۱	که متناظر با گام رو به جلو است. برای توضیحات بیشتر به متن مراجعه کنید.	
	تابع آستانه‌گذاری نرم صفر هموار شده (۵-۱۶) به همراه تابع آستانه‌گذاری سخت.	۱-۵
۵۸	تابع آستانه‌گذاری نرم یک هموار شده به همراه تابع آستانه‌گذاری نرم.	۲-۵
۵۹	الگوریتم تکراری تُنک‌سازی-تصویر کردن (ISP).	۳-۵
۶۲	الگوریتم پیشنهادی برای حل مسأله‌ی (۵-۲۰).	۴-۵
۶۵	مقادیر NMSE برحسب تکرار الگوریتم‌های مختلف ISP.	۵-۵
۶۶	مقادیر اندیس جینی بر حسب شماره‌ی تکرار الگوریتم‌های مختلف ISP.	۶-۵
	(آ) مقادیر NMSE و (ب) زمان اجرای الگوریتم $ISP-l$ که یکبار مجهز به الگوریتم معرفی	۷-۵
	شده در [۵۰] است و یکبار با الگوریتم پیشنهادی در شکل ۴-۵. سطح نویز برابر با σ_{noise}	
۶۷	۰/۰۰۵ بوده و $s = ۵۰$	
	منحنی‌های گذر فاز برای الگوریتم‌های مختلف در بازایی سیگنال‌های تُنک به طول $n = ۱۰۰۰$	۸-۵
۶۸	در این شکل $\rho \triangleq s/m$ و $\delta \triangleq m/n$	

- ۹-۵ منحنی‌های گذر فاز که نواحی موفقیت‌آمیز را از نواحی شکست که در شکل ۵-۸ رسم شده‌اند جدا می‌کنند. منحنی تئوری گذر فاز برای الگوریتم LASSO نیز رسم شده است [۴۷]. معیار
- ۶۹ جداسازی نواحی موفقیت و شکست نرخ موفقیت برابر با $0/5$ بوده است.
- ۷۰ ۱۰-۵ زمان متوسط اجرای الگوریتم‌های مختلف برحسب پارامترهای (آ) $\rho \triangleq s/m$ و (ب) $\delta \triangleq m/n$.
- ۷۱ ۱۱-۵ مقادیر NMSE در بازیابی سیگنال‌های تُنک به طول 1000 از روی تعداد 500 اندازه‌گیری تصادفی با استفاده از ماتریس‌های اندازه‌گیری (آ) تُنک، (ب) بدحالت، (ج) میانگین غیرصفر و (د) کم‌رُتبه. در این شکل، ISP-Hard1 و ISP-Hard2، به ترتیب، نشان‌دهنده‌ی الگوریتم ISP-Hard با تصویر کردن پیشنهادی در الگوریتم Robust-SL0 و با تصویر کردن پیشنهادی هستند.
- ۷۲ ۱۲-۵ مقادیر NMSE برای الگوریتم‌های مختلف در بازیابی سیگنال‌های فشرده‌پذیر به طول $n = 1000$ از روی $m = 500$ اندازه‌گیری گوسی. محور افقی پارامتر q ، که معیاری از میزان فشرده‌پذیری است، را نشان می‌دهد. همچنین، در محور عمودی سمت راست، مقادیر معادل ضریب جینی نشان داده شده است.
- ۷۴ ۱۳-۵ نمودار تابع (۵-۴۱) برای مقادیر مختلف β
- ۷۶ ۱۴-۵ الگوریتم پیشنهادی برای حل مسأله‌ی (۵-۴۲).
- ۷۶ ۱۵-۵ الگوریتم پیشنهادی LOsoft برای حل مسأله‌ی (۵-۴۲).
- ۷۸ ۱۶-۵ مقادیر متوسط MSE برحسب دسی‌بل برای الگوریتم‌های مختلف. مقادیر مختلف تُنکی، s ، در نظر گرفته شده است. همچنین، طول سیگنال‌های تُنک برابر با $n = 1000$ بوده و بازسازی از روی تعداد $m = 400$ اندازه‌گیری گوسی انجام شده است.
- ۷۹ ۱۷-۵ نتایج بازسازی تصاویر 32×32 از روی اندازه‌گیری‌های گوسی و برای مقادیر مختلف نرخ نمونه‌برداری، m/n
- ۸۰ ۱۸-۵ زمان متوسط (برحسب ثانیه) برای الگوریتم‌های مختلف.
- ۸۲ ۱۹-۵ الگوریتم پیشنهادی IPP برای حل مسأله‌ی (۵-۵۸).
- ۸۵ ۲۰-۵ نمودارهای تابع آستانه‌گذاری SCAD که در (۵-۶۷) تعریف شده است برای مقادیر مختلف a .
- ۸۵ ۲۱-۵ نمودارهای آستانه‌گذاری SL0 به همراه توابع آستانه‌گذاری سخت و نرم. در این شکل، $\sigma = \lambda\sqrt{2e}$
- ۸۹ ۲۲-۵ بررسی تاثیر پارامتر a در الگوریتم IPP-SCAD (ردیف بالا). برای هر مقدار a ، بهترین مقدار w نیز نشان داده شده است. محور افقی تعداد درایه‌های غیرصفر در سیگنال تُنک را نشان می‌دهد. ردیف پائین نیز اثر پارامتر w را نشان می‌دهد.
- ۹۰ ۲۳-۵ مقایسه‌ی الگوریتم‌های IPP و ISP با توابع آستانه‌گذاری متفاوت و در دو حالت: با وزن مخالف صفر ($w = 0/95$)، نمودارهای خط‌چین، و با وزن صفر ($w = 0$)، نمودارهای توپر.
- ۹۰ ۲۴-۵ مقایسه عملکرد الگوریتم IPP-SCAD با الگوریتم‌های دیگر برای حالت نویزی و بدون نویز.
- ۹۳ ۲۵-۵ مقایسه‌ی دیداری تصاویر بازسازی شده توسط الگوریتم‌های مختلف با نرخ نمونه‌برداری $\delta = 0/4$. از راست به چپ، هر ستون متناظر است با: تصویر اصلی، تصویر بازسازی شده با الگوریتم SCSA-FIT، $IPP(w = 0)$ ، و $IPP(w = 0/85)$
- ۹۹ ۱-۶ الگوریتم پیشنهادی HighDim-DL برای حل مسأله‌ی یادگیری دیکشنری.

- ۲-۶ مقادیر نهایی خطای نمایش برای الگوریتم‌های مختلف و دو حالت ماتریس sketching . . . ۱۰۱
- ۳-۶ الگوریتم پیشنهادی، RINC-DL، برای حل مسأله‌ی (۶-۲۹). در این شکل، منظور از SA یافتن نمایش تُنک داده‌ها است. ۱۰۹
- ۴-۶ الگوریتم پیشنهادی، CINC-DL، برای حل مسأله‌ی (۶-۲۹). در این شکل، منظور از SA یافتن نمایش تُنک داده‌ها است. ۱۱۱
- ۵-۶ نمودارهای مقادیر نهایی خطا و همبستگی متقابل برای الگوریتم‌های پیشنهادی و مقادیر مختلف پارامترهای آنها. مقادیر پیش‌فرض برای پارامترها برابر است با $c = 0.85$ ، $J = 3$ ، $L_2 = 20$ ، $\epsilon = 0.05$. برای بررسی تاثیر هر کدام از پارامترها، بقیه‌ی پارامترها به مقادیر پیش‌فرض خود تنظیم شده‌اند. ۱۱۵
- ۶-۶ زمان اجرای الگوریتم‌های پیشنهادی برحسب مقادیر مختلف پارامترهای آنها. بقیه‌ی تنظیمات شبیه شکل ۵-۶ است. ۱۱۶
- ۷-۶ مقادیر نهایی خطای نمایش و همبستگی متقابل دیکشنری‌های یادگیری شده، برای دو الگوریتم BSC-DL و RINC-DL و بعنوان توابعی از پارمتر λ . مقدار باند ولج هم با μ_{min} مشخص شده است. ۱۱۷
- ۸-۶ مقایسه‌ی الگوریتم‌های مختلف برحسب (آ) مقدار نهایی خطای نمایش و (ب) زمان اجرا، و بازای مقادیر همبستگی‌های متقابل. ۱۱۸
- ۹-۶ الگوریتم پیشنهادی IFD-AMPM برای طراحی فریم‌های ناهمبسته. ۱۲۱
- ۱۰-۶ مقادیر همبستگی متقابل برحسب تکرار الگوریتم‌های SIDCO [۱۰۳]، [۷۸] LZVCB و IFD-AMPM برای طراحی فریم‌های با ابعاد (آ) 120×15 و (ب) 120×25 ۱۲۳

فهرست کلمات اختصاری

ADMM	Alternating Direction Method of Multipliers
BP	Basis Pursuit
CS	Compressed Sensing
DCT	Discrete Cosine Transform
EVD	Eigenvalue Value Decomposition
FBS	Forward Backward Splitting
IHT	Iterative Hard Thresholding
IMAT	Iterative Method with Adaptive Thresholding
IPP	Iterative Proximal Projection
ISP	Iterative Sparsification Projection
IST	Iterative Shrinkage-Thresholding
MC	Mutual Coherence
MOD	Method of Optimal Directions
MP	Matching Pursuit
MSE	Mean Squared Error
NP	Non Polynomial
OMP	Orthogonal Matching Pursuit
RIP	Restricted Isometry Property
SL0	Smoothed ℓ_1 (norm)
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition

فصل ۱

پیشگفتار

در این فصل، مباحث مورد بررسی در این رساله را به طور مختصر بررسی می‌کنیم. این مباحث به طور عمده حول محور نمایش تَنک و یادگیری دیکشنری قرار می‌گیرد. نمایش تَنک یک توصیف ساده و کارآمد از سیگنال‌ها برحسب مجموعه‌ی از سیگنال‌های پایه، که اتم نامیده می‌شوند، فراهم می‌کند [۵۲]. به مجموعه‌ی اتم‌ها، دیکشنری گفته می‌شود. یادگیری دیکشنری برای کاربردهای نمایش تَنک مورد استفاده قرار می‌گیرد که عموماً شامل بازیابی و یا بهبود کیفیت سیگنال و یا تصویر است. در واقع، نمایش تَنک یک مدل برای سیگنال معرفی می‌کند که طبق آن می‌توان عملیات پردازش سیگنال از جمله بهبود کیفیت، جداسازی نویز از سیگنال (نویززدائی) [۵۲]، جداسازی منابع ترکیب شده [۳۷]، نمونه‌برداری و بازسازی تصاویر [۱۱۶] و بازسازی تصاویر پزشکی از روی نمونه‌های محدود حوزه‌ی فوریه‌ی آنها [۸۱] را با کیفیتی عموماً بهتر از روش‌های کلاسیک انجام داد.

برای یک مرور اجمالی بر بحث نمایش تَنک و یادگیری دیکشنری، فرض کنید $\mathbf{y} \in \mathbb{R}^n$ یک سیگنال داده شده، و $\mathbf{D} \in \mathbb{R}^{n \times m}$ یک دیکشنری معلوم شامل m اتم در فضای \mathbb{R}^n باشد. حال، نمایش تَنک به صورت تقریب زدن سیگنال \mathbf{y} بر حسب یک ترکیب خطی شامل تعداد معدودی اتم از دیکشنری \mathbf{D} تعریف می‌شود. این مسأله را می‌توان به فرم زیر نوشت:

$$(P_0) : \min_{\mathbf{x}} \|\mathbf{x}\|. \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \quad (1-1)$$

که در آن، $\|\cdot\|$ اصطلاحاً شبه‌نرم^۱ صفر بوده که تعداد درایه‌های ناصفر را می‌شمارد و $\mathbf{x} \in \mathbb{R}^m$ بردار نمایش تَنک

^۱Pseudo norm

است. با توجه به این که شبه‌نرم صفر تابعی گسسته و مشتق‌ناپذیر است، حل مسأله فوق در زمان چندجمله‌ای^۱ ممکن نبوده و نشان داده شده است که این مسأله NP-hard است [۴۲]. در نتیجه، تعداد زیادی جایگزین برای این تابع معرفی شده است که یا مشتق‌پذیر هستند و یا توابعی پیوسته و محدب و در نتیجه مناسب برای اعمال الگوریتم‌های بهینه‌سازی هستند [۱۲۱]. یکی از معروف‌ترین این توابع نرم یک بوده که محدب و پیوسته است و مسأله بازیابی نمایش تُنک با استفاده از آن به فرم زیر تبدیل می‌شود [۵۲]:

$$(P_1) : \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}. \quad (2-1)$$

در کاربردهای نمایش تُنک، یک عامل تعیین‌کننده برای موفقیت و کیفیت بالا، دیکشنری مورد استفاده است. اگر چه دیکشنری‌های از پیش‌ساخته شده‌ی زیادی برای این منظور وجود دارد، از جمله دیکشنری فوریه یا تبدیل کسینوسی گسسته^۲ (DCT)، اما نشان داده شده است که طراحی دیکشنری از روی خود داده‌ها عملکرد به مراتب بهتری دارد [۳] (برای جزئیات بیشتر، به قسمت ۲-۳ مراجعه نمایید). مسأله‌ی یادگیری دیکشنری به صورت زیر فرمولبندی می‌شود:

$$\min_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T_0, \quad i = 1, \dots, L, \quad (3-1)$$

که در آن، \mathbf{Y} ماتریس حاوی داده‌های آموزشی است و ستون‌های دیکشنری قید واحد بودن نرم دو را دارند. الگوریتم‌های متنوعی برای حل این مسأله پیشنهاد شده‌اند [۱۱۸]. اساس همه‌ی این روش‌ها مبتنی بر حل مسأله‌ی فوق بصورت تکراری با ثابت گرفتن یک متغیر و حداقل‌سازی تابع هدف روی متغیر دیگر است. به عبارت دقیق‌تر، با شروع از یک دیکشنری و ماتریس ضرایب اولیه‌ی $(\mathbf{D}_0, \mathbf{X}_0)$ تکرارهای زیر برای حل (۳-۱) استفاده می‌شود ($k \geq 0$)

$$\begin{cases} \mathbf{X}_{k+1} = \operatorname{argmin}_{\mathbf{X}} \|\mathbf{Y} - \mathbf{D}_k \mathbf{X}\|_F \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T_0, \quad i = 1, \dots, L \\ \mathbf{D}_{k+1} = \operatorname{argmin}_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D} \mathbf{X}_{k+1}\|_F \quad \text{subject to} \quad \|\mathbf{d}_i\|_2 = 1, \quad i = 1, \dots, m \end{cases} \quad (4-1)$$

همانطور که ملاحظه می‌شود، مسأله‌ی کلی یادگیری دیکشنری خود شامل دو مسأله‌ی جدا است. اولی مسأله‌ی بازیابی نمایش تُنک سیگنال‌های آموزشی با استفاده از دیکشنری فعلی است. مسأله‌ی دوم، به‌روز رسانی دیکشنری با استفاده از ماتریس نمایش‌های تُنک است. در واقع تفاوت اصلی الگوریتم‌های یادگیری دیکشنری در نحوه‌ی حل مسأله‌ی به‌روز رسانی دیکشنری است [۱۱۸]. از آنجایی که هم بهینه‌سازی الگوریتم‌های بازیابی

^۱Polynomial time

^۲Discrete Cosine Transform

نمایش تُنک و هم بهینه‌سازی الگوریتم‌های به‌روزرسانی برای کارایی الگوریتم‌های یادگیری دیکشنری ضروری هستند، در این رساله روی هر دوی این موارد تمرکز خواهیم داشت.

اگرچه تاکنون روش‌های متنوعی برای یادگیری دیکشنری و بازیابی نمایش تُنک ارائه شده است، اما هنوز نیاز به روش‌های جدیدتر و مؤثرتر احساس می‌شود. بعبارت دقیق‌تر، در حوزه‌ی بازیابی نمایش تُنک، کمینه‌سازی توابع ناهموار و نامحدب تشویق‌کننده‌ی تُنکی، مانند تابع نُرم صفر، با توجه به دشواری حل مسأله‌ی متناظر، کمتر مورد توجه قرار گرفته است. از طرف دیگر، چنین توابعی منجر به جواب‌های تُنک‌تر و عملکرد بهتر در بسیاری از کاربردها می‌شوند. در این رساله و بعنوان راه‌حلی برای این مشکل، الگوریتم‌هایی مبتنی بر روش‌های پراکسیمال^۱ ارائه می‌شوند. الگوریتم‌های پیشنهادی به شیوه‌ای کارآمد مسأله‌ی دشوار کمینه‌سای توابع ناهموار و نامحدب را حل می‌کنند. بعلاوه، این روش‌ها دید روشن‌تری نسبت به الگوریتم نُرم صفر هموار شده (SL0) [۹۲]، که تقریبی هموار اما نامحدب از نُرم صفر پیشنهاد می‌کند، ارائه می‌دهند. این موارد شامل ارائه‌ی یک اثبات همگرایی برای الگوریتم SL0 و نشان دادن این موضوع است که تابع آستانه‌گذاری متناظر با این الگوریتم، تابعی بین تابع آستانه‌گذاری سخت^۲ و تابع آستانه‌گذاری نُرم^۳ است. علاوه بر این، روش جدید دیگری برای کمینه‌سازی تابع نُرم صفر پیشنهاد می‌شود که مبتنی بر استفاده از یک تعریف معادل تابع نُرم صفر بر مبنای تابع نُرم یک است. حل این مسأله نیز به کمک روش‌های پراکسیمال انجام می‌شود. در مورد یادگیری دیکشنری، دو هدف اصلی در این رساله دنبال می‌شود. اولی ارائه‌ی روش‌های کارآمدتری برای یادگیری دیکشنری‌های با همبستگی متقابل پائین است. توضیح این که، روش‌های موجود برای این منظور از تقریب‌هایی استفاده می‌کنند که در نهایت قادر نیستند مصالحه‌ای مطلوب بین کاهش خطای نمایش (قدرت وفق کردن دیکشنری به داده‌ها) و کاهش همبستگی بین اتم‌های آن (حداکثر عدم شباهت اتم‌ها به یکدیگر) برقرار کنند. برای حل این مشکل، در این رساله روش‌هایی معرفی می‌شوند که بصورت مستقیم از تابع همبستگی متقابل در یادگیری دیکشنری استفاده می‌کنند. این امر با توجه به ناهموار بودن تابع همبستگی متقابل، حل مسأله‌ی یادگیری دیکشنری را با چالش مواجه می‌کند که برای این منظور از روش‌های پراکسیمال استفاده می‌کنیم. همچنین، این ایده را به طراحی ماتریس‌های حسگر نیز تعمیم می‌دهیم. هدف دومی که در این رساله پیگیری می‌شود، ارائه روش‌هایی کارآمدتری برای یادگیری توزیع‌شده‌ی

^۱ Proximal algorithms

^۲ Hard-thresholding

^۳ Soft-thresholding

دیکشنری^۱ است. این دسته از روش‌ها مبتنی بر توزیع داده‌های آموزشی بین چند پردازنده است که در تعامل با یکدیگر، سعی در یادگیری یک دیکشنری واحد برای کل داده‌های آموزشی دارند. اما این تعامل بصورت تبادل اطلاعاتی بین این پردازنده‌ها است که در بعضی موارد حجم آن می‌تواند بالا باشد. برای رفع این مشکل، یک ساختار تُنک برای دیکشنری در نظر گرفته می‌شود که باعث کاهش حجم داده‌ی انتقالی می‌شود. بعلاوه، برای صرفه‌جویی بیشتر در میزان محاسبات انجام شده روی هر یک از پردازنده‌ها، از روش‌های کاهش بُعد روی داده‌های آموزشی استفاده می‌شود.

مطالب این رساله به این شیوه دسته‌بندی می‌شود که ابتدا در فصل دوم، نمایش تُنک سیگنال‌ها را به همراه تعدادی از الگوریتم‌های موجود مرور می‌کنیم. فصل سوم به مرور مسأله‌ی یادگیری دیکشنری و الگوریتم‌های مهم موجود در این زمینه اختصاص دارد. همچنین، در این فصل مروری خواهیم داشت بر روش‌های کارآمد برای پردازش داده‌های با ابعاد بالا. سپس، در فصل چهارم الگوریتم‌های پراکسیمال را بعنوان ابزاری مهم برای حل بسیاری از مسائل بهینه‌سازی پردازش سیگنال و یادگیری ماشین مرور می‌کنیم. این الگوریتم‌ها از اهمیت ویژه‌ای در این رساله برخوردار هستند؛ چرا که عمده‌ی الگوریتم‌های پیشنهادی مبتنی بر آنها هستند. در ادامه‌ی رساله، فصل پنجم به معرفی الگوریتم‌های پیشنهادی برای بازیابی نمایش تُنک اختصاص دارد. در این فصل، با الهام از ساختار الگوریتم SL0، الگوریتم‌های جدیدی برای بازیابی نمایش تُنک ارائه می‌شوند که نسبت به الگوریتم SL0 سرعت همگرایی و کیفیت بهتری دارند. سپس، فصل ششم الگوریتم‌های پیشنهادی برای یادگیری دیکشنری و نیز طراحی ماتریس حسگر را تشریح می‌کند. در این فصل، ابتدا الگوریتمی برای یادگیری دیکشنری از روی داده‌های با ابعاد بالا معرفی می‌شود که مبتنی بر کاهش بُعد داده‌ها و پردازش توزیعی آنها است. سپس، الگوریتم‌هایی برای یادگیری دیکشنری‌های با همبستگی متقابل کم ارائه می‌شوند. همبستگی متقابل یک ماتریس عبارت است از حداکثر همبستگی بین دوجه‌دوی ستون‌های آن که پارامتری مهم برای موفقیت عملکرد الگوریتم‌های بازیابی نمایش تُنک سیگنال‌ها است. همانطور که خواهیم دید، الگوریتم‌های پیشنهادی نسبت به الگوریتم‌های موجود حجم محاسبات کمتری داشته و منجر به یادگیری دیکشنری‌های با همبستگی متقابل کمتر و با میزان خطای قابل مقایسه با الگوریتم‌های دیگر می‌شوند. در نهایت، فصل هفتم به جمع‌بندی و کارهای پیشنهادی برای آینده اختصاص دارد.

مطالب این رساله منجر به مقالات زیر شده است:

¹Distributed Dictionary Learning

1. M. Sadeghi and M. Babaie-Zadeh, "Iterative Sparsification-projection: Fast and Robust Sparse Signal Approximation," *IEEE Transactions on Signal Processing*, vol 64, no. 21, pp. 5536–5548, Nov. 2016.
2. M. Sadeghi and M. Babaie-Zadeh, "Incoherent Unit-norm Frame Design via an Alternating Minimization Penalty Method," *IEEE Signal Processing Letters*, vol. 24, no. 1, pp. 32–36, Jan. 2017.
3. F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "Sparse Signal Recovery Using Iterative Proximal Projection," *IEEE Transactions on Signal Processing*, vol. 66, no. 44, pp. 879–894, Feb. 2018.
4. M. Sadeghi, and M. Babaie-Zadeh, "Learning Low-coherence Dictionaries for Sparse Signal Approximation," *Signal Processing (Elsevier)*, 2018 (submitted).
5. M. Sadeghi, F. Ghayem, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "L0Soft: ℓ_0 Minimization via Soft Thresholding," *Signal Processing (Elsevier)*, 2018 (submitted).
6. M. Sadeghi, M. Babaie-Zadeh, C. Jutten, "Regularized Low Coherence Overcomplete Dictionary Learning for Sparse Signal Decomposition," *EUSIPCO 2016 conference*, Budapest, Aug. 2016.

فصل ۲

نمایش تُنک سیگنال

۱-۲ مقدمه

در این فصل، نمایش تُنک را به همراه تعدادی از الگوریتم‌های موجود مرور می‌کنیم. ابتدا مسأله‌ی بازیابی تُنک‌ترین نمایش سیگنال را بیان کرده، سپس قضایای یکتایی تُنک‌ترین جواب را مرور می‌کنیم. در ادامه، تعدادی از الگوریتم‌های بدست آوردن نمایش تُنک سیگنال‌ها را که تاکنون معرفی شده‌اند به اختصار مرور می‌کنیم. در نهایت، مروری مختصر خواهیم داشت بر حسگری فشرده، که یک کاربرد مهم از نمایش تُنک و تحوُّلی در نمونه‌برداری سیگنال‌ها است.

۲-۲ معرفی مسأله

ماتریس $D \in \mathbb{R}^{n \times m}$ را که $m > n$ در نظر بگیرید. در این صورت دستگاه معادلات خطی $y = Dx$ فرومعین^۱ بوده و در حالت کلی بی‌شمار جواب دارد. در مسائل مختلف پردازش سیگنال به چنین دستگاه معادلات خطی فرومعینی برمی‌خوریم؛ به عنوان مثال، بسیاری از مسائل معکوس^۲ در این دسته قرار دارند [۵۲]. در این گونه مسائل، ما به دنبال یافتن سیگنال مطلوب x هستیم که تحت تبدیل خطی D تغییر شکل یافته و ما تنها نسخه‌ی با تفکیک‌پذیری پائین و احتمالاً تخریب شده‌ی آن، یعنی y ، را در اختیار داریم.

^۱Underdetermined

^۲Inverse problems

اگرچه معادله‌ی خطی فرومیین ذکر شده در حالت کلی بی‌شمار جواب دارد، اما آنچه برای ما مهم است تنها یکی از این جواب‌ها است. بسته به کاربرد می‌توان جوابی را انتخاب کرد که یک ویژگی منحصر به فرد داشته باشد. به عنوان مثال، جواب با حداقل انرژی. انتخاب یک جواب با ویژگی خاص همان تنظیم^۱ این دستگاه معادلات است. به عبارت دقیق‌تر، اگر تابع $J: \mathbb{R}^m \rightarrow \mathbb{R}$ تشویق‌کننده‌ی ویژگی مورد نظر ما باشد، برای بدست آوردن این جواب باید مسأله‌ی زیر را حل کنیم:

$$P_J: \min_{\mathbf{x}} J(\mathbf{x}) \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x}. \quad (1-2)$$

در مثال جواب با حداقل انرژی داریم $J(\mathbf{x}) = \|\mathbf{x}\|_2^2$. این همان تنظیم معروف تیخونوف^۲ است [۵۲]. توجه کنید که در این حالت مسأله محدب بوده و در نتیجه این جواب یکتا است. در بسیاری از کاربردها اما مطلوب ما جوابی است که تا حد ممکن درایه‌ی صفر (یا خیلی نزدیک صفر) داشته باشد. از جمله‌ی این کاربردها می‌توان به فشرده‌سازی داده‌ها اشاره کرد [۲۴]. به علاوه، بسیاری از سیگنال‌های طبیعی از جمله تصویر و سیگنال صحبت در یک دیکشنری مناسب نمایشی تُنک دارند؛ به عبارت دیگر بُعد واقعی آن‌ها خیلی کمتر از بُعد ظاهری (تعداد مؤلفه‌ها در نمایش برداری) آن‌ها است. این خود می‌تواند نشانه‌ای مناسب برای بازیابی این سیگنال‌ها در مسائل معکوس از جمله نویززدائی باشد.

برای بدست آوردن تُنک‌ترین جواب، باید مسأله‌ی زیر را حل کنیم:

$$P_0: \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \quad (2-2)$$

که در آن $\|\mathbf{x}\|_0 \triangleq |\text{supp}(\mathbf{x})| = \{i: x_i \neq 0\}$ نشان‌دهنده‌ی تعداد درایه‌های غیر صفر \mathbf{x} است. مسأله‌ی (۲-۲) غیر محدب است و از طرفی تابع $\|\mathbf{x}\|_0$ ناهموار و مشتق‌ناپذیر بوده و علاوه بر این، حل این مسأله مستلزم یک جستجوی ترکیباتی است که در ابعاد بالا امکان‌پذیر نیست. به عبارت دیگر، برای پیدا کردن جوابی با s درایه‌ی غیر صفر، لازم است در بدترین حالت تعداد $N = \binom{m}{s}$ زیر ماتریس از \mathbf{D} را بررسی کنیم. یکی از جایگزین‌های جالب برای این تابع، نُرم یک است که محدب بوده و مسأله‌ی حاصل را می‌توان با الگوریتم‌های گوناگونی و در زمان چندجمله‌ای حل نمود [۳۴]. تابع نُرم یک در حقیقت نزدیک‌ترین تابع محدب به تابع شبه نُرم صفر است [۵۲]. در نهایت مسأله‌ی زیر را داریم:

$$P_1: \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x}. \quad (3-2)$$

^۱Regularization

^۲Tykhonov

به طور کلی، یک دسته از توابع تشویق کننده‌ی تُنک بودن بصورت $J(\mathbf{x}) = \|\mathbf{x}\|_p^p = \sum_i |x_i|^p$ با $0 \leq p \leq 1$ است [۵۹، ۳۱]. به استثنای $p = 1$ که متناظر با نُرم یک است، بقیه‌ی این توابع همه‌ی خواص یک تابع نُرم را نداشته و عموماً «شبه نُرم» خوانده می‌شوند. با این انتخاب، مسأله‌ی متناظر به فرم زیر است:

$$P_p : \min_{\mathbf{x}} \|\mathbf{x}\|_p^p \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x}. \quad (4-2)$$

در انتها، توجه به این نکته ضروری است که در حضور نویز اندازه‌گیری، یعنی هنگامی که اندازه‌گیری‌های \mathbf{y} با یک نویز (عموماً گوسی) جمع شده‌اند، مسائل مقاوم به نویز زیر حل می‌شوند [۴۵]

$$P_p^\epsilon : \min_{\mathbf{x}} \|\mathbf{x}\|_p. \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon, \quad (5-2)$$

و

$$P_1^\epsilon : \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon. \quad (6-2)$$

که در آن، $\epsilon > 0$ پارامتری وابسته به سطح نویز است [۵۲].

۳-۲ یکتایی تُنک‌ترین جواب

در این قسمت، قضایای مربوط به یکتایی تُنک‌ترین نمایش یک سیگنال را به نقل از [۵۲] بیان می‌کنیم. ابتدا قضیه‌ی یکتایی را با استفاده از مفهوم اسپارک^۱ دیکشنری بیان می‌کنیم. اسپارک یک ماتریس روشی برای توصیف فضای پوچ^۲ آن ماتریس است. طبق تعریف، اسپارک یک ماتریس عبارت است از کمترین تعداد ستون‌هایی از آن که وابسته‌ی خطی‌اند. با این تعریف، قضیه‌ی یکتایی به صورت زیر بیان می‌شود:

قضیه ۱-۲ اگر جوابی از دستگاه فرومعی $\mathbf{y} = \mathbf{D}\mathbf{x}$ در شرط $\|\mathbf{x}\|_0 < \text{spark}(\mathbf{D})/2$ صدق کند، آنگاه لزوماً تُنک‌ترین جواب این دستگاه معادلات خواهد بود.

همانطور که از تعریف فوق برمی‌آید، بدست آوردن اسپارک یک ماتریس خود نیازمند یک جستجوی ترکیبیاتی است. بنابراین به روش‌های ساده‌تری برای تضمین یکتایی تُنک‌ترین جواب نیاز داریم. یک راه بسیار ساده استفاده از همبستگی متقابل^۳ بین اتم‌های دیکشنری است [۴۶]. طبق تعریف، همبستگی متقابل یک ماتریس عبارت است از ماکسیمم قدر مطلق ضریب همبستگی بین ستون‌های متمایز آن. به بیان ریاضی، همبستگی متقابل برای ماتریس

^۱Spark

^۲Null-space

^۳Mutual-coherence

\mathbf{A} به صورت زیر تعریف می‌شود:

$$\mu(\mathbf{A}) = \max_{i \neq j} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2} \quad (7-2)$$

به این ترتیب، همبستگی متقابل، کمیتی برای ارزیابی وابستگی خطی بین ستون‌های یک ماتریس است. می‌توان نشان داد که رابطه‌ی بین اسپارک و همبستگی متقابل یک ماتریس بصورت $\text{spark}(\mathbf{A}) \geq 1 + 1/\mu(\mathbf{A})$ است [۵۲]. قضیه‌ی یکتائی با استفاده از همبستگی متقابل دیکشنری بصورت زیر بیان می‌شود:

قضیه ۲-۲ اگر جوابی از دستگاه فرومعی $\mathbf{y} = \mathbf{D}\mathbf{x}$ در شرط $\|\mathbf{x}\|_0 < \frac{1}{\mu} (1 + 1/\mu(\mathbf{D}))$ صدق کند، آنگاه لزوماً تُنک‌ترین جواب این دستگاه معادلات خواهد بود.

همانطور که اشاره شد، مسأله‌ی P ماهیتی ترکیباتی داشته و یک مسأله‌ی NP-hard است. به همین دلیل، مسأله‌ی P_1 که محدب بوده و قابل حل است معرفی شد [۳۴]. نشان داده شده است که اگر جوابی از $\mathbf{y} = \mathbf{D}\mathbf{x}$ در شرط قضیه‌ی ۲-۲ صدق کند، آنگاه هم جواب مسأله‌ی P است و هم جواب مسأله‌ی P_1 ؛ به عبارت دیگر، در این وضعیت این دو مسأله معادل‌اند [۴۴]. در حضور نویز، اگر جواب مسائل (۲-۵) و (۲-۶) را به ترتیب با \mathbf{x}^ϵ و \mathbf{x}^ϵ نشان دهیم، آنگاه قضایای زیر رابطه‌ی این جواب‌ها با سیگنال اصلی \mathbf{x}_0 را که در $\mathbf{y}_0 = \mathbf{D}\mathbf{x}_0$ صدق می‌کند، مشخص می‌سازند [۴۵]:

قضیه ۳-۲ سیگنال تمیز $\mathbf{y}_0 = \mathbf{D}\mathbf{x}_0$ را در نظر بگیرید. در اینصورت اگر $\|\mathbf{x}_0\|_0 = s < \frac{1}{\mu} (1 + 1/\mu(\mathbf{D}))$ آنگاه

$$\|\mathbf{x}_0^\epsilon - \mathbf{x}_0\|_2 \leq \frac{\epsilon + \delta}{\sqrt{1 - \mu(\mathbf{D})(2s - 1)}}, \quad \forall \epsilon \geq \delta > 0 \quad (8-2)$$

قضیه ۴-۲ سیگنال تمیز $\mathbf{y}_0 = \mathbf{D}\mathbf{x}_0$ را در نظر بگیرید. در اینصورت اگر $\|\mathbf{x}_0\|_0 = s < \frac{1}{\mu} (1 + 1/\mu(\mathbf{D}))$ آنگاه

$$\|\mathbf{x}_1^\epsilon - \mathbf{x}_0\|_2 \leq \frac{\epsilon + \delta}{\sqrt{1 - \mu(\mathbf{D})(4s - 1)}}, \quad \forall \epsilon \geq \delta > 0 \quad (9-2)$$

لازم به ذکر است که علاوه بر همبستگی متقابل و اسپارک، یک ابزار قوی و مهم در بحث تئوری نمایش تُنک، ویژگی همسانی محدود شده^۱ (RIP) [۲۵] است. می‌گوئیم ماتریس $\mathbf{A} \in \mathbb{R}^{n \times m}$ با ثابت δ_k ($0 \leq \delta_k < 1$)

را ارضا می‌کند هرگاه برای هر بردار k -تُنک (یعنی با حداکثر k درایه‌ی غیرصفر) $\mathbf{x} \in \mathbb{R}^m$ داشته باشیم [۲۵]

$$(1 - \delta_k) \cdot \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_k) \cdot \|\mathbf{x}\|_2^2 \quad (10-2)$$

با وجود این که بسیاری از تضمین‌های بازیابی برای الگوریتم‌های مختلف بر مبنای RIP است، اما تعیین این که ثابت RIP برای یک ماتریس داده شده چقدر است، یک مسأله‌ی NP-hard است [۱۱۷].

^۱Restricted isometry property

۴-۲ الگوریتم‌های بازیابی نمایش تُنک

طی دهه‌ی اخیر الگوریتم‌های زیادی برای بدست آوردن تُنک‌ترین نمایش یک سیگنال معرفی شده است [۱۲۱، ۵۲]. این الگوریتم‌ها را می‌توان به سه دسته‌ی کلی تقسیم کرد: الگوریتم‌های حریص، روش‌های مبتنی بر آستانه‌گذاری و روش‌های تقریب نُرم صفر. الگوریتم‌های حریص به صورت قدم به قدم، یک یا چند اتم را که بیشترین همبستگی با باقی‌مانده‌ی مربوط به نمایش سیگنال دارند انتخاب کرده و با استفاده از آن‌ها این باقی‌مانده را به‌روز می‌کنند. اساس این روش‌ها مبتنی بر تخمین مرحله به مرحله‌ی سیگنال با استفاده از اتم‌های دیکشنری است. دسته‌ی دوم و سوم الگوریتم‌ها هدفشان حل یک مسأله‌ی بهینه‌سازی است که حالت کلی آن مسأله‌ی (۱-۲) (یا نسخه‌ی مقاوم به نویز آن) است. به طور کلی، الگوریتم‌های حریص نسبت به دو گروه دیگر الگوریتم‌ها سریع‌تر هستند؛ اما هزینه‌ی این سرعت بالا، دقت کمتر این الگوریتم‌ها است. در قسمت بعد، تعدادی از این الگوریتم‌ها را به اختصار مرور می‌کنیم.

۱-۴-۲ روش‌های حریص

الگوریتم‌های حریص را می‌توان به دو دسته‌ی کلی تقسیم کرد: دسته‌ی اول در هر گام تنها یک اتم را به عنوان اتمی که بیشترین شباهت را به باقی‌مانده نمایش سیگنال دارد انتخاب می‌کنند. دسته‌ی دوم بیش از یک اتم را انتخاب کرده، سپس طی پیشروی الگوریتم تعدادی از این اتم‌ها را حذف کرده یا اتم‌های جدیدی را به این مجموعه اضافه می‌کنند. الگوریتم‌های 1MP و 2OMP در دسته‌ی اول، و الگوریتم‌های 3CoSaMP و 4GOMP [۱۲۷] در دسته‌ی دوم قرار دارند.

۱-۱-۴-۲ الگوریتم MP

الگوریتم جستجوی تطابق یا به اختصار MP به عنوان روشی برای تجزیه‌ی اتمی و در [۸۶] معرفی شد. در این الگوریتم، بصورت قدم به قدم اتمی انتخاب می‌شود که بیشترین شباهت (یا به بیان دیگر کم‌ترین فاصله) را

^۱ Matching Pursuit

^۲ Orthogonal Matching Pursuit

^۳ Compressive Sampling Matching Pursuit

^۴ Generalized OMP

- هدف: محاسبه‌ی نمایش تنک y برحسب اتم‌های D
- مقداردهی اولیه: $r^0 = y$ و $x^0 = 0$
- شروع الگوریتم: قرار بده $k = 1$ و گام‌های زیر را تا رسیدن به شرط توقف انجام بده:
 ۱. محاسبه‌ی همبستگی اتم‌ها با باقی‌مانده: $c^k = D^T r^{k-1}$
 ۲. انتخاب بهترین اتم: $i^k = \arg \max_i |c_i^k|$
 ۳. به‌روز کردن نمایش تنک: $x_{i^k}^k = x_{i^k}^{k-1} + c_{i^k}^k$
 ۴. به‌روز کردن باقی‌مانده: $r^k = r^{k-1} - c_{i^k}^k d_{i^k}$
 ۵. چک کردن شرط توقف: اگر شرط توقف برآورده نشده است، قرار بده $k = k + 1$ و برگرد به گام ۱
- خروجی: x^k

شکل ۲-۱: الگوریتم جستجوی تطابق (MP) [۸۶].

با باقی‌مانده‌ی نمایش سیگنال دارد. این باقی‌مانده در قدم اول خود سیگنال بوده و سپس در قدم‌های بعدی از تفاضل سیگنال با تقریب آن بدست می‌آید. معیار توقف این الگوریتم (و سایر الگوریتم‌های حریص) یا رسیدن نرُم باقی‌مانده (یا خطای نمایش) به یک حد از پیش تعیین شده و یا معیار حداکثر تعداد اتم‌ها در نمایش سیگنال است. شبه کُد این الگوریتم در شکل ۲-۱ آورده شده است.

۲-۱-۴-۲ الگوریتم OMP

الگوریتم OMP یکی دیگر از الگوریتم‌های حریص است که در [۹۸] معرفی شده است. اساس این الگوریتم شبیه الگوریتم MP است ولی با یک تفاوت مهم که منجر به بهبود زیاد عملکرد آن می‌شود. در این الگوریتم بر خلاف الگوریتم MP، برای به‌روز کردن نمایش تنک، سیگنال روی زیرفضای تولید شده توسط اتم‌های انتخاب شده تا آن مرحله، تصویر می‌شود. به عبارت دیگر، در گام ۳ شکل ۲-۱ داریم $x^k = D_{\Gamma^k}^\dagger y$ که در آن Γ^k حاوی اندیس اتم‌هایی است که تا آن گام انتخاب شده‌اند. در گام ۴ نیز داریم $r^k = r^{k-1} - D x^k$. توجه کنید که این کار حجم محاسبات این الگوریتم را نسبت به الگوریتم MP افزایش می‌دهد؛ ولی عملکرد آن بهبود زیادی پیدا می‌کند. شایان ذکر است که الگوریتم GOMP، که یک نسخه‌ی تعمیم‌یافته از الگوریتم OMP است [۱۲۷]، به جای انتخاب تنها یک اتم در هر تکرار، بیش از یک اتم انتخاب می‌کند. نشان داده شده است که این الگوریتم نسبت به الگوریتم OMP همگرایی سریعتری داشته و کیفیت جواب‌های آن بهتر است [۱۲۷].

- هدف: محاسبه‌ی یک نمایش s -تُنک از سیگنال y در دیکشنری D
- مقداردهی اولیه: $\mathbf{r}^0 = y$ و $\mathbf{x}^0 = \mathbf{0}$
- شروع الگوریتم: قرار بده $k = 1$ و گام‌های زیر را تا رسیدن به شرط توقف انجام بده:
 ۱. محاسبه‌ی همبستگی اتم‌ها با باقی‌مانده: $\mathbf{c}^k = D^T \mathbf{r}^{k-1}$
 ۲. انتخاب s اتم با بیشترین شباهت به باقی‌مانده: $\Omega = \text{supp}([\mathbf{c}^k]_s)$
 ۳. ترکیب اندیس اتم‌های جدید با اتم‌های قدیم: $T = \Omega \cup \text{supp}(\mathbf{x}^{k-1})$
 ۴. تخمین اولیه از نمایش تُنک: $\mathbf{x}_T^k = D_T^\dagger y$ و $\mathbf{x}_{T^c}^k = \mathbf{0}$
 ۵. تخمین نهائی نمایش تُنک: $\mathbf{x}^k = [\mathbf{x}^k]_s$
 ۶. به‌روز کردن باقی‌مانده: $\mathbf{r}^k = y - D\mathbf{x}^k$
 ۷. چک کردن شرط توقف: اگر شرط توقف برآورده نشده است، قرار بده $k = k + 1$ و برگرد به گام ۱
- خروجی: \mathbf{x}^k

شکل ۲-۲: الگوریتم CoSaMP [۹۳]. منظور از $[\cdot]_s$ ، انتخاب s تا از مؤلفه‌ها با بزرگترین قدرمطلق و صفر کردن بقیه است.

۲-۴-۱-۳ الگوریتم CoSaMP

در الگوریتم‌های حریص مبتنی بر انتخاب تنها یک اتم در هر گام، مانند دو الگوریتم MP و OMP، اتمی که در هر گام انتخاب می‌شود، به طور قطع تا پایان الگوریتم در نمایش تُنک سیگنال حضور دارد. به عبارت دیگر، اگر این الگوریتم‌ها در انتخاب یک اتم اشتباه کنند، اثر آن تا پایان الگوریتم وجود داشته و در نتیجه منجر به جواب‌های نادرست می‌شود. احتمال بروز این اشتباه به خصوص زمانی که همبستگی (شباهت) بین اتم‌های دیکشنری زیاد است، بالا می‌رود. این امر به دلیل «حرص زیاد» این دسته از الگوریتم‌ها است. یک راه‌حل برای غلبه بر این مشکل، انتخاب چندین اتم در گام انتخاب بهترین اتم‌ها، به عنوان کاندید حضور در نمایش تُنک سیگنال، و سپس استفاده از تعدادی از این اتم‌ها برای به‌روز کردن نمایش سیگنال است. علاوه بر الگوریتم GOMP که در قسمت قبل به آن اشاره شد، الگوریتم CoSaMP یکی دیگر از روش‌هایی است که این کار را انجام می‌دهد [۹۳]. این الگوریتم یک نمایش s -تُنک، یعنی حداکثر با s مؤلفه‌ی غیر صفر، از سیگنال بدست می‌آورد. به عبارت دیگر، مقدار s باید معلوم باشد؛ هر چند همانطور که در [۹۳] پیشنهاد شده است، می‌توان از رابطه‌ی $s \leq n / (2 \log(1 + m/s))$ به عنوان تخمینی از s استفاده کرد. شبه‌کُد این الگوریتم در شکل ۲-۲ آورده شده است.

۲-۴-۲ روش‌های آستانه‌گذاری

در این قسمت، تعدادی از الگوریتم‌های معروف بازیابی نمایش تَنک، که مبتنی بر آستانه‌گذاری هستند، مرور می‌شوند. این الگوریتم‌ها عموماً ساختاری ساده داشته و عملکرد مطلوبی از خود نشان می‌دهند.

۱-۲-۴-۲ الگوریتم IDE

الگوریتم آشکارسازی-تخمین پی‌درپی^۱ (IDE) یکی از اولین الگوریتم‌های آستانه‌گذاری است که سعی دارد تَنک‌ترین جواب معادله‌ی $\mathbf{y} = \mathbf{D}\mathbf{x}$ را با انجام متوالی دو گام آشکارسازی و تخمین زدن پیدا کند [۴، ۵]. در مرحله‌ی آشکارسازی، درایه‌های غیرفعال^۲ جواب، یعنی آنهایی که دامنه‌ی کوچکی (نسبت به یک سطح آستانه‌ی مشخص) دارند، شناسایی می‌شوند. اگر مجموعه‌ی این اندیس‌ها را با \mathcal{I} ، و تخمین مرحله‌ی k از جواب تَنک را با \mathbf{x}^k نشان دهیم، در اینصورت داریم [۵]

$$\mathcal{I} = \{1 \leq i \leq m : g_i(\mathbf{y}, \mathbf{x}^k) < \epsilon_{k+1}\}, \quad (11-2)$$

که در آن، $g_i(\mathbf{y}, \mathbf{x}^k) \triangleq |\mathbf{d}_i^T(\mathbf{y} - \sum_{j \neq i} x_j^k \mathbf{d}_j)|$ تابع آستانه‌گذاری بوده و x_k^i درایه‌ی i ام بردار \mathbf{x}^k را نشان می‌دهد. همچنین، ϵ_{k+1} سطح آستانه‌ی تکرار $(k+1)$ ام را نشان می‌دهد. تابع آستانه‌گذاری را می‌توان به فرم برداری زیر

$$\mathbf{g}(\mathbf{y}, \mathbf{x}^k) = |\mathbf{x}^k + \mathbf{D}^T(\mathbf{D}\mathbf{x}^k - \mathbf{y})| \quad (12-2)$$

هم نوشت [۴]، که در آن، تابع قدرمطلق و تابع g بصورت درایه‌به‌درایه عمل می‌کنند. برای سطوح آستانه، یک دنباله‌ی کاهش‌ی در نظر گرفته می‌شود [۵]. بعد از شناسایی اندیس‌های غیرفعال، مسأله‌ی زیر برای بدست آوردن

تخمین جدید سیگنال حل می‌شود^۳

$$\begin{cases} \mathbf{x}_{\mathcal{I}^c}^{k+1} = \operatorname{argmin}_{\mathbf{x}_{\mathcal{I}^c}} \|\mathbf{y} - \mathbf{D}_{\mathcal{I}^c} \mathbf{x}_{\mathcal{I}^c}\|_2 = (\mathbf{D}_{\mathcal{I}^c}^T \mathbf{D}_{\mathcal{I}^c})^{-1} \mathbf{D}_{\mathcal{I}^c}^T \mathbf{y} & (|\mathcal{I}^c| \leq n) \\ \mathbf{x}_{\mathcal{I}}^{k+1} = \mathbf{o} \end{cases} \quad (13-2)$$

که در آن، \mathcal{I}^c مجموعه‌ی مکمل \mathcal{I} بوده، $\mathbf{x}_{\mathcal{I}^c}^{k+1}$ درایه‌های \mathbf{x}_{k+1} محدود به اندیس‌های موجود در \mathcal{I}^c را نشان می‌دهد، و $\mathbf{D}_{\mathcal{I}^c}$ شامل تنها ستون‌هایی از \mathbf{D} است که اندیس آنها در \mathcal{I}^c وجود دارد. در واقع، با حل مسأله‌ی فوق، سیگنال

^۱Iterative Detection-Estimation

^۲Inactive entries

لازم به ذکر است که روش دیگری نیز برای انجام مرحله‌ی تخمین در [۵] پیشنهاد شده است که تصویر کردن را در حوزه‌ی \mathbf{x} انجام می‌دهد.

y روی زیرفضای ایجاد شده توسط ستون‌های ماتریس D_α تصویر می‌شود [۵]. الگوریتم نهایی عبارت است از انجام متوالی (۲-۱۱) و (۲-۱۳).

۲-۲-۴-۲ الگوریتم‌های IST

نمونه‌ی دیگری از الگوریتم‌های مبتنی بر آستانه‌گذاری، معروف به الگوریتم‌های تکراری آستانه‌گذاری-انقباض^۱

(IST) [۴۱، ۵۳] هستند که سعی در حل مسأله‌ی غیر مقید زیر دارند:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{c} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1, \quad (14-2)$$

که در آن $\lambda > 0$ نشان‌دهنده‌ی مصالحه‌ی بین میزان تَنک بودن و خطای نمایش است. به عبارت دیگر، مقادیر بزرگ‌تر این پارامتر منجر به نمایش‌های تَنک‌تری می‌شود. ایده‌ی کلی این الگوریتم‌ها تبدیل مسأله‌ی فوق به m مسأله‌ی اسکالر و سپس حل این مسائل است. برای این منظور، جمله‌ی زیر به تابع هدف فوق اضافه می‌شود:

$$d(\mathbf{x}, \mathbf{x}_0) = \frac{c}{2} \|\mathbf{x} - \mathbf{x}_0\|_2^2 - \frac{1}{2} \|\mathbf{D}\mathbf{x} - \mathbf{D}\mathbf{x}_0\|_2^2 \quad (15-2)$$

که در آن $c \geq \lambda_{\max}(\mathbf{D}^T \mathbf{D})$ یک عدد ثابت است. نقش بردار \mathbf{x}_0 در ادامه مشخص خواهد شد. در این صورت تابع هدف جدید عبارت است از $\tilde{f}(\mathbf{x}, \mathbf{x}_0) = f(\mathbf{x}) + d(\mathbf{x}, \mathbf{x}_0)$. توجه کنید که هدف از اضافه کردن این جمله، تبدیل تابع هدف به گونه‌ای است که بر حسب مؤلفه‌های بردار \mathbf{x} جداپذیر شود؛ این موضوع به راحتی با ساده کردن تابع $\tilde{f}(\mathbf{x}, \mathbf{x}_0)$ قابل مشاهده است. به علاوه، همانطور که در ادامه می‌بینیم، این جمله معیاری از نزدیکی دو جواب متوالی در حل تکراری مسأله‌ی (۲-۱۴) است [۴۱]. به عبارت دیگر، $\tilde{f}(\mathbf{x}, \mathbf{x}_0) \rightarrow f(\mathbf{x})$ به عبارت دیگر، $\mathbf{x} \rightarrow \mathbf{x}_0$. برای بدست آوردن مینیمم تابع $\tilde{f}(\mathbf{x}, \mathbf{x}_0)$ ، مشتق آن را برابر صفر قرار می‌دهیم و معادله‌ی حاصل را برای \mathbf{x} حل می‌کنیم. توجه به این نکته ضروری است که چون تابع قدرمطلق مشتق‌پذیر نیست، باید از زیر-گرادیان^۲ آن استفاده کرد. به این ترتیب به معادله‌ی زیر می‌رسیم:

$$\mathbf{x} - \mathbf{z}_0 + \frac{\lambda}{c} \text{sgn}(\mathbf{x}) = \mathbf{0}, \quad (16-2)$$

که در آن $\mathbf{z}_0 = \frac{1}{c} \mathbf{D}^T (\mathbf{y} - \mathbf{D}\mathbf{x}_0) + \mathbf{x}_0$ است. در نهایت، جواب نهایی که مینیمم کننده‌ی سراسری تابع $\tilde{f}(\mathbf{x}, \mathbf{x}_0)$ است بصورت زیر بدست می‌آید:

$$\mathbf{x}_{opt} = \mathcal{S}_{\frac{\lambda}{c}}(\mathbf{z}_0), \quad (17-2)$$

^۱ Iterative Shrinkage-Thresholding

^۲ Sub-gradient

که در آن $\mathcal{S}_\lambda(\cdot)$ تابع آستانه‌گذاری نرم^۱ است که مؤلفه به مؤلفه عمل کرده و بصورت زیر تعریف می‌شود:

$$\mathcal{S}_\lambda(a) = \begin{cases} a + \lambda & a \leq -\lambda \\ 0 & |a| \leq \lambda \\ a - \lambda & a \geq \lambda \end{cases} \quad (18-2)$$

تا اینجا، تابع هدف اصلی به تابعی جدید تبدیل شد که برای آن یک مینیمم‌کننده‌ی سراسری به فرم بسته بدست آمد. حال برای حل مسأله‌ی (۲-۱۴)، تابع $f(\mathbf{x})$ به صورت تکراری مینیمم می‌شود؛ طوری که برای بدست آوردن

\mathbf{x}_{i+1} ، تابع $\tilde{f}(\mathbf{x}, \mathbf{x}_0)$ با انتخاب $\mathbf{x}_0 = \mathbf{x}_k$ روی \mathbf{x} مینیمم می‌شود. این عملیات بصورت زیر خلاصه می‌شود:

$$\mathbf{x}_{k+1} = \mathcal{S}_\lambda\left(\frac{1}{c}\mathbf{D}^T(\mathbf{y} - \mathbf{D}\mathbf{x}_k) + \mathbf{x}_k\right) \quad (19-2)$$

ایده‌ی فوق، یعنی استفاده از تابع $\tilde{f}(\mathbf{x}, \mathbf{x}_0)$ برای بهینه‌کردن تابع $f(\mathbf{x})$ ، به روش تابع جایگزین^۲ یا روش Majorization (MM) معروف است. برای جزئیات بیشتر درباره‌ی این الگوریتم به [۴۱، ۵۲] مراجعه کنید. نکته‌ی قابل ذکر دیگر این است که اگر در مسأله‌ی (۲-۱۴)، به جای نرم یک از نرم صفر استفاده کرده و مراحل فوق را تکرار کنیم، به الگوریتم آستانه‌گذاری سخت پیاپی^۳ (IHT) [۱۸] می‌رسیم که تابع آستانه‌گذاری سخت بصورت زیر تعریف می‌شود:

$$H_\lambda(x) = \begin{cases} x & |x| \geq \sqrt{2\lambda} \\ 0 & |x| < \sqrt{2\lambda} \end{cases} \quad (20-2)$$

لازم به ذکر است که اگر $\lambda_{\max}(\mathbf{D}^T\mathbf{D}) = 1$ باشد، با انتخاب $c = 1$ برای الگوریتم IHT خواهیم داشت

$$\mathbf{x}_{k+1} = H_\lambda(\mathbf{D}^T(\mathbf{y} - \mathbf{D}\mathbf{x}_k) + \mathbf{x}_k). \quad (21-2)$$

با مقایسه‌ی رابطه‌ی فوق با تابع آستانه‌گذاری (۲-۱۲) می‌توان دید که رابطه‌ای که در IHT آستانه‌گذاری می‌شود همانی است که در الگوریتم IDE تابع آستانه بر روی آن اعمال می‌شود. با این حال، رهیافت این دو الگوریتم برای به‌روزرسانی تخمین متفاوت است.

۳-۲-۴-۲ الگوریتم IMAT

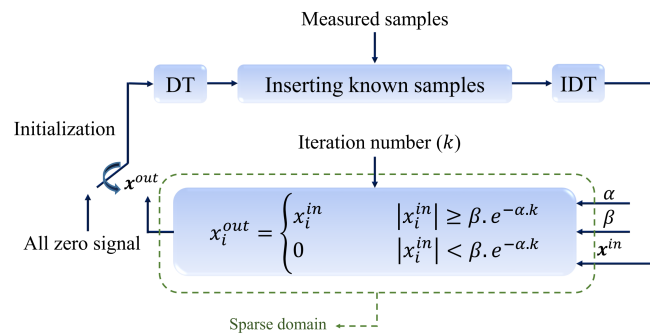
الگوریتم دیگری که در دسته‌ی الگوریتم‌های آستانه‌گذاری قرار می‌گیرد، معروف به الگوریتم تکراری با آستانه‌گذاری وفقی^۴، یا به اختصار IMAT، است [۹، ۸۹، ۸۷، ۸۸]. این الگوریتم، که نمودار بلوکی آن در شکل ۲-۳ رسم شده است، برای بازسازی سیگنال‌ها از روی نمونه‌های تصادفی یا فشرده استفاده می‌شود. اساس الگوریتم IMAT به این

^۱Soft-thresholding function

^۲Surrogate function

^۳Iterative hard-thresholding

^۴Iterative Method with Adaptive Thresholding (IMAT)



شکل ۲-۳: نمودار بلوکی الگوریتم IMAT [۸۸]. بلوک‌های DT و IDT، به ترتیب، نشان‌دهنده‌ی یک تبدیل و عکس آن هستند.

ترتیب است که یک تخمین اولیه از جواب را بصورت تکراری، با رفتن به یک حوزه‌ی تُنک و حوزه‌ی اطلاعات، به‌روز می‌کند. در حوزه‌ی تُنک، با اعمال یک تبدیل تُنک‌کننده مانند تبدیل DCT برای تصاویر طبیعی، و در ادامه، آستانه‌گذاری سخت روی ضرایب تبدیل، عملیات تُنک‌سازی انجام می‌شود. سپس، با استفاده از عکس تبدیل مورد استفاده، دوباره به حوزه‌ی اصلی برگشته و برای مطابقت با اطلاعات موجود، سیگنال تبدیل یافته روی مجموعه‌ی مجاز اطلاعات تصویر می‌شود. برای تُنک‌سازی ضرایب تبدیل، پارامتر آستانه به صورت نمایی در طول تکرارهای الگوریتم کم می‌شود. این ایده، یعنی کاهش سطح آستانه، قبلاً در الگوریتم IDE نیز استفاده شده است. بعلاوه، آستانه‌گذاری IMAT مشابه الگوریتم IHT بر مبنای آستانه‌گذاری سخت، یعنی رابطه‌ی (۲-۲۰)، است. الگوریتم IMAT عملکرد بسیار خوبی، هم از نظر کیفیت بازسازی و هم از نظر پیچیدگی محاسباتی دارد.

۲-۴-۳ روش‌های تقریب نُرم صفر

دسته‌ی سوم الگوریتم‌ها سعی در حل مستقیم مسأله‌ی P دارند [۸۴، ۹۲]. ایده‌ی کلی این روش‌ها تقریب تابع ناهموار و مشتق‌ناپذیر نُرم صفر با یک تابع هموار و مشتق‌پذیر است. یکی از معروف‌ترین این الگوریتم‌ها، الگوریتم ^1SLO [۹۲] است. این الگوریتم به عنوان تابع تشویق‌کننده‌ی تُنکی از تابع زیر استفاده می‌کند:

$$J(\mathbf{x}; \sigma) = \sum_i (1 - f_\sigma(x_k)) = \sum_i (1 - \exp(-\frac{x_k^2}{2\sigma^2})), \quad (2-22)$$

که در آن، $\sigma > 0$ یک پارامتر تنظیم است. توجه کنید که این تابع، تقریبی هموار و مشتق‌پذیر از تابع شبه نُرم صفر است. همچنین، داریم $\|\mathbf{x}\|_0 = \lim_{\sigma \rightarrow 0} J(\mathbf{x}; \sigma)$. با این توضیح، بازای σ های خیلی کوچک تقریب خوبی از شبه

¹Smoothed L0

نُرم صفر بدست می‌آید که مشتق‌پذیر بوده و براحتی می‌توان مسأله‌ی بهینه‌سازی متناظر را با الگوریتم‌های مبتنی بر گرادینان، مثل الگوریتم ^۱SD حل کرد. مشکلی که وجود دارد این است که تابع $J(\mathbf{x}; \sigma)$ در این حالت تعداد زیادی مینیمم محلی داشته، امکان توقف الگوریتم در این مینیمم‌های عموماً نامطلوب زیاد است. راه‌حلی که در [۹۲] پیشنهاد شده است، استفاده از یک دنباله‌ی کاهشی از مقادیر σ و حل مسائل متناظر به صورت تکراری و بعلاوه، شروع هر مسأله با استفاده از جواب نهائی مسأله‌ی قبلی است (که به آن شروع گرم ^۲ گفته می‌شود [۵۷]). این ایده به «عدم تحدب تدریجی» یا GNC^۳ معروف است [۱۷]. در این وضعیت اگرچه احتمال توقف الگوریتم در یک مینیمم محلی نامطلوب کاسته می‌شود؛ ولی مسأله‌ی کلی هم‌چنان غیرمحدب بوده و لذا تضمینی به همگرایی به یک جواب حداقل‌کننده‌ی سراسری وجود ندارد. با این حال، همانطور که در [۹۲] نشان داده شده است، این الگوریتم سرعت و دقت نسبتاً بالایی دارد.

۵-۲ مقدمه‌ای بر حسگری فشرده

حسگری فشرده^۴ روشی نوین برای ضبط و نمونه‌برداری از داده‌ها است که نخستین بار در [۲۵، ۴۳] معرفی شد. روش‌های کلاسیک نمونه‌برداری مبتنی بر نظریه‌ی شانون-نایکوئیست^۵ است، که بیان می‌کند برای بازسازی کامل یک سیگنال، نرخ نمونه‌برداری باید حداقل به اندازه‌ی دو برابر بزرگترین فرکانس موجود در آن باشد. اساس حسگری فشرده بر این نکته استوار است که بسیاری از سیگنال‌ها از جمله تصویر، صوت و سیگنال‌های زلزله در یک پایه‌ی معین (یا یک دیکشنری)، تُنک یا فشرده‌پذیر^۶ هستند. بر این اساس، حسگری فشرده بیان می‌کند که نرخ اطلاعات (میزان اطلاعات موجود در واحد زمان) یک سیگنال پیوسته می‌تواند خیلی کمتر از پهنای باند آن باشد، یا تعداد درجات آزادی یک سیگنال گسسته خیلی کمتر از طول (محدود) آن است.

ابزارهای متداول ضبط (و انتقال) داده، مانند دوربین عکاسی، به این ترتیب عمل می‌کنند که ابتدا حجم زیادی از داده (طبق تئوری شانون-نایکوئیست) را ضبط کرده، سپس به کمک یک روش فشرده‌سازی، مانند روش‌های

^۱Steepest Descent

^۲Warm start

^۳Graduated Non-Convexity

^۴Compressed Sensing

^۵Shannon-Nyquist

^۶به این معنی که دامنه‌ی ضرایب آن‌ها در آن پایه به طور نمایی کاهش می‌یابد.

کُدینگ حوزه‌ی تبدیل^۱، حجم داده را کاهش می‌دهند. به عبارت بهتر، ابتدا داده‌ای با طول زیاد ضبط می‌شود، سپس به حوزه‌ای دیگر (مثلاً با استفاده از تبدیل ویولت) منتقل شده، در آنجا روی ضرایب آستانه گذاری انجام می‌شود، و نهایتاً ضرایب باقی مانده به همراه موقعیت آن‌ها کُد می‌شوند. این روش اما به دلایل زیر ناکارآمد است [۱۳]. اولاً، تعداد نمونه‌های اولیه‌ی سیگنال که ضبط می‌شود در بسیاری از کاربردها، مانند تصویربرداری، بسیار زیاد است که خود منجر به هزینه‌ی زیاد به دلیل استفاده از حسگرهای ظریف‌تر می‌شود. ثانیاً، باید همه‌ی ضرایب تبدیل برای کُد کردن محاسبه شوند؛ اگرچه نهایتاً تنها تعداد کمی از آن‌ها نگه داشته می‌شوند. همچنین، علاوه بر مقدار ضرایب، موقعیت آن‌ها نیز باید کُد شود. حسگری فشرده برای غلبه بر این مشکلات، یک سیستم اندازه‌گیری (یا ضبط داده) معرفی می‌کند که همزمان عمل فشرده‌سازی را نیز انجام می‌دهد؛ به عبارت دیگر، تنها «اطلاعات مفید» موجود در سیگنال را اندازه می‌گیرد. حسگری فشرده در بسیاری از کاربردها از جمله تصویربرداری پزشکی، به خصوص روش MRI^۲ [۸۱]، نقشی اساسی ایفا می‌کند.

سیگنال $\mathbf{x} \in \mathbb{R}^m$ را در نظر بگیرید که در پایه‌ی $\Psi = \{\psi_i\}_{i=1}^m$ دارای نمایش s -تُنک است؛ یعنی $\mathbf{x} = \Psi \mathbf{s}$ و به علاوه، تنها $m \ll n$ ضریب s غیر صفر است. حسگری فشرده برای ضبط سیگنال \mathbf{x} ، تعداد $n \ll m$ اندازه‌گیری از این سیگنال را ضبط می‌کند. هر اندازه‌گیری با استفاده از ضرب داخلی سیگنال با یک بردار مشخص از ستون‌های $\Phi = \{\phi_i\}_{i=1}^n$ بدست می‌آید، طوریکه برای اندازه‌گیری i ام داریم $y_i = \langle \mathbf{x}, \phi_i \rangle$. این عملیات به فرم ماتریسی زیر است:

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi \mathbf{s} = \Theta \mathbf{s}. \quad (۲۳-۲)$$

ماتریس اندازه‌گیری Φ باید به گونه‌ای طراحی شود که قادر باشد اطلاعات مفید سیگنال را ضبط کند و به علاوه، به عنوان یکی از ملزومات حسگری فشرده، باید نسبت به ماتریس Ψ ناهم‌بسته باشد. به بیان دیگر، حداکثر ضریب همبستگی ستون‌های دو ماتریس Φ و Ψ باید به اندازه‌ی کافی کوچک باشد. یک دسته از این ماتریس‌ها، که با دقت خوبی شرایط بازسازی سیگنال را فراهم می‌کنند، ماتریس‌های تصادفی است. به عبارت دیگر، ما برای ضبط سیگنال یک سری اندازه‌گیری تصادفی از آن را ضبط می‌کنیم. در نهایت، برای بازسازی سیگنال از روی اندازه‌گیری‌های آن، لازم است یک مسأله‌ی بازسازی نمایش تُنک حل شود.

^۱Transform Coding

^۲Magnetic Resonance Imaging

۶-۲ جمع‌بندی

در این فصل مباحث تئوری نمایش تُنک را به همراه تعدادی از الگوریتم‌های موجود برای بازیابی این نمایش، به اختصار مرور کردیم. گفتیم که هسته‌ی مرکزی بحث نمایش تُنک، یک دستگاه معادلات خطی فرومعیّن است که در حالت کلی بیشمار جواب دارد، اما تُنک‌ترین جواب آن تحت شرایطی یکتا است. تعدادی از الگوریتم‌های موجود را، که در حالت کلی شامل دسته‌ی الگوریتم‌های حریص، روش‌های مبتنی بر آستانه‌گذاری و روش‌های تقریب نُرم صفر است، به اختصار مرور کردیم. در انتها، حسگری فشرده را به طور مختصر معرفی کردیم. دیدیم که طبق این رویکرد، می‌توان به نرخ فشرده‌سازی کمتر از نرخ نایکوئیست دست پیدا کرد. هزینه‌ی این کار اما نیاز به حلّ یک مسأله‌ی بازیابی نمایش تُنک است.

در فصل بعد، بحث «آموزش دیکشنری» را برای پردازش تُنک سیگنال‌ها مرور می‌کنیم. همچنین، تعدادی از الگوریتم‌هایی را که تاکنون معرفی شده‌اند به اختصار مرور خواهیم کرد. بعلاوه، روش‌های موجود برای پردازش داده‌های بُعد بالا را نیز مرور می‌کنیم.

یادگیری دیکشنری و پردازش داده‌های بُعد بالا

۱-۳ مقدمه

در فصل قبل مباحث پایه‌ای نمایش تُنک را به همراه تعدادی از الگوریتم‌های موجود به طور مختصر مرور کردیم. دیدیم که هدف، تقریب زدن یک سیگنال به صورت یک ترکیب خطی متشکل از کم‌ترین تعداد اتم‌ها از یک دیکشنری فوق‌کامل^۱ مشخص است. فرض ما تا این‌جا بر این بود که این دیکشنری کاملاً معلوم است. اما لازم است که برای هر کاربردی مانند فشرده‌سازی، نوپزدائی و تشخیص الگو^۲، یک دیکشنری مناسب، یا اصطلاحاً تُنک‌کننده^۳، داشته باشیم. تمرکز این فصل روی طراحی دیکشنری برای یک کلاس مشخص از داده‌ها است. ابتدا مروری مختصر خواهیم داشت بر تاریخچه‌ی دیکشنری‌ها یا همان تبدیل‌ها. سپس مسأله‌ی آموزش دیکشنری را معرفی کرده و در ادامه، تعدادی از الگوریتم‌های معرفی شده برای آموزش دیکشنری را مرور خواهیم کرد. در انتها، تعدادی از روش‌های معروف مورد استفاده برای پردازش داده‌های با ابعاد بالا را نیز مرور خواهیم کرد.

۲-۳ یادگیری دیکشنری

نمایش سیگنال در یک پایه‌ی مناسب از دیرباز مورد توجه بوده است [۱۱۸]. برای این منظور، سیگنال بر حسب تعدادی سیگنال پایه یا اتم بسط داده می‌شود. هر کدام از این اتم‌ها در حقیقت یک ویژگی از سیگنال را توصیف

^۱Overcomplete

^۲Pattern recognition

^۳Sparsifying

می‌کنند. در بسیاری از کاربردها لازم است که «ساده‌ترین نمایش» را برای یک سیگنال داشته باشیم. همانطور که در فصل قبل گفتیم، بسیاری از سیگنال‌های (گسسته‌ی) طبیعی اطلاعاتی به مراتب کمتر از طولشان دارند. به عبارت دیگر، «بُعد ذاتی» آن‌ها از «بُعد ظاهری» آن‌ها خیلی کوچکتر است. بنابراین، اتم‌های مورد استفاده برای نمایش یک سیگنال باید ویژگی‌های مهم و به طور کلی اطلاعات مفید آن را استخراج کنند. بعنوان مثال، در استخراج ویژگی^۱ برای تشخیص الگو، نمایش سیگنال باید ویژگی‌های برجسته‌ی آن را توصیف کند.

در ساده‌ترین حالت، یک دیکشنری عبارت است از یک پایه‌ی متعامد. در این حالت، ضرایب بسط یا نمایش سیگنال به سادگی با استفاده از ضرب داخلی آن با اتم‌ها بدست می‌آید [۵۲، ۱۰۰]. در کلی‌ترین حالت یک «پایه»، که شامل اتم‌های مستقل خطی ولی نه لزوماً متعامد است، ضرایب نمایش سیگنال از ضرب داخلی آن با اتم‌های دوگان^۲ بدست می‌آید. از معروف‌ترین این تبدیل‌ها می‌توان به تبدیل فوریه^۳ اشاره کرد. در حالت گسسته، که موسوم به DFT^۴ است، اتم‌های این تبدیل عبارتند از $\{\mathbf{d}_k \in \mathbb{R}^n : \mathbf{d}_k(\ell) = \exp(-j2\pi k(\ell - 1)/n), \ell = 1, 2, \dots, n\}_{k=1}^n$. هر اتم DFT عبارت است از یک سینوسی مختلط با یک فرکانس منحصر به فرد و بعلاوه، اتم‌های متمایز بر هم عمودند. به این ترتیب، این اتم‌ها محتوای فرکانسی یک سیگنال را توصیف می‌کنند. این تبدیل و البته تمامی تبدیل‌های به اصطلاح «کامل»^۵، یعنی تبدیل‌هایی که در آن تعداد اتم‌ها با بُعد سیگنال برابر است، تنها برای سیگنال‌هایی مشخص مناسب‌اند. به عبارت دیگر، توانایی ارائه‌ی نمایش تُنک برای این تبدیل‌ها محدود است [۱۰۰].

برای غلبه بر این مشکل، دیکشنری‌های «فوقِ کامل» معرفی شدند. یک دسته‌ی مهم از این تبدیل‌ها در جامعه‌ی پردازش سیگنال به «فریم»^۶ معروف‌اند [۷۴]. در این دیکشنری‌ها، تعداد اتم‌ها از بُعد سیگنال بیشتر است و بنابراین قادراند ویژگی‌های بیشتری از سیگنال را توصیف کنند. واضح است که در این وضعیت، اتم‌ها وابسته‌ی خطی‌اند. به عنوان یک تعریف دقیق‌تر، گوئیم خانواده‌ی اتم‌های $\mathbf{D} = \{\mathbf{d}_i\}_{i=1}^m$ تشکیل یک فریم برای فضای \mathbb{R}^n می‌دهند، هر گاه دو عدد ثابت $0 < \alpha \leq \beta$ وجود داشته باشند بطوری که:

^۱ Feature Extraction

^۲ Dual

^۳ Fourier transform

^۴ Discrete Fourier Transform

^۵ Complete

^۶ Frame

$$\forall \mathbf{y} \in \mathbb{R}^n : \alpha \|\mathbf{y}\|_2^2 \leq \sum_{i=1}^m |\langle \mathbf{d}_i, \mathbf{y} \rangle|^2 \leq \beta \|\mathbf{y}\|_2^2, \quad (1-3)$$

که در آن، $\langle \cdot, \cdot \rangle$ ضرب داخلی دو بردار را نشان می‌دهد. همانطور که پایه‌های متعامد (یا متعامد یگه؛ که علاوه بر تعامد اتم‌ها، نُرم همگی برابر یک است) ساده‌ترین محاسبات را در بین تبدیل‌های کامل دارند، در بین فریم‌ها نیز چنین تبدیل‌هایی به نام Tight Frame وجود دارد که ساده‌ترین محاسبات را دارند. برای این دسته از فریم‌ها داریم $\alpha = \beta$. به عبارت دیگر، $\sum_{i=1}^m |\langle \mathbf{d}_i, \mathbf{y} \rangle|^2 = \alpha \|\mathbf{y}\|_2^2$. در نتیجه، ضرایب نمایش یک سیگنال در یک Tight Frame به سادگی از ضرب داخلی سیگنال با هر یک از اتم‌ها بدست می‌آید؛ به عبارت دیگر $\mathbf{y} = \frac{1}{\alpha} \sum_{i=1}^m \langle \mathbf{d}_i, \mathbf{y} \rangle \mathbf{d}_i$. لازم به ذکر است که این نمایش کمترین نُرم دو را بین تمام نمایش‌های ممکن داشته و یکتا است [۷۴]. به عنوان مثالی از یک Tight Frame می‌توان به تبدیل DCT فوق کامل اشاره کرد.

تبدیل‌هایی که تا این‌جا بررسی کردیم، ثابت بوده و به عبارتی وابسته به سیگنال نیستند. به این تبدیل‌ها، «دیکشنری‌های از پیش تعریف شده»^۱ یا «دیکشنری‌های غیروفقی»^۲ می‌گویند. اگرچه فریم‌ها نسبت به تبدیل‌های کامل قابلیت بیش‌تری در نمایش تُنک سیگنال‌ها دارند؛ اما با این وجود، قابلیت وفق کردن به محتویات سیگنال‌های تحت بررسی را ندارند. به عنوان مثال، اگرچه یک تبدیل DCT فوق کامل نسبت به یک تبدیل DCT کامل نمایش تُنک‌تری برای یک تصویر بدست می‌دهد، اما برای بسیاری از تصاویر نمی‌تواند نمایش به اندازه‌ی کافی تُنکی داشته باشد. بعنوان یک نمونه، این تبدیل برای تصاویری که شامل فقط بافت^۳ هستند، نمایش تُنکی دارد، ولی برای تصاویر دارای ساختارهای پیچیده‌تر مانند انواع لبه‌ها، نمایش مناسبی ندارد. یک راه برای بدست آوردن دیکشنری‌های فوق کامل ثابت کارآتر، ترکیب پایه‌های مختلف است. هر پایه یک یا چند ویژگی از سیگنال را توصیف می‌کند. بعنوان مثال، برای سیگنال‌های همواری که شامل تعدادی نقطه‌ی یگه یا تکینی^۴ هستند، می‌توانیم دیکشنری‌های DFT و دلتای دیراک را (که همان ماتریس همانی است) با هم ترکیب کنیم. این ایده نخستین بار در [۳۴] مطرح شد. بعنوان یک راه حل بهتر، بحث «آموزش دیکشنری» طی دهه‌ی اخیر مورد توجه قرار گرفته است. در این رهیافت، ابتدا تعدادی «داده‌ی آموزشی»^۵ شبیه به سیگنال‌های تحت بررسی انتخاب می‌شود. سپس طی یک الگوریتم آموزش، اتم‌های دیکشنری طوری بهینه می‌شوند که تا حد امکان، نمایان‌گر برجسته‌ترین ویژگی‌های

^۱Predefined Dictionaries^۲Non-Adaptive Dictionaries^۳Texture^۴Singularity^۵Training data

(مشترک) داده‌های آموزشی بوده و در نتیجه، نمایشی به اندازه‌ی کافی تُنک برای سیگنال‌های آموزشی ارائه دهند.

رابطه‌ای نزدیک بین آموزش دیکشنری برای نمایش تُنک و کمی‌سازی بُرداری^۱ [۶۲] وجود دارد. در کمی‌سازی بُرداری، هدف طراحی یک کتاب کُد^۲ شامل m بُردار کُد^۳، $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m]$ ، به گونه‌ای است که هر یک از داده‌های آموزشی، یعنی ستون‌های ماتریس $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L]$ که $\mathbf{y}_i \in \mathbb{R}^n$ ، برحسب دقیقاً یکی از بُردارهای کُد، که نزدیک‌ترین فاصله را (عموماً بر مبنای نُرم ℓ_2) با آن دارد، قابل نمایش باشد. این نمایش در حقیقت حالت حدی نمایش تُنک است. بعبارت دیگر، در حالی که در نمایش تُنک هر داده می‌تواند از بیش از یک اتم برای توصیف خود استفاده کند، در کمی‌سازی بُرداری، هر داده تنها از یک بُردار کُد (در اینجا همان اتم) استفاده می‌کند و بعلاوه، ضریب نمایش نیز برابر با ۱ است. در نتیجه، در این وضعیت تُنک‌ترین نمایش ممکن را خواهیم داشت. برای یک کتاب کُد معلوم \mathbf{C} ، هر کدام از داده‌ها با بُردار کُدی نمایش داده می‌شود که بیش‌ترین شباهت را نسبت به بقیه‌ی اتم‌ها با آن دارد. در اینصورت اگر بُردار کُد متناظر با نمونه‌ی \mathbf{y}_i برابر با \mathbf{c}_j باشد، می‌توان نوشت $\mathbf{y}_i = \mathbf{C}\mathbf{x}_i$ ، که در آن $\mathbf{x}_i = \mathbf{e}_j$ و ستون j ‌ام ماتریس همانی $\mathbf{I}_{m \times m}$ است. اگر خطای این نمایش

$$E = \sum_{i=1}^L e_i = \|\mathbf{Y} - \mathbf{C}\mathbf{X}\|_F^2 \quad (2-3)$$

را بصورت $e_i = \|\mathbf{y}_i - \mathbf{C}\mathbf{x}_i\|_2^2$ نشان دهیم، خطای کلی برابر است با:

به این ترتیب، هدف کمی‌سازی بُرداری یافتن یک کتاب کُد به گونه‌ای است که خطای فوق کمینه شود. به بیان دقیق‌تر، مسأله‌ی زیر حل می‌شود:

$$\min_{\mathbf{C}, \mathbf{X}} \|\mathbf{Y} - \mathbf{C}\mathbf{X}\|_F \quad \text{subject to} \quad \mathbf{x}_i = \mathbf{e}_j, \quad i = 1, \dots, L. \quad (3-3)$$

با توجه به اینکه مسأله‌ی فوق نسبت به دو متغیر \mathbf{C} و \mathbf{X} توأمأً محدب نیست، بنابراین در حالت کلی تعداد زیادی کمینه‌ساز محلی دارد. یک الگوریتم مهم برای طراحی کتاب کُد، الگوریتم K-means است [۶۲]. این الگوریتم با شروع از یک کتاب کُد اولیه، دو گام را به صورت تکراری انجام می‌دهد. گام نخست، «خوشه‌بندی»^۴ یا «کُدینگ تُنک»^۵ است. در این گام، با استفاده از کتاب کُد فعلی، هر داده به بُردار کُد مناسب خود تخصیص داده می‌شود. در نتیجه تعدادی خوشه متناظر با هر بُردار کُد بدست می‌آید. گام دوم، «به روز کردن بُردارهای کُد» است. در این

^۱ Vector Quantization

^۲ Code Book

^۳ Code Vector

^۴ Clustering

^۵ Sparse Coding

گام، هر بُردار کُد با استفاده از میانگین داده‌های موجود در خوشه‌ی خود به روز می‌شود. این دو گام تا رسیدن به یک خطای قابل قبول تکرار می‌شود. تضمینی وجود ندارد که این الگوریتم به کمینه‌ساز سراسری^۱ مسأله‌ی (۳-۳) همگرا شود، ولی بعد از هر تکرار، خطای کلی یا کم می‌شود و یا ثابت می‌ماند [۶۲].

با توضیحات فوق، طبیعی به نظر می‌رسد که آموزش دیکشنری برای نمایش تُنک در حقیقت تعمیمی از کمی‌سازی بُرداری و الگوریتم‌های آموزش دیکشنری هم تعمیمی از الگوریتم K-means است. بعبارت دیگر، قید سنگین وجود تنها یک اتم در نمایش سیگنال‌ها، تبدیل به «نمایش تُنک» می‌شود (یعنی استفاده‌ی از کم‌ترین تعداد اتم‌ها؛ نه الزاماً یکی). در نتیجه، مسأله‌ی (۳-۳) به مسأله‌ی کلی‌تر زیر تبدیل می‌شود:

$$\min_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T_0, \quad i = 1, \dots, L, \quad (4-3)$$

که T_0 برابر است با حداکثر تعداد مجاز اتم‌ها برای نمایش هر سیگنال آموزشی. همچنین، اکثر الگوریتم‌های آموزش دیکشنری مبتنی بر ماهیت تکراری الگوریتم K-means و دو گام موجود در آن هستند. در گام اول، نمایش تُنک سیگنال‌ها در دیکشنری فعلی محاسبه می‌شود. در گام دوم، دیکشنری طوری به‌روز می‌شود که خطای کلی نمایش تُنک سیگنال‌ها در گام اول، یا شود یا لااقل کاهش یابد. اکثر الگوریتم‌هایی که طی سال‌های گذشته برای آموزش دیکشنری معرفی شده‌اند در نحوه‌ی انجام این دو گام اختلاف دارند [۱۰۰].

۳-۳ الگوریتم‌ها

تاکنون، روش‌های زیادی برای یادگیری دیکشنری معرفی شده است. برای یک مرور جامع، مرجع [۴۹] را ببینید. در این بخش، دو نمونه از معروف‌ترین الگوریتم‌های آموزش دیکشنری را که تاکنون معرفی شده‌اند، به اختصار مرور می‌کنیم. همانطور که پیشتر توضیح دادیم، اساس این الگوریتم‌ها بر مبنای تکرار دو گام «نمایش تُنک» و «به‌روز کردن دیکشنری» است. تفاوت بین این الگوریتم‌ها در روشی است که هر یک برای محاسبه‌ی نمایش تُنک سیگنال‌ها و از آن مهم‌تر، برای به‌روز کردن دیکشنری استفاده می‌کند. بعبارت دیگر، تفاوت این الگوریتم‌ها بیش‌تر در گام به‌روز کردن دیکشنری است.

^۱Global Minimum

۱-۳-۳ الگوریتم MOD

الگوریتم جهت‌های بهینه یا به اختصار MOD^۱ یکی از ابتدائی‌ترین و البته ساده‌ترین الگوریتم‌های آموزش دیکشنری است [۵۴]. در گام بدست آوردن نمایش تُنک، از هر الگوریتمی می‌توان استفاده کرد (در [۵۴] از الگوریتم OMP استفاده شده است). در گام دوم و با استفاده از ماتریس ضرایب که در گام نخست بدست آمده است، اتم‌ها در جهتی تغییر داده می‌شوند، که خطای کلی نمایش یعنی، $E = \|Y - DX\|_F$ ، حداقل شود. این کار معادل با حلّ مسأله‌ی (۳-۴) با فرض X ثابت است. برای این منظور، کافی است مشتق این عبارت را نسبت به D برابر با صفر قرار دهیم. در اینصورت بدست می‌آوریم $(Y - DX)X^T = 0$. نهایتاً برای دیکشنری به‌روز شده داریم:

$$D = YX^T(XX^T)^{-1} = YX^\dagger. \quad (۵-۳)$$

دقت کنید که بعد از به‌روز کردن دیکشنری لازم است ستون‌های آن تُرمالیزه شود. ممکن است طی پیشروی الگوریتم در برخی تکرارها ماتریس XX^T بد حالت^۲ شود. این وضعیت می‌تواند اثر سوئی روی دیکشنری به‌روز شده داشته باشد. برای رفع این مشکل، ماتریس λI به XX^T اضافه می‌شود، که در آن λ یک عدد مثبت کوچک است. در اینصورت داریم:

$$D = YX^T(XX^T + \lambda I)^{-1}. \quad (۶-۳)$$

به راحتی می‌توان نشان داد که دیکشنری فوق در واقع جواب مسأله‌ی زیر است [۵۲]:

$$\min_D \|Y - DX\|_F + \lambda \|D\|_F. \quad (۷-۳)$$

۲-۳-۳ الگوریتم KSVD

الگوریتم K-SVD [۳] یکی از موفق‌ترین الگوریتم‌های آموزشی دیکشنری بوده که نسبت به بقیه‌ی الگوریتم‌ها رابطه‌ای نزدیک‌تر با الگوریتم K-means دارد. K-SVD در گام نمایش تُنک از الگوریتم OMP استفاده می‌کند. انتخاب این الگوریتم به دلیل سرعت بالای آن نسبت به بقیه‌ی الگوریتم‌های کُدینگ تُنک است؛ اگرچه به دلیل ماهیت حریص آن ممکن است جواب‌های نادرستی داشته باشد.

برای به‌روز کردن دیکشنری، بر خلاف MOD که کل اتم‌ها را یک‌جا به‌روز می‌کند، K-SVD اتم‌ها را یک‌به‌یک و به‌طور متوالی به‌روز می‌کند. به یاد بیاورید که در K-means هم، بردارهای کُد یک‌به‌یک به‌روز می‌شوند.

^۱Method of Optimal Directions

^۲Ill-conditioned

برای به‌روز کردن اتم \mathbf{d}_k ، بقیه‌ی اتم‌ها را ثابت می‌گیریم. در اینصورت داریم:

$$E = \|\mathbf{Y} - \mathbf{DX}\|_F^2 = \|\mathbf{Y} - \sum_{i=1}^m \mathbf{d}_i \mathbf{x}_T^i\|_F^2 = \left\| \left(\mathbf{Y} - \sum_{i \neq k}^m \mathbf{d}_i \mathbf{x}_T^i \right) - \mathbf{d}_k \mathbf{x}_T^k \right\|_F^2 = \|\mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k\|_F^2, \quad (8-3)$$

که در آن منظور از \mathbf{x}_T^i سطر i ام ماتریس \mathbf{X} است. در نتیجه، برای به‌روز کردن اتم \mathbf{d}_k باید مسأله‌ی زیر را حل کنیم:

$$\min_{\mathbf{d}_k} \|\mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k\|_F^2 \quad \text{subject to} \quad \|\mathbf{d}_k\|_2 = 1. \quad (9-3)$$

با اندکی محاسبات، جواب این مسأله به صورت زیر بدست می‌آید:

$$\mathbf{d}_k \leftarrow \mathbf{E}_k (\mathbf{x}_T^k)^T / \|\mathbf{E}_k (\mathbf{x}_T^k)^T\|_2. \quad (10-3)$$

ایده‌ی K-SVD اما فراتر از این است. K-SVD علاوه بر هر اتم، سطر متناظر آن در ماتریس ضرایب را نیز به‌روز می‌کند؛ و بعلاوه، در به‌روز کردن هر اتم تنها سیگنال‌هایی به کار می‌روند که قبلاً (در گام اول) از این اتم در نمایش خود استفاده کرده‌اند. برای توضیح بیشتر، اولاً دقت کنید که اگر در محاسبه‌ی ماتریس \mathbf{E}_k ، از مقادیر به‌روز شده‌ی \mathbf{d}_i و \mathbf{x}_T^i ها استفاده کنیم، سرعت همگرایی الگوریتم بیش‌تر خواهد شد. به عنوان مثال، بعد از به‌روز کردن \mathbf{d}_1 و \mathbf{x}_T^1 ، برای به‌روز کردن اتم دوم و سطر متناظر آن، از مقادیر به‌روز شده‌ی \mathbf{d}_1 و \mathbf{x}_T^1 در محاسبه‌ی \mathbf{E}_2 استفاده کنیم و به همین ترتیب برای بقیه اتم‌ها و سطر متناظرشان. مسأله‌ی به‌روز کردن همزمان اتم \mathbf{d}_k و \mathbf{x}_T^k به صورت زیر است:

$$\min_{\mathbf{d}_k, \mathbf{x}_T^k} \|\mathbf{E}_k - \mathbf{d}_k \mathbf{x}_T^k\|_F^2. \quad (11-3)$$

مسأله‌ی فوق در واقع تقریب رتبه-۱ ماتریس \mathbf{E}_k است. به طور کلی، اگر تجزیه به مقادیر تکین^۱ (SVD) ماتریس دلخواه $\mathbf{A} \in \mathbb{R}^{n \times m}$ بصورت $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ باشد، تقریب رتبه- r این ماتریس که $r \leq \min(n, m)$ بصورت $\mathbf{A}_r = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ است [۹۱].

مشکلی که در اینجا بوجود می‌آید این است که طی این فرآیند به احتمال زیاد تمام درایه‌های بُردار \mathbf{x}_T^k پُر خواهد شد. به همین دلیل تنها از سیگنال‌هایی استفاده می‌شود که در گام قبلی از اتم \mathbf{d}_k در نمایش خود استفاده کرده‌اند (به شباهت این کار با به‌روز کردن بُردارهای کُد در K-means دقت کنید). به این ترتیب، تنها درایه‌های غیر صفر بُردار \mathbf{x}_T^k به‌روز شده و بقیه‌ی درایه‌های آن صفر می‌مانند. اگر تعریف کنیم $\omega_k = \{i : 1 \leq i \leq m, \mathbf{x}_T^k(i) \neq 0\}$ ، در نهایت مسأله‌ی زیر را برای به‌روز کردن اتم \mathbf{d}_k و درایه‌های غیر صفر سطر

^۱Singular Value Decomposition

داریم: \mathbf{x}_T^k

$$\min_{\mathbf{d}_k, \mathbf{x}_T^k} \|\mathbf{E}_k^{\omega_k} - \mathbf{d}_k \mathbf{x}_T^k\|_F^2 \quad \text{subject to} \quad \|\mathbf{d}_k\|_2 = 1, \quad (12-3)$$

که در آن $\mathbf{E}_k^{\omega_k}$ شامل تنها ستون‌های متناظر با ω_k از ماتریس \mathbf{E}_k و \mathbf{x}_T^k نیز برداری به طول $|\omega_k|$ است. اگر تجزیه به مقادیر تکین $\mathbf{E}_k^{\omega_k}$ بصورت $\mathbf{E}_k^{\omega_k} = \mathbf{U}\Sigma\mathbf{V}^T$ باشد، داریم:

$$\mathbf{d}_k \leftarrow \mathbf{u}_1, \quad \mathbf{x}_T^k(\omega_k) \leftarrow \sigma_1 \mathbf{v}_1, \quad (13-3)$$

که σ_1 بزرگترین مقدار تکین است. توجه کنید که چون نُرم ستون‌های \mathbf{U} واحد است، بنابراین نیازی به نرمالیزه کردن \mathbf{d}_k نیست. همه‌ی اتم‌های دیکشنری به این ترتیب به‌روز می‌شوند.

۳-۴ روش‌های پردازش داده‌های با ابعاد بالا

داده‌هایی که امروزه برای پردازش و تحلیل در دسترس قرار می‌گیرند عموماً حجم و بُعد بسیار بالایی دارند. به عنوان مثال، داده‌هایی که هر روزه در شبکه‌های اجتماعی رد و بدل می‌شود، داده‌های موجود در اینترنت، تصاویر با تفکیک‌پذیری بالایی که از ماهواره‌ها دریافت می‌شود، داده‌های دریافتی از شبکه‌های حسگر با ابعاد و گستردگی بالا و ذخیره‌سازی، پردازش و تحلیل این حجم وسیع از داده‌ها حجم محاسبات و حافظه‌ی ذخیره‌سازی بسیار بالایی می‌طلبد که الگوریتم‌های فعلی برای این منظور کارآ نیستند. به علاوه، در برخی کاربردها با جریانی از داده‌ها^۱ روبرو هستیم، یعنی داده‌ها بصورت لحظه‌ای در دسترس قرار می‌گیرند. بنابراین، باید الگوریتم‌هایی پیشنهاد و استفاده کرد که قادر به پردازش چنین داده‌هایی به صورت زمان-حقیقی^۲ باشند. همچنین، در این حجم وسیع از داده‌ها امکان وجود داده‌های نامربوط و نیز نقصان اطلاعات وجود دارد. در نتیجه، الگوریتم‌های ارائه شده باید قادر به کنترل و غلبه بر چنین مشکلاتی باشند. این حوزه به «داده‌های با ابعاد بالا» معروف است [۲۹، ۱۱۳].

روش‌های موجود برای پردازش داده‌های با ابعاد بالا شامل الگوریتم‌های برخط، روش‌های کاهش بُعد، روش‌های توزیع شده، روش‌های انتخاب داده، استفاده از تنسورها و استفاده از الگوریتم‌های غربالگری^۳ در نمایش تُنک است [۲۹، ۱۱۳] که در ادامه به تعدادی از مهمترین آنها، به طور خاص در کاربرد نمایش‌های وفقی سیگنالها، اشاره‌ی مختصری خواهیم داشت.

^۱Data Stream

^۲Real-time

^۳Screening

۳-۴-۱ روش‌های مبتنی بر گرادیان تصادفی

الگوریتم‌های مبتنی بر گرادیان تصادفی^۱ را شاید بتوان ساده‌ترین و پرکاربردترین روش‌ها برای پردازش داده‌های با ابعاد بالا و نیز داده‌های جریانی دانست [۸۳، ۲۹، ۷۶]. برای توضیح این روش‌ها، مسأله‌ی بهینه‌سازی زیر را که در آن تابع هدف به صورت مجموعی از توابع مشتق‌پذیر روی داده‌های مورد پردازش تجزیه می‌شود در نظر بگیرید

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N f_i(\mathbf{x}) \quad (۱۴-۳)$$

که در آن، f و f_i توابعی حقیقی مقدار و تعریف شده روی مجموعه‌ی \mathcal{X} هستند. متغیر \mathbf{x} می‌تواند بعنوان مثال پارامتر یک مدل باشد که ما به دنبال تخمین زدن آن از روی تعدادی داده‌ی آموزشی هستیم. یک روش ساده و سراسر برای حل این مسأله استفاده از الگوریتم مرتبه‌ی اول Gradient Descent است. این الگوریتم با شروع از یک نقطه‌ی اولیه‌ی \mathbf{x}_0 ، تکرارهای زیر را برای رسیدن به جواب نهائی انجام می‌دهد ($k \geq 0$)

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mu \nabla f(\mathbf{x}_k) \quad (۱۵-۳)$$

به دلیل استفاده از تنها اطلاعات مرتبه‌ی اول تابع هدف، یعنی مشتق مرتبه‌ی اول، این روش پیچیدگی محاسباتی کمی دارد. اما در کاربرد داده‌های با ابعاد بالا، تعداد داده‌ها و نیز بُعد آنها بسیار زیاد است. در نتیجه، ارزیابی گرادیان فوق در هر تکرار ممکن است حجم محاسبات و حافظه‌ی زیادی بطلبد. همچنین، در کاربردهای داده‌های زمان-حقیقی به همه‌ی داده‌ها به صورت یکجا دسترسی نداریم. برای غلبه بر این مشکل، از الگوریتم گرادیان تصادفی (SGD)^۲ استفاده می‌شود [۲۰]. در الگوریتم SGD به صورت تکراری، گرادیان تابع f ، که در واقع عبارت است از متوسط گرادیان توابع f_i ، با یک گرادیان یکی از f_i ها تقریب زده می‌شود. بعبارت دیگر، این الگوریتم با

$$\left\{ \begin{array}{l} \text{انتخاب اندیس تصادفی } i \text{ از مجموعه } \{1, \dots, N\} \\ \mathbf{x}_{k+1} = \mathbf{x}_k - \mu \nabla f_i(\mathbf{x}_k) \end{array} \right. \quad (۱۶-۳)$$

رفت و برگشت روی همه‌ی داده‌های آموزشی بصورت زیر عمل می‌کند

الگوریتم SGD به شیوه‌ی فوق چندبار روی کل داده‌ها عمل کرده تا به یک نقطه‌ی همگرایی برسد. برای همگرایی بهتر، ترتیب داده‌ها برای هر دور بصورت تصادفی عوض می‌شود. در این الگوریتم، گرادیان واقعی که در (۱۵-۳) آمده است با گرادیان محاسبه شده در یک داده تقریب زده می‌شود. برای تقریب بهتر گرادیان و احتمالاً همگرایی سریعتر، می‌توان به جای یک داده، گرادیان را در چند داده، که یک دسته‌ی کوچک (mini-batch) نامیده می‌شود،

^۱ Stochastic Gradient^۲ Stochastic Gradient Descent

محاسبه می‌گردد [۲۰]. نکته‌ی مهم دیگر در این الگوریتم این است که باید مقدار طول گام، که نرخ یادگیری نیز نامیده می‌شود، در طول تکرارها کاهش داده شود. در این وضعیت، نشان داده شده است که اگر تابع هدف محدب باشد، تحت شرایطی نه چندان محدودکننده، الگوریتم SGD تقریباً با اطمینان^۱ به حداقل‌کننده سراسری همگرا می‌شود. در مورد توابع غیرمحدب هم همگرایی به یک نقطه‌ی حداقل‌کننده‌ی محلی اثبات می‌شود [۲۰].

الگوریتم SGD در کاربرد یادگیری دیکشنری به این ترتیب است که با ورود هر داده جدید (با دسته‌ی کوچکی از داده‌ها)، ابتدا نمایش \mathbf{y}_i آن در دیکشنری فعلی محاسبه شده و سپس این داده دیکشنری را اندکی در جهت بهبود خطای نمایش خود به‌روز می‌کند. اگر اندیس این داده را با i نمایش دهیم، این عملیات به صورت زیر خواهد بود [۸۳]:

$$\begin{cases} \mathbf{x}_i = \text{sparse-code}(\mathbf{y}_i, \mathbf{D}_{i-1}) \\ \mathbf{D}_i = \mathbf{D}_{i-1} - \mu_i \nabla f_i(\mathbf{x}_i) \end{cases} \quad (17-3)$$

در این روابط، \mathbf{D}_{i-1} دیکشنری به‌روز شده توسط داده‌ی $i-1$ بوده و μ_i طول گامی است که داده‌ی جدید i دیکشنری را برای بهبود خطای نمایش خود به‌روز می‌کند. بعلاوه، عبارت $\text{sparse-code}(\mathbf{y}_i, \mathbf{D}_{i-1})$ به معنی نمایش \mathbf{y}_i سیگنال روی دیکشنری \mathbf{D}_{i-1} است. همچنین، تابع f_i عموماً بصورت $f_i(\mathbf{x}) = \|\mathbf{y}_i - \mathbf{D}\mathbf{x}\|_2^2$ تعریف می‌شود. این روال تا تمام شدن همه‌ی داده‌ها انجام می‌شود. تا اینجا تنها معادل یک تکرار از الگوریتم‌های عادی یادگیری دیکشنری انجام شده است. برای تکرارهای بعدی مشابه همین فرآیند تکرار می‌شود. برای جزئیات بیشتر، مقالات [۷، ۷۶، ۱۱۲، ۸۳] را ببینید.

۳-۴-۲ روش‌های یادگیری توزیع‌شده^۲

روش‌های توزیع شده برای پردازش داده‌های با ابعاد بالا مبتنی بر پردازش بلوکی داده‌ها توسط پردازشگرهای کوچک، که گره^۳ هم نامیده می‌شوند، است [۲۲]. به بیان دقیقتر، فرض کنید تعدادی واحد پردازشی داریم که در یک شبکه توزیع شده‌اند اما هر یک به تنهایی قادر به پردازش تمام داده‌های در دسترس نیستند. به عنوان مثال، تعدادی کامپیوتر کوچک داریم که قدرت پردازشی و حافظه‌ی هر یک به اندازه‌ای نیست که بتوان همه‌ی داده‌ها را با یکی از آنها پردازش کرد. اما می‌توان برای برخی کاربردها که قابلیت موازی‌سازی دارند (از جمله نمایش‌های

^۱ Almost surely

^۲ Distributed learning methods

^۳ Node

وفقی)، هر واحد پردازشی را مسئول پردازش قسمت کوچکی از داده‌ها قرار داد. این روال به گونه‌ای است که همه‌ی واحدهای پردازشی، بصورت محلی یا سراسری، در تماس و مشارکت با یکدیگر بوده و داده‌های مورد نیاز را میان خود به اشتراک می‌گذارند. شبکه‌ی پردازشی می‌تواند متمرکز^۱ و یا غیرمتمرکز^۲ باشد. در شبکه‌ی متمرکز، یک واحد پردازش مرکزی^۳، یا مرکز ترکیب^۴، وجود دارد که گره‌های شبکه اطلاعات خود را با آن به اشتراک می‌گذارند. واحد پردازش مرکزی نیز اطلاعات بدست آمده از همه‌ی گره‌ها را پردازش کرده و داده‌های مورد نیاز را به گره‌ها ارسال می‌کند. در شبکه‌ی غیرمتمرکز، واحد پردازش مرکزی وجود نداشته و گره‌ها اطلاعات خود را با همسایه‌هایشان به اشتراک می‌گذارند.

یکی از روش‌های معروف و پرکاربرد یادگیری توزیع‌شده مبتنی بر الگوریتم ADMM^۵ [۲۲] است. برای

توضیح این روش، مسأله‌ی (۳-۱۴) را به فرم زیر بازنویسی می‌کنیم

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N f_i(\mathbf{x}) + g(\mathbf{x}) \quad (۱۸-۳)$$

که در آن، g تابع نشانگر^۶ مجموعه‌ی \mathcal{X} است. این تابع برای \mathbf{x} هایی که داخل \mathcal{X} باشند مقدار صفر و در غیر اینصورت مقدار بینهایت برمی‌گرداند. حال، با تعریف متغیرهای محلی \mathbf{x}_i و یک متغیر سراسری \mathbf{z} ، این مسأله را می‌توان به شکل معادل زیر نوشت

$$\min_{\{\mathbf{x}_i\}_{i=1}^N, \mathbf{z}} \frac{1}{N} \sum_{i=1}^N f_i(\mathbf{x}_i) + g(\mathbf{z}) \quad \text{s.t.} \quad \mathbf{x}_i - \mathbf{z} = \mathbf{0}, \forall i \quad (۱۹-۳)$$

به این ترتیب، تابع هدف روی متغیرهای \mathbf{x}_i تجزیه شده و راه برای پردازش توزیع‌شده هموار می‌گردد. قدم بعدی، تشکیل تابع لاگرانژین افزوده^۷ به شکل زیر است

$$L(\{\mathbf{x}_i\}_{i=1}^N, \mathbf{z}, \{\boldsymbol{\lambda}_i\}_{i=1}^N) = \frac{1}{N} \sum_{i=1}^N \left\{ f_i(\mathbf{x}_i) + \boldsymbol{\lambda}_i^T (\mathbf{x}_i - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}\|^2 \right\} + g(\mathbf{z}) \quad (۲۰-۳)$$

که در آن، $\rho > 0$ پارامتر جریمه^۸ بوده و \mathbf{y}_i ها ضرایب لاگرانژ یا متغیرهای دوگان نامیده می‌شوند. تابع فوق سپس بصورت تکراری روی متغیرهای محلی و متغیر سراسری کمینه شده، به همراه آن متغیرهای دوگان^۹ نیز به‌روز

^۱Centralized

^۲Decentralized

^۳Central Processing Unit

^۴Fusion center

^۵Alternating Direction Method of Multipliers

^۶Indicator function

^۷Augmented Lagrangian function

^۸Penalty parameter

^۹Dual variables

می‌شوند. به بیان دقیقتر، الگوریتم کلی ADMM بصورت زیر عمل می‌کند

$$\begin{cases} \mathbf{x}_i^{k+1} = \operatorname{argmin}_{\mathbf{x}_i} f_i(\mathbf{x}_i) + (\boldsymbol{\lambda}_i^k)^T (\mathbf{x}_i - \mathbf{z}^k) + \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}^k\|_2^2, & i = 1, \dots, N \\ \mathbf{z}^{k+1} = \operatorname{argmin}_{\mathbf{z}} \frac{\rho}{2} \|\mathbf{x}_i^{k+1} - \mathbf{z}\|_2^2 + g(\mathbf{z}) \\ \boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\mathbf{x}_i^{k+1} - \mathbf{z}^{k+1}) \end{cases} \quad (21-3)$$

همانطور که ملاحظه می‌شود، می‌توان به‌روزرسانی متغیرهای محلی \mathbf{x}_i را بصورت توزیع شده و مستقل از هم انجام داد. در واقع، هر گره می‌تواند متغیر مربوط به خود را با داشتن یک کپی از مقدار متغیر سراسری \mathbf{z} به‌روز کند. متغیر \mathbf{z} نیز در واحد پردازش مرکزی و با دریافت مقادیر به‌روز شده‌ی متغیرهای محلی از تمامی گره‌ها به‌روز شده و سپس در اختیار گره‌ها قرار می‌گیرد. از آنجایی که عموماً تابع f_i بصورت نرم دو تعریف می‌شود، مسائل به‌روزرسانی متغیرهای محلی جواب فرم بسته دارد. به‌روزرسانی متغیر سراسری نیز با استفاده از روش‌های پراکسیمال^۱، که در فصل آینده مرور می‌شوند، انجام می‌گردد. روش‌های توزیع شده تاکنون برای حل مسأله‌ی یادگیری دیکشنری نیز استفاده شده‌اند [۷۹، ۳۳، ۳۹، ۴۰].

۳-۴-۳ روش‌های مبتنی بر Data Sketching

با توجه به این که بُعد داده‌های امروزی بسیار بالا بوده و از طرفی، عموماً محتوای اطلاعاتی این داده‌ها به مراتب کمتر از بُعد ظاهری آنها است، روش‌های کاهش بُعد طی سالیان اخیر به‌طور گسترده‌ای برای پردازش داده‌های با ابعاد بالا مورد استفاده قرار گرفته است [۲۹]. در واقع، در مورد داده‌های با ابعاد بالا، عملیات جبری پرکاربرد از قبیل ضرب ماتریس در ماتریس، تجزیه به مقادیر ویژه و تجزیه به مقادیر تکین حجم محاسبات بالایی طلب می‌کند [۷۰]. روش‌های کاهش بُعد با تصویر کردن داده‌ها به فضاهایی با ابعاد به مراتب پائین‌تر همزمان با حفظ حداکثری اطلاعات مفید آنها، بار محاسباتی الگوریتم‌های پردازشی را به مراتب پائین می‌آورند. بعلاوه، با توجه به این که ماتریس داده‌ها معمولاً رتبه‌ی بسیار کمتری نسبت به ابعاد خود دارند^۲، می‌توان الگوریتم‌های کارآمدی را با استفاده از این ویژگی طراحی کرد. بعنوان مثال، می‌توان تقریب رتبه-پائین یک ماتریس^۳ را به طرز کارآمدی با استفاده از روش‌های تصادفی انجام داد [۷۰].

علاوه بر بُعد داده‌ها، با توجه به این که در کاربردهای امروزی تعداد داده‌های آموزشی نیز معمولاً بسیار زیاد است، پردازش آنها حافظه و توان پردازشی زیادی می‌طلبد. برای غلبه بر این مشکل، روش‌های انتخاب

^۱ Proximal methods

^۲ Low rank matrices

^۳ Low rank matrix approximation

داده به دنبال کاهش تعداد داده‌ها هستند. این روش‌ها، که به روش‌های sketching نیز معروف هستند [۱۲۹]، یا انتخاب داده‌ها را با یک توزیع احتمالاتی یکنواخت انجام می‌دهند یا با استفاده از ضرب یک ماتریس تصادفی با توزیع مشخص در ماتریس داده‌ها (مشابه روالی که برای کاهش بُعد خطی انجام می‌شود) متناهی برای کاهش تعداد داده‌ها). به این ترتیب، ماتریس داده بصورت یک ماتریس کوچکتر فشرده شده و سپس عملیات پردازشی روی این ماتریس، به جای ماتریس بزرگ اولیه، انجام می‌شود. به عبارت دقیقتر، فرض کنید ماتریس داده‌ها برابر با $\mathbf{Y} \in \mathbb{R}^{n \times L}$ که L عددی بزرگ است. روش sketching برای کاهش تعداد ستون‌های ماتریس به صورت زیر عمل می‌کند

$$\bar{\mathbf{Y}} = \mathbf{Y}\mathbf{S}$$

که در آن $\mathbf{S} \in \mathbb{R}^{L \times N}$ و $N \ll L$. این ماتریس یا با انتخاب تعدادی از سطرهای ماتریس همانی $\mathbf{I}_{L \times L}$ بدست می‌آید که در این صورت معادل با انتخاب تعدادی از ستونهای ماتریس داده‌ها است، و یا یک ماتریس تصادفی مشابه آنچه در حسگری فشرده^۱ وجود دارد است که در این صورت این کار معادل با انتخاب یکسری اندازه‌گیری فشرده از ستون‌های ماتریس است. معروف‌ترین مورد استفاده از sketching در مسائل حداقل مربعات^۲ است. برای توضیح بیشتر، فرض کنید مشاهدات $\{(\mathbf{a}_i, b_i)\}_{i=1}^n$ ، که در آن $\mathbf{a}_i \in \mathbb{R}^d$ و $n \gg d$ از یک سیستم موجودند. این مجموعه در واقع داده‌های آموزشی را تشکیل می‌دهد. سپس، برای برازش یک مدل خطی به این داده‌ها به مفهوم نرم دو، مسأله‌ی زیر را باید حل کنیم:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 \quad (۲۲-۳)$$

که در آن، $\mathbf{A} \in \mathbb{R}^{n \times d}$ بوده و سطرهای آن بردارهای \mathbf{a}_i است. همچنین، درایه‌های بردار $\mathbf{b} \in \mathbb{R}^n$ عبارتند از b_i ‌ها. این مسأله جواب فرم بسته دارد که با حل معادله‌ی $\mathbf{A}^T \mathbf{A} \mathbf{x}^* = \mathbf{A}^T \mathbf{b}$ بدست می‌آید. اگر ستون‌های ماتریس \mathbf{A} مستقل خطی باشند، ماتریس $\mathbf{A}^T \mathbf{A}$ رتبه-کامل^۳ بوده و جواب دستگاه معادلات فوق بصورت $\mathbf{x}^* = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ خواهد بود. مشکل این عبارت در مورد داده‌های با ابعاد بالا حجم محاسبات زیادی است که طلب می‌کند. در واقع، استفاده از ضرب عادی ماتریس در ماتریس برای حل این معادلات پیچیدگی متناسب با $n \cdot d^2$ دارد [۱۲۹]. اما، با استفاده از روش sketching می‌توان محاسبات را تا حد زیادی کم کرد. در واقع، در این رهیافت، ما به دنبال

^۱ Compressed Sensing

^۲ Least squares problems

^۳ Full-rank

یافتن یک جواب x هستیم که برای آن داریم $\|b - Ax\|_2 \leq (1 + \epsilon) \cdot \|b - Ax^*\|_2$. روش sketching برای حل این مشکل بصورت زیر عمل می‌کند:

- عدد $r \ll n$ را انتخاب کن.

- ماتریس $S \in \mathbb{R}^{r \times n}$ را از یک توزیع تصادفی تشکیل بده.

- $S \cdot A$ و $S \cdot b$ را تشکیل بده.

- جواب دقیق مسأله‌ی $\min_x \|S \cdot b - S \cdot Ax\|_2$ را محاسبه کن.

به این ترتیب، یک مسأله‌ی با ابعاد کوچکتر نسبت به مسأله‌ی اصلی حل می‌شود. عبارت $S \cdot A$ یک sketch خطی از ماتریس A نامیده می‌شود [۱۲۹]. انتخاب‌های متعددی برای ماتریس نمونه‌برداری S وجود دارد که معروف‌ترین آن، ماتریس تصادفی با توزیع گوسی است [۱۲۹]. فرض کنید $r = O(d/\epsilon^2)$ داده شده است و S ماتریسی با درایه‌های i.i.d. از توزیع گوسی با میانگین صفر و واریانس $1/r$ است، یعنی $\mathcal{N}(0, 1/r)$. همچنین، فرض کنید ماتریس متعامد یکه‌ی $U \in \mathbb{R}^{n \times (d+1)}$ بگونه‌ای است که فضای ستونی U آن برابر با فضای ستونی ماتریس $[A, b]$ است. حال، با استفاده از خواص توزیع گوسی، می‌توان نشان داد که درایه‌های ماتریس $S \cdot U$ دارای توزیع $\mathcal{N}(0, 1/r)$ هستند [۱۲۹]. از طرفی، طبق قضایای تجمع اندازه 3 [۱۱۹]، با احتمال $1 - \exp(-d)$ ، همه‌ی مقادیر تکین $S \cdot U$ در بازه‌ی $[1 - \epsilon, 1 + \epsilon]$ قرار می‌گیرند. بنابراین برای هر بردار ثابت x داریم $\|S \cdot Ux\|_2 = (1 \pm \epsilon) \cdot \|x\|_2$. حال، برای مسأله‌ی $\min_x \|S \cdot (b - Ax)\|_2 = \min_x \|S \cdot b - S \cdot Ax\|_2$ برای هر بردار x در فضای ستونی U قرار می‌گیرد. در نتیجه، $\|S \cdot (b - Ax)\|_2 = (1 \pm \epsilon) \cdot \|Ax - b\|_2$ [۱۲۹]. بنابراین، با حل مسأله‌ی $\min_x \|S \cdot (b - Ax)\|_2$ با احتمال $1 - \exp(-d)$ ، یک تقریب $(1 + \epsilon)$ به مسأله‌ی اصلی بدست می‌آوریم.

موضوع مهمی که در اینجا مطرح می‌شود این است که طراحی ماتریس S باید بگونه‌ای باشد که ضرب آن در ماتریس و یا بردار حجم محاسبات کمی داشته باشد. در غیر اینصورت، ممکن است محاسبات کلی روش sketching قابل مقایسه با حل عادی مسأله‌ی اصلی باشد. برای غلبه بر این مشکل، پیشنهادها‌ی متعددی داده شده است، از جمله استفاده از ماتریس‌های تصادفی با ساختاری که برای ضرب ماتریس-ماتریس کارآمد هستند. برای

^۱Orthonormal

^۲Column space

^۳Concentration of measure

جزئیات بیشتر در این مورد، به [۱۲۹] مراجعه نمایید.

۳-۵ جمع‌بندی

این فصل مروری داشت بر بحث یادگیری دیکشنری و الگوریتم‌های موجود برای آن. دیدیم که دیکشنری نقش مهمی در ارائه‌ی نمایش تُنک برای سیگنال‌ها و کیفیت الگوریتم‌های مبتنی بر نمایش تُنک دارد. برای همین منظور، یادگیری دیکشنری از روی داده‌های آموزشی به موضوعی جذاب تبدیل شده است. در زمینه‌ی یادگیری دیکشنری، مانند هر ابزار دیگری، با گسترش داده‌ها، مشکل حجم محاسبات مطرح می‌شود. برای رفع این مشکل، روش‌های زیادی مطرح شده‌اند که به تعدادی از موارد مهم و معروف آنها اشاره شد. همچنین، تعدادی از روش‌های کارآمد موجود برای پردازش داده‌های با ابعاد بالا را مرور کردیم.

در فصل بعدی، الگوریتم‌های پراکسیمال را مرور خواهیم کرد. این الگوریتم‌های دسته‌ی مهم و بزرگی از الگوریتم‌ها را در برمی‌گیرند که امروزه به گزینه‌ای کارآمد برای حل بسیاری از مسائل بهینه‌سازی تبدیل شده است. بعلاوه، این الگوریتم‌ها اساس عمده‌ی روش‌های پیشنهادی این رساله هستند.

مروری بر الگوریتم‌های پراکسیمال برای کمینه‌سازی غیرمحدب

۱-۴ مقدمه

امروزه، در بسیاری از کاربردهای پردازش سیگنال و یادگیری ماشین، ما با حجم فزاینده‌ای از داده مواجه هستیم که نیازمند الگوریتم‌های ویژه‌ای برای تحلیل و پردازش هستند. الگوریتم‌های موسوم به پراکسیمال^۱ با توجه به حجم محاسبات نسبتاً پائین توانسته‌اند توجه زیادی را در دهه‌ی اخیر به خود جلب کنند [۹۶، ۱۹، ۳۶]. این الگوریتم‌ها مرتبه‌ی اول هستند، به این معنی که تنها به اطلاعات تابع هدف و مشتق مرتبه‌ی اول آن نیاز دارند. بعلاوه، ساختاری بسیار ساده و در عین حال عملکردی بسیار مطلوب دارند به گونه‌ای که در خیلی از کاربردها بعنوان الگوریتم‌هایی کارآمد مورد استفاده قرار می‌گیرند [۹۶]. مهم‌تر از این، طیف وسیعی از مسائل پردازشی امروزی، با توابع هدف هموار، ناهموار، محدب و حتی غیرمحدب را می‌توان به راحتی با الگوریتم‌های پراکسیمال و با دقت خوبی حل نمود.

با توجه به اهمیت این دسته از الگوریتم‌ها و نیز این نکته که تقریباً همه‌ی روش‌های پیشنهاد شده در این رساله مبتنی بر این الگوریتم‌ها هستند، در این فصل مروری مختصر خواهیم داشت بر مفاهیم اساسی، نسخه‌های اصلی و نیز مبانی اثبات همگرایی این الگوریتم‌ها، به طور خاص برای مسائل غیرمحدب. برای این منظور، ابتدا

^۱Proximal algorithms

برخی تعاریف ضروری را که در این فصل استفاده خواهند شد مرور می‌کنیم. در ادامه، دو نمونه از کاربردهای این الگوریتم‌ها در بهینه‌سازی توابع متشکل از دو قسمت هموار و ناهموار و نیز توابع به کلی ناهموار را مرور می‌کنیم. در نهایت، نگاهی خواهیم داشت بر گام‌های اساسی در اثبات همگرایی این الگوریتم‌ها.

۲-۴ عملگر پراکسیمال

یک عنصر اساسی در الگوریتم‌های پراکسیمال، عملگر پراکسیمال^۱ است. قبل از تعریف این عملگر، ابتدا برخی تعاریف مقدماتی مورد نیاز را مرور می‌کنیم.

تعریف ۱-۴ (دامنه‌ی تابع [۲۳]) برای تابع $f: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ ، دامنه به صورت زیر تعریف می‌شود

$$\text{dom}_f \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) < +\infty\}.$$

تعریف ۲-۴ (تابع سره [۲۳]) می‌گوئیم تابع f سره است هرگاه دامنه‌ی آن مخالف تهی باشد.

تعریف ۳-۴ (گراف یک تابع [۲۳]) گراف تابع $f: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ به صورت مجموعه‌ی زیر تعریف می‌شود

$$\text{graph}(f) \triangleq \{(\mathbf{x}, \lambda) \in \mathbb{R}^{n+1} \mid f(\mathbf{x}) = \lambda\}.$$

تعریف ۴-۴ (تابع از پائین نیمه‌پیوسته [۱۲۳]) گوئیم تابع $f: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ در نقطه‌ی \mathbf{x}_0 از پائین نیمه‌پیوسته است هرگاه برای هر $\epsilon > 0$ ، یک همسایگی U از \mathbf{x}_0 وجود داشته باشد به نحوی که

$$\bullet \text{ اگر } f(\mathbf{x}_0) < +\infty \text{، آنگاه برای همه‌ی } \mathbf{x} \in U \text{، } f(\mathbf{x}) \geq f(\mathbf{x}_0) - \epsilon \text{،}$$

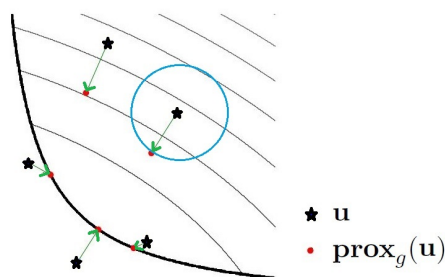
$$\bullet \text{ اگر } f(\mathbf{x}_0) = +\infty \text{، } f(\mathbf{x}) \rightarrow +\infty \text{ وقتی } \mathbf{x} \rightarrow \mathbf{x}_0 \text{.$$

این تعریف همچنین معادل است با عبارت زیر

$$\liminf_{\mathbf{x} \rightarrow \mathbf{x}_0} f(\mathbf{x}) \geq f(\mathbf{x}_0).$$

در ادامه، تعریف عملگر پراکسیمال را مرور می‌کنیم.

^۱Proximal operator



شکل ۴-۱: نمایش مفهوم عملگر پراکسیمال. در این شکل، منحنی تیره نشان‌دهنده دامنه تابع بوده و منحنی‌های کمرنگ تعدادی از سطوح تراز آن را نشان می‌دهند [۹۶].

تعریف ۴-۵ (عملگر پراکسیمال [۹۶]) تابع سره^۱ و از پائین نیمه-پیوسته^۲ $g: \text{dom}_g \rightarrow (-\infty, +\infty]$ را که دارای دامنه dom_g است در نظر بگیرید. عملگر پراکسیمال این تابع در نقطه‌ی $\mathbf{x} \in \mathbb{R}^m$ به صورت زیر تعریف می‌شود:

$$\text{prox}_g(\mathbf{x}) \triangleq \underset{\mathbf{u} \in \text{dom}_g}{\text{argmin}} \quad g(\mathbf{u}) + \frac{1}{\tau} \|\mathbf{x} - \mathbf{u}\|_2. \quad (1-4)$$

به عبارت دیگر، مقدار عملگر پراکسیمال تابع در یک نقطه‌ی داده شده برابر است با حداقل‌کننده‌ی تابع اصلی به اضافه‌ی یک جمله که نزدیکی حداقل‌کننده به نقطه‌ی مورد محاسبه را تضمین می‌کند. در نتیجه، برای فهم بهتر می‌توان معادل زیر را برای عملگر پراکسیمال نوشت:

$$\text{prox}_g(\mathbf{x}) = \underset{\mathbf{u} \in \text{dom}_g}{\text{argmin}} \quad g(\mathbf{u}) \quad \text{s.t.} \quad \|\mathbf{x} - \mathbf{u}\|_2 \leq \tau \quad (2-4)$$

که در آن، $\tau > 0$ به گونه‌ای است که دو مسأله‌ی (۱-۴) و (۲-۴) معادل باشند. طبق این تعریف، مقدار عملگر پراکسیمال تابع g در یک نقطه‌ی \mathbf{x} داده شده برابر با حداقل‌کننده‌ی g در داخل یک گوی حول \mathbf{x} است. این موضوع در شکل ۴-۱ نشان داده شده است. بعنوان یک مثال مهم و پرکاربرد، فرض کنید تابع g تابع مشخصه یک مجموعه محدب و بسته‌ی \mathcal{C} باشد:

$$g(\mathbf{x}) = \delta_{\mathcal{C}}(\mathbf{x}) \triangleq \begin{cases} 0 & \mathbf{x} \in \mathcal{C} \\ \infty & \mathbf{x} \notin \mathcal{C} \end{cases} \quad (3-4)$$

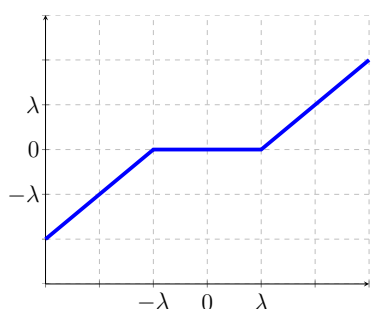
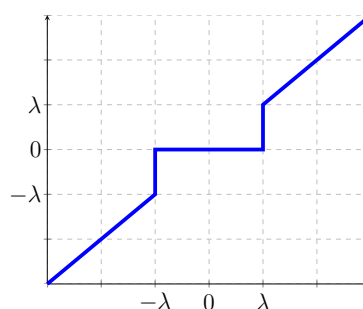
در اینصورت، بسادگی می‌توان نشان داد که عملگر پراکسیمال این تابع عبارت است از عملگر تصویر^۳ روی مجموعه‌ی \mathcal{C} . یعنی داریم $\text{prox}_g(\mathbf{x}) = \mathcal{P}_{\mathcal{C}}(\mathbf{x}) \triangleq \underset{\mathbf{v} \in \mathcal{C}}{\text{argmin}} \|\mathbf{v} - \mathbf{x}\|_2$. به این دلیل، عملگر پراکسیمال را گاه عملگر تصویرکردن تعمیم‌یافته^۴ می‌نامند [۹۶]. به عنوان دو نمونه‌ی معروف دیگر از عملگر پراکسیمال، می‌توان

^۱ Proper

^۲ Lower semi-continuous

^۳ Projection operator

^۴ Generalized projection operator

(ب) عملگر پراکسیمال تابع $g(x) = \lambda \|x\|_1$ (الف) عملگر پراکسیمال تابع $g(x) = \lambda \|x\|_0$

شکل ۴-۲: عملگر پراکسیمال مربوط به دو تابع معروف.

به آستانه‌گذاری سخت و آستانه‌گذاری نرم اشاره کرد که به ترتیب متناظر با عملگر پراکسیمال تابع نرم صفر و تابع نرم یک هستند [۵۲]. این عملگرها در شکل ۴-۲ نشان داده شده‌اند.

عملگر پراکسیمال برای توابع هموار و مشتق‌پذیر یک تعبیر جالب دیگری هم دارد و آن عبارت است از یک گام از گرادیان کاهشی^۱. به بیان دقیق‌تر، تابع هموار g را در نظر بگیرید. در این صورت برای یک $\lambda > 0$ به اندازه‌ی کافی کوچک می‌توان نوشت [۹۶]:

$$\text{prox}_{\lambda g}(\mathbf{x}) = \underset{\mathbf{u} \in \text{dom}_g}{\text{argmin}} \lambda g(\mathbf{u}) + \frac{1}{\lambda} \|\mathbf{x} - \mathbf{u}\|_2^2 \simeq \mathbf{x} - \lambda \nabla g(\mathbf{x}). \quad (4-4)$$

این تعبیر البته با شهود هم هماهنگی دارد، چرا که طبق تعریف عملگر پراکسیمال، تابع اصلی تنها اندکی (حول نقطه‌ای که قرار است محاسبه شود) به سمت حداقل‌کننده‌ی خود حرکت می‌کند. این امر از گام کوچک گرادیان کاهشی فوق مشهود است.

عملگر پراکسیمال تعدادی ویژگی مهم دارد که محاسبه‌ی آن را برای برخی توابع پرکاربرد آسان می‌کند. این ویژگی‌ها در ادامه بیان می‌شوند (برای جزئیات بیشتر به [۹۶] مراجعه نمایید). ویژگی اول، ویژگی ترکیب خطی تابع است. به عبارت دیگر،

$$f(\mathbf{x}) = \alpha \phi(\mathbf{x}) + \beta \rightarrow \text{prox}_f(\mathbf{x}) = \text{prox}_{\alpha \phi}(\mathbf{x}). \quad (5-4)$$

به این ترتیب، براحتی می‌توان عملگر پراکسیمال را برای توابعی که حاصل ترکیب خطی توابع دیگر هستند محاسبه نمود. ویژگی دیگر، موسوم به تغییر آرگومان تابع است که به فرم زیر برقرار است

$$f(\mathbf{x}) = \phi(\alpha \mathbf{x} + \beta) \rightarrow \text{prox}_f(\mathbf{x}) = \frac{1}{\alpha} (\text{prox}_{\alpha \phi}(\alpha \mathbf{x} + \beta) - \beta). \quad (6-4)$$

^۱ Gradient descent

اما مهمترین ویژگی عملگر پراکسیمال معروف به تجزیه مورو^۱ است که عملگر پراکسیمال یک تابع و عملگر پراکسیمال تابع مزدوج^۲ آن را به هم مربوط می‌کند. قبل از بیان این ویژگی، ابتدا تعریف تابع مزدوج را یادآوری می‌کنیم. تابع $f : \mathbb{R} \rightarrow (-\infty, +\infty]$ را در نظر بگیرید. در این صورت، تابع مزدوج آن که با نماد f^* نشان داده می‌شود، به صورت زیر تعریف می‌شود [۲۳]

$$f^*(\mathbf{y}) = \sup_{\mathbf{x}} \{\mathbf{y}^T \mathbf{x} - f(\mathbf{x})\} \quad (۷-۴)$$

بعنوان مثال، می‌توان نشان داد که تابع مزدوج نُرم l_p عبارت است از نُرم l_q که $1/p + 1/q = 1$. با این تعریف، ویژگی مروری عملگر پراکسیمال به صورت زیر بیان می‌شود:

$$\mathbf{x} = \text{prox}_f(\mathbf{x}) + \text{prox}_{f^*}(\mathbf{x}). \quad (۸-۴)$$

در نتیجه، با استفاده از این ویژگی می‌توان به راحتی عملگر پراکسیمال تابع مزدوج یک تابع را به دست آورد.

۳-۴ کمینه‌سازی توابع ترکیبی (هموار + ناهموار)

در این قسمت، کمینه‌سازی توابع ترکیبی را، که بصورت مجموع یک تابع هموار و یک تابع ناهموار بیان می‌شوند، به کمک الگوریتم‌های پراکسیمال مرور می‌کنیم. برای این منظور، مسأله‌ی زیر را در نظر بگیرید:

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + g(\mathbf{x}), \quad (۹-۴)$$

که در آن، $f : \text{dom}_f \rightarrow \mathbb{R}$ تابعی هموار (محدب و یا غیرمحدب) بوده، و $g : \text{dom}_g \rightarrow \mathbb{R} \cup \{+\infty\}$ تابعی ناهموار (محدب و یا غیرمحدب) است. به عنوان مثالی معروف از این دست مسائل می‌توان به مسأله‌ی بازیابی نمایش

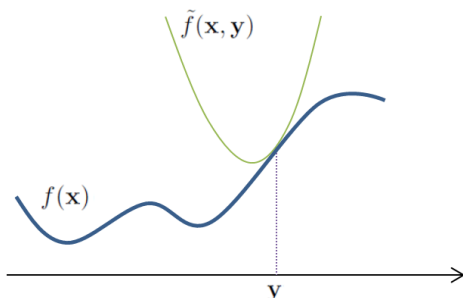
$$\text{تُنک و یا حسگری فشرده اشاره کرده که در آن، } f(\mathbf{x}) = 1/2 \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \text{ و } g(\mathbf{x}) = \|\mathbf{x}\|_1$$

الگوریتم‌های پراکسیمال برای حل مسأله‌ی (۹-۴) به این ترتیب عمل می‌کنند که به صورت تکراری، با شروع از یک جواب (حدس) اولیه، قسمت هموار تابع هدف را با تقریب مرتبه‌ی دوم آن در حول جواب فعلی جایگزین کرده و حداقل‌کننده‌ی مسأله‌ی جدید را به عنوان به‌روزرسانی جواب در نظر می‌گیرند [۱۹]. قبل از این که وارد جزئیات این روش بشویم، لازم است ابتدا مفهوم ثابت لیشیتز^۳ برای گرادیان یک تابع هموار را یادآور شویم. تابع هموار $f : \text{dom}_f \rightarrow \mathbb{R}$ را در نظر بگیرید. در این صورت، می‌گوئیم گرادیان این تابع، یعنی ∇f

^۱Moreau decomposition

^۲Conjugate function

^۳Lipschitz constant



شکل ۴-۳: نمایش تابع هموار f و تقریب مرتبه‌ی دوم آن $\tilde{f}(x, y)$ در نقطه‌ی y .

دارای ثابت لپشیتزی برابر با L است هر گاه داشته باشیم [۹۱]:

$$\forall x, y \in \text{dom}_f : \quad \|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2. \quad (10-4)$$

با این توضیحات، الگوریتم‌های پراکسیمال تکرارهای زیر را برای حل مسأله‌ی (۹-۴) انجام می‌دهند [۱۹]:

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{argmin}} \tilde{f}(\mathbf{x}, \mathbf{x}_k) + g(\mathbf{x}) \quad (11-4)$$

که در آن،

$$\tilde{f}(\mathbf{x}, \mathbf{x}_k) \triangleq f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\mu} \|\mathbf{x} - \mathbf{x}_k\|_2^2 \quad (12-4)$$

تقریب مرتبه‌ی دوم تابع f حول نقطه‌ی \mathbf{x}_k بوده (شکل ۴-۳ را ببینید) و $\mu \in (0, 1/L]$. همچنین، نماد $\langle \cdot, \cdot \rangle$ نشان‌گر

ضرب داخلی است. می‌توان نشان داد که [۱۹]

$$\forall \mathbf{x}, \mathbf{y} \in \text{dom}_f : \quad f(\mathbf{x}) \leq \tilde{f}(\mathbf{x}, \mathbf{y}) \quad (13-4)$$

به عبارت دیگر، در هر تکرار، یک حد بالای مرتبه‌ی دوم از قسمت هموار تابع هدف جایگزین آن می‌شود (شکل

۴-۴ را ببینید). عملیات تکراری (۱۱-۴) را می‌توان به صورت زیر هم بازنویسی کرد [۱۹]

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{argmin}} \frac{1}{2\mu} \|\mathbf{x} - (\mathbf{x}_k - \mu \nabla f(\mathbf{x}_k))\|_2^2 + \mu \cdot g(\mathbf{x}) \quad (14-4)$$

با این بازنویسی، آشکار است که سمت راست رابطه‌ی فوق چیزی نیست جز عملگر پراکسیمال تابع $\mu \cdot g$ که در

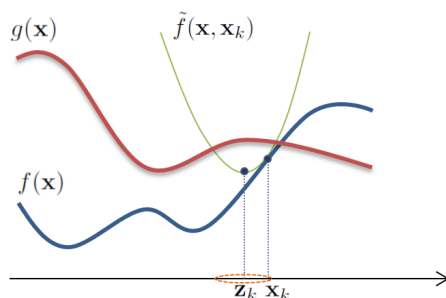
نقطه‌ی $\mathbf{x}_k - \mu \nabla f(\mathbf{x}_k)$ ارزیابی شده است. به این ترتیب، می‌توان نوشت

$$\mathbf{x}_{k+1} = \text{prox}_{\mu g}(\mathbf{x}_k - \mu \nabla f(\mathbf{x}_k)). \quad (15-4)$$

به صورت خلاصه، الگوریتم‌های پراکسیمال برای حل مسائل به فرم (۹-۴)، که تابع هدف آن‌ها مجموع یک

عبارت هموار و ناهموار است، به شیوه‌ای تکراری، با شروع از یک نقطه‌ی اولیه، ابتدا یک گام گرادیان کاهشی

روی قسمت هموار اعمال کرده، و عملگر پراکسیمال قسمت ناهموار را در نقطه‌ی به دست آمده ارزیابی می‌کنند.



شکل ۴-۴: تکرارهای الگوریتم FBS که در معادله‌ی (۴-۱۵) آمده است در این شکل نشان داده شده است. در این شکل، نقطه‌ی z_k عبارت است از نتیجه‌ی اعمال یک گام گرادیان روی تابع f که متناظر با گام رو به جلو است. برای توضیحات بیشتر به متن مراجعه کنید.

گام اول این عملیات، یعنی گرادیان کاهشی، به گام رو به جلو^۱، و گام دوم، یعنی اعمال تابع پراکسیمال، به گام رو به عقب^۲ معروف است [۱۹]. به همین دلیل، این الگوریتم‌ها با نام الگوریتم‌های تجزیه‌ی روبه جلو-رو به عقب^۳ یا به اختصار، الگوریتم‌های FBS، شناخته می‌شوند. این دو گام در شکل ۴-۴ به نمایش در آمده است.

۴-۴ کمینه‌سازی توابع ترکیبی (ناهموار + ناهموار)

مسأله‌ی زیر را در نظر بگیرید

$$\min_{\mathbf{x} \in \mathbb{R}^n} g(\mathbf{x}) + h(\mathbf{x}), \quad (۴-۱۶)$$

که در آن، $g: \text{dom}_g \rightarrow \mathbb{R} \cup \{+\infty\}$ و $h: \text{dom}_h \rightarrow \mathbb{R} \cup \{+\infty\}$ توابعی ناهموار بوده که می‌توانند محدب و یا غیرمحدب باشند. الگوریتم‌های پراکسیمال برای کمینه‌سازی چنین توابعی تکرارهای زیر را انجام می‌دهند [۳۶]

$$\mathbf{x}_{k+1} = \text{prox}_g \left(\text{prox}_h(\mathbf{x}_k) \right). \quad (۴-۱۷)$$

به عبارت دیگر، در این الگوریتم‌ها، با شروع از یک نقطه‌ی اولیه، به صورت تکراری، عملگر پراکسیمال دو تابع اعمال می‌شوند. به دلیل این که در این الگوریتم‌ها هر دو گام عبارت است از ارزیابی یک تابع پراکسیمال، این الگوریتم‌ها به الگوریتم‌های تجزیه‌ی رو به عقب-رو به عقب^۴ یا به طور خلاصه BBS معروف هستند [۳۶]. به عنوان مثال‌هایی از کاربرد این الگوریتم‌ها، مسأله‌ی یافتن یک نقطه در فصل مشترک دو مجموعه‌ی \mathcal{C}_1 و \mathcal{C}_2 را در نظر بگیرید. این مسأله را می‌توان به فرم زیر فرمول‌بندی کرد [۳۶]

^۱Forward step

^۲Backward step

^۳Forward-backward splitting

^۴Backward-backward splitting

$$\min_{\mathbf{x}} \delta_{C_l}(\mathbf{x}) + \delta_{C_r}(\mathbf{x}). \quad (18-4)$$

واضح است که هر دو قسمت از تابع هدف، توابعی ناهموار هستند. در نتیجه، با توجه به این که عملگر پراکسیمال برای تابع مشخصه‌ی یک مجموعه برابر است با عملگر تصویر روی آن مجموعه [۹۶]، الگوریتم BBS برای حل مسأله‌ی فوق به صورت زیر خواهد بود

$$\mathbf{x}_{k+1} = \mathcal{P}_{C_r} \left(\mathcal{P}_{C_l}(\mathbf{x}_k) \right). \quad (19-4)$$

به عنوان مثالی دیگر، مسأله‌ی حداقل‌سازی نرَم صفر را در نظر بگیرید

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon. \quad (20-4)$$

این مسأله را می‌توان به فرم معادل زیر نوشت

$$\min_{\mathbf{x} \in \mathbb{R}^n} \underbrace{\|\mathbf{x}\|_0}_{g(\mathbf{x})} + \underbrace{\delta_C(\mathbf{x})}_{h(\mathbf{x})}, \quad (21-4)$$

که در آن، $C = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon\}$. در نتیجه، الگوریتم BBS برای حل مسأله‌ی (۲۰-۴) به صورت زیر خواهد بود^۱

$$\mathbf{x}_{k+1} = \underbrace{\text{prox}_h}_{\text{تصویر}} \left(\underbrace{\text{prox}_g}_{\text{آستانه‌گذاری سخت}}(\mathbf{x}_k) \right). \quad (22-4)$$

۴-۵ تعمیم به بیش از یک بلوک

مسائلی که در قسمت‌های قبل بررسی کردیم همگی متشکل از یک بلوک از متغیر، یعنی \mathbf{x} ، بودند. اما در بسیاری از مسائل کاربردی، متغیرها در بیش از یک بلوک ظاهر می‌شوند. به عنوان مثال، در مسأله‌ی یادگیری دیکشنری، ما دو متغیر \mathbf{X} و \mathbf{D} داریم. در نتیجه، حل مسأله‌ی یادگیری دیکشنری با فرمول‌بندی‌های قسمت‌های قبل مقدور نیست. برای حل این مشکل، تعمیم الگوریتم‌های قسمت‌های قبل به حالت بیش از یک بلوک از متغیرها با استفاده از کمینه‌سازی نوبتی انجام شده است [۱۹]. برای سادگی، ما تنها تعمیم به دو بلوک را از [۱۹] مرور می‌کنیم. برای این منظور، مسأله‌ی کلی زیر را در نظر بگیرید

$$\min_{\mathbf{x}, \mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{y}) + H(\mathbf{x}, \mathbf{y}), \quad \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m, \quad (23-4)$$

که در آن، $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ و $g: \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ توابعی سره و از پائین نیمه‌هموار هستند و $H: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ تابعی با مشتق اول پیوسته است. همچنین، ثابت‌های لپشیتز گرادیان این تابع نسبت به \mathbf{x}

^۱ این الگوریتم با جزئیات بیشتری در قسمت ۴-۵ بررسی خواهد شد.

و \mathbf{y} را به ترتیب با L_x و L_y نشان می‌دهیم. علاوه بر این، این تابع روی زیرمجموعه‌های کراندار از $\mathbb{R}^n \times \mathbb{R}^m$

لیپشیتز فرض می‌شود. یعنی، $M > 0$ وجود دارد به گونه‌ای که برای هر $(\mathbf{x}_i, \mathbf{y}_i)$ ، $i = 1, 2$ ، داریم [۱۹]

$$\|(\nabla_x H(\mathbf{x}_1, \mathbf{y}_1) - \nabla_x H(\mathbf{x}_2, \mathbf{y}_2), \nabla_y H(\mathbf{x}_1, \mathbf{y}_1) - \nabla_y H(\mathbf{x}_2, \mathbf{y}_2))\|_2 \leq M \|(\mathbf{x}_1 - \mathbf{x}_2, \mathbf{y}_1 - \mathbf{y}_2)\|_2 \quad (24-4)$$

در اینجا فرض بر این است که همه توابع موجود در (۴-۲۳) از پائین کراندار هستند. حل مسأله‌ی (۴-۲۳) با

استفاده از ایده‌های پراکسیمال به این ترتیب است که به صورت نوبتی، مسأله یک بار روی بلوک \mathbf{x} و در نوبت بعد

روی بلوک \mathbf{y} حداقل می‌شود. به علاوه، هر بار، تابع مشتق‌پذیر H روی متغیر مورد کمینه‌سازی با تقریب مرتبه‌ی

دوم خود (با ثابت در نظر گرفتن بلوک دیگر) جایگزین می‌شود. به عبارت دقیق‌تر، داریم

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) + H(\mathbf{x}_k, \mathbf{y}_k) + \langle \nabla_x H(\mathbf{x}_k, \mathbf{y}_k), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{\sqrt{\mu_x}} \|\mathbf{x} - \mathbf{x}_k\|_2^2 \\ \mathbf{y}_{k+1} = \operatorname{argmin}_{\mathbf{y}} g(\mathbf{y}) + H(\mathbf{x}_{k+1}, \mathbf{y}_k) + \langle \nabla_y H(\mathbf{x}_{k+1}, \mathbf{y}_k), \mathbf{y} - \mathbf{y}_k \rangle + \frac{1}{\sqrt{\mu_y}} \|\mathbf{y} - \mathbf{y}_k\|_2^2, \end{cases} \quad (25-4)$$

که در آن، $\mu_x \in (0, 1/L_x]$ و $\mu_y \in (0, 1/L_y]$. مشابه قبل، می‌توان نشان داد که به‌روزرسانی‌های فوق معادل با

روابط زیر هستند

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x}) + \frac{1}{\sqrt{\mu_x}} \|\mathbf{x} - (\mathbf{x}_k - \mu_x \nabla_x H(\mathbf{x}_k, \mathbf{y}_k))\|_2^2 \\ \mathbf{y}_{k+1} = \operatorname{argmin}_{\mathbf{y}} g(\mathbf{y}) + \frac{1}{\sqrt{\mu_y}} \|\mathbf{y} - (\mathbf{y}_k - \mu_y \nabla_y H(\mathbf{x}_{k+1}, \mathbf{y}_k))\|_2^2, \end{cases} \quad (26-4)$$

که با توجه به تعریف عملگر پراکسیمال، به روابط نهایی زیر می‌رسیم

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{prox}_{\mu_x f}(\mathbf{x}_k - \mu_x \nabla_x H(\mathbf{x}_k, \mathbf{y}_k)) \\ \mathbf{y}_{k+1} = \operatorname{prox}_{\mu_y g}(\mathbf{y}_k - \mu_y \nabla_y H(\mathbf{x}_{k+1}, \mathbf{y}_k)). \end{cases} \quad (27-4)$$

الگوریتم پراکسیمال نهایی با شروع از یک نقطه‌ی اولیه $(\mathbf{x}_0, \mathbf{y}_0)$ تکرارهای فوق را تا رسیدن به یک شرط توقف

انجام می‌دهد. این الگوریتم، کمینه‌سازی خطی‌سازی تناوبی پراکسیمال^۱ یا به اختصار PALM نامیده می‌شود [۱۹].

۴-۶ مبانی اثبات همگرایی

همگرایی الگوریتم‌های پراکسیمال برای مسائل محدب به عنوان یک مسأله استاندارد شناخته شده است [۹۶]. با

این حال، بررسی همگرایی این الگوریتم‌ها برای مسائل غیرمحدب چند سالی است که مورد توجه قرار گرفته

است. دلیل این امر عمدتاً این موضوع است که خیلی از مسائل مهم در بسیاری از کاربردها ماهیتی غیرمحدب

دارند. در این بخش، ما به اختصار اصول و گام‌های اساسی در اثبات همگرایی الگوریتم PALM برای دسته‌ی کلی

مسائل غیرمحدب که در رابطه‌ی (۴-۲۳) فرمول‌بندی شده‌اند، مرور می‌کنیم. برای این منظور، ابتدا مقدمات این

بحث را که شامل تعاریف و لم‌های ضروری است مرور کرده، در ادامه، مراحل اثبات را با توجه به مرجع [۱۹]

پی می‌گیریم.

^۱Proximal Alternating Linearized Minimization

۴-۶-۱ تعاریف و لم‌های مورد نیاز

برخی از این تعاریف را قبلاً در این فصل استفاده کردیم. برای کامل بودن این بحث، این تعاریف را به همراه تعاریف و لم‌های ضروری دیگر مرور می‌کنیم.

تعریف ۴-۶ (زیر-دیفرانسیل^۱ [۱۲۳]) فرض کنید $g: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ یک تابع صحیح و از پائین نیمه‌پیوسته

است. در این صورت، زیر-دیفرانسیل آن در یک نقطه‌ی $\mathbf{x} \in \text{dom}_g$ عبارت است از مجموعه‌ی زیر

$$\partial g(\mathbf{x}) \triangleq \left\{ \zeta \in \mathbb{R}^n \mid \exists \mathbf{x}_k \rightarrow \mathbf{x}, g(\mathbf{x}_k) \rightarrow g(\mathbf{x}), \zeta_k \rightarrow \zeta, \zeta_k \in \hat{\partial} g(\mathbf{x}_k) \right\},$$

که در آن، $\hat{\partial} g(\mathbf{x})$ ، زیر-دیفرانسیل فرجه^۲ تابع g در نقطه‌ی $\mathbf{x} \in \text{dom}_g$ ، به صورت زیر تعریف می‌شود

$$\hat{\partial} g(\mathbf{x}) \triangleq \left\{ \zeta \in \mathbb{R}^n \mid \liminf_{\mathbf{v} \rightarrow \mathbf{x}, \mathbf{v} \neq \mathbf{x}} \frac{1}{\|\mathbf{x} - \mathbf{v}\|} \cdot \left(g(\mathbf{v}) - g(\mathbf{x}) - \langle \mathbf{v} - \mathbf{x}, \zeta \rangle \right) \geq 0 \right\}.$$

با این تعریف، اگر \mathbf{x} یک حداقل‌کننده‌ی محلی تابع f باشد، در این صورت، $\mathbf{o} \in \partial f(\mathbf{x})$ [۱۲۳].

لم ۴-۱ (دنباله‌ی همگرا در زیر-دیفرانسیل [۱۹]) فرض کنید $\{(\mathbf{x}_k, u_k)\}_{k \in \mathbb{N}}$ دنباله‌ای در گراف ∂g باشد که به

(\mathbf{x}, u) میل می‌کند وقتی که $k \rightarrow \infty$. در این صورت، طبق تعریف زیر-دیفرانسیل، اگر $g(\mathbf{x}_k)$ به $g(\mathbf{x})$ میل کند،

در این صورت، $(\mathbf{x}, u) \in \text{graph}(\partial f)$.

تعریف ۴-۷ (نقطه‌ی بحرانی^۳ [۱۲۳]) گوئیم \mathbf{x} یک نقطه‌ی بحرانی برای تابع f است هر گاه $\mathbf{o} \in \partial f(\mathbf{x})$.

لم ۴-۲ (زیر-دیفرانسیل برای توابع ترکیبی [۱۹]) تابع $\Psi(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{y}) + H(\mathbf{x}, \mathbf{y})$ را که H هموار و

f و g ناهموار هستند در نظر بگیرید. در این صورت،

$$\partial \Psi(\mathbf{x}, \mathbf{y}) = \left(\partial f(\mathbf{x}) + \nabla_{\mathbf{x}} H(\mathbf{x}, \mathbf{y}), \partial g(\mathbf{y}) + \nabla_{\mathbf{y}} H(\mathbf{x}, \mathbf{y}) \right) = \left(\partial_{\mathbf{x}} \Psi(\mathbf{x}, \mathbf{y}), \partial_{\mathbf{y}} \Psi(\mathbf{x}, \mathbf{y}) \right). \quad (۲۸-۴)$$

یک ویژگی کلیدی در مورد یک تابع، که نقشی اساسی در اثبات همگرایی الگوریتم‌های پراکسیمال برای مسائل

غیرمحدب دارد، موسوم به ویژگی کوردیکا-لواشویچ^۴ یا به اختصار، ویژگی KL، است که به صورت زیر تعریف

می‌شود.

تعریف ۴-۸ (ویژگی KL [۱۹]) فرض کنید $f: \mathbb{R}^n \rightarrow (-\infty, +\infty]$ یک تابع صحیح و از پائین نیمه‌پیوسته

است. در این صورت،

^۱Fréchet subdifferential

^۴Kurdyka-Lojasiewicz property

• گفته می‌شود که تابع f دارای ویژگی KL در نقطه‌ی \mathbf{x}_0 متعلق به دامنه‌ی ∂f است هر گاه وجود داشته باشد

$\eta \in (0, \infty]$ یک همسایگی U از \mathbf{x}_0 و یک تابع مقعر و پیوسته‌ی $\psi: [0, \eta) \rightarrow \mathbb{R}^+$ به نحوی که برای هر \mathbf{x}

متعلق به مجموعه‌ی $U \cup \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}_0) < f(\mathbf{x}) < f(\mathbf{x}_0) + \eta\}$ داشته باشیم

$$\psi'(f(\mathbf{x}) - f(\mathbf{x}_0)) \cdot \text{dist}(\mathbf{x}_0, \partial f(\mathbf{x})) \geq 1 \quad (29-4)$$

که در آن، $\text{dist}(\mathbf{x}_0, \partial f(\mathbf{x}))$ نشانگر فاصله‌ی بردار صفر از $\partial f(\mathbf{x})$ است.

• اگر تابع f ویژگی KL را در تمام نقاط متعلق به ∂f داشته باشد، یک تابع KL خوانده می‌شود.

تعریف ۹-۴ (مجموعه‌ی نیمه‌جبری^۱ [۱۹]) زیرمجموعه‌ی S از \mathbb{R}^n نیمه‌جبری خوانده می‌شود هر گاه تعداد شمارایی

تابع چندجمله‌ای حقیقی P_{ij} و Q_{ij} وجود داشته باشند به گونه‌ای که $S = \bigcup_{j=1}^p \bigcap_{i=1}^q \{\mathbf{x} \in \mathbb{R}^n \mid P_{ij} = 0, Q_{ij} < 0\}$

تعریف ۱۰-۴ (تابع نیمه‌جبری^۲ [۱۲۳]) تابع f نیمه‌جبری است هر گاه گراف آن یک مجموعه‌ی نیمه‌جبری باشد.

یک ویژگی مهم توابع نیمه‌جبری این است که این توابع، KL هستند [۱۲۳]. لم زیر، که قبلاً به نحوی آن را معرفی

کردیم، نیز نقشی کلیدی در توسعه‌ی الگوریتم‌های پراکسیمال بازی می‌کند.

لم ۳-۴ (لم کاهش^۳ [۱۲۳]) فرض کنید تابع $f: \mathbb{R}^n \rightarrow \mathbb{R}$ به صورت پیوسته مشتق‌پذیر بوده و دارای گرادیان

∇f ، که ثابت لپشیتز L_f دارد، است. در این صورت، نامساوی زیر برای همه‌ی $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ برقرار است

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L_f}{2} \|\mathbf{y} - \mathbf{x}\|_2^2. \quad (30-4)$$

۲-۶-۴ ویژگی‌های همگرایی پایه

در این قسمت، چند ویژگی در مورد همگرایی دنباله‌ی $\{\mathbf{x}_k, \mathbf{y}_k\}_{k \geq 0}$ تولید شده توسط الگوریتم پراکسیمال و

نیز مقادیر تابع هدف متناظر، یعنی $\{\Psi(\mathbf{x}_k, \mathbf{y}_k)\}_{k \geq 0}$ ، از [۱۹] مرور می‌شود. برای این منظور، تعریف می‌کنیم

$\mathbf{z}_k = (\mathbf{x}_k, \mathbf{y}_k)$. در اینجا فرض بر این است که دنباله‌ی $\{\mathbf{z}_k\}_{k \geq 0}$ کراندار است. سپس، مسأله‌ی به‌روزرسانی \mathbf{x} را

در معادله‌ی (۲۵-۴) در نظر بگیرید. در این صورت، می‌توان نوشت

$$f(\mathbf{x}_{k+1}) + \langle \nabla_x H(\mathbf{x}_k, \mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{1}{\sqrt{\mu_x}} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 \leq f(\mathbf{x}_k). \quad (31-4)$$

از طرف دیگر، با استفاده از خاصیت لم کاهش برای تابع $H(\cdot, \mathbf{y}_k)$ ، می‌توان نوشت

$$H(\mathbf{x}_{k+1}, \mathbf{y}_k) \leq H(\mathbf{x}_k, \mathbf{y}_k) + \langle \nabla_x H(\mathbf{x}_k, \mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{L_x}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2. \quad (32-4)$$

با جمع کردن دو سمت روابط (۳۱-۴) و (۳۲-۴) خواهیم داشت

$$\left(\frac{1}{\mu_x} - \frac{L_x}{\mu_x}\right) \|x_{k+1} - x_k\|_2^2 \leq f(x_k) + H(x_k, y_k) - f(x_{k+1}) - H(x_{k+1}, y_k). \quad (33-4)$$

با دنبال کردن روالی مشابه برای y ، بدست خواهیم آورد

$$\left(\frac{1}{\mu_y} - \frac{L_y}{\mu_y}\right) \|y_{k+1} - y_k\|_2^2 \leq g(y_k) + H(x_{k+1}, y_k) - g(y_{k+1}) - H(x_{k+1}, y_{k+1}). \quad (34-4)$$

تعریف می‌کنیم

$$\gamma_x = \frac{1}{\mu_x} - \frac{L_x}{\mu_x}, \quad \gamma_y = \frac{1}{\mu_y} - \frac{L_y}{\mu_y}. \quad (35-4)$$

حال، با جمع کردن هر دو سمت روابط (33-4) و (34-4) داریم

$$\gamma_x \|x_{k+1} - x_k\|_2^2 + \gamma_y \|y_{k+1} - y_k\|_2^2 \leq \Psi(x_k, y_k) - \Psi(x_{k+1}, y_{k+1}), \quad (36-4)$$

که به صورت زیر نیز قابل خلاصه‌سازی است

$$\rho \|z_{k+1} - z_k\|_2^2 \leq \Psi(x_k, y_k) - \Psi(x_{k+1}, y_{k+1}), \quad (37-4)$$

که در آن، $\rho = \min(\gamma_x, \gamma_y)$. از رابطه‌ی فوق نتیجه می‌شود که دنباله‌ی مقادیر تابع هدف، یعنی $\{\Psi(x_k, y_k)\}_{k \geq 0}$ ، دنباله‌ای غیرافزایشی است. از طرفی، چون تابع Ψ از پائین کراندار است، نتیجه می‌شود که این دنباله همگرا است. حال، رابطه‌ی (37-4) را برای همه‌ی $k \geq 0$ نوشته و دو سمت آن‌ها را با هم جمع می‌کنیم. در این صورت،

خواهیم داشت

$$\sum_{k=0}^{\infty} \rho \|z_{k+1} - z_k\|_2^2 \leq \Psi(x_0, y_0) - \Psi(x_{\infty}, y_{\infty}). \quad (38-4)$$

با توجه به این که $\gamma_x, \gamma_y \geq 0$ و لذا $\rho > 0$ ، رابطه‌ی فوق نتیجه می‌دهد که

$$\lim_{k \rightarrow \infty} \|z_{k+1} - z_k\|_2 = 0. \quad (39-4)$$

۳-۶-۴ همگرایی زیردنباله به نقطه‌ی بحرانی

در این قسمت، یک باند پائین برای نرم اختلاف دو تکرار متوالی، یعنی $\|z_{k+1} - z_k\|_2$ ، به دست می‌آید [۱۹] که در ادامه، برای اثبات همگرایی دنباله‌ی تولید شده توسط الگوریتم پراکسیمال به یک نقطه‌ی بحرانی تابع هدف مفید خواهد بود. برای این منظور، توجه داریم که شرط بهینگی مرتبه‌ی اول برای مسائل بهینه‌سازی (۲۵-۴) لازم می‌دارد که بردار صفر عضو مجموعه‌ی زیر-دیفرانسیل توابع هدف مربوطه در نقاط حداقل کننده باشد. به عبارت

دیگر، برای مسأله‌ی به‌روزرسانی x خواهیم داشت

$$0 \in \partial f(x_{k+1}) + \nabla_x H(x_k, y_k) + \frac{1}{\mu_x} (x_{k+1} - x_k). \quad (40-4)$$

تعریف می‌کنیم

$$A_x^k \triangleq \frac{1}{\mu_x}(\mathbf{x}_k - \mathbf{x}_{k+1}) + \nabla_x H(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - \nabla_x H(\mathbf{x}_k, \mathbf{y}_k). \quad (41-4)$$

سپس، با در نظر گرفتن رابطه‌ی (۴-۴۰)، می‌توان دید که $A_x^k \in \partial_x \Psi(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$. با طی روالی مشابه برای \mathbf{y} و تعریف کمیتی مشابه (۴-۴۱)، می‌توان نوشت

$$(A_x^k, A_y^k) \in \partial \Psi(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}). \quad (42-4)$$

از طرفی، با استفاده از ویژگی توپلیتز بودن گرادیان تابع H و با بکارگیری نامساوی مثلثی برای نُرم، از رابطه‌ی

(۴-۴۱) می‌توان نتیجه گرفت

$$\begin{aligned} \|A_x^k\|_2 &\leq \frac{1}{\mu_x} \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2 + \|\nabla_x H(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - \nabla_x H(\mathbf{x}_k, \mathbf{y}_k)\|_2 \\ &\leq \frac{1}{\mu_x} \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2 + M(\|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2 + \|\mathbf{y}_k - \mathbf{y}_{k+1}\|_2) \\ &= (M + \frac{1}{\mu_x}) \|\mathbf{x}_k - \mathbf{x}_{k+1}\|_2 + M \|\mathbf{y}_k - \mathbf{y}_{k+1}\|_2 \\ &\leq \sqrt{2} (M + \frac{1}{\mu_x}) \|\mathbf{z}_k - \mathbf{z}_{k+1}\|_2. \end{aligned} \quad (43-4)$$

به طرز مشابه، با توجه به مسأله‌ی به‌روزرسانی \mathbf{y} و با در نظر گرفتن شرط بهینگی مرتبه‌ی اول بدست می‌آوریم

$$\mathbf{o} \in \partial g(\mathbf{y}_{k+1}) + \nabla_y H(\mathbf{x}_{k+1}, \mathbf{y}_k) + \frac{1}{\mu_y}(\mathbf{y}_{k+1} - \mathbf{y}_k), \quad (44-4)$$

که با تعریف

$$A_y^k \triangleq \frac{1}{\mu_y}(\mathbf{y}_k - \mathbf{y}_{k+1}) + \nabla_y H(\mathbf{x}_{k+1}, \mathbf{y}_{k+1}) - \nabla_y H(\mathbf{x}_{k+1}, \mathbf{y}_k), \quad (45-4)$$

و با توجه به لیبشیتز بودن $\nabla_y H$ ، می‌توان نوشت

$$\|A_y^k\|_2 \leq \frac{2}{\mu_y} \|\mathbf{z}_k - \mathbf{z}_{k+1}\|_2. \quad (46-4)$$

در نتیجه، از روابط (۴-۴۳) و (۴-۴۶) داریم

$$\|(A_x^k, A_y^k)\|_2 \leq \|A_x^k\|_2 + \|A_y^k\|_2 \leq \rho_{xy} \|\mathbf{z}_k - \mathbf{z}_{k+1}\|_2. \quad (47-4)$$

که در آن، $\rho_{xy} = \max(\sqrt{2}(M + 1/\mu_x), 2/\mu_y)$.

با توجه به فرض کراندار بودن دنباله‌ی $\{\mathbf{z}_k\}_{k \geq 0}$ ، این دنباله حاوی زیردنباله‌ای همگرا است. فرض کنید

$\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*)$ یک نقطه‌ی حدی از این دنباله باشد. در نتیجه، زیردنباله‌ای مانند $\{(\mathbf{x}_{k_q}, \mathbf{y}_{k_q})\}_{q \in \mathbb{N}}$ در $\{\mathbf{z}_k\}_{k \geq 0}$ وجود دارد که

$(\mathbf{x}_{k_q}, \mathbf{y}_{k_q}) \rightarrow (\mathbf{x}^*, \mathbf{y}^*)$ وقتی $q \rightarrow \infty$. با توجه به اینکه توابع f و g از پائین نیمه‌پیوسته هستند،

داریم

$$\liminf_{q \rightarrow \infty} f(\mathbf{x}_{k_q}) \geq f(\mathbf{x}^*), \quad \liminf_{q \rightarrow \infty} g(\mathbf{y}_{k_q}) \geq g(\mathbf{y}^*). \quad (۴۸-۴)$$

از طرف دیگر، با قرار دادن $\mathbf{x} = \mathbf{x}^*$ در رابطه‌ی (۴-۲۵)، داریم

$$\begin{aligned} f(\mathbf{x}_{k+1}) + \langle \nabla_x H(\mathbf{x}_k, \mathbf{y}_k), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{1}{\sqrt{\mu_x}} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 \leq \\ f(\mathbf{x}^*) + \langle \nabla_x H(\mathbf{x}_k, \mathbf{y}_k), \mathbf{x}^* - \mathbf{x}_k \rangle + \frac{1}{\sqrt{\mu_x}} \|\mathbf{x}^* - \mathbf{x}_k\|_2^2. \end{aligned} \quad (۴۹-۴)$$

با قرار دادن $k = k_q - 1$ در نامساوی فوق و میل دادن q به سمت بی‌نهایت، خواهیم داشت

$$\limsup_{q \rightarrow \infty} f(\mathbf{x}_{k_q}) \leq \limsup_{q \rightarrow \infty} \left(\langle \nabla_x H(\mathbf{x}_{k_q-1}, \mathbf{y}_{k_q-1}), \mathbf{x}^* - \mathbf{x}_{k_q-1} \rangle + \frac{1}{\sqrt{\mu_x}} \|\mathbf{x}^* - \mathbf{x}_{k_q-1}\|_2^2 \right) + f(\mathbf{x}^*). \quad (۵۰-۴)$$

حال، با توجه به این که $\mathbf{x}_{k_q-1} \rightarrow \mathbf{x}^*$ برای $q \rightarrow \infty$ ، در نهایت به این نتیجه می‌رسیم که

$$\limsup_{q \rightarrow \infty} f(\mathbf{x}_{k_q}) \leq f(\mathbf{x}^*). \quad (۵۱-۴)$$

ترکیب دو رابطه‌ی (۴-۴۸) و (۴-۵۱) نتیجه می‌دهد که $f(\mathbf{x}_{k_q}) \rightarrow f(\mathbf{x}^*)$. استدلال مشابهی را برای تابع g و \mathbf{y}

می‌توان نوشت. در نهایت، داریم

$$\begin{aligned} \lim_{q \rightarrow \infty} \Psi(\mathbf{x}_{k_q}, \mathbf{y}_{k_q}) &= \lim_{q \rightarrow \infty} \{f(\mathbf{x}_{k_q}) + g(\mathbf{y}_{k_q}) + H(\mathbf{x}_{k_q}, \mathbf{y}_{k_q})\} \\ &= f(\mathbf{x}^*) + g(\mathbf{y}^*) + H(\mathbf{x}^*, \mathbf{y}^*) \\ &= \Psi(\mathbf{x}^*, \mathbf{y}^*). \end{aligned} \quad (۵۲-۴)$$

از طرف دیگر، با در نظر گرفتن روابط (۴-۴۲)، (۴-۴۷) و (۴-۳۹)، از لم ۴-۱ نتیجه می‌شود که $(\mathbf{o}, \mathbf{o}) \in$

$\partial \Psi(\mathbf{x}^*, \mathbf{y}^*)$. بنابراین، $(\mathbf{x}^*, \mathbf{y}^*)$ یک نقطه‌ی بحرانی Ψ است.

۴-۶-۴ همگرایی کلی به نقطه‌ی بحرانی

در این قسمت، اثبات این موضوع که دنباله‌ی تولید شده توسط الگوریتم پراکسیمال به یک نقطه‌ی بحرانی Ψ همگرا می‌شود، مرور می‌گردد. اینجا زمانی است که ویژگی مهم KL بودن وارد عمل می‌شود. در واقع، اثبات‌هایی که تا به اینجا در مورد همگرایی مرور شد هیچکدام این ویژگی را لازم نداشتند. این نتیجه در قضیه‌ی زیر مرور می‌شود.

قضیه ۴-۱ ([۱۹]) فرض کنید Ψ تابعی KL است. همچنین، کلیه‌ی مفروضاتی که تا به اینجا مطرح شد نیز همچنان

برقرار هستند. در اینصورت، داریم

• دنباله‌ی $\{\mathbf{z}_k\}_{k \in \mathbb{N}}$ طول محدودی دارد، یعنی

$$\sum_{k=1}^{\infty} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2 < \infty \quad (۵۳-۴)$$

• دنباله‌ی $\{\mathbf{z}_k\}_{k \in \mathbb{N}}$ به یک نقطه‌ی بحرانی $\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*)$ از Ψ همگرا می‌شود.

برای مرور مختصر اثبات این قضیه، ابتدا توجه کنید که فرض کراندار بودن دنباله‌ی $\{\mathbf{z}_k\}_{k \in \mathbb{N}}$ نتیجه می‌دهد که

زیردنباله‌ای مانند $\{\mathbf{z}_{k_q}\}_{q \in \mathbb{N}}$ از این دنباله وجود دارد که برای $q \rightarrow \infty$ به $\bar{\mathbf{z}}$ همگرا می‌شود. با طی کردن گام‌هایی

مشابه آنچه در بخش قبلی انجام شد، می‌توان نوشت

$$\lim_{k \rightarrow \infty} \Psi(\mathbf{x}_k, \mathbf{y}_k) = \Psi(\bar{\mathbf{x}}, \bar{\mathbf{y}}). \quad (۵۴-۴)$$

از طرفی، با توجه به KL بودن تابع Ψ ، از رابطه‌ی (۲۹-۴) می‌توان نوشت

$$\psi'(\Psi(\mathbf{z}_k) - \Psi(\bar{\mathbf{z}})) \cdot \text{dist}(\mathbf{o}, \partial\Psi(\mathbf{z}_k)) \geq 1. \quad (۵۵-۴)$$

حال، استفاده از رابطه‌ی (۴۷-۴) نتیجه می‌دهد که

$$\psi'(\Psi(\mathbf{z}_k) - \Psi(\bar{\mathbf{z}})) \geq \rho_{xy}^{-1} \cdot \|\mathbf{z}_k - \mathbf{z}_{k-1}\|_2^{-1}. \quad (۵۶-۴)$$

با توجه به مقعر بودن تابع ψ ، داریم

$$\psi(\Psi(\mathbf{z}_k) - \Psi(\bar{\mathbf{z}})) - \psi(\Psi(\mathbf{z}_{k+1}) - \Psi(\bar{\mathbf{z}})) \geq \psi'(\Psi(\mathbf{z}_k) - \Psi(\bar{\mathbf{z}}))(\Psi(\mathbf{z}_k) - \Psi(\mathbf{z}_{k+1})). \quad (۵۷-۴)$$

برای راحتی ادامه‌ی بحث، تعریف می‌کنیم

$$\Delta_{p,q} \triangleq \psi(\Psi(\mathbf{z}_p) - \Psi(\bar{\mathbf{z}})) - \psi(\Psi(\mathbf{z}_q) - \Psi(\bar{\mathbf{z}})), \quad p, q \in \mathbb{N}, \quad (۵۸-۴)$$

و

$$C \triangleq \frac{\gamma \rho_{xy}}{\rho}. \quad (۵۹-۴)$$

در این صورت، با استفاده از رابطه‌ی (۳۷-۴) و با توجه به روابط بالا، می‌توان نوشت

$$\Delta_{k,k+1} \geq \frac{\|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^\gamma}{C \cdot \|\mathbf{z}_k - \mathbf{z}_{k-1}\|_2}, \quad (۶۰-۴)$$

یا

$$\|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^\gamma \leq C \cdot \Delta_{k,k+1} \cdot \|\mathbf{z}_k - \mathbf{z}_{k-1}\|_2. \quad (۶۱-۴)$$

حال، با استفاده از نامساوی $\gamma\sqrt{\alpha\beta} \leq \alpha + \beta$ برای $\alpha, \beta \geq 0$ ، بدست می‌آوریم

$$\gamma\|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2 \leq \|\mathbf{z}_k - \mathbf{z}_{k-1}\|_2 + C \cdot \Delta_{k,k+1}. \quad (۶۲-۴)$$

در ادامه، اثبات رابطه‌ی زیر مرور می‌شود

$$\sum_{i=l+1}^k \|\mathbf{z}_{i+1} - \mathbf{z}_i\|_2 \leq \|\mathbf{z}_{l+1} - \mathbf{z}_l\|_2 + C \cdot \Delta_{l+1, k+1}, \quad \forall k > l \quad (۶۳-۴)$$

که در آن، $l > 0$ اندیسی است به اندازه‌ی کافی بزرگ. برای این منظور، رابطه‌ی (۶۲-۴) را برای $i = l+1, \dots, k$ جمع می‌کنیم. خواهیم داشت

$$\begin{aligned} 2 \sum_{i=l+1}^k \|\mathbf{z}_{i+1} - \mathbf{z}_i\|_2 &\leq \sum_{i=l+1}^k \|\mathbf{z}_i - \mathbf{z}_{i-1}\|_2 + C \sum_{i=l+1}^k \Delta_{i, i+1} \\ &\leq \sum_{i=l+1}^k \|\mathbf{z}_{i+1} - \mathbf{z}_i\|_2 + \|\mathbf{z}_{l+1} - \mathbf{z}_l\|_2 + C \sum_{i=l+1}^k \Delta_{i, i+1} \\ &= \sum_{i=l+1}^k \|\mathbf{z}_{i+1} - \mathbf{z}_i\|_2 + \|\mathbf{z}_{l+1} - \mathbf{z}_l\|_2 + C \cdot \Delta_{l+1, k+1}, \end{aligned} \quad (۶۴-۴)$$

که در آن از این نکته استفاده شده است که $\Delta_{p, q} + \Delta_{q, r} = \Delta_{p, r}$ برای تمام $p, q, r \in \mathbb{N}$ از آنجایی که $\psi \geq 0$ برای هر $k > l$ داریم

$$\sum_{i=l+1}^k \|\mathbf{z}_{i+1} - \mathbf{z}_i\|_2 \leq \|\mathbf{z}_{l+1} - \mathbf{z}_l\|_2 + C \cdot \psi(\Psi(\mathbf{z}_{l+1}) - \Psi(\bar{\mathbf{z}})), \quad (۶۵-۴)$$

که (۶۳-۴) را نتیجه می‌دهد. رابطه‌ی فوق همچنین نشان می‌دهد که دنباله‌ی $\{\mathbf{z}_k\}_{k \in \mathbb{N}}$ یک دنباله‌ی کوشی^۱ است، یعنی از یک اندیسی به بعد، فاصله‌ی هر دو عضو دنباله به طور دلخواه کم می‌شود. به عبارت دقیق‌تر، برای هر $q > p > l$ داریم

$$\mathbf{z}_q - \mathbf{z}_p = \sum_{k=p}^{q-1} (\mathbf{z}_{k+1} - \mathbf{z}_k), \quad (۶۶-۴)$$

و بنابراین،

$$\|\mathbf{z}_q - \mathbf{z}_p\|_2 = \left\| \sum_{k=p}^{q-1} (\mathbf{z}_{k+1} - \mathbf{z}_k) \right\|_2 \leq \sum_{k=p}^{q-1} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2. \quad (۶۷-۴)$$

در اینصورت، با میل دادن l به سمت بی‌نهایت و با توجه به رابطه‌ی (۶۳-۴)، کوشی بودن دنباله اثبات می‌شود. از طرفی، می‌دانیم هر دنباله‌ی کوشی، همگرا است [۱۲۳]. بنابراین، دنباله‌ی $\{\mathbf{z}_k\}_{k \in \mathbb{N}}$ همگرا است. طبق روابط بدست آمده در (۶۲-۴)، می‌دانیم این نقطه‌ی همگرایی یک نقطه‌ی بحرانی از تابع Ψ است. به این ترتیب، اثبات قضیه کامل می‌شود.

۴-۶-۵ نرخ همگرایی

در این قسمت، نرخ همگرایی را برای الگوریتم PALM از [۸] مرور می‌کنیم. برای این منظور، ابتدا باید توجه داشت که برای توابع نیمه‌جبری، تابع مقعر ψ را می‌توان به فرم زیر انتخاب کرد [۸]

^۱Cauchy sequence

$$\psi(s) = c \cdot s^{1-\theta}, \quad (۴-۶۸)$$

که در آن، $c > 0$ و $0 \leq \theta < 1$. در اینصورت، برای نرخ همگرایی دنباله‌ی $\{z_k\}_{k \in \mathbb{N}}$ تولید شده توسط الگوریتم پراکسیمال می‌توان نوشت [۸]

• اگر $\theta = 0$ ، دنباله‌ی $\{z_k\}_{k \in \mathbb{N}}$ در یک تعداد محدودی تکرار همگرا می‌شود.

• اگر $0 < \theta \leq 0.5$ ، در این صورت $\omega > 0$ و $0 \leq \theta < 1$ وجود دارند به نحوی که $\|z_k - \bar{z}\|_2 \leq \omega T^k$.

• اگر $0.5 < \theta < 1$ ، در این صورت $\omega > 0$ وجود دارد به نحوی که

$$\|z_k - \bar{z}\|_2 \leq \omega k^{\frac{\theta-1}{1-\theta}}.$$

۷-۴ جمع‌بندی

در این فصل، یک دسته‌ی بسیار مهم از الگوریتم‌های کمینه‌سازی، معروف به الگوریتم‌های پراکسیمال را مرور کردیم. این الگوریتم‌ها به ویژه امروزه برای حل مسائل با ابعاد بالا محبوبیت فراوانی کسب کرده‌اند و عملکرد بسیار مطلوبی از خود نشان می‌دهند. از این مهم‌تر، این الگوریتم‌ها در چند سال گذشته به مسائل غیرمحدب نیز تعمیم داده شده‌اند. در سلسله مطالبی که در این فصل گذشت، ابتدا با مفهوم عملگر پراکسیمال یک تابع آشنا شده و با ذکر چند مثال مهم در پردازش تئیک، اهمیت آن را یادآور شدیم. سپس، ایده‌ی الگوریتم‌های پراکسیمال را برای حل مسائل با تابع هدف ترکیبی، یعنی هم شامل قسمت هموار و هم غیرهموار، یادآوری نموده، و در نهایت، گام‌های اساسی اثبات همگرایی این الگوریتم‌ها را مرور نمودیم.

در فصل بعد، الگوریتم‌های پیشنهادی خودمان برای بازیابی نمایش تئیک را، که عمدتاً بر مبنای الگوریتم‌های

پراکسیمال هستند، معرفی می‌کنیم.

الگوریتم‌های پیشنهادی برای کُدینگ تُنک

۱-۵ مقدمه

این فصل شروع معرفی الگوریتم‌های پیشنهادی این رساله است. به طور خاص، این فصل به الگوریتم‌های پیشنهادی برای نمایش تُنک اختصاص دارد که خود یک گام مهم از آموزش دیکشنری را تشکیل می‌دهد. الگوریتم‌های متنوعی در این فصل معرفی می‌شوند که در ساختار و ماهیت عموماً تفاوت‌هایی دارند اما همگی سعی در حل مسأله‌ی مقید بازیابی نمایش تُنک دارند.

این فصل با معرفی خانواده‌ی الگوریتم‌های تکراری تُنک‌سازی-تصویر^۱ یا به اختصار ISP آغاز می‌شود [۱۰۴]. الگوریتم‌های ISP در واقع تعمیمی از الگوریتم SL0 بوده و ایده‌ی زیربنایی مشابهی دارند. نکته‌ی کلیدی که برای توسعه این الگوریتم‌ها استفاده شده است مبتنی بر تعبیر گرادیان کاهش از عملگر پراکسیمال است. در واقع، همانطور که در ادامه خواهیم دید، الگوریتم‌های ISP گام گرادیان کاهش الگوریتم SL0 را با عملگر پراکسیمال متناظر با یک تابع تشویق‌کننده‌ی تُنکی کلی، مانند نُرم صفر و یا نُرم یک، جایگزین می‌کند. بعلاوه، این الگوریتم‌ها نسخه‌ای مقاوم به نویز دارند که برخلاف الگوریتم SL0 مقاومت خوبی نسبت به نویز از خود نشان می‌دهند.

سپس، الگوریتم تکراری پراکسیمال-تصویر^۲، یا IPP معرفی می‌شود. این الگوریتم مسأله‌ی کمینه‌سازی یک تابع تشویق‌کننده تُنک بودن ناهموار را برای بازیابی نمایش تُنک حل می‌کند. ساختار کلی این الگوریتم مشابه

^۱Iterative Sparsification-Projection

^۲Iterative Proximal-Projection

الگوریتم ISP است، اما ناهموار بودن تابع هدف تفاوت‌هایی را ایجاد می‌کند. همچنین، نسخه‌ای با همگرایی سریع‌تر از این الگوریتم‌ها معرفی می‌شود که مبتنی بر استفاده از ایده‌های سرعت‌بخشی الگوریتم‌های پراکسیمال برای مسائل محدب است. علاوه بر این، اثبات همگرایی این الگوریتم‌ها نیز ارائه می‌شود.

در ادامه، الگوریتم دیگری به نام الگوریتم نُرم صفر-آستانه‌گذاری نُرم یا L0Soft معرفی می‌شود. ساختار این الگوریتم با الگوریتم‌های قبلی فرق داشته و مبتنی بر ایده‌ای متفاوت است. در واقع، این الگوریتم مانند الگوریتم SL0 قصد دارد تابع ناهموار نُرم صفر را با یک تابع پیوسته جایگزین نماید که برخلاف تابع SL0، مشتق‌پذیر نیست. در این الگوریتم، با استفاده از یک تعریف خاص از تابع نُرم صفر، مسأله‌ی کمینه‌سازی نُرم صفر به یک مسأله‌ی کمینه‌سازی نُرم یک تبدیل شده و با استفاده از الگوریتم‌های پراکسیمال حل می‌شود.

۲-۵ الگوریتم تُنک‌سازی-تصویرسازی پیاپی (ISP)

در این قسمت، الگوریتم‌های ISP معرفی می‌شوند. برای این منظور، ابتدا لازم است تا مروری دیگر داشته باشیم بر الگوریتم SL0 که در واقع نقطه‌ی شروع و الهام‌بخش الگوریتم ISP است. این مرور شامل بدست آوردن الگوریتم SL0 با استفاده از الگوریتم‌های پراکسیمال است که در نتیجه‌ی آن، بازه‌ی طول گام‌گرایان کاهش می‌یابد، که همگرایی دنباله را تضمین می‌کند، بدست می‌آید. بعلاوه، نشان داده می‌شود که مرحله‌ی گرایان کاهش در واقع نوعی عملیات آستانه‌گذاری را پیاده‌سازی می‌کند که منجر به تُنک شدن جواب می‌شود.

۱-۲-۵ بازینی الگوریتم نُرم صفر هموار شده (SL0)

همانطور که در فصل ۲ بیان شد، ایده‌ی اصلی الگوریتم SL0 عبارت است از تقریب زدن تابع ناهموار نُرم صفر با یک تابع مشتق‌پذیر و خوش‌رفتار. ما این تابع را با $\|\cdot\|_\sigma$ نشان می‌دهیم که به صورت زیر تعریف می‌شود

$$\|\mathbf{x}\|_\sigma \triangleq n - \sum_{i=1}^n \exp\left(-\frac{x_i^2}{\sigma^2}\right). \quad (1-5)$$

زمانی که $\sigma \rightarrow 0$ ، تابع فوق به سمت تابع نُرم صفر میل می‌کند. با این تعریف، مسأله‌ی متناظر با الگوریتم SL0 به صورت زیر فرمول‌بندی می‌شود

$$\min_{\mathbf{x}} \|\mathbf{x}\|_\sigma \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}. \quad (2-5)$$

این الگوریتم سپس دنباله‌ای از مسائل فوق را به ازای مقادیر کاهشی از σ حل می‌کند، که هر مسأله با جواب نهایی مسأله‌ی قبلی مقداردهی می‌شود. بعلاوه، برای هر مسأله از یک استراتژی دو مرحله‌ای گرادیان و تصویر استفاده می‌شود. در گام گرادیان، اندکی در جهت عکس گرادیان تابع هدف حرکت کرده و سپس نتیجه را روی مجموعه‌ی قید (۲-۵) تصویر می‌کنیم. برای راحتی ادامه‌ی بحث، این مجموعه‌ی قید را بصورت زیر نشان می‌دهیم

$$C \triangleq \{\mathbf{x} \mid \mathbf{y} = \mathbf{D}\mathbf{x}\}. \quad (۳-۵)$$

در حالت کلی، مسأله‌ی حل شده توسط الگوریتم SL0 را می‌توان به فرم زیر نوشت

$$\min_{\mathbf{x}} \{F_{\sigma}(\mathbf{x}) \triangleq \|\mathbf{x}\|_{\sigma} + \mathcal{I}(\mathbf{x})\}, \quad (۴-۵)$$

که در آن، \mathcal{I} تابع مشخصه‌ی مجموعه‌ی C است که بصورت زیر تعریف می‌شود

$$\mathcal{I}(\mathbf{x}) \triangleq \begin{cases} 0 & \mathbf{x} \in C \\ +\infty & \mathbf{x} \notin C \end{cases} \quad (۵-۵)$$

حال اگر تعریف کنیم $f_{\sigma}(\mathbf{x}) \triangleq \|\mathbf{x}\|_{\sigma}$ و $g(\mathbf{x}) \triangleq \mathcal{I}(\mathbf{x})$ ، در اینصورت به راحتی می‌توان دید که مسأله‌ی (۴-۵) در قالب کلی مسائل قابل حل توسط الگوریتم‌های پراکسیمال است (مسأله‌ی (۴-۹) از فصل قبل را ببینید). نکته‌ای که باقی می‌ماند این است که نشان دهیم گرادیان تابع f_{σ} لیپشیتز است. این موضوع در لم زیر اثبات می‌شود.

لم ۱-۵ گرادیان تابع $f_{\sigma}(\mathbf{x})$ لیپشیتز بوده و ثابت لیپشیتز آن عبارت است از $L = \frac{1}{\sigma^2}$.

اثبات. برای بدست آوردن ثابت لیپشیتز ∇f_{σ} ، ابتدا ثابت می‌کنیم که اگر داشته باشیم $F(\mathbf{x}) \triangleq \sum_{i=1}^n f(x_i)$ ، که در آن $\text{dom}_F = \mathbb{R}^n$ و $f: \mathbb{R} \rightarrow \mathbb{R}$ لیپشیتز با ثابت L است، در اینصورت، تابع F هم گرادیانی لیپشیتز با همان ثابت L خواهد داشت. برای اثبات این موضوع، توجه داریم که گرادیان F عبارت است از

$$\nabla F(\mathbf{x}) = [f'(x_1), \dots, f'(x_n)]^T.$$

$$\begin{aligned} \|\nabla F(\mathbf{x}) - \nabla F(\mathbf{z})\|_2 &= \sqrt{\sum_{i=1}^n (f'(x_i) - f'(z_i))^2} \\ &\leq \sqrt{\sum_{i=1}^n L^2 \cdot (x_i - z_i)^2} \\ &= L \cdot \|\mathbf{x} - \mathbf{z}\|_2, \end{aligned} \quad (۶-۵)$$

سپس، برای همه‌ی $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$ خواهیم داشت

حال، دقت کنید که $f_\sigma(\mathbf{x}) = \sum_{i=1}^n (1 - \exp(-\frac{x_i^2}{\sigma^2}))$. بنابراین، با استفاده از نتیجه‌ی فوق کافی است ثابت لپشیتز

تابع $f(x) = 1 - \exp(-\frac{x^2}{\sigma^2})$ را محاسبه کنیم. برای مشتق دوم تابع f که برابر است با

$$f''(x) = \frac{2}{\sigma^2} (1 - \frac{2x^2}{\sigma^2}) \exp(-\frac{x^2}{\sigma^2}),$$

داریم $\forall x: |f''(x)| \leq (2/\sigma^2)$ که با استفاده از قضیه مقدار میانی^۱ به این نتیجه می‌رسیم که ثابت لپشیتز f' برابر

است با $L = 2/\sigma^2$. در نتیجه، ثابت لپشیتز ∇f_σ هم عبارت است از $2/\sigma^2$.

حال برمی‌گردیم به مسأله‌ی (۴-۵). حل این مسأله به کمک الگوریتم‌های پراکسیمال بصورت زیر خواهد

بود

$$\mathbf{x}_{k+1} = \text{prox}_{\mu g}(\mathbf{x}_k - \mu \sigma \nabla f_\sigma(\mathbf{x}_k)). \quad (۷-۵)$$

توجه داریم که تابع g یک تابع مشخصه است و از نتایج فصل گذشته می‌دانیم که عملگر پراکسیمال یک تابع

مشخصه عبارت است از عملگر تصویر روی مجموعه‌ی متناظر، که در مورد تابع g می‌شود عملگر تصویر روی

مجموعه‌ی C . در نتیجه، این الگوریتم دقیقاً همان دو گام گرادیان کاهشی و تصویر کردن الگوریتم SL0 است.

همگرایی دنباله‌ی تولید شده در (۷-۵) در قضیه‌ی زیر بیان شده است.

قضیه ۱-۵ فرض کنید $\{\mathbf{x}_k\}_{k \geq 0}$ دنباله‌ی تولید شده در (۷-۵) باشد. در اینصورت، این دنباله به یک نقطه‌ی

بحرانی تابع هدف (۴-۵) همگرا می‌شود. بعلاوه، اگر $\mu, \sigma \in (0, \sigma^2/2]$ یا عبارتی $\mu \in (0, 1/2]$ ، در اینصورت،

دنباله‌ی مقادیر تابع هدف، یعنی $\{F_\sigma(\mathbf{x}_k)\}_{k \geq 0}$ غیرافزایشی است.

اثبات. ابتدا توجه داریم که دنباله‌ی تولید شده در (۷-۵)، در واقع جواب مسأله‌ی زیر است

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\text{argmin}} \left\{ \nabla f_\sigma(\mathbf{x}_k)^T (\mathbf{x} - \mathbf{x}_k) + \frac{1}{\sqrt{\mu\sigma}} \|\mathbf{x} - \mathbf{x}_k\|_2 + g(\mathbf{x}) \right\} \quad (۸-۵)$$

در نتیجه، شروط لازم برای بهینگی \mathbf{x}_{k+1} را می‌توان بصورت زیر نوشت

$$\nabla f_\sigma(\mathbf{x}_k)^T (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{1}{\sqrt{\mu\sigma}} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 + g(\mathbf{x}_{k+1}) \leq g(\mathbf{x}_k) \quad (۹-۵)$$

و

$$0 \in \partial g(\mathbf{x}_{k+1}) + \nabla f_\sigma(\mathbf{x}_k) + \frac{1}{\mu\sigma} (\mathbf{x}_{k+1} - \mathbf{x}_k), \quad (۱۰-۵)$$

که در رابطه‌ی (۱۰-۵) از این نکته استفاده شده است که برای تابع $h = f + g$ ، که f بصورت پیوسته مشتق‌پذیر

و g محدب است، می‌توان نوشت $\partial h(\mathbf{x}) = \nabla f(\mathbf{x}) + \partial g(\mathbf{x})$. [۹۵]. از طرف دیگر، چون ∇f_σ لپشیتز است (لم

^۱Mean value theorem

(۱-۵)، می‌توان نوشت

$$f_{\sigma}(\mathbf{x}_{k+1}) \leq f_{\sigma}(\mathbf{x}_k) + \nabla f_{\sigma}(\mathbf{x}_k)^T (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{L}{\nu} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\nu}^2. \quad (11-5)$$

حال، جمع زدن دو سمت روابط (۹-۵) و (۱۱-۵) نتیجه می‌دهد

$$f_{\sigma}(\mathbf{x}_{k+1}) + g(\mathbf{x}_{k+1}) \leq f_{\sigma}(\mathbf{x}_k) + g(\mathbf{x}_k) - \left(\frac{1}{\nu\mu_{\sigma}} - \frac{L}{\nu}\right) \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\nu}^2. \quad (12-5)$$

رابطه‌ی فوق نشان می‌دهد که اگر $\mu_{\sigma} \in (0, \frac{1}{L}]$ ، در اینصورت دنباله‌ی مقادیر تابع هدف، یعنی $\{F_{\sigma}(\mathbf{x}_k)\}_{k=0}^{\infty}$ ، غیرافزایشی خواهد بود. از آنجایی که F_{σ} از پائین کراندار است، بنابراین دنباله‌ی $\{F_{\sigma}(\mathbf{x}_k)\}_{k=0}^{\infty}$ همگرا می‌شود.

جمع زدن رابطه‌ی (۱۲-۵) برای $k = 0, \dots, \infty$ نتیجه می‌دهد

$$\sum_{k=0}^{\infty} \left\{ \left(\frac{1}{\nu\mu_{\sigma}} - \frac{L}{\nu} \right) \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\nu}^2 \right\} \leq F_{\sigma}(\mathbf{x}_0) - F_{\sigma}(\mathbf{x}_{\infty}) \quad (13-5)$$

که با توجه به نامنفی بودن سمت راست این رابطه، نتیجه می‌شود که $\mathbf{x}_{k+1} \rightarrow \mathbf{x}_k$. از طرف دیگر، طبق رابطه‌ی

(۷-۵)، بردار تخمین در هر تکرار روی مجموعه‌ی C ، که یک مجموعه‌ی کراندار است، تصویر می‌شود. در نتیجه،

دنباله‌ی $\{\mathbf{x}_k\}_{k=0}^{\infty}$ کراندار است. بنابراین، طبق قضیه‌ی Bolzano–Weierstrass [۶۸]، یک زیردنباله‌ی همگرا مانند

$\{\mathbf{x}_{k_j}\}_{j=0}^{\infty}$ از $\{\mathbf{x}_k\}_{k=0}^{\infty}$ وجود دارد که به یک نقطه‌ی \mathbf{x}^* همگرا می‌شود. در ادامه، نشان می‌دهیم که \mathbf{x}^* یک نقطه‌ی

بحرانی F_{σ} است. برای این منظور، تعریف می‌کنیم

$$\mathbf{u}_j \triangleq \nabla f_{\sigma}(\mathbf{x}_{k_j}) - \nabla f_{\sigma}(\mathbf{x}_{k_j-1}) - \frac{1}{\mu_{\sigma}} (\mathbf{x}_{k_j} - \mathbf{x}_{k_j-1}). \quad (14-5)$$

حال، از (۱۰-۵) نتیجه می‌شود که $\mathbf{u}_j \in \partial F_{\sigma}(\mathbf{x}_{k_j})$. سپس، با استفاده از ویژگی لیشیتز بودن ∇f_{σ} داریم

$$\|\mathbf{u}_j\|_2 \leq L \|\mathbf{x}_{k_j} - \mathbf{x}_{k_j-1}\|_2 + \frac{1}{\mu_{\sigma}} \|\mathbf{x}_{k_j} - \mathbf{x}_{k_j-1}\|_2 \rightarrow 0 \quad (15-5)$$

که در آن از این نکته استفاده شده است که $\mathbf{x}_{k+1} - \mathbf{x}_k \rightarrow 0$. بنابراین، $\mathbf{u}_j \rightarrow 0$. از طرف دیگر، با توجه به پیوسته

بودن f_{σ} و از پائین نیمه-پیوسته بودن g ، داریم $F_{\sigma}(\mathbf{x}_{k_j}) \rightarrow F_{\sigma}(\mathbf{x}^*)$. در نهایت، با استفاده از لم (۱-۴) داریم

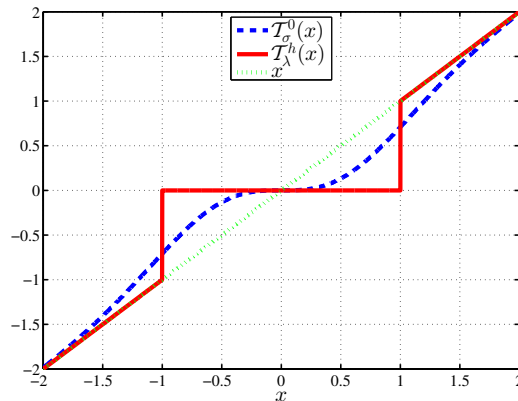
$$0 \in \partial F_{\sigma}(\mathbf{x}^*),$$

که اثبات را کامل می‌کند.

یک نتیجه‌ی مهمی که از قضیه فوق استخراج می‌شود شرط لازمی برای طول گام‌گرایان کاهش‌ی است به

گونه‌ای که همگرایی دنباله را تضمین می‌کند. الگوریتم SL0 چنین شرطی نداشته و بنابراین برای تعیین طول گام

دستورالعمل خاصی ارائه نکرده است.



شکل ۵-۱: تابع آستانه‌گذاری نرم صفر هموار شده (۵-۱۶) به همراه تابع آستانه‌گذاری سخت.

۵-۲-۲ الگوریتم پیشنهادی

در این قسمت، الگوریتم پیشنهادی برای بازیابی تُنک ارائه می‌شود. برای این منظور، ابتدا توجه کنید که مرحله‌ی گرادین کاهشی الگوریتم SL0 در حقیقت عبارت است از اعمال تابع اسکالر زیر به تک تک مؤلفه‌های بردار مورد

تخمین

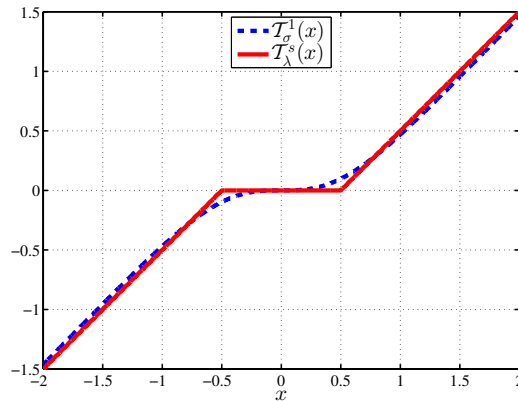
$$x \leftarrow T_\sigma(x) \triangleq x \cdot (1 - \exp(-\frac{x^2}{\sigma^2})), \quad (16-5)$$

که برای آن فرض شده است $\mu = 1/2$. همچنین، با توجه به جدائی‌پذیر بودن تابع نرم صفر هموار شده، تنها حالت اسکالر بیان شده است. دقت کنید که تابع T_σ در واقع یک عملگر تُنک‌ساز است. برای یک دید بهتر، این تابع به همراه تابع آستانه‌گذاری سخت در شکل ۵-۱ رسم شده است. توجه داریم که در این شکل، $T_\lambda^h(x) = \text{prox}_{\lambda \|\cdot\|_1}(x)$. همانطور که از این شکل مشخص است، تابع آستانه‌گذاری نرم صفر هموار شده در جهت تُنک‌سازی ورودی خود عمل می‌کند، به این ترتیب که ورودی‌هایی را که دامنه‌ی آن‌ها از حدی بزرگتر است بدون تغییر می‌گذارد، اما ورودی‌های دیگر را متناسب با دامنه‌ی آن‌ها منقبض می‌کند. در واقع، برخلاف تابع آستانه‌گذاری سخت که ورودی‌های از حدی کمتر را به طور کامل صفر می‌کند، تابع (۵-۱۶) رفتاری هموارتر برای چنین ورودی‌هایی دارد.

به عنوان یک مثال دیگر از توابع هموار تشویق‌کننده‌ی تُنک بودن، تابع زیر را در نظر بگیرید

$$f_\sigma^1(x) = \frac{x^2}{\sqrt{x^2 + \sigma^2}}. \quad (17-5)$$

فرض کنید این تابع را به صورت مؤلفه به مؤلفه روی یک بردار \mathbf{x} اعمال کنیم، یعنی $F_\sigma^1(\mathbf{x}) \triangleq \sum_i f_\sigma^1(x_i)$. به این



شکل ۵-۲: تابع آستانه‌گذاری نُرم یک هموار شده به همراه تابع آستانه‌گذاری نُرم.

ترتیب، می‌توان نشان داد که تابع F_σ^1 برای $\sigma \rightarrow 0$ به سمت نُرم یک میل می‌کند. در نتیجه، تابع فوق یک نسخه‌ی هموار شده از تابع نُرم یک است. به همین دلیل، ما این تابع را در ادامه، تابع نُرم یک هموار شده می‌نامیم و آن را به اختصار با SL1 نمایش می‌دهیم.

مشابه روالی که در لم ۵-۱ طی شد، می‌توان نشان داد که گرادیان تابع f_σ^1 با ثابتی برابر با $L = 2/\sigma$ لیبشیتز است. تابع آستانه‌گذاری متناظر با تابع SL1 عبارت است از $T_\sigma^1(x) = x - \mu_\sigma \nabla f_\sigma^1(x)$ که برای $\mu_\sigma = 1/L$ به همراه تابع آستانه‌گذاری نُرم یک، در شکل ۵-۲ رسم شده است. در این شکل، $T_\lambda^s(x) = \text{prox}_{\lambda \|\cdot\|_1}(x)$ همانطور که از این شکل مشخص است، تابع $T_\sigma^1(x)$ به خوبی تابع آستانه‌گذاری نُرم یک را تقریب می‌زند و در واقع تقریبی هموار از آن ارائه می‌دهد. برای بدست آوردن رابطه‌ی بین پارامتر σ در این تابع، و پارامتر آستانه‌ی λ در تابع آستانه‌گذاری نُرم، دقت داریم که در بی‌نهایت رابطه‌ی $x - \mu_\sigma = x - \lambda$ برقرار است. بنابراین، داریم $\lambda = \mu_\sigma = \sigma/2$.

این رفتارها در واقع نشان‌دهنده‌ی یک رابطه‌ی جالب میان توابع هموار تشویق‌کننده‌ی تُنک‌بودن و توابع آستانه‌گذاری (عملگرهای پراکسیمال) توابع ناهموار متناظر آنها است. در واقع، این موضوع از تعبیر گرادیان کاهشی برای عملگر پراکسیمال توابع هموار ناشی می‌شود که در فصل قبل آن را بررسی کردیم. برای یادآوری، عملگر پراکسیمال یک تابع هموار f به صورت تقریبی برابر است با یک گام گرادیان کاهشی با فرض یک طول گام به اندازه‌ی کافی کوچک. به عبارت دیگر،

$$\text{prox}_{\mu f}(x) \approx x - \mu \nabla f(x). \quad (18-5)$$

اکنون، تابع $f(x) = |x|$ را در نظر بگیرید. این تابع در نقطه‌ی صفر هموار نیست و بنابراین ویژگی فوق را نمی‌توان در مورد آن به کار گرفت. اما اگر از تقریب هموار آن یعنی $f_\sigma^1(x)$ استفاده کنیم، انتظار داریم که ویژگی فوق

- هدف: محاسبه‌ی نمایش تُنک y برحسب اتم‌های D
- ورودی: $y, D, \tau_f, \tau_0, T_\tau^*(\cdot), I, 0 < c < 1$
- مقداردهی اولیه: $\tau = \tau_0, x = D^+ y$
- شروع الگوریتم: گام‌های زیر را تا رسیدن به شرط توقف $\tau > \tau_f$ انجام بده
- برای $i = 1, 2, \dots, I$ انجام بده
- ۱. تُنک‌سازی: $\bar{x} = T_\tau^*(x)$
- ۲. تصویر کردن: $x = \operatorname{argmin}_{u \in \mathcal{A}_\epsilon} \|u - \bar{x}\|_2$
- $\tau \leftarrow c \cdot \tau$
- خروجی: x^k

شکل ۵-۳: الگوریتم تکراری تُنک‌سازی-تصویر کردن (ISP).

همچنان با دقت خوبی و برای یک σ به اندازه‌ی کافی کوچک برقرار باشد. به عبارت دیگر،

$$\operatorname{prox}_{\mu_\sigma f}(x) = T_{\mu_\sigma}^s(x) \approx x - \mu_\sigma \nabla f_\sigma^1(x). \quad (19-5)$$

دقت این تقریب را بار دیگر می‌توان از شکل ۵-۲ مشاهده نمود. نتیجه‌ای که می‌توان از این مشاهدات گرفت این است که یک گام گرادیان کاهشی روی تقریب هموار تابع l_1 با دقت خوبی مانند آستانه‌گذاری نرم عمل می‌کند، که پارامتر آستانه برای آن برابر با طول گام گرادیان کاهشی است. نتیجه‌گیری مشابهی در مورد تابع نرم صفر و نسخه‌ی هموار آن برقرار است. رابطه‌ی بین پارامتر آستانه و طول گام نیز مشابه قبل است، یعنی $\lambda = \mu_\sigma$.

به این ترتیب، می‌توان از این بحث نتیجه گرفت که همان روال الگوریتم SL0 با توابع آستانه‌گذاری مختلف قابل اعمال است. این مشاهده ما را به الگوریتم‌های پیشنهادی، که آن را ISP^۱ نامیده‌ایم، رهنمون می‌کند. در واقع، ایده‌ی اصلی الگوریتم‌های ISP این است که گام گرادیان کاهشی الگوریتم SL0 را با یک تابع آستانه‌گذاری کلی جایگزین کنیم. این تابع، یا عملگر پراکسیمال یک تابع ناهموار است، مانند تابع آستانه‌گذاری سخت و یا نرم، یا عبارت است از یک گام گرادیان کاهشی روی یک تابع هموار مانند تابع SL0 یا تابع SL1. این الگوریتم در شکل ۵-۳ خلاصه شده است. در این شکل، T_τ^* عبارت است از یک تابع آستانه‌گذاری، که به نحوی که اشاره شد می‌تواند به طرق گوناگون انتخاب شود. علاوه بر این، τ_0 و τ_f به ترتیب، مقدار اولیه و مقدار نهایی پارامتر آستانه τ بوده، $0 < c < 1$ یک ضریب کاهش برای τ است، و I نشان‌دهنده‌ی تعداد تکرارهای حلقه داخلی الگوریتم است. لازم به ذکر است که حلقه‌ی بیرونی الگوریتم متناظر با پارامترهای τ بوده و حلقه‌ی درونی عبارت است از

^۱Iterative Sparsification-Projection

اعمال دو گام تُنک‌سازی و تصویر کردن برای یک مقدار خاص از پارامتر آستانه.

بسته به انتخاب توابع آستانه‌گذاری مختلف در الگوریتم ISP، الگوریتم‌های متنوعی با عملکردهای مختلف تحقق می‌یابند. به طور خاص، الگوریتم‌های ISP-Soft، ISP-Hard، ISP- l_0 و ISP- l_1 به ترتیب متناظر با توابع آستانه T_r^1 ، T_r^0 ، T_r^s ، T_r^h هستند. الگوریتم‌های مبتنی بر گرادیان، مانند ISP- l_0 ، یک پارامتر اضافی نسبت به الگوریتم‌های مبتنی بر عملگر پراکسیمال دارند. این پارامتر اضافی همان طول گام گرادیان کاهش‌ی است که می‌توان آن را برای داشتن یک عملکرد مطلوب تنظیم کرد.

نکته‌ی دیگری که لازم است بررسی شود، گام تصویر کردن در الگوریتم ISP است. برای این گام، باید

مسأله‌ی زیر حل شود

$$\min_{\mathbf{x}} \frac{1}{\nu} \|\mathbf{x} - \bar{\mathbf{x}}\|_{\nu}^2 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon, \quad (5-20)$$

که در آن، $\bar{\mathbf{x}}$ خروجی گام تُنک‌سازی است و ϵ پارامتری است که به سطح نویز بستگی دارد. نسخه‌ای مقاوم برای الگوریتم SL0 توسط افتخاری و دیگران [۵۰] معرفی شده است. برای توضیح این روش، که الگوریتم Robust-SL0 نامیده می‌شود، ابتدا مجموعه‌ی زیر را تعریف می‌کنیم

$$C_{\epsilon} \triangleq \{\mathbf{x} : \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon\}. \quad (5-21)$$

در این صورت، برای انجام گام تصویر کردن، نویسندگان [۵۰] پیشنهاد می‌دهند که اگر $\bar{\mathbf{x}} \notin C_{\epsilon}$ آنگاه $\bar{\mathbf{x}}$ را روی مجموعه‌ی $C = \{\mathbf{x} : \mathbf{y} = \mathbf{D}\mathbf{x}\}$ تصویر کرده و در غیر اینصورت، آن را بعنوان به‌روزرسانی تخمین قبول می‌کنیم. تصویر $\bar{\mathbf{x}}$ روی C با عبارت $(\mathbf{D}\mathbf{D}^T)^{-1}(\mathbf{D}\bar{\mathbf{x}} - \mathbf{y})$ - $\bar{\mathbf{x}}$ داده می‌شود [۹۲].

ایده‌ی فوق در واقع یک راه تقریبی برای حل (۵-۲۰) ارائه می‌دهد. ما در اینجا یک الگوریتم دیگر برای حل این مسأله معرفی می‌کنیم. الگوریتم پیشنهادی مبتنی بر روش معروف ADMM^۱ است [۲۲]. این الگوریتم برای حالت کلی دیکشنری است، یعنی زمانی که دیکشنری ساختار خاصی ندارد. اما اگر دیکشنری \mathbf{D} به گونه‌ای باشد که $\mathbf{D}\mathbf{D}^T = \alpha\mathbf{I}$ ، برای یک $\alpha > 0$ ، در این صورت، همانطور که در ادامه نشان داده می‌شود، مسأله‌ی (۵-۲۰) جواب فُرم بسته دارد. لازم به ذکر است که چنین قیدی روی دیکشنری چندان دور از کاربردهای عملی نیست. بعنوان مثال، اگر دیکشنری زیرماتریسی از ماتریس تبدیل فوریه^۳ باشد، این ویژگی برقرار است؛ چرا که

^۱ Alternating Direction Method of Multipliers

^۲ در این حالت، ماتریس یک Tight Frame نامیده می‌شود [۷۴]. با این مفهوم در فصل ۳ آشنا شدیم.

^۳ Discrete Fourier Transform

ماتریس تبدیل فوریه گسسته یک ماتریس متعامد نُرمال است. مثال‌های دیگر عبارتند از تبدیل کسینوسی گسسته^۱ و تبدیل هادامارد^۲ [۱۵]. علاوه بر این، نشان داده شده است که چنین دیکشنری‌هایی عملکرد خوبی هم از خود نشان می‌دهند [۳۵].

برای حل مسأله‌ی (۵-۲۰) در حالتی که دیکشنری، یک Tight Frame است، ابتدا تابع لاگرانژ^۳ متناظر با

آن را بصورت زیر تشکیل می‌دهیم

$$L(\mathbf{x}, \lambda) = \frac{1}{\alpha} \|\mathbf{x} - \bar{\mathbf{x}}\|_{\alpha}^{\alpha} + \lambda (\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_{\alpha}^{\alpha} - \epsilon^{\alpha}), \quad (22-5)$$

که در آن، λ ضریب لاگرانژ است. شرایط KKT^۴ برای مسأله‌ی (۵-۲۰) بصورت زیر خواهند بود

$$\begin{cases} \mathbf{x}^* = (\mathbf{I} + \lambda^* \cdot \mathbf{D}^T \mathbf{D})^{-1} (\bar{\mathbf{x}} + \lambda^* \cdot \mathbf{D}^T \mathbf{y}) \\ \|\mathbf{y} - \mathbf{D}\mathbf{x}^*\|_{\alpha}^{\alpha} = \epsilon^{\alpha} \\ \lambda^* \geq 0 \end{cases} \quad (23-5)$$

که منجر به حل مسأله‌ی زیر برای تعیین λ^* می‌شود

$$\|\mathbf{y} - \mathbf{D}(\mathbf{I} + \lambda^* \mathbf{D}^T \mathbf{D})^{-1} (\bar{\mathbf{x}} + \lambda^* \mathbf{D}^T \mathbf{y})\|_{\alpha}^{\alpha} = \epsilon^{\alpha}. \quad (24-5)$$

برای حل مسأله‌ی فوق، از فرض Tight Frame بودن برای دیکشنری و نیز لم معکوس ماتریس^۵ [۶۹] استفاده

می‌کنیم. به این ترتیب، جمله‌ی حاوی معکوس ماتریس در عبارت \mathbf{x}^* به فرم زیر ساده می‌شود

$$(\mathbf{I} + \lambda^* \mathbf{D}^T \mathbf{D})^{-1} = \mathbf{I} - \frac{\lambda^*}{(1 + \lambda^* \alpha)} \mathbf{D}^T \mathbf{D}, \quad (25-5)$$

که اگر آن را در عبارت (۲۴-۵) قرار داده و نسبت به λ^* حل کنیم خواهیم داشت

$$\begin{cases} \lambda^* = \frac{1}{\alpha} \max(\frac{\|\mathbf{y} - \mathbf{D}\bar{\mathbf{x}}\|_{\alpha}^{\alpha}}{\epsilon^{\alpha}} - 1, 0) \\ \mathbf{x}^* = \bar{\mathbf{x}} + \frac{\lambda^*}{1 + \lambda^* \alpha} \mathbf{D}^T (\mathbf{y} - \mathbf{D}\bar{\mathbf{x}}) \end{cases}. \quad (26-5)$$

در ادامه، حالت کلی دیکشنری را در نظر می‌گیریم. برای این منظور، فرمول‌بندی معادل زیر را برای (۵-۲۰) در

نظر بگیرید

$$\min_{\mathbf{x}, \mathbf{z}} \frac{1}{\alpha} \|\mathbf{x} - \bar{\mathbf{x}}\|_{\alpha}^{\alpha} \quad \text{s.t.} \quad \|\mathbf{z}\|_{\alpha}^{\alpha} \leq \epsilon, \quad \mathbf{z} = \mathbf{y} - \mathbf{D}\mathbf{x}, \quad (27-5)$$

که در آن، \mathbf{z} یک متغیر کمکی است. برای حل مسأله‌ی فوق به کمک الگوریتم ADMM، ابتدا تابع لاگرانژین

افزوده^۶ را به فرم زیر تشکیل می‌دهیم [۲۲]

^۱Discrete Cosine Transform

^۲Hadamard Transform

^۳Lagrangian function

^۴Karush-Kuhn-Tucker

^۵Woodbury matrix inversion lemma

^۶Augmented Lagrangian function

• هدف: حل مسأله‌ی (۲۰-۵) با استفاده از ADMM

• ورودی: $y, D, \bar{x}, \gamma > 0$

• مقداردهی اولیه: $\lambda = 0, x = \bar{x}, A = (I + \gamma D^T D)^{-1}$

• شروع الگوریتم: گام‌های زیر را تا رسیدن به شرط توقف $\|y - Dx\|_2 \leq \epsilon$ انجام بده

۱. $z = \mathcal{P}_{\mathcal{A}_z}(y - Dx + \frac{1}{\gamma}\lambda)$

۲. $x = A(\bar{x} + \gamma D^T(y - z + \frac{1}{\gamma}\lambda))$

۳. $\lambda = \lambda - \gamma(z - y + Dx)$

• خروجی: x

شکل ۴-۵: الگوریتم پیشنهادی برای حل مسأله‌ی (۲۰-۵).

$$L(x, z, \lambda) = \frac{1}{\gamma} \|x - \bar{x}\|_2^2 - \lambda^T (z - y + Dx) + \frac{\gamma}{2} \|z - y + Dx\|_2^2, \quad (28-5)$$

که در آن، $\gamma > 0$ نرخ همگرایی الگوریتم کلی را تعیین می‌کند. دقت کنید که در عبارت فوق، قید $\|z\|_2 \leq \epsilon$ را لحاظ نکرده‌ایم. در واقع، این قید زمانی که بهینه‌سازی روی z انجام می‌شود در نظر گرفته خواهد شد. در ادامه، تابع فوق بصورت تکراری، یکبار روی x و بار دیگر روی z کمینه می‌شود. بردار ضرایب لاگرانژ λ هم طبق استاندارد روش ADMM فرمول به‌روزرسانی خودش را دارد. این به‌روزرسانی‌ها به فرم زیر هستند

$$\begin{cases} z_{k+1} = \operatorname{argmin}_{z: \|z\|_2 \leq \epsilon} L(x_k, z, \lambda_k) \\ x_{k+1} = \operatorname{argmin}_x L(x, z_{k+1}, \lambda_k) \\ \lambda_{k+1} = \lambda_k - \gamma(z_{k+1} - y + Dx_{k+1}) \end{cases} \quad (29-5)$$

به‌روزرسانی z بعد از کمی ساده‌سازی بصورت معادل زیر تبدیل می‌شود

$$z_{k+1} = \operatorname{argmin}_{z: \|z\|_2 \leq \epsilon} \frac{1}{\gamma} \|z - y + Dx_k - \frac{1}{\gamma}\lambda_k\|_2^2, \quad (30-5)$$

که دارای جواب فرم بسته‌ی زیر است

$$z_{k+1} = \mathcal{P}_{\mathcal{A}_z}(y - Dx_k + \frac{1}{\gamma}\lambda_k), \quad (31-5)$$

که در آن، $\mathcal{P}_{\mathcal{A}_z}(\cdot)$ عملگر تصویر روی مجموعه‌ی $\{z : \|z\|_2 \leq \epsilon\}$ است. به عبارت دیگر،

$$\mathcal{P}_{\mathcal{A}_z}(z) \triangleq \begin{cases} z & z \in \mathcal{A}_z \\ \frac{\epsilon}{\|z\|_2} \cdot z & \text{oth.} \end{cases}$$

برای به‌روزرسانی x هم داریم

$$x_{k+1} = (I + \gamma D^T D)^{-1}(\bar{x} + \gamma D^T(y - z_{k+1} + \frac{1}{\gamma}\lambda_k)). \quad (32-5)$$

الگوریتم نهایی برای حل مسأله‌ی (۲۰-۵) در شکل ۴-۵ خلاصه شده است.

۳-۲-۵ آنالیز همگرایی

تحلیل همگرایی الگوریتم‌های ISP که مبتنی بر گرادیان کاهشی هستند را به راحتی می‌توان با استفاده از اثبات همگرایی الگوریتم‌های پراکسیمال که در فصل قبل مرور شدند انجام داد. در واقع، در حالت هموار، الگوریتم‌های ISP ساختاری کاملاً مشابه الگوریتم SL0 دارند و اگر شرط لپشیتز بودن برای گرادیان آن‌ها برقرار باشد (که عمدتاً چنین نیز هست)، همگرایی آن‌ها روالی کاملاً مشابه آنچه در فصل پیش دیدیم خواهد داشت.

در حالتی که تابع آستانه‌گذاری عبارت است از عملگر پراکسیمال یک تابع ناهموار، اثبات همگرایی به سادگی حالت هموار نخواهد بود؛ چرا که در این وضعیت مسأله‌ی اولیه‌ای که الگوریتم در پی حل آن است به طور کامل مشخص نیست. در واقع، در الگوریتم‌های ISP ما عموماً تمرکز زیادی روی مسأله‌ی اولیه نداشته و تنها الگوریتم را، که در واقع الهام‌بخش از الگوریتم SL0 است، مورد توجه قرار می‌دهیم. با این وجود، ما در قسمت ۴-۵ مسأله‌ی متناظر با حالت ناهموار را فرمول‌بندی کرده و علاوه بر آن، به صورت مستقیم الگوریتمی برای حل آن پیشنهاد می‌دهیم.

۴-۲-۵ شبیه‌سازی

در این بخش، نتایج شبیه‌سازی برای بررسی عملکرد الگوریتم‌های پیشنهادی و مقایسه‌ی آن‌ها با برخی از الگوریتم‌های موجود ارائه می‌شود. الگوریتم‌های مورد مقایسه عبارتند از الگوریتم GOMP [۱۲۷]، الگوریتم EM-GM-AMP [۱۲۵]، و الگوریتم NESTA [۱۵]. GOMP تعمیمی از الگوریتم OMP است که در هر تکرار، به جای یک اتم، بیش از یک اتم کاندید انتخاب می‌شوند. نشان داده شده است که این الگوریتم سرعت همگرایی بیشتری نسبت به الگوریتم OMP دارد [۱۲۷]. خانواده الگوریتم‌های AMP^۱ [۴۱] تعمیمی از الگوریتم‌های تکراری-آستانه‌گذاری (IST) هستند که سرعت بالا و کیفیت خوبی از خود نشان می‌دهند. الگوریتم EM-GM-AMP^۲ الگوریتمی مبتنی بر AMP است که ماهیتی آماری دارد و پارامترهای آزاد موجود در مسأله‌ی بازیابی نمایش تُنک را با استفاده از روش‌های بیزین^۳ تخمین می‌زند. در نهایت، الگوریتم NESTA روشی برای کمینه‌سازی نُرم یک

^۱ Approximate message passing

^۲ Expectation-maximization Gaussian-mixture approximate message passing

^۳ Bayesian

بوده که مبتنی بر ایده‌ی هموارسازی نستروف^۱ است. برای پیاده‌سازی این الگوریتم‌ها از کدهای در دسترس متلب آن‌ها استفاده شده است.^۲

بعنوان معیارهایی برای ارزیابی عملکرد الگوریتم‌ها، ما از کمیت‌های زیر استفاده کرده‌ایم:

- متوسط خطای مربعات نرمالیزه‌شده (NMSE)^۳ بین سیگنال اصلی \mathbf{x}^* و سیگنال تخمین‌زده شده $\hat{\mathbf{x}}$:

$$\text{NMSE}(\mathbf{x}^*, \hat{\mathbf{x}}) \triangleq \frac{\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2}{\|\mathbf{x}^*\|_2}. \quad (33-5)$$

- ضریب جینی (GI)^۴ [۷۳] که معیاری برای سنجش میزان تُنک بودن یک سیگنال گسسته $\mathbf{x} = [x_1, \dots, x_n]^T$

است. برای تعریف این کمیت، ابتدا نسخه‌ی مرتب‌شده‌ی سیگنال \mathbf{x} را که با $\bar{\mathbf{x}} = [\bar{x}_1, \dots, \bar{x}_n]$ نشان می‌دهیم،

در نظر بگیرید بگونه‌ای که $|\bar{x}_1| \leq |\bar{x}_2| \leq \dots \leq |\bar{x}_n|$. سپس، ضریب جینی \mathbf{x} به فرم زیر تعریف می‌شود

$$\text{GI}(\mathbf{x}) \triangleq 1 - \frac{2}{\|\mathbf{x}\|_1} \sum_{i=1}^n \frac{n-i+0.5}{n} \cdot |\bar{x}_i|. \quad (34-5)$$

تابع فوق برای سیگنالی که تنها یک درایه‌ی ناصفر دارد (تُنک‌ترین سیگنال) مقدار عددی ۱ را برگرداننده و

برای سیگنالی که همه‌ی درایه‌های آن ناصفر و برابر هستند (کم‌ترین درجه‌ی تُنکی) مقدار ۰ برمی‌گرداند.

علاوه بر معیارهای فوق، برای ارزیابی تقریبی پیچیدگی محاسباتی الگوریتم‌ها، زمان متوسط اجرای آن‌ها را نیز

گزارش خواهیم کرد. تمام شبیه‌سازی‌ها روی یک سیستم با ویندوز ۷، پردازشگر Core i7، و با RAM برابر با ۸

گیگابایت انجام شده است.

پارامترهای الگوریتم‌های مختلف به شیوه‌ی زیر مقداردهی و انتخاب شده‌اند. برای الگوریتم‌های JSP

$\tau_0 = 5 \max_i |x_i^2|$ قرار داده شده است، که در آن، \mathbf{x}^0 جواب حداقل نُرم دوی دستگاه $\mathbf{y} = \mathbf{D}\mathbf{x}$ است. همچنین،

$\tau_f = 5 \times 10^{-4}$ ، $c = 0.9$ و $I = 3$. پارامتر γ در الگوریتم ۴-۵ برابر ۰/۴ انتخاب شده است. برای الگوریتم

GOMP، تعداد اتم‌های انتخاب شده در هر تکرار $N = 4$ قرار داده شده است. برای EM-GM-AMP، پارامترهای

پیش‌فرض پیشنهاد شده در مقاله‌ی اصلی مورد استفاده قرار گرفته است. برای الگوریتم NESTA، مقدار نهایی برای

پارامتر هموارکننده $\mu_f = 10^{-5}$ انتخاب شده است و بقیه پارامترها همان مقادیر پیش‌فرض در نظر گرفته شده‌اند.

^۱Nesterov's smoothing idea

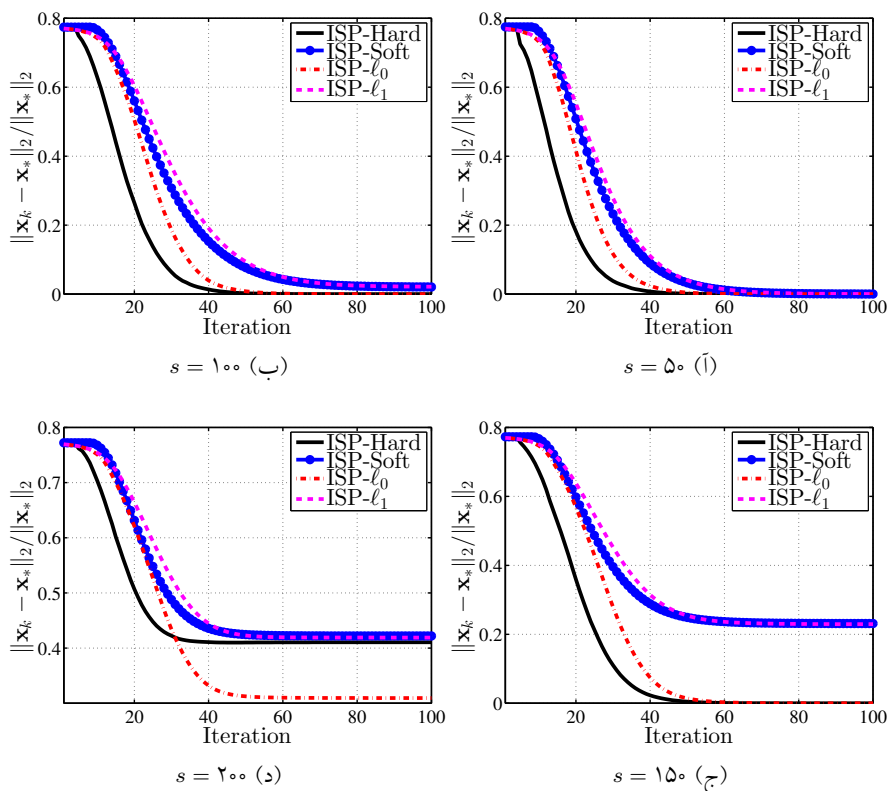
^۲این کدها را می‌توان از آدرس‌های زیر دریافت کرد: GOMP: <http://islab.snu.ac.kr/paper/gOMP.zip>، EM-GM-

AMP: <http://statweb.stanford.edu/~candes/nesta/>، NESTA و <http://www.ece.ohio-state.edu/~schniter/EMGMAMP/EMGMAMP.html>

~candes/nesta/

^۳Normalized mean squared error

^۴Gini index

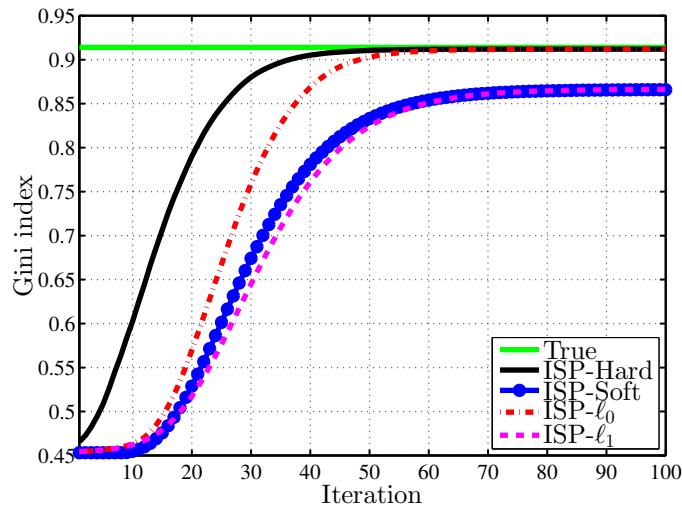


شکل ۵-۵: مقادیر NMSE برحسب تکرار الگوریتم‌های مختلف ISP.

مقایسه‌ی الگوریتم‌های ISP

در اینجا، عملکرد نسخه‌های گوناگون الگوریتم ISP مقایسه می‌شوند. برای این منظور، آزمایشی طراحی کرده‌ایم که عموماً برای ارزیابی عملکرد الگوریتم‌های بازیابی نمایش تُنک مورد استفاده قرار می‌گیرد [۵۲]. برای این منظور، ابتدا یک سیگنال تُنک تصادفی با توزیع گوسی-برنولی تولید می‌شود، بصورتی که مکان درایه‌های ناصفر با توزیع یکنواخت و مقدار این درایه‌ها از توزیع نُرمال $\mathcal{N}(0, 1)$ است. طول این سیگنال $n = 1000$ انتخاب می‌شود. همچنین، میزان مختلف تُنکی $s \in \{50, 100, 150, 200\}$ در نظر گرفته شده است. در ادامه، ماتریس \mathbf{D} با ابعاد 1000×400 با درایه‌های تصادفی از توزیع $\mathcal{N}(0, 1)$ تولید می‌شود. در نهایت، بردار اندازه‌گیری \mathbf{y} به طول $m = 400$ بصورت $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{e}$ ایجاد می‌شود، که \mathbf{e} بردار نویزی با توزیع درایه‌های $\mathcal{N}(0, \sigma_{noise}^2)$ است.

^۱ Bernoulli-Gaussian



شکل ۵-۶: مقادیر اندیس جینی بر حسب شماره‌ی تکرار الگوریتم‌های مختلف ISP.

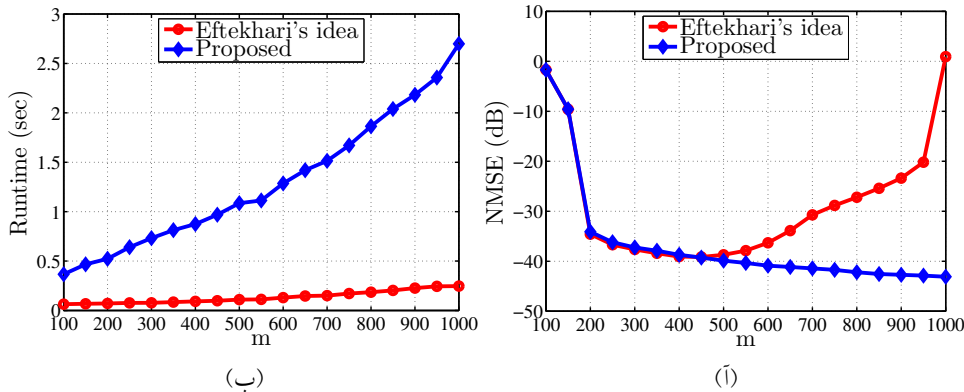
جدول ۵-۱: مقادیر نهایی NMSE برای الگوریتم‌های مختلف.

الگوریتم	$s = 50$	$s = 100$	$s = 150$	$s = 200$
ISP-Hard	$3/4623e-8$	$1/9449e-8$	$4/6082e-7$	$0/4105$
ISP- l_0	$9/1372e-8$	$6/3223e-7$	$1/5820e-6$	$0/3095$
ISP-Soft	$1/7035e-4$	$0/0206$	$0/2312$	$0/4219$
ISP- l_1	$2/1137e-4$	$0/0214$	$0/2294$	$0/4191$

شکل ۵-۵ رفتار مقادیر NMSE بر حسب شماره‌ی تکرار را برای الگوریتم‌های ISP گوناگون و حالت بدون نویز نشان می‌دهد. مقادیر نهایی NMSE نیز در جدول ۵-۱ آورده شده‌اند. بعلاوه، نمودارهای اندیس جینی بر حسب شماره‌ی تکرار در شکل ۵-۶ رسم شده‌اند. همانطور که این نتایج نشان می‌دهند، الگوریتم ISP-Hard عملکرد بهتری از نظر سرعت همگرایی و نیز مقادیر نهایی NMSE نسبت به سایر نسخه‌های ISP دارد. نکته‌ی جالب توجه دیگر این که، برای تمام الگوریتم‌ها، میزان اندیس جینی بر حسب شماره‌ی تکرار افزایشی است، که این معنی را می‌دهد که همه‌ی نسخه‌های ISP میزان تُنکی جواب را در حین پیشرفت الگوریتم بهبود می‌دهند.

بررسی عملکرد در حضور نویز

در این قسمت، الگوریتم پیشنهادی برای پیاده‌سازی گام تصویر کردن، که در شکل ۵-۴ آمده است، با الگوریتم پیشنهادی در [۵۰] مقایسه می‌شود. برای این منظور، مشابه روال قبل، سیگنالی تُنک با ۵۰ درایه‌ی غیرصفر تولید



شکل ۵-۷: (آ) مقادیر NMSE و (ب) زمان اجرای الگوریتم $ISP-l$ که یکبار مجهز به الگوریتم معرفی شده در [۵۰] است و یکبار با الگوریتم پیشنهادی در شکل ۵-۴. سطح نویز برابر با $\sigma_{noise} = 0.005$ بوده و $s = 50$.

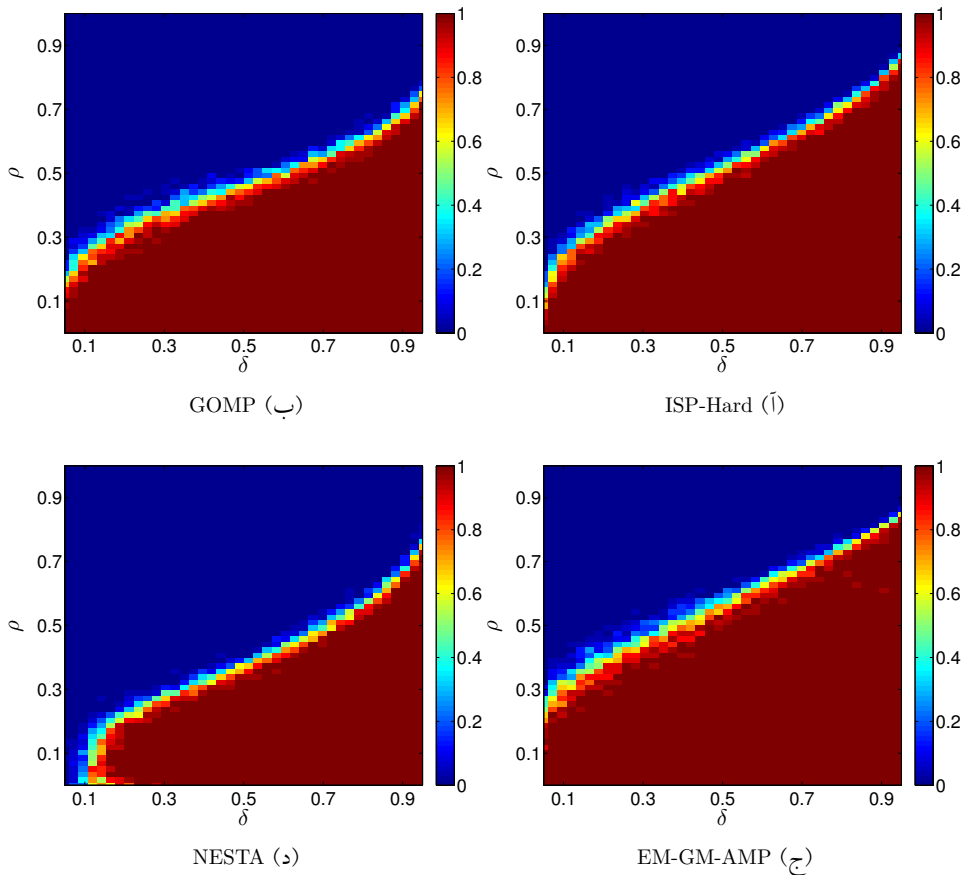
می‌شود. سطح نویز گوسی را برابر با $\sigma_{noise} = 0.005$ را انتخاب کرده و تعداد اندازه‌گیری‌ها، یعنی m ، را متغیر می‌گیریم. سپس، الگوریتم $ISP-l$ را یکبار با تصویرکردن پیشنهادی در [۵۰] و بار دیگر با تصویرکردن پیشنهادی در این رساله، که در شکل ۵-۴ خلاصه شده است، اجرا می‌کنیم. نتایج مربوط به این آزمایش در شکل ۷-۵ نشان داده شده است. همانطور که دیده می‌شود، بر خلاف الگوریتم پیشنهادی، عملکرد الگوریتم Robust-SL0 با افزایش تعداد اندازه‌گیری‌ها بدتر می‌شود. با این وجود، این الگوریتم زمان اجرای بهتری نسبت به الگوریتم پیشنهادی دارد. نکته‌ی مورد توجه دیگر در مورد این دو الگوریتم این است که عملکرد آن‌ها وقتی که ماتریس اندازه‌گیری تعداد سطرها به مراتب کمتری نسبت به تعداد ستون‌ها دارد تقریباً یکسان است.

مقایسه با الگوریتم‌های دیگر

در این قسمت، عملکرد الگوریتم پیشنهادی $ISP-l$ با الگوریتم‌های GOMP، EM-GM-AMP و NESTA مقایسه می‌شود. برای این منظور، ابتدا منحنی‌های گذر فاز^۱ [۴۷] این الگوریتم‌ها را مقایسه می‌کنیم. نمودار گذر فاز، نواحی شکست و موفقیت یک الگوریتم را در بازیابی سیگنال‌های تُنک با تُنکی‌های متفاوت و از روی تعداد متفاوت اندازه‌گیری تصادفی مشخص می‌کند. نسبت تُنکی را با $\rho \triangleq s/m$ و نسبت تعداد اندازه‌گیری‌ها به طول سیگنال (نرخ زیرنمونه‌برداری) را با m/n نشان می‌دهیم. برای تولید این منحنی‌ها، ما صفحه‌ی تُنکی-زیرنمونه‌برداری^۲ را بصورت شبکه‌ای 40×60 تقسیم کردیم که با $\delta \in [0.05, 0.95]$ و $\rho \in [0.01, 0.99]$ مشخص

^۱Phase transition diagram

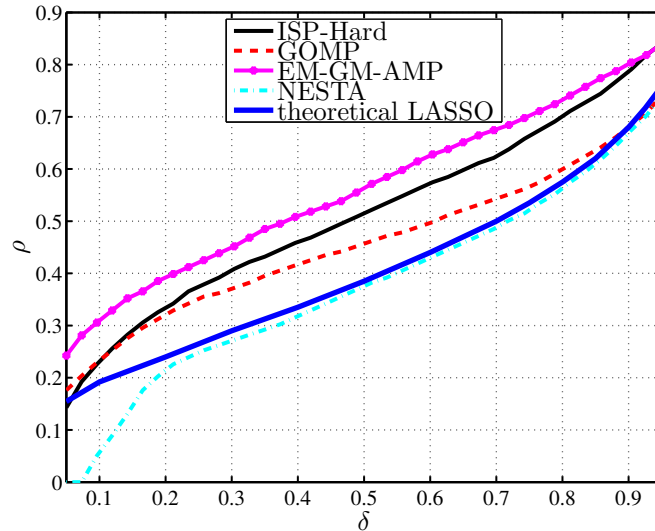
^۲sparsity-undersampling parameter space



شکل ۵-۸: منحنی‌های گذر فاز برای الگوریتم‌های مختلف در بازیابی سیگنال‌های تَنک به طول $n = 1000$. در این شکل $\delta \triangleq m/n$ و $\rho \triangleq s/m$

می‌شود. سیگنال تَنک \mathbf{x} و ماتریس اندازه‌گیری \mathbf{D} به همان ترتیب قبل تولید می‌شوند. برای تولید منحنی گذر فاز، یک بازیابی را موفقیت‌آمیز می‌گوئیم اگر برای آن داشته باشیم $\text{NMSE} \leq 0.001$. برای هر نقطه از این شبکه، نرخ بازیابی موفقیت‌آمیز با صدبار انجام متفاوت آزمایش، یعنی تولید سیگنال تَنک و ماتریس اندازه‌گیری و سپس اعمال الگوریتم‌های مختلف، بدست می‌آید.

نواحی گذر فاز به همراه منحنی‌های گذر فاز، که نواحی شکست و موفقیت را از هم جدا می‌کنند، به ترتیب در شکل‌های ۵-۸ و ۵-۹ رسم شده‌اند. همانگونه که از این شکل‌ها مشخص است، الگوریتم EM-GM-AMP نمودار گذر فاز بالاتری نسبت به سایر الگوریتم‌ها دارد، که به این معنی است که این الگوریتم می‌تواند در گستره‌های وسیعتری از تَنکی و زیرنمونه‌برداری، سیگنال‌های تَنک را بازیابی کند. علاوه بر این، دو الگوریتم $\text{ISP-}l$ و GOMP عملکرد مشابهی برای مقادیر کوچک δ و ρ دارند، حال آنکه برای سایر مقادیر، الگوریتم $\text{ISP-}l$.



شکل ۵-۹: منحنی‌های گذر فاز که نواحی موفقیت‌آمیز را از نواحی شکست که در شکل ۵-۸ رسم شده‌اند جدا می‌کنند. منحنی تئوری گذر فاز برای الگوریتم LASSO نیز رسم شده است [۴۷]. معیار جداسازی نواحی موفقیت و شکست نرخ موفقیت برابر با ۰/۵ بوده است.

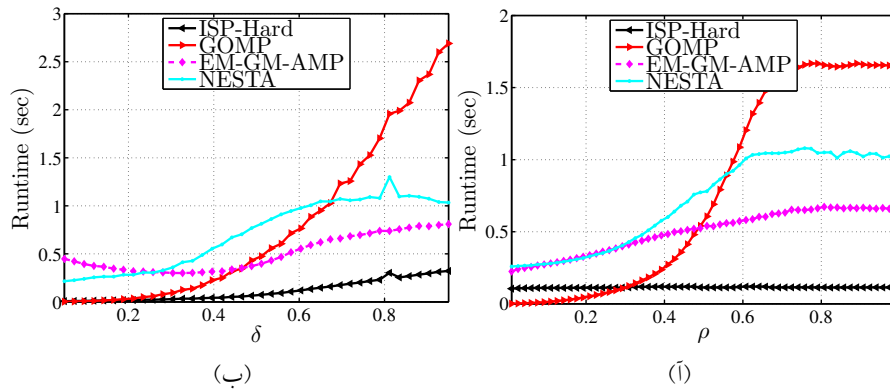
وضعیت بهتری نسبت به GOMP دارد. نکته‌ی دیگر این که عملکرد NESTA مشابه عملکرد تئوری LASSO است.

شکل ۵-۱۰ زمان متوسط اجرای الگوریتم‌ها برحسب پارامترهای $\delta \triangleq m/n$ و $\rho \triangleq s/m$ را نشان می‌دهد. همانطور که مشاهده می‌شود، الگوریتم $ISP-l$ زمان اجرای تقریباً ثابتی برای مقادیر متفاوت تُنکی دارد، حال آنکه الگوریتم‌های دیگر زمان اجرای بالاتری برای تُنکی‌های کمتر دارند. اما برحسب تعداد اندازه‌گیری‌ها، همه‌ی الگوریتم‌ها زمان اجرای افزایشی با زیاد شدن تعداد اندازه‌گیری‌ها دارند. در مجموع، الگوریتم پیشنهادی زمان اجرای کمتری دارد.

ماتریس‌های اندازه‌گیری متفاوت

در این قسمت، مشابه با [۱۲۴]، عملکرد الگوریتم‌های مختلف را در وضعیت‌های مختلف ماتریس اندازه‌گیری مورد بررسی قرار می‌دهیم. برای این منظور، چهار حالت برای ماتریس اندازه‌گیری، که در عمل باعث ایجاد چالش برای بازیابی سیگنال‌های تُنک می‌شوند، در نظر گرفته می‌شود که عبارتند از

- **ماتریس تُنک.** یک ماتریس با توزیع درایه‌های گوسی نرمال که تعدادی از درایه‌های آن بصورت یکنواخت صفر شده است.



شکل ۵-۱: زمان متوسط اجرای الگوریتم‌های مختلف بر حسب پارامترهای (آ) $\rho \triangleq s/m$ و (ب) $\delta \triangleq m/n$

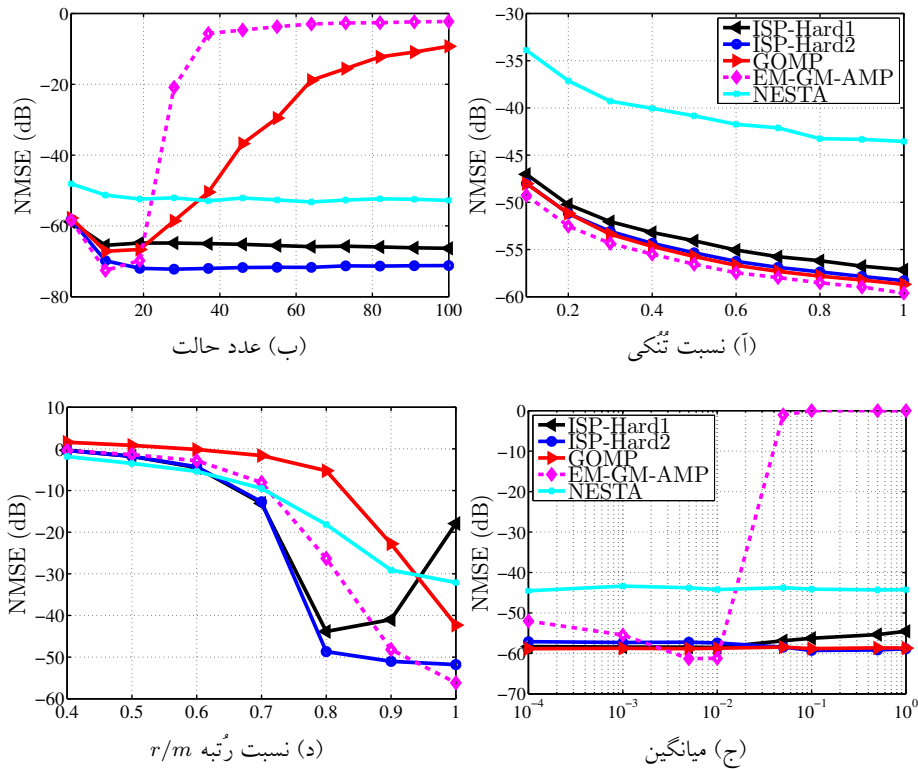
- ماتریس با میانگین ناصفر. ماتریسی با درایه‌های گوسی با میانگین غیرصفر و واریانس واحد.
- ماتریس بدحالت^۱. ماتریس \mathbf{D} با تجزیه SVD به فرم $\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^T$ که درایه‌های روی قطر Σ بصورت $\Sigma_{ii} = \kappa^{(m-i)/m}$ هستند، که در آن، $\kappa > 1$ عدد حالت^۲ ماتریس \mathbf{D} است.
- ماتریس کم‌رتبه^۳. ماتریس $\mathbf{D} = \frac{1}{m}\mathbf{U}_{m \times r}\mathbf{V}_{r \times n}^T$ که در آن، r رتبه‌ی ماتریس بوده و درایه‌های \mathbf{U} و \mathbf{V} از توزیع گوسی نرمال انتخاب می‌شوند.

برای بررسی عملکرد الگوریتم‌ها در این زمینه، سیگنال تُنک را مشابه قبل با طول $n = 1000$ و با 150 درایه‌ی غیرصفر انتخاب کردیم. همچنین، تعداد اندازه‌گیری‌ها را برابر با 500 و سطح نویز را $\sigma_{noise} = 0.0005$ گرفتیم. مقادیر NMSE برای الگوریتم‌های مختلف و ماتریس‌های اندازه‌گیری با وضعیت‌های ذکر شده در فوق، در شکل ۵-۱۱ نشان داده شده‌اند. این نتایج حاصل میانگین‌گیری روی 100 بار انجام شبیه‌سازی هستند. در این شکل، ISP-Hard1 و ISP-Hard2، به ترتیب، نشان‌دهنده‌ی الگوریتم ISP-Hard با تصویر کردن پیشنهادی در الگوریتم Robust-SL0 و با تصویر کردن پیشنهادی هستند. همانطور که از این شکل مشخص است، الگوریتم EM-GM-AMP حساسیت زیادی به ماتریس‌های بدحالت و ماتریس‌های با میانگین غیرصفر دارد. علاوه بر این، الگوریتم GOMP عملکرد خوبی در مورد ماتریس‌های کم‌رتبه و بدحالت ندارد، حال آنکه، در سایر موارد عملکرد مشابهی با ISP-Hard2 دارد. نکته‌ی قابل ذکر دیگر اینکه، الگوریتم ISP-Hard1 عملکرد ضعیفتری نسبت به الگوریتم

^۱ Ill-conditioned

^۲ Condition number

^۳ Low rank



شکل ۵-۱۱: مقادیر NMSE در بازیابی سیگنال‌های تُنک به طول ۱۰۰۰ از روی تعداد ۵۰۰ اندازه‌گیری تصادفی با استفاده از ماتریس‌های اندازه‌گیری (آ) تُنک، (ب) بدحالت، (ج) میانگین غیرصفر و (د) کم‌رتبه. در این شکل، ISP-Hard1 و ISP-Hard2، به ترتیب، نشان‌دهنده الگوریتم ISP-Hard با تصویر کردن پیشنهادی در الگوریتم Robust-SL0 و با تصویر کردن پیشنهادی هستند.

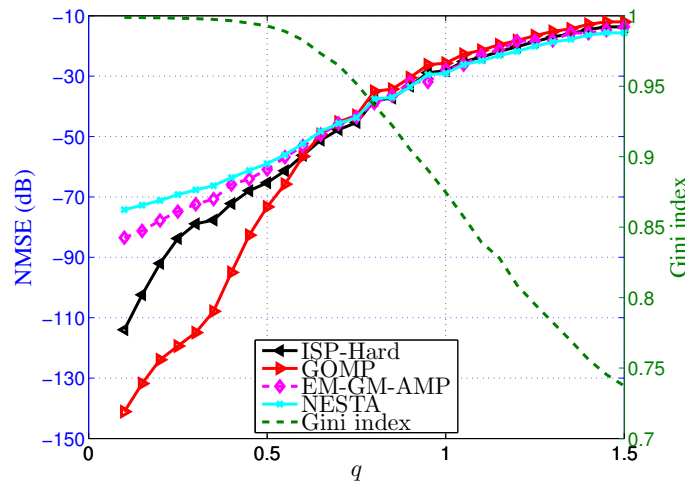
ISP-Hard2 دارد، بخصوص برای ماتریس‌های کم‌رتبه.

سیگنال‌های فشرده‌پذیر

سیگنال‌های طبیعی، از جمله تصویر، عموماً نمایش دقیقاً تُنکی ندارند. بعبارت دیگر، این سیگنال‌ها نمایشی دارند که اکثر درایه‌های آن بسیار کوچک بوده ولی دقیقاً صفر نیستند. به چنین سیگنال‌هایی، فشرده‌پذیر^۱ می‌گویند [۲۸]. بعنوان یک تعریف دقیقتر، سیگنال $\mathbf{x} \in \mathbb{R}^n$ فشرده‌پذیر است هرگاه درایه‌های آن از نظر دامنه مرتب‌شده‌ی آن، یعنی $\{\bar{x}_i\}$ ، یک سیر نزولی نمایی بصورت $\bar{x}_i \lesssim R \cdot i^{-d}$ از خود نشان دهند، که در آن، $R > 0$ است و $d > 0$ نرخ کاهش را نشان می‌دهد [۲۸]. تابع توزیع GPD^۲ توزیعی است که سیگنال‌های فشرده تولید می‌کند [۲۸]. تابع چگالی احتمال این توزیع به فرم زیر است

^۱ Compressible

^۲ Generalized Pareto distribution



شکل ۵-۱۲: مقادیر NMSE برای الگوریتم‌های مختلف در بازیابی سیگنال‌های فشرده‌پذیر به طول $n = 1000$ از روی $m = 500$ اندازه‌گیری گوسی. محور افقی پارامتر q ، که معیاری از میزان فشرده‌پذیری است، را نشان می‌دهد. همچنین، در محور عمودی سمت راست، مقادیر معادل ضریب جینی نشان داده شده است.

$$P(x; q, \lambda) = \frac{q}{\Gamma(\lambda)} \left(1 + \frac{|x|}{\lambda}\right)^{-(q+1)}. \quad (5-35)$$

نشان داده شده است که $R = \lambda n^{1/q}$ و $d = 1/q$ و این که، ضرایب موجک تصاویر طبیعی را می‌توان با این توزیع مدل کرد [۲۸].

برای بررسی عملکرد الگوریتم‌های مورد مقایسه در بازیابی سیگنال‌های فشرده‌پذیر از روی اندازه‌گیری‌های گوسی، ما از توزیع GPD استفاده کرده و سیگنال‌هایی به طول $n = 1000$ و با درجات مختلفی از فشرده‌پذیری تولید کردیم. به بیان دقیقتر، پارامتر q را از ۱ تا ۱/۵ تغییر دادیم و همچنین قرار دادیم $\lambda = 1$. تعداد اندازه‌گیری‌ها را نیز $m = 500$ در نظر گرفتیم.

مقادیر NMSE برای الگوریتم‌های مختلف در شکل ۵-۱۲ نمایش داده شده است. همانطور که می‌توان دید، الگوریتم GOMP بهترین عملکرد را دارد. همچنین، الگوریتم ISP-Hard عملکرد بهتری نسبت به الگوریتم EM-GM-AMP از خود نشان می‌دهد. نکته‌ی آخر هم این که، همانطور که انتظار می‌رفت، عملکرد همه‌ی الگوریتم‌ها با کاهش میزان فشرده‌پذیری سیگنال سیری نزولی دارد.

۳-۵ الگوریتم نُرم صفر-آستانه‌گذاری نُرم (L0Soft)

در این قسمت، الگوریتم دیگری برای بازیابی سیگنال‌های تُنک از روی اندازه‌گیری‌های محدود آن‌ها پیشنهاد می‌دهیم [۱۱۰]. ایده‌ی این الگوریتم، مشابه الگوریتم SL0، تقریب زدن تابع نُرم صفر با یک تابع مشتق‌پذیر و خوش‌رفتار است. اما برخلاف الگوریتم‌هایی مانند SL0 که کل تابع نُرم صفر را با یک تابع هموار جایگزین می‌کنند، روش پیشنهادی، با استفاده از یک تعریف خاص از تابع نُرم صفر، تنها قسمتی از این تابع را با یک تابع هموار جایگزین می‌کند. در ادامه، مسأله‌ی حداقل‌سازی تابع نُرم صفر به یک مسأله‌ی معادل حداقل‌سازی نُرم یک تبدیل شده و با استفاده از روش‌های پراکسیمال حل می‌شود. نتایج شبیه‌سازی حکایت از برتری روش پیشنهادی نسبت به الگوریتم‌هایی مانند SL0 و ISP-Hard، که در قسمت قبل معرفی کردیم، دارد.

۱-۳-۵ فرمول‌بندی مسأله

مسأله‌ای که ما برای بازیابی سیگنال‌های تُنک در نظر می‌گیریم عبارت است از کمینه‌سازی نُرم صفر، که برای یادآوری در زیر آورده شده است:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon, \quad (36-5)$$

که $\mathbf{y} \in \mathbb{R}^m$ ، $\mathbf{A} \in \mathbb{R}^{m \times n}$ و $\epsilon \geq 0$. نقطه‌ی شروع روش پیشنهادی توجه به این نکته است که برای یک بردار

$\mathbf{x} \in \mathbb{R}^n$ تابع نُرم صفر را می‌توان به فرم معادل زیر تعریف کرد

$$\|\mathbf{x}\|_0 = \sum_{i=1}^n |\text{sgn}(x_i)|, \quad (37-5)$$

که در آن، sgn تابع علامت بوده و $\text{sgn}(0) \triangleq 0$. تعریف فوق را به طرز معادلی می‌توان بصورت زیر نوشت

$$\|\mathbf{x}\|_0 = \|\mathbf{z}\|_1, \quad \mathbf{z} = \text{sgn}(\mathbf{x}), \quad (38-5)$$

که در آن فرض می‌شود تابع sgn بصورت مولفه به مولفه روی یک بردار عمل می‌کند. ایده‌ی پیشنهادی برای حل

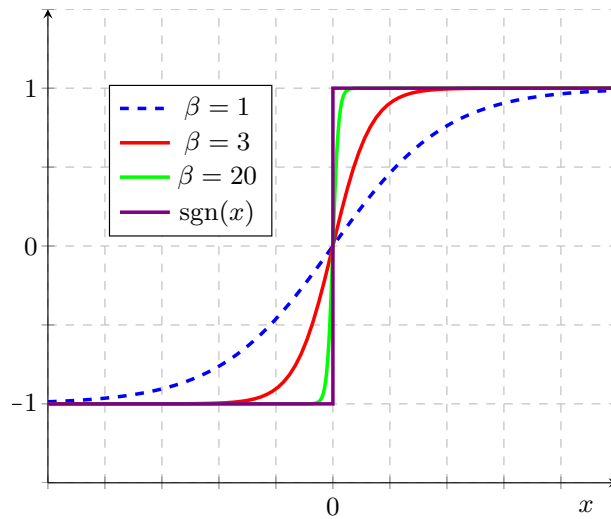
(۳۶-۵) عبارت است از استفاده از رابطه‌ی فوق میان \mathbf{x} و \mathbf{z} . به این ترتیب، به مسأله‌ی زیر می‌رسیم

$$\min_{\mathbf{x}, \mathbf{z}} \|\mathbf{z}\|_1 \quad \text{s.t.} \quad \begin{cases} \mathbf{z} = \text{sgn}(\mathbf{x}) \\ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon. \end{cases} \quad (39-5)$$

برای حل مسأله‌ی فوق، ما از روش جریمه^۱ استفاده می‌کنیم [۹۴]. با این روش، مسأله‌ی جایگزین زیر باید حل

شود

^۱Penalty methods

شکل ۵-۱۳: نمودار تابع (۴۱-۵) برای مقادیر مختلف β .

$$\min_{\mathbf{x}, \mathbf{z}} \|\mathbf{z}\|_1 + \frac{1}{\sqrt{\alpha}} \|\mathbf{z} - \text{sgn}(\mathbf{x})\|_2^2 + \delta_{C_\epsilon}(\mathbf{x}), \quad (40-5)$$

که در آن، $\delta_{C_\epsilon}(\mathbf{x})$ تابع مشخصه‌ی مجموعه‌ی $C_\epsilon(\mathbf{x}) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon\}$ بوده و $\alpha > 0$ یک ضریب جریمه است. توجه داریم که برای $\alpha \rightarrow 0$ ، دو مسأله‌ی (۴۰-۵) و (۳۹-۵) معادل خواهند بود. از آنجایی که تابع علامت مشتق‌پذیر نیست، ما از یک تابع هموار به عنوان جایگزینی برای آن استفاده می‌کنیم. برای این منظور، توابع مختلفی وجود دارند، از جمله تابع زیر

$$f_\beta(x) \triangleq \tanh(\beta x) = \frac{\exp(2\beta x) - 1}{\exp(2\beta x) + 1}, \quad (41-5)$$

که در آن، $\beta > 0$ پارامتری است که کیفیت تقریب را تعیین می‌کند. تابع فوق برای چند مقدار مختلف β در شکل رسم شده است. همانطور که مشاهده می‌شود، برای مقادیر بزرگ β تابع فوق تقریب بهتری از تابع علامت ارائه

می‌دهد. با استفاده از تابع (۴۱-۵)، مسأله‌ی (۴۰-۵) به فرم زیر تقریب زده می‌شود

$$\min_{\mathbf{x}, \mathbf{z}} \alpha \|\mathbf{z}\|_1 + \frac{1}{\sqrt{\alpha}} \|\mathbf{z} - f_\beta(\mathbf{x})\|_2^2 + \delta_{C_\epsilon}(\mathbf{x}), \quad (42-5)$$

که تابع f_β به صورت مولفه به مولفه عمل می‌کند. برای حل مسأله‌ی فوق، از الگوریتم PALM، که در فصل قبل

مرور شد، استفاده می‌کنیم. برای این منظور، تعریف می‌کنیم

$$\begin{cases} F(\mathbf{x}, \mathbf{z}) = \frac{1}{\sqrt{\alpha}} \|\mathbf{z} - f_\beta(\mathbf{x})\|_2^2 \\ g(\mathbf{x}) = \delta_{C_\epsilon}(\mathbf{x}) \\ h(\mathbf{z}) = \alpha \|\mathbf{z}\|_1. \end{cases} \quad (43-5)$$

در نتیجه، مسأله‌ی (۴۲-۵) به فرم معادل زیر نوشته می‌شود

$$\min_{\mathbf{x}, \mathbf{z}} \{H(\mathbf{x}, \mathbf{z}) \triangleq F(\mathbf{x}, \mathbf{z}) + g(\mathbf{x}) + h(\mathbf{z})\}. \quad (44-5)$$

همانطور که در فصل گذشته دیدیم، برای استفاده از الگوریتم PALM، تابع هموار $F(\mathbf{x}, \mathbf{z})$ باید نسبت به \mathbf{x} و \mathbf{z} گرادینانی لیپشیتز داشته باشد. به عبارت دیگر، باید ثابت‌هایی مانند $L_x, L_z > 0$ وجود داشته باشند به نحوی که

برای هر $\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{v}$ داشته باشیم

$$\begin{cases} \|\nabla_{\mathbf{x}} F(\mathbf{x}, \mathbf{z}) - \nabla_{\mathbf{x}} F(\mathbf{u}, \mathbf{z})\|_2 \leq L_x \|\mathbf{x} - \mathbf{u}\|_2 \\ \|\nabla_{\mathbf{z}} F(\mathbf{x}, \mathbf{z}) - \nabla_{\mathbf{z}} F(\mathbf{x}, \mathbf{v})\|_2 \leq L_z \|\mathbf{z} - \mathbf{v}\|_2. \end{cases} \quad (45-5)$$

در رابطه‌ی فوق، گرادینان‌ها نسبت به \mathbf{x} و \mathbf{z} با روابط زیر داده می‌شوند

$$\begin{cases} \nabla_{\mathbf{x}} F(\mathbf{x}, \mathbf{z}) = f'_\beta(\mathbf{x}) \odot (f_\beta(\mathbf{x}) - \mathbf{z}) \\ \nabla_{\mathbf{z}} F(\mathbf{x}, \mathbf{z}) = \mathbf{z} - f_\beta(\mathbf{x}), \end{cases} \quad (46-5)$$

که در آن، علامت \odot نشان‌دهنده‌ی ضرب هادامارد^۱ (یا ضرب درایه به درایه) است و

$$f'_\beta(x) = \beta \operatorname{sech}^\gamma(\beta x), \quad \operatorname{sech}(x) = \frac{2}{\exp(x) + \exp(-x)}. \quad (47-5)$$

واضح است که $L_z = 1$. برای بدست آوردن یک L_x ، تابع $J(x) = f'_\beta(x) \cdot (f_\beta(x) - z)$ را در نظر بگیرید. در

ادامه، حد بالایی برای دامنه‌ی مشتق این تابع، یعنی

$$J'(x) = -2\beta^2 \cdot \operatorname{sech}^\gamma(\beta x) \cdot \tanh(\beta x) \cdot (\tanh(\beta x) - z) + \beta^2 \cdot \operatorname{sech}^4(\beta x) \quad (48-5)$$

بدست می‌آوریم. توجه کنید که برای هر x داریم $|\tanh(\beta x)| \leq 1$ که با استفاده از رابطه‌ی $\operatorname{sech}^\gamma(\beta x) =$

$$1 - \tanh^2(\beta x) \quad |J'(x)| \leq \beta^2(3 + 2|z|). \quad \text{بنابراین، عبارت زیر}$$

$$L_x = (3 + 2|z|) \cdot \beta^2 \quad (49-5)$$

رابطه‌ی (45-5) را ارضا می‌کند. با استفاده از روش PALM، الگوریتم پیشنهادی از یک نقطه‌ی اولیه $(\mathbf{x}_0, \mathbf{z}_0)$

شروع کرده، دنباله‌ی $\{\mathbf{x}_k, \mathbf{z}_k\}_{k \geq 1}$ را برای به‌روزرسانی \mathbf{x} و \mathbf{z} تولید می‌کند. ابتدا مسأله‌ی به‌روزرسانی \mathbf{z} را که

بصورت زیر است، در نظر بگیرید

$$\mathbf{z}_{k+1} = \operatorname{argmin}_{\mathbf{z}} \langle \nabla_{\mathbf{z}} F(\mathbf{x}_k, \mathbf{z}_k), \mathbf{z} - \mathbf{z}_k \rangle + \frac{1}{\mu_z} \|\mathbf{z} - \mathbf{z}_k\|_2^2 + h(\mathbf{z}). \quad (50-5)$$

در عبارت فوق، $\mu_z \in (0, 1/L_z]$. مسأله‌ی بالا را می‌توان به شکل زیر ساده کرد

$$\mathbf{z}_{k+1} = \operatorname{argmin}_{\mathbf{z}} \frac{1}{\mu_z} \|\mathbf{z} - \tilde{\mathbf{z}}_k\|_2^2 + \mu_z \cdot h(\mathbf{z}), \quad (51-5)$$

که در آن، $\tilde{\mathbf{z}}_k = \mathbf{z}_k - \mu_z \nabla_{\mathbf{z}} F(\mathbf{x}_k, \mathbf{z}_k) = (1 - \mu_z)\mathbf{z}_k + \mu_z f_\beta(\mathbf{x}_k)$ ، جواب بهینه‌ی مسأله‌ی (51-5)، همانطور که

در فصل گذشته دیدیم، با تابع آستانه‌گذاری نرم و بصورت زیر داده می‌شود

$$\mathbf{z}_{k+1} = \mathcal{T}_{\mu_z, \alpha}^s((1 - \mu_z) \cdot \mathbf{z}_k + \mu_z \cdot f_\beta(\mathbf{x}_k)). \quad (52-5)$$

برای به‌روزرسانی \mathbf{x} ، باید مسأله‌ی زیر حل شود

^۱Hadamard (entry-wise) product

• هدف: حل مسأله‌ی (۴۲-۵) با استفاده از PALM

• ورودی: $w, \alpha, \epsilon, (\mathbf{x}_0, \mathbf{z}_0), \mathbf{A}, \mathbf{y}$

• شروع الگوریتم: گام‌های زیر را برای $k = 0, 1, \dots$ انجام بده

۱. $\mathbf{z}_{k+1} = S_{\mu_z \cdot \alpha}((1 - \mu_z) \cdot \mathbf{z}_k + \mu_z \cdot f_\beta(\mathbf{x}_k))$
۲. $\hat{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1})$
۳. $\mathbf{x}_{k+1} = \mathcal{P}_{C_\epsilon}(\hat{\mathbf{x}}_k - \mu_x \cdot \nabla_x F(\mathbf{x}_k, \mathbf{z}_{k+1}))$

• خروجی: $(\mathbf{x}_k, \mathbf{z}_k)$

شکل ۵-۱۴: الگوریتم پیشنهادی برای حل مسأله‌ی (۴۲-۵).

• توصیف: الگوریتم L0Soft برای حل مسأله‌ی (۳۹-۵)

• ورودی: $c, w, \alpha_1, \epsilon, (\mathbf{x}_0, \mathbf{z}_0), \mathbf{A}, \mathbf{y}$

• شروع الگوریتم: گام‌های زیر را برای $j = 0, 1, \dots$ انجام بده

۱. $(\mathbf{x}_j, \mathbf{z}_j) = \text{PALM}(\mathbf{y}, \mathbf{A}, (\mathbf{x}_{j-1}, \mathbf{z}_{j-1}), \epsilon, \alpha_j, w)$
۲. $\alpha_{j+1} = c \cdot \alpha_j$

• خروجی: \mathbf{x}_j

شکل ۵-۱۵: الگوریتم پیشنهادی L0Soft برای حل مسأله‌ی (۴۲-۵).

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \langle \nabla_x F(\mathbf{x}_k, \mathbf{z}_{k+1}), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\mu_x} \|\mathbf{x} - \hat{\mathbf{x}}_k\|_2^2 + g(\mathbf{x}), \quad (53-5)$$

که $\mu_x \in (0, 1/L_x]$ و $\hat{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1})$ همچنین، $w \in [0, 1]$ پارامتری برای بالابردن کیفیت الگوریتم است که در الگوریتم‌های پراکسیمال استفاده می‌شود [۱۳۳]. با انجام محاسباتی ساده می‌توان نشان داد که جواب

مسأله‌ی (۵۳-۵) عبارت است از

$$\mathbf{x}_{k+1} = \mathcal{P}_{C_\epsilon}(\hat{\mathbf{x}}_k - \mu_x \cdot \nabla_x F(\mathbf{x}_k, \mathbf{z}_{k+1})). \quad (54-5)$$

الگوریتم کلی برای حل (۴۲-۵) در شکل ۵-۱۴ خلاصه شده است. حال، برای حل تقریبی (۳۹-۵)، باید مسأله‌ی (۴۲-۵) را برای یک دنباله‌ی کاهشی از α حل کنیم. برای این منظور، دنباله‌ی $\{\alpha_j\}_{j \geq 1}$ را $\alpha_{j+1} = c \cdot \alpha_j$ که $0 < c < 1$ در نظر بگیرید. در اینصورت، شکل ۵-۱۵ توصیفی از الگوریتم پیشنهادی، که ما آن را L0Soft می‌نامیم، ارائه می‌دهد.

جزئیات پیاده‌سازی

برای مقداردهی اولیه الگوریتم L0Soft، قرار می‌دهیم $\mathbf{x}_0 = \mathbf{A}^\dagger \mathbf{y}$ ، $\mathbf{x}_{-1} = \mathbf{0}$ و $\mathbf{z}_0 = f_\beta(\mathbf{x}_0)$ همچنین، $\alpha_1 = 1$ در عمل منجر به نتایج خوبی می‌شود. برای پارامتر μ_z ، مقداری نزدیک به ۱، مثلاً ۰/۹۵، عملکرد خوبی دارد. برای انتخاب مقدار μ_x ، ابتدا لم زیر را اثبات می‌کنیم.

لم ۲-۵ در الگوریتم شکل ۵-۱۴ و با مقداردهی \mathbf{z} بصورتی که در بالا گفته شد، داریم

$$\forall i, k: |z_k^i| \leq 1, \quad (55-5)$$

که در آن، z_k^i مولفه‌ی i -ام از \mathbf{z}_k را نشان می‌دهد.

برای اثبات این لم، ابتدا توجه کنید که برای همی x و $\lambda, \beta \geq 0$ داریم $|T_\lambda(x)| \leq |x|$ و $|f_\beta(x)| \leq 1$. بنابراین، از (۵۲-۵) می‌توان نوشت

$$\forall i, k: |z_{k+1}^i| \leq (1 - \mu_z) \cdot |z_k^i| + \mu_z. \quad (56-5)$$

استفاده‌ی مکرر از نامساوی فوق منجر به روابط زیر می‌شود

$$\begin{aligned} \forall i, K: |z_{K+1}^i| &\leq (1 - \mu_z)^K + \sum_{k=0}^K \mu_z \cdot (1 - \mu_z)^k \\ &\leq (1 - \mu_z)^K + \mu_z \frac{1 - (1 - \mu_z)^{K+1}}{1 - (1 - \mu_z)} \\ &= 1. \end{aligned} \quad (57-5)$$

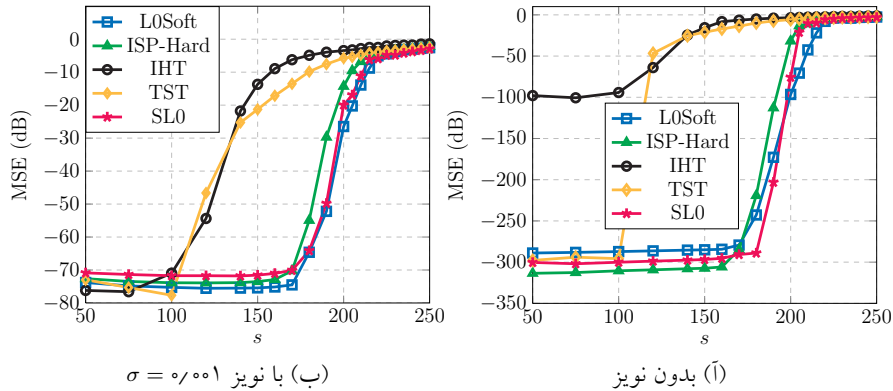
با استفاده از نتیجه‌ی فوق، می‌توان یک باند بهتر برای L_x در معادله‌ی (۴۹-۵) بصورت $L_x \leq 5\beta^2$ بدست آورد، و چون $\mu_x \in (0, 1/L_x]$ می‌توان یک باند مجاز برای μ_x بدست آورد.

۲-۳-۵ شبیه‌سازی

این قسمت، نتایج شبیه‌سازی و مقایسه عملکرد الگوریتم L0Soft با الگوریتم SL0، ISP-Hard و الگوریتم IHT [۱۸] ارائه می‌دهد. نقطه‌ی اشتراک این الگوریتم‌ها این است که همگی سعی در کمینه‌سازی نُرم صفر را دارند. برای الگوریتم IHT ما از پیاده‌سازی بهینه‌شده آن که نیازی به دانستن ضریب تنظیم‌کننده‌ی λ (که بین کمینه‌سازی نُرم صفر و خطا مصالحه برقرار می‌کند) ندارد [۸۵]، استفاده کردیم^۱. علاوه بر IHT، از یک نسخه‌ی پیشرفته‌تر آن به نام TST^۲، که یک آستانه‌گذاری دو مرحله‌ای انجام می‌دهد، نیز استفاده کرده‌ایم [۸۵]. پارامترهای ISP-hard و

^۱<http://sparselab.stanford.edu/OptimalTuning/code.htm>

^۲Two stage thresholding



شکل ۵-۱۶: مقادیر متوسط MSE برحسب دسی‌بل برای الگوریتم‌های مختلف. مقادیر مختلف تُنکی، s ، در نظر گرفته شده است. همچنین، طول سیگنال‌های تُنک برابر با $n = 1000$ بوده و بازسازی از روی تعداد $m = 400$ اندازه‌گیری گوسی انجام شده است.

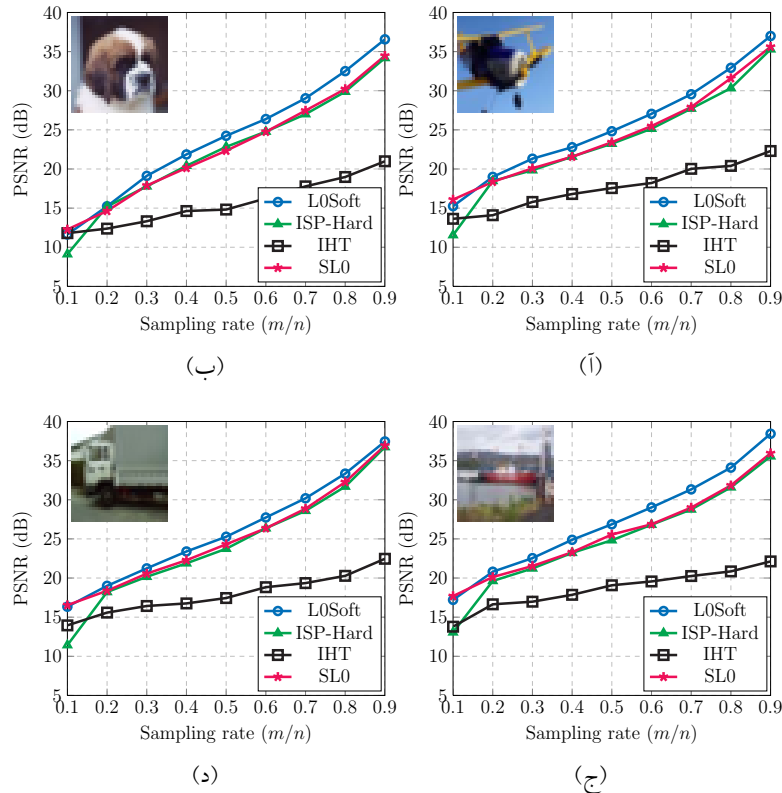
SL0 به مقادیر پیش‌فرض آن‌ها تنظیم شده است. برای L0Soft، قرار داده‌ایم $\beta = 5$ ، $w = 0.9$ و $c = 0.9$. علاوه بر این، تمامی الگوریتم‌ها برای تعداد ۳۰۰ تکرار اجرا شدند. دو مجموعه شبیه‌سازی صورت گرفته است؛ یکی روی داده‌های مصنوعی و دیگری روی داده‌های طبیعی، که در ادامه نتایج آن‌ها ارائه می‌شود.

داده‌های مصنوعی

در این قسمت، قابلیت بازیابی الگوریتم‌های مختلف در مورد داده‌های مصنوعی مقایسه می‌شود. شیوه‌ی انجام این شبیه‌سازی مشابه قسمت ۵-۲-۴ است. برای اندازه‌گیری کیفیت بازسازی، از معیار MSE که به صورت $MSE(\mathbf{x}, \hat{\mathbf{x}}) = 20 \log(\|\mathbf{x} - \hat{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2)$ تعریف می‌شود، استفاده می‌کنیم. در این رابطه، \mathbf{x} سیگنال تُنک اصلی و $\hat{\mathbf{x}}$ سیگنال بازسازی شده است. نتایج، برای هم بازسازی نویزی (با توزیع گوسی) و هم بدون نویز، در شکل ۵-۱۶ رسم شده‌اند. همانطور که مشاهده می‌شود، الگوریتم‌های IHT و TST عملکرد ضعیف‌تری نسبت به سایر الگوریتم‌ها دارند. همچنین، L0Soft عملکرد بهتری در مقابل ISP-hard برای تُنکی‌های کم نشان می‌دهد.

داده‌های طبیعی

در این قسمت، بازسازی تصاویر طبیعی از روی اندازه‌گیری‌های تصادفی آن‌ها را بررسی کرده و عملکرد الگوریتم‌های مختلف را مقایسه می‌کنیم. برای توضیح در مورد چگونگی انجام این کار، فرض کنید $\mathbf{X} \in \mathbb{R}^{\sqrt{n} \times \sqrt{n}}$ یک تصویر طبیعی بوده و $\mathbf{x} \in \mathbb{R}^n$ نسخه‌ی برداری شده‌ی آن را نشان دهد. در اینصورت، هدف بازسازی \mathbf{x} از روی $\mathbf{y} = \Phi \mathbf{x}$

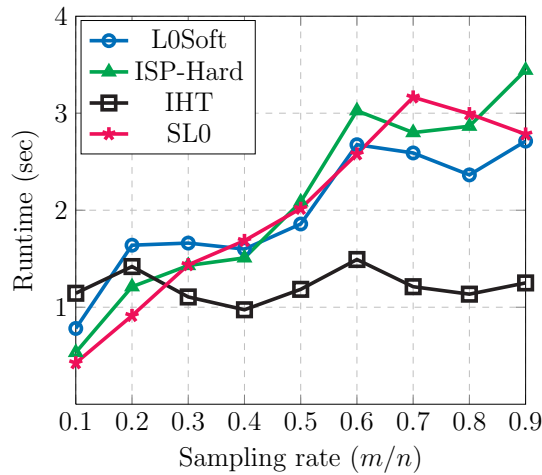


شکل ۵-۱۷: نتایج بازسازی تصاویر 32×32 از روی اندازه‌گیری‌های گوسی و برای مقادیر مختلف نرخ نمونه‌برداری، m/n .

است که در آن $\Phi \in \mathbb{R}^{m \times n}$ با $m < n$ ماتریسی تصادفی بوده که درایه‌های آن توزیع نُرمال دارند. همانطور که در قسمت ۲-۵ مرور شد، طبق روال حسگری فشرده، برای انجام بازسازی از فرض تُنک بودن تصاویر طبیعی در یک حوزه مانند $\Psi \in \mathbb{R}^{n \times p}$ ، با $p > n$ استفاده می‌شود. در واقع، فرض می‌شود در رابطه‌ی $\mathbf{x} = \Psi \mathbf{a}$ ، تُنک است. در نتیجه، ابتدا باید تُنک‌ترین جواب $\mathbf{y} = \Phi \Psi \mathbf{a} = \mathbf{A} \mathbf{a}$ را یافته و سپس با ضرب در Ψ ، تخمینی از \mathbf{x} را بدست آورد.

برای این منظور، ما تصاویری 32×32 از مجموعه‌ی دادگان CIFAR-10^۱ انتخاب کردیم. بنابراین، $n = 1024$. دیکشنری تُنک‌کننده‌ی Ψ را یک ماتریس DCT فوق کامل با ابعاد 1024×4096 گرفتیم. سپس، الگوریتم‌های مختلف را برای بازسازی تصاویر تست از روی اندازه‌گیری‌های گوسی با نرخ‌های مختلف نمونه‌برداری، که بصورت m/n تعریف می‌شود، اجرا کردیم. نتایج در شکل ۵-۱۷ نشان داده شده‌اند. بار دیگر، این نتایج حاکی از عملکرد نسبتاً ضعیف IHT در قیاس با سایر الگوریتم‌ها هستند. علاوه بر این، الگوریتم L0Soft قادر است تصاویر را با

^۱<https://www.cs.toronto.edu/~kriz/cifar.html>



شکل ۵-۱۸: زمان متوسط (برحسب ثانیه) برای الگوریتم‌های مختلف.

کیفیت بیشتری بازسازی کند، که در مواردی این برتری به میزان ۲-۳ دسی‌بل بهبود در کیفیت است. شکل ۵-۱۸ زمان اجرای متوسط الگوریتم‌ها را برحسب نرخ نمونه‌برداری نشان می‌دهد. همانطور که مشخص است، الگوریتم‌های L0Soft و SL0، ISP-Hard، IHT نیز اگر چه زمان اجرای کمتری دارد، اما کیفیت نهایی بازسازی آن مناسب نیست.

۴-۵ الگوریتم پراکسیمال-تصویرسازی پیاپی (IPP)

در این قسمت، الگوریتمی برای کمینه‌سازی یک تابع ناهموار تشویق‌کننده^۱ تُنکی ارائه می‌شود. الگوریتم پیشنهادی که IPP^۱ نام دارد [۶۳]، از روش جریمه به همراه ایده‌های پراکسیمال استفاده می‌کند. همچنین، این الگوریتم مجهز به یک گام درونیابی نیز هست که سرعت همگرایی و کیفیت آن را بهبود می‌دهد. در ادامه، ابتدا این الگوریتم را با جزئیات بررسی کرده، و سپس همگرایی آن را اثبات می‌کنیم. در نهایت، برای ارزیابی و مقایسه‌ی عملکرد آن، نتایج شبیه‌سازی روی داده‌های مصنوعی و نیز داده‌های طبیعی ارائه می‌شود.

۱-۴-۵ فرمول‌بندی مسأله

مسأله‌ای که برای بازیابی نمایش تُنک مورد بررسی قرار می‌دهیم عبارت است از

$$\min_{\mathbf{x}} J(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{Ax}\|_2 \leq \epsilon, \quad (58-5)$$

^۱Iterative Proximal Projection

که در آن، J یک تابع ناهموار و غیرمحدب تشویق‌کنندهی تُنکی است، مانند نُرم صفر، و

$$\mathcal{A}_\epsilon \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon\}. \quad (59-5)$$

از آنجائی که تابع J ناهموار و غیرمحدب فرض می‌شود، حل مسأله‌ی فوق در حالت کلی دشوار است. با این حال، ساختار ویژه‌ی تابع هدف باعث می‌شود که بتوان از روش‌های پراکسیمال استفاده کرد. برای این منظور، فرم معادل زیر از (58-5) را در نظر بگیرید

$$\min_{\mathbf{x}, \mathbf{z}} J(\mathbf{z}) + \delta_{\mathcal{A}_\epsilon}(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{z} = \mathbf{x}, \quad (60-5)$$

که در آن، متغیر کمکی \mathbf{z} تعریف شده است. سپس، این مسأله را با استفاده از روش جریمه حل می‌کنیم که منجر به تعریف مسأله‌ی زیر می‌شود

$$(P_\alpha) : \quad \min_{\mathbf{x}, \mathbf{z}} J(\mathbf{z}) + \delta_{\mathcal{A}_\epsilon}(\mathbf{x}) + \frac{1}{\alpha} \|\mathbf{x} - \mathbf{z}\|_2, \quad (61-5)$$

که $\alpha > 0$ ضریب جریمه است. این مسأله را در ادامه به کمک روش بهینه‌سازی نوبتی حل می‌کنیم. برای نیل به

این هدف، تکرارهای زیر باید انجام بشود

$$\begin{cases} \mathbf{z}_{k+1} = \operatorname{argmin}_{\mathbf{z}} \alpha J(\mathbf{z}) + \frac{1}{\alpha} \|\mathbf{z} - \mathbf{x}_k\|_2 = \operatorname{prox}_{\alpha J}(\mathbf{x}_k) \\ \mathbf{x}_{k+1} = \operatorname{argmin}_{\mathbf{x}} \delta_{\mathcal{A}_\epsilon}(\mathbf{x}) + \frac{1}{\alpha} \|\mathbf{x} - \mathbf{z}_{k+1}\|_2 = \operatorname{prox}_{\delta_{\mathcal{A}_\epsilon}}(\mathbf{z}_{k+1}) \end{cases} \quad (62-5)$$

و یا به فرم خلاصه‌تر

$$\mathbf{x}_{k+1} = \operatorname{prox}_{\delta_{\mathcal{A}_\epsilon}}(\operatorname{prox}_{\alpha J}(\mathbf{x}_k)) = \mathcal{P}_{\mathcal{A}_\epsilon}(\operatorname{prox}_{\alpha J}(\mathbf{x}_k)). \quad (63-5)$$

سپس، مشابه روش جریمه، یک دنباله‌ی کاهشی از ضرایب جریمه در نظر گرفته شده و تکرارهای فوق برای هر مقدار ضریب جریمه انجام می‌شود. برای افزایش کیفیت الگوریتم، از روش‌های سرعت‌بخشی پراکسیمال استفاده

می‌کنیم [۱۳۲، ۱۳۳] که در اینجا بصورت زیر است

$$\begin{cases} \tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1}) \\ \mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{A}_\epsilon}(\operatorname{prox}_{\alpha J}(\tilde{\mathbf{x}}_k)) \end{cases} \quad (64-5)$$

در رابطه‌ی فوق، w یک پارامتر وزن‌دهی است. الگوریتم نهایی، که آن را IPP نام گذاشته‌ایم، در شکل 5-۱۹ خلاصه شده است. در این الگوریتم، μ_x و μ_z پارامترهایی بین صفر و یک هستند که با میل کردن به سمت یک، باعث می‌شوند تا الگوریتم IPP به (64-5) میل کند. در واقع، دلیل معرفی این پارامترها سهولت در اثبات همگرایی الگوریتم پیشنهادی است. همگرایی این الگوریتم در قضیه‌ی زیر بیان شده است.

قضیه 5-۲ در الگوریتم 5-۱۹ فرض کنید $0 \leq w < \frac{1}{\max(\mu_x, \mu_z)} - 1$. در اینصورت، دنباله‌ی $\{\mathbf{u}_k \triangleq (\mathbf{x}_k, \mathbf{z}_k)\}_{k=0}^\infty$ تولید شده توسط این الگوریتم برای یک مقدار ثابت α به یک نقطه‌ی بحرانی، \mathbf{u}^* ، از تابع هدف تعریف شده در

- توصیف: الگوریتم IPP برای حل مسأله‌ی (۵-۵۸)
- ورودی: $c, w, \alpha_1, \epsilon, (\mathbf{x}_0, \mathbf{z}_0), \mathbf{A}, \mathbf{y}$
- مقداردهی اولیه: $\alpha = \alpha_1, \mathbf{x}_0 = \mathbf{z}_0 = \mathbf{A}^T \mathbf{y}, k = 0$
- شروع الگوریتم: گام‌های زیر را تا زمانی که $\alpha > \alpha_f$ انجام بده
- تا زمانی که $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2 > \tau$ انجام بده:
 ۱. $\tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1})$
 ۲. $\mathbf{z}_{k+1} = \text{prox}_{\mu_z \cdot \alpha \cdot J}(\mathbf{z}_k + \mu_z (\tilde{\mathbf{x}}_k - \mathbf{z}_k))$
 ۳. $\mathbf{x}_{k+1} = \mathcal{P}_{A_\epsilon}(\mathbf{x}_k + \mu_x (\mathbf{z}_{k+1} - \mathbf{x}_k))$
 ۴. $k \rightarrow k + 1$
- $\alpha \leftarrow c \cdot \alpha$
- خروجی: \mathbf{x}_k

شکل ۵-۱۹: الگوریتم پیشنهادی IPP برای حل مسأله‌ی (۵-۵۸).

(۵-۶۱) میل می‌کند.

اثبات این قضیه در قسمت ۵-۴-۴ شرح داده می‌شود. لازم به ذکر است که قضیه‌ی فوق، همگرایی الگوریتم IPP را برای یک مقدار خاص از α (در واقع، حلقه‌ی داخلی این الگوریتم، که متناظر با گام‌های ۱ تا ۴ در شکل ۵-۱۹ است) تضمین می‌کند. اما باید توجه داشت که این همچنین معادل با همگرایی کل الگوریتم است. برای توضیح دقیق‌تر، دقت کنید که تنها آخرین حلقه‌ی داخلی الگوریتم، یعنی حلقه‌ی متناظر با $\alpha = \alpha_f$ ، مهم بوده و حلقه‌های داخلی قبل از آن صرفاً نقش مقداردهی کارآمد این حلقه را برعهده دارند. در نتیجه، از آنجایی که همگرایی تک تک این حلقه‌ها در قضیه‌ی فوق تضمین می‌شود، می‌توان گفت کل الگوریتم، که در واقع همان آخرین حلقه‌ی داخلی است، نیز همگرا می‌شود.

۵-۴-۲ ارتباط با الگوریتم‌های دیگر

در این قسمت، ارتباط الگوریتم IPP را با الگوریتم‌های موجود بررسی می‌کنیم. نزدیک‌ترین الگوریتم به IPP، الگوریتم ISP است که در قسمت‌های قبل معرفی کردیم. در واقع، با دقت در این دو الگوریتم می‌توان فهمید که ساختار کلی هر دو یکی است. یک تفاوت بین این دو الگوریتم در این است که فرمول‌بندی الگوریتم IPP بر خلاف الگوریتم ISP بصورت مستقیم بدست آمد. در حقیقت، روال بدست آوردن IPP کاملاً مشخص است.

اما الگوریتم ISP تنها برای حالت هموار بررسی شده است و فرمول‌بندی حالت ناهموار تنها با الهام از حالت هموار بدست آمده است. تفاوت دیگر هم در فرمول‌بندی نهایی دو الگوریتم است که الگوریتم IPP را به دلیل وجود پارامترهای μ_x, μ_z و ضریب وزنی w متفاوت می‌کند. همچنین، برای الگوریتم ISP در حالت ناهموار اثبات همگرایی ارائه نشده است در حالی که الگوریتم IPP اثبات همگرایی دارد.

الگوریتم مشابه دیگر، الگوریتم^۱ SCSA است [۸۴] که تابع زیر را برای اعمال تُنکی استفاده می‌کند:

$$f_{\sigma}^{\text{scsa}}(x) = \lambda \left\{ 1 - \exp\left(-\frac{|x|}{\sigma}\right) \right\}. \quad (۶۵-۵)$$

مشابه تابع SL0، تابع جریمه‌ی SCSA نیز یک پارامتر هموارکننده‌ی σ دارد. اما تابع f_{σ}^{scsa} یک تقریب بهتر از تابع نُرم صفر نسبت به SL0 بدست می‌دهد [۸۴]. این الگوریتم مسأله‌ی زیر را به کمک روش‌های پراکسیمال حل می‌کند:

$$\min_{\mathbf{x}} \left\{ \frac{1}{\nu} \|\mathbf{y} - \mathbf{Ax}\|_{\nu} + f_{\sigma}^{\text{scsa}}(\mathbf{x}) \right\}. \quad (۶۶-۵)$$

مشابه الگوریتم SL0، الگوریتم SCSA نیز از یک دنباله‌ی کاهشی برای σ استفاده می‌کند. بعلاوه، نشان داده شده است که برای مقادیر بزرگ σ ، عملگر پراکسیمال عبارت است از تابع آستانه‌گذاری نرم، و برای مقادیر کوچک این پارامتر، برابر با تابع آستانه‌گذاری سخت است. در نتیجه، فرم تابع آستانه‌گذاری در طول تکرارهای الگوریتم تغییر می‌کند حال آن که در مورد الگوریتم IPP شکل کلی تابع آستانه‌گذاری ثابت است و تنها مقدار سطح آستانه تغییر می‌کند.

۳-۴-۵ توابع ناهموار مختلف

بعنوان مثال‌هایی از تابع J می‌توان به توابع نُرم صفر و نُرم یک اشاره کرد. علاوه بر این دو، یک تابع کارآمدتری وجود دارد که عمدتاً در کاربردهای آماری مطرح شده است. این تابع، SCAD^۲ نام دارد [۵۵] که عملگر پراکسیمال آن با عبارت زیر داده می‌شود [۵۵]

^۱Successive Concave Sparsity Approximation

^۲Smoothly clipped absolute deviation penalty

$$\mathcal{T}_{\lambda,a}^{scad}(x) \triangleq \begin{cases} \operatorname{sgn}(x)(|x| - \lambda)_+ & |x| \leq 2\lambda \\ \frac{(a-1)x - \operatorname{sgn}(x)a\lambda}{a-2} & 2\lambda < |x| \leq a\lambda \\ x & |x| > a\lambda \end{cases} \quad (۶۷-۵)$$

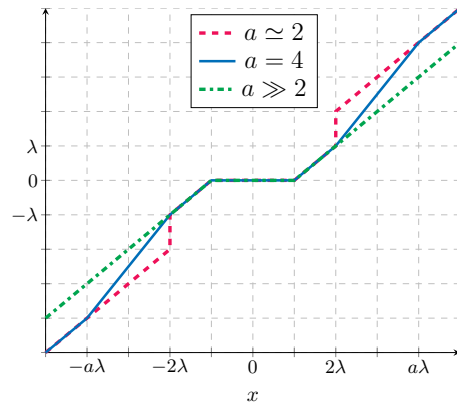
همانطور که این عبارت نشان می‌دهد، تابع آستانه‌گذاری SCAD برای مقادیر کوچک ورودی متناظر با تابع آستانه‌گذاری نرم است و هرچه دامنه‌ی ورودی بزرگتر می‌شود، رفتار آن به سمت تابع آستانه‌گذاری سخت میل می‌کند. این تابع یک پارامتر اضافی a دارد که در واقع شیب گذار از آستانه‌گذاری نرم به آستانه‌گذاری سخت را مشخص می‌کند. این تابع برای مقادیر مختلف a در شکل ۵-۲۰ رسم شده است. در کاربردهای رگرسیون و انتخاب متغیر نشان داده شده است که تابع SCAD عملکرد بهتری نسبت به توابع آستانه‌گذاری سخت و نرم دارد؛ چرا که بصورت همزمان سه ویژگی مهم بدون بایاس بودن، اعمال تُنکی و پیوستگی را برآورده می‌کند [۵۵].

در ادامه، نشان می‌دهیم که تابع آستانه‌گذاری SL0، شباهت بسیاری به تابع آستانه‌گذاری SCAD داشته، و ترکیبی از توابع آستانه‌گذاری سخت و نرم است. این توابع در شکل ۵-۲۱ رسم شده‌اند. رابطه‌ی بین λ و σ را به راحتی می‌توان از روی نقطه‌ی تماس توابع آستانه‌گذاری SL0 و تابع آستانه‌گذاری نرم بدست آورد. بعد از محاسباتی سراسری، خواهیم داشت $\lambda = \frac{\sigma}{\sqrt{2e}}$. در نتیجه، همانطور که از شکل ۵-۲۱ می‌توان دید، برای $|x| \leq 2\lambda$ و یا بصورت معادل $|x| \leq \sigma\sqrt{2/e}$ ، تابع آستانه‌گذاری SL0 تقریبی هموار از تابع آستانه‌گذاری نرم ارائه می‌دهد، حال آنکه هرچه دامنه‌ی ورودی زیاد می‌شود، این تابع به سمت تابع آستانه‌گذاری سخت میل می‌کند. با مقایسه این رفتار با تابع آستانه‌گذاری SCAD می‌توان دید که تابع آستانه‌گذاری SL0 در واقع تقریبی هموار برای این تابع است، اما تنها برای یک مقدار مشخص از پارامتر a . در واقع، یک مزیت مهم تابع آستانه‌گذاری SCAD این است که با تغییر پارامتر a می‌توان فرم‌های مختلفی از آستانه‌گذاری را داشت که این موضوع برخلاف آستانه‌گذاری SL0 است.

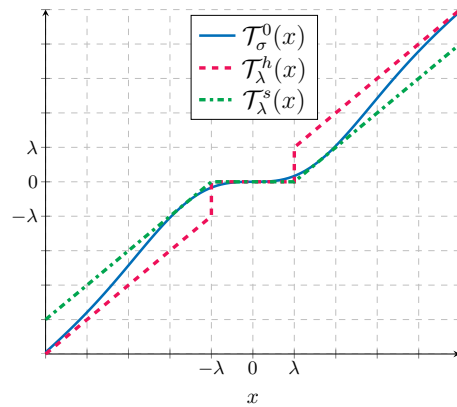
۴-۴-۵ آنالیز همگرایی

برای اثبات همگرایی الگوریتم IPP که در قضیه‌ی ۵-۲ بیان شده است، فرم معادل زیر از مسأله‌ی (۴۴) را در نظر بگیرید

$$\min_{\mathbf{z}, \mathbf{x}} r_1(\mathbf{z}) + r_2(\mathbf{x}) + Q(\mathbf{x}, \mathbf{z}), \quad (۶۸-۵)$$



شکل ۵-۲۰: نمودارهای تابع آستانه‌گذاری SCAD که در (۶۷-۵) تعریف شده است برای مقادیر مختلف a .



شکل ۵-۲۱: نمودارهای آستانه‌گذاری SL0 به همراه توابع آستانه‌گذاری سخت و نرم. در این شکل، $\sigma = \lambda\sqrt{2e}$.

که $Q(\mathbf{x}, \mathbf{z}) \triangleq \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2$ ، $r_1(\mathbf{z}) \triangleq \alpha J(\mathbf{z})$ و $r_2(\mathbf{x}) \triangleq \delta_{A_\epsilon}(\mathbf{x})$. در اینصورت، استفاده از روش‌های پراکسیمال منجر به رابطه‌ی زیر

$$\mathbf{z}_{k+1} = \underset{\mathbf{z}}{\operatorname{argmin}} r_1(\mathbf{z}) + \langle \nabla_z Q(\tilde{\mathbf{x}}_k, \mathbf{z}_k), \mathbf{z} - \mathbf{z}_k \rangle + \frac{1}{2\mu_z} \|\mathbf{z} - \mathbf{z}_k\|_2^2 \quad (۶۹-۵)$$

برای به‌روزرسانی \mathbf{z} می‌شود که $\tilde{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1})$ و $0 < \mu_x, \mu_z < 1$. همچنین، رابطه‌ی به‌روزرسانی \mathbf{x} به فرم زیر خواهد بود

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} r_2(\mathbf{x}) + \langle \nabla_x Q(\mathbf{x}_k, \mathbf{z}_{k+1}), \mathbf{x} - \mathbf{x}_k \rangle + \frac{1}{2\mu_x} \|\mathbf{x} - \mathbf{x}_k\|_2^2. \quad (۷۰-۵)$$

سپس، چون \mathbf{z}_{k+1} حداقل‌کننده‌ی (۶۹-۵) است، داریم

$$r_1(\mathbf{z}_{k+1}) + \langle \nabla_z Q(\tilde{\mathbf{x}}_k, \mathbf{z}_k), \mathbf{z}_{k+1} - \mathbf{z}_k \rangle + \frac{1}{2\mu_z} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^2 \leq r_1(\mathbf{z}_k). \quad (۷۱-۵)$$

از طرف دیگر، لم کاهشی (لم ۳-۴) می‌گوید که

$$Q(\mathbf{x}_k, \mathbf{z}_{k+1}) \leq Q(\mathbf{x}_k, \mathbf{z}_k) + \langle \nabla_z Q(\mathbf{x}_k, \mathbf{z}_k), \mathbf{z}_{k+1} - \mathbf{z}_k \rangle + \frac{1}{2} \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^2. \quad (۷۲-۵)$$

جمع دو رابطه‌ی فوق نتیجه می‌دهد

$$r_1(\mathbf{z}_{k+1}) + Q(\mathbf{x}_k, \mathbf{z}_{k+1}) \leq r_1(\mathbf{z}_k) + Q(\mathbf{x}_k, \mathbf{z}_k) + \langle \nabla_z Q(\mathbf{x}_k, \mathbf{z}_k) - \nabla_z Q(\tilde{\mathbf{x}}_k, \mathbf{z}_k), \mathbf{z}_{k+1} - \mathbf{z}_k \rangle + \left(\frac{1}{\gamma} - \frac{1}{\gamma\mu_z}\right) \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma}^2. \quad (۷۳-۵)$$

حال، استفاده از نامساوی کوشی-شوارتز در رابطه‌ی فوق منجر به رابطه‌ی زیر می‌شود

$$r_1(\mathbf{z}_{k+1}) + Q(\mathbf{x}_k, \mathbf{z}_{k+1}) \leq r_1(\mathbf{z}_k) + Q(\mathbf{x}_k, \mathbf{z}_k) + \|\nabla_z Q(\mathbf{x}_k, \mathbf{z}_k) - \nabla_z Q(\tilde{\mathbf{x}}_k, \mathbf{z}_k)\|_{\gamma} \cdot \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma} + \left(\frac{1}{\gamma} - \frac{1}{\gamma\mu_z}\right) \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma}^2. \quad (۷۴-۵)$$

سپس، مقدارگذاری گرادیان‌های رابطه‌ی فوق و استفاده از نامساوی یانگ [۱۳۶]، یعنی نامساوی زیر

$$a \cdot b \leq \frac{a^2}{\gamma\epsilon} + \frac{\epsilon b^2}{\gamma}, \quad \forall a, b \geq 0, \epsilon > 0 \quad (۷۵-۵)$$

با $\epsilon = 1$ ، رابطه‌ی زیر را بدست می‌دهد

$$r_1(\mathbf{z}_{k+1}) + Q(\mathbf{x}_k, \mathbf{z}_{k+1}) \leq r_1(\mathbf{z}_k) + Q(\mathbf{x}_k, \mathbf{z}_k) + w \cdot \|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\gamma} \cdot \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma} + \left(\frac{1}{\gamma} - \frac{1}{\gamma\mu_z}\right) \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma}^2 \\ \leq r_1(\mathbf{z}_k) + Q(\mathbf{x}_k, \mathbf{z}_k) + \frac{w}{\gamma} \cdot \|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\gamma}^2 + \frac{w}{\gamma} \cdot \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma}^2 + \left(\frac{1}{\gamma} - \frac{1}{\gamma\mu_z}\right) \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma}^2, \quad (۷۶-۵)$$

که به فرم زیر ساده می‌شود

$$r_1(\mathbf{z}_{k+1}) + Q(\mathbf{x}_k, \mathbf{z}_{k+1}) \leq r_1(\mathbf{z}_k) + Q(\mathbf{x}_k, \mathbf{z}_k) + \frac{w}{\gamma} \cdot \|\mathbf{x}_k - \mathbf{x}_{k-1}\|_{\gamma}^2 + \left(\frac{w}{\gamma} + \frac{1}{\gamma} - \frac{1}{\gamma\mu_z}\right) \cdot \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_{\gamma}^2. \quad (۷۷-۵)$$

با دنبال کردن روالی مشابه، برای \mathbf{x} بدست می‌آوریم

$$r_2(\mathbf{x}_{k+1}) + \langle \nabla_x Q(\mathbf{x}_k, \mathbf{z}_{k+1}), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{1}{\gamma\mu_x} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\gamma}^2 \leq r_2(\mathbf{x}_k). \quad (۷۸-۵)$$

همچنین، لم کاهشی نتیجه می‌دهد

$$Q(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}) \leq Q(\mathbf{x}_k, \mathbf{z}_{k+1}) + \langle \nabla_x Q(\mathbf{x}_k, \mathbf{z}_{k+1}), \mathbf{x}_{k+1} - \mathbf{x}_k \rangle + \frac{1}{\gamma} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\gamma}^2. \quad (۷۹-۵)$$

اضافه کردن دو سمت روابط فوق به رابطه‌ی زیر منتهی می‌شود

$$r_2(\mathbf{x}_{k+1}) + Q(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}) \leq \left(\frac{1}{\gamma} - \frac{1}{\gamma\mu_x}\right) \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\gamma}^2 + r_2(\mathbf{x}_k) + Q(\mathbf{x}_k, \mathbf{z}_{k+1}). \quad (۸۰-۵)$$

حال، با تعریف $F(\mathbf{x}, \mathbf{z}) \triangleq r_1(\mathbf{z}) + r_2(\mathbf{x}) + Q(\mathbf{x}, \mathbf{z})$ و جمع زدن دو رابطه‌ی (۷۷-۵) و (۸۰-۵) خواهیم داشت

$$\gamma_x \cdot \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 + \gamma_z \cdot \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^2 \leq F(\mathbf{x}_k, \mathbf{z}_k) - F(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}), \quad (۸۱-۵)$$

که در آن

$$\gamma_x = \left(\frac{1}{\sqrt{\mu_x}} - \frac{w}{\sqrt{\mu_x}} - \frac{1}{\sqrt{\mu_x}} \right), \quad \gamma_z = \left(\frac{1}{\sqrt{\mu_z}} - \frac{w}{\sqrt{\mu_z}} - \frac{1}{\sqrt{\mu_z}} \right). \quad (۸۲-۵)$$

طبق فرض، $w \leq \frac{1}{\max(\mu_x, \mu_z)} - 1$. بنابراین $\gamma_x \geq 0$ و $\gamma_z \geq 0$. سپس، نوشتن عبارت (۸۱-۵) برای $k \geq 0$ و جمع

زدن روابط حاصل منجر می‌شود به

$$\sum_{k=0}^{\infty} \gamma_x \cdot \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 + \gamma_z \cdot \|\mathbf{z}_{k+1} - \mathbf{z}_k\|_2^2 \leq F(\mathbf{x}_0, \mathbf{z}_0) - F(\mathbf{x}_{\infty}, \mathbf{z}_{\infty}). \quad (۸۳-۵)$$

از طرف دیگر، رابطه‌ی (۸۱-۵) نشان می‌دهد که دنباله‌ی $\{F(\mathbf{x}_k, \mathbf{z}_k)\}_{k=0}^{\infty}$ غیرافزایشی است. همچنین، چون

$F(\mathbf{x}, \mathbf{z}) \geq 0$ ، این دنباله به یک مقدار محدود $F(\mathbf{x}_{\infty}, \mathbf{z}_{\infty})$ میل می‌کند. بنابراین، سمت راست رابطه‌ی (۸۳-۵)

محدود و نامنفی است که با توجه به آن می‌توان نوشت

$$\mathbf{x}_{k+1} \rightarrow \mathbf{x}_k, \quad \mathbf{z}_{k+1} \rightarrow \mathbf{z}_k, \quad (۸۴-۵)$$

از طرفی، می‌توان نشان داد که دنباله‌ی $\{(\mathbf{x}_k, \mathbf{z}_k)\}_{k=0}^{\infty}$ کراندار است. در واقع، چون $\mathbf{x}_{k+1} = \text{prox}_{\mu_x r_1}(\tilde{\mathbf{x}}_k) =$

یک مجموعه‌ی کراندار است، نتیجه می‌شود که دنباله‌ی \mathbf{x}_k کراندار است. برای اثبات کراندار بودن

\mathbf{z}_k هم کافی است به این نکته دقت کنیم که $|\text{prox}_{\mu_z r_1}(x)| \leq |x| \forall x$. سپس، استفاده از رابطه‌ی به‌روزرسانی \mathbf{z}

و انجام محاسبات ساده، نتیجه‌ی مطلوب بدست می‌آید.

گام بعدی این است که نشان دهیم دنباله‌ی $\{(\mathbf{x}_k, \mathbf{z}_k)\}_{k=0}^{\infty}$ به سمت نقاط بحرانی تابع هدف میل می‌کند.

برای این منظور، از (۶۹-۵) در مورد شرط بهینگی \mathbf{z}_{k+1} داریم

$$0 \in \partial r_1(\mathbf{z}_{k+1}) + \nabla_z Q(\tilde{\mathbf{x}}_k, \mathbf{z}_k) + \frac{1}{\mu_z}(\mathbf{z}_{k+1} - \mathbf{z}_k), \quad (۸۵-۵)$$

که از روی آن می‌توان نوشت

$$A_z^k \in \partial_z F(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}), \quad (۸۶-۵)$$

که در آن

$$A_z^k \triangleq (\mathbf{x}_k - \mathbf{x}_{k+1}) + w(\mathbf{x}_k - \mathbf{x}_{k-1}) + \left(1 - \frac{1}{\mu_z}\right)(\mathbf{z}_{k+1} - \mathbf{z}_k). \quad (۸۷-۵)$$

به طرز مشابه بدست می‌آوریم

$$A_x^k \in \partial_x F(\mathbf{x}_{k+1}, \mathbf{z}_{k+1}) \quad (۸۸-۵)$$

که

$$A_x^k \triangleq \left(1 - \frac{1}{\mu_z}\right)(\mathbf{x}_{k+1} - \mathbf{x}_k). \quad (۸۹-۵)$$

سپس، داریم $(A_z^k, A_x^k) \in \partial F(\mathbf{x}_{k+1}, \mathbf{z}_{k+1})$. حال، با توجه به (۸۴-۵) داریم

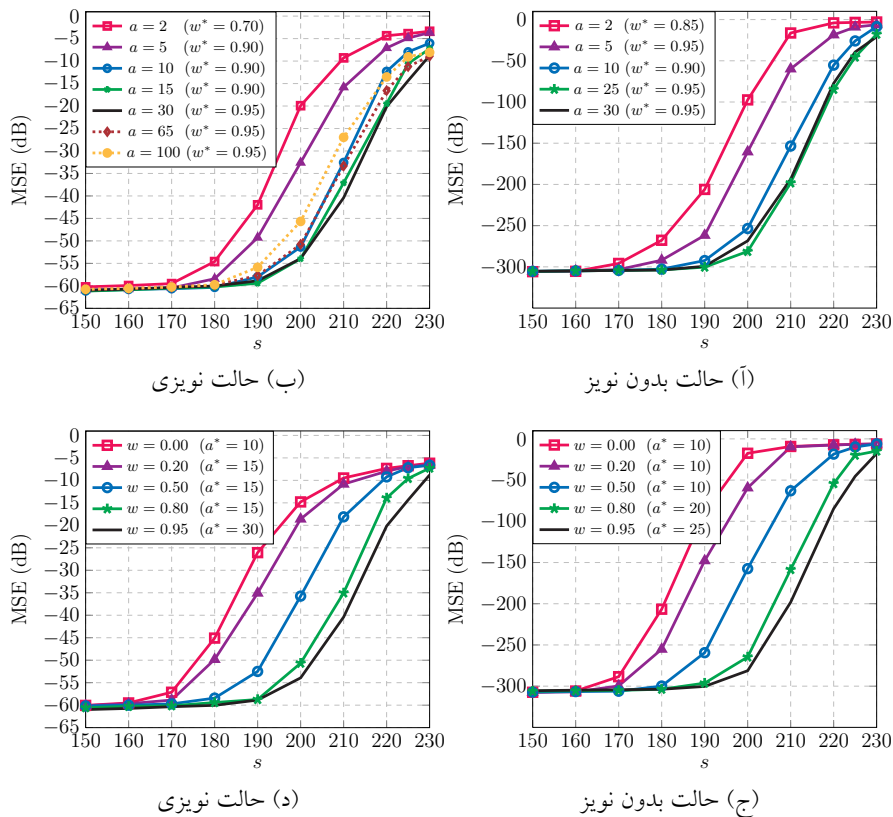
$$A_z^k, A_x^k \rightarrow \mathbf{o} \quad (۹۰-۵)$$

برای $k \rightarrow \infty$ فرض کنید $(\mathbf{x}^*, \mathbf{z}^*)$ یک نقطه‌ی حدی $\{\mathbf{x}_k, \mathbf{z}_k\}_{k=0}^{\infty}$ باشد. در اینصورت، با توجه به پیوسته بودن $\nabla_x Q$ و $\nabla_z Q$ و از پائین نیمه‌پیوسته بودن r_1 و r_2 و با استدلال‌های مشابه لم ۵ و قضیه‌ی ۱ از [۱۹]، نتیجه می‌گیریم که $(\mathbf{x}^*, \mathbf{z}^*)$ یک نقطه‌ی بحرانی از F است. عبارت دیگر، $\mathbf{o} \in \partial F(\mathbf{x}^*, \mathbf{z}^*)$. همچنین، دنباله‌ی $\{\mathbf{x}_k, \mathbf{z}_k\}_{k=0}^{\infty}$ به $(\mathbf{x}^*, \mathbf{z}^*)$ میل می‌کند.

۵-۴-۵ شبیه‌سازی

در این قسمت، عملکرد الگوریتم IPP با الگوریتم‌های خانواده‌ی ISP و تعدادی دیگر از الگوریتم‌های موجود، از جمله الگوریتم کمینه‌سازی نُرم ℓ_q [۵۹] برای $q = 0.5$ ، الگوریتم کمینه‌سازی نامقید تابع SCAD [۶۱] که آن را الگوریتم SCAD می‌نامیم، و الگوریتم SCSA-FIT [۸۴] مقایسه می‌شود. ابتدا، تاثیر دو پارامتر وزن، یعنی w و a بررسی می‌شود. سپس، نتایج مقایسه با الگوریتم ISP ارائه می‌شود. همچنین، نسخه‌ی بهبود داده شده‌ی الگوریتم‌های ISP را، با بکارگیری ایده‌ی وزن‌دهی توسط w ، مشابه آنچه در الگوریتم IPP انجام می‌شود، نیز مورد بررسی و مقایسه قرار می‌دهیم. این نسخه از ISP را ImpISP نامگذاری می‌کنیم. در ادامه، الگوریتم پیشنهادی با تعدادی از الگوریتم‌های معروف موجود، برای هم داده‌های مصنوعی و هم داده‌های طبیعی، مورد ارزیابی قرار می‌گیرد.

برای دیدن تاثیر پارامترهای IPP روی عملکرد آن، داده‌هایی مصنوعی با همان روال و ابعاد موجود در قسمت ۵-۲-۴ تولید می‌کنیم. همچنین، تابع J را تابع SCAD انتخاب کرده‌ایم. در نتیجه، الگوریتم حاصل را IPP-SCAD می‌نامیم. نتایج برای هر دو حالت نویزی (با نویز گوسی) و بدون نویز در شکل‌های ۵-۲۲ خلاصه شده است. در این شکل‌ها، الگوریتم IPP-SCAD برای مقادیر مختلف تُنکی و نیز مقادیر متفاوت پارامترهای w و a اجرا شده است. هنگام تغییر یک پارامتر، مقدار مناسب پارامتر دیگر نیز در داخل پرانتز درج شده است. با دقت در این شکل‌ها می‌توان دید که هر دوی پارامترهای a و w تاثیر زیادی روی عملکرد الگوریتم دارند. در واقع، همانطور که نتایج نشان می‌دهد، برای a مقداری حوالی ۲۰ و برای w مقداری حدود ۰/۹ عملکرد متوسط خوبی

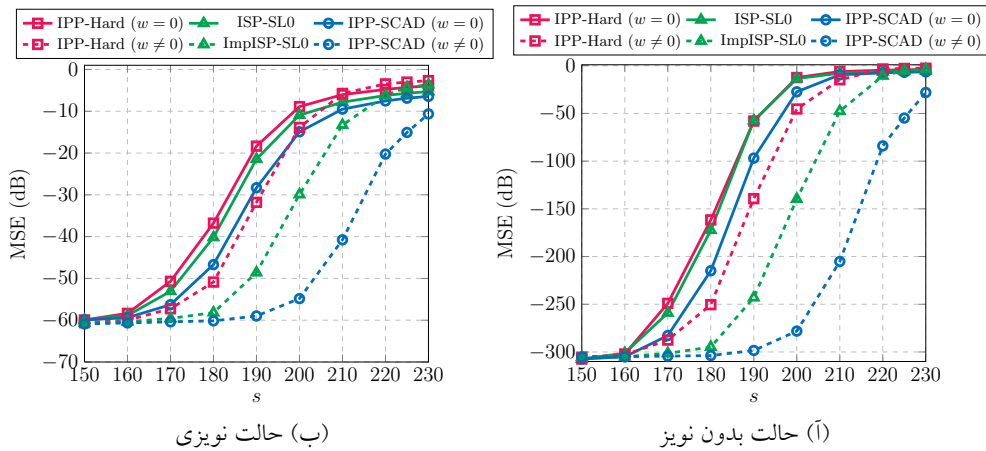


شکل ۵-۲۲: بررسی تاثیر پارامتر a در الگوریتم IPP-SCAD (ردیف بالا). برای هر مقدار a ، بهترین مقدار w نیز نشان داده شده است. محور افقی تعداد درایه‌های غیرصفر در سیگنال تُنک را نشان می‌دهد. ردیف پایین نیز اثر پارامتر w را نشان می‌دهد.

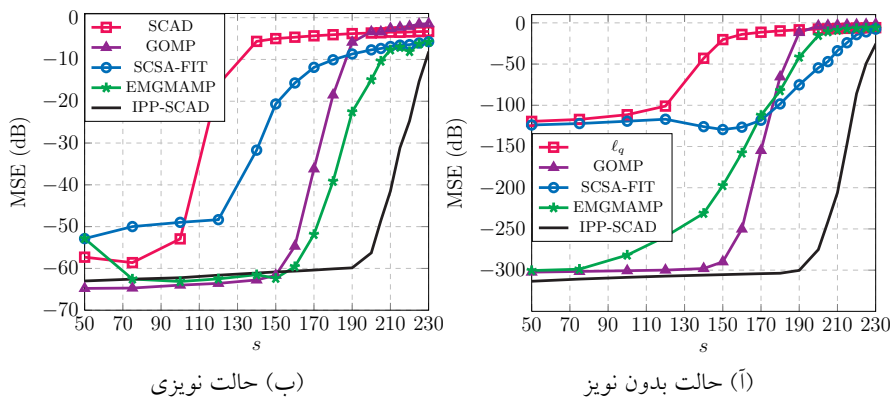
ارائه می‌دهد.

عملکرد الگوریتم‌های IPP و ISP با توابع آستانه‌گذاری مختلف و برای دو حالت اندازه‌گیری‌های نویزی و بدون نویز در شکل ۵-۲۳ مقایسه شده است. همانطور که دیده می‌شود، تابع آستانه‌گذاری SCAD عملکرد بسیار بهتری نسبت به سایر آستانه‌گذاری‌ها دارد. همچنین، تاثیر پارامتر w را بار دیگر می‌توان مشاهده نمود.

عملکرد الگوریتم‌های مختلف در بازیابی سیگنال‌های تُنک در حالت‌های بدون نویز و با نویز گوسی در شکل ۵-۲۴ بررسی شده است. همانطور که این شکل نشان می‌دهد، الگوریتم پیشنهادی عملکرد بسیار بهتری نسبت به سایر الگوریتم‌ها، به خصوص برای مقادیر بالاتر تعداد درایه‌های غیرصفر، دارد. برای مقایسه تقریبی پیچیدگی محاسباتی این الگوریتم‌ها، زمان متوسط دستیابی به سطوح MSE مشخص را در نظر گرفته‌ایم. نتایج مربوط به این آزمایش در جدول‌های ۵-۲ و ۵-۳، متناظر با حالت‌های بدون نویز و نویزی، خلاصه شده است.



شکل ۵-۲۳: مقایسه‌ی الگوریتم‌های IPP و ISP با توابع آستانه‌گذاری متفاوت و در دو حالت: با وزن مخالف صفر ($w = 0/95$ ، نمودارهای خط‌چین)، و با وزن صفر ($w = 0$ ، نمودارهای توپر).



شکل ۵-۲۴: مقایسه عملکرد الگوریتم IPP-SCAD با الگوریتم‌های دیگر برای حالت نویزی و بدون نویز.

مواردی که با خط تیره مشخص شده است مربوط به وضعیتی است که الگوریتم موردنظر قادر نبوده به سطح خطای مطلوب برسد. بررسی داده‌های این جدول‌ها عملکرد برتر الگوریتم IPP را نشان می‌دهد. بعلاوه، تاثیر پارامتر w در این الگوریتم با مقایسه‌ی دو حالت $w = 0$ و $w = 0/95$ آشکار است.

در ادامه، عملکرد الگوریتم‌های مختلف را در بازیابی بلوکی تصاویر طبیعی از روی اندازه‌گیری‌های گوسی آن‌ها ارزیابی می‌کنیم. برای این منظور، فرض کنید \mathbf{X} نشان‌دهنده‌ی ماتریس مقادیر پیکسل یک تصویر هدف باشد. تمام بلوک‌های با ابعاد 8×8 این ماتریس را با هم‌پوشانی پنجاه درصد جدا کرده و سپس آن‌ها را به بردارهای معادل با طول ۶۴ تبدیل می‌کنیم. اگر بردار i -ام را نشان دهد، در اینصورت، تعداد m اندازه‌گیری گوسی تصادفی از این بردار با استفاده از یک ماتریس گوسی $\Phi \in \mathbb{R}^{m \times 64}$ که درایه‌های آن از توزیع استاندارد گوسی انتخاب

جدول ۵-۲: زمان متوسط (برحسب ثانیه) الگوریتم‌های مختلف برای سطوح تُنکی گوناگون و در حالت بدون نویز. هر خانه از این جدول زمان متوسطی را نشان می‌دهد که هر الگوریتم صرف می‌کند تا به یک سطح خاص از MSE برسد. علامت خط تیره نشان می‌دهد که الگوریتم متناظر قادر نبوده به آن سطح از MSE برسد. بهترین نتیجه با فونت برجسته مشخص شده است.

میزان تُنکی		$s = 50$				$s = 100$				$s = 150$				$s = 200$			
الگوریتم \ MSE*(dB)																	
IPP($w = 0$)	۰/۱۹	۰/۱۷	۰/۱۵	۰/۱۰	۰/۲۰۰	۰/۱۸	۰/۱۶	۰/۱۱	۰/۲۴	۰/۲۱	۰/۱۷	۰/۱۱	-	-	-		
IPP($w = 0/95$)	۰/۲۱	۰/۱۸	۰/۱۵	۰/۱۰	۰/۲۲	۰/۱۸	۰/۱۵	۰/۱۰	۰/۲۵	۰/۲۰	۰/۱۶	۰/۱۱	۰/۸۱	۰/۷۶	۰/۶۵		
GOMP	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	۰/۰۶	-	-		
EMGMAMP	-	۰/۴۷	۰/۳۹	۰/۳۴	۰/۳۴	۰/۴۷	۰/۳۴	۰/۲۹	۰/۳۴	۰/۲۹	۰/۲۹	۰/۲۹	۰/۲۹	۰/۲۹	۰/۲۹		
SCSA-FIT	-	-	-	۰/۲۴	-	-	-	۰/۲۵	-	-	-	-	۱/۱۲	-	-		

جدول ۵-۳: مشابه جدول ۵-۲ ولی برای حالت نویزی.

میزان تُنکی		$s = 50$				$s = 140$				$s = 160$				$s = 200$			
الگوریتم \ MSE*(dB)																	
IPP($w = 0$)	۱/۶۰	۰/۵۰	۰/۴۲	۰/۳۵	۰/۳۹	۰/۵۳	۰/۴۵	۰/۳۹	۰/۶۱	۱/۰۲	۰/۹۵	۰/۸۱	-	-	-		
IPP($w = 0/95$)	۰/۷۲	۰/۴۹	۰/۴۰	۰/۳۴	۰/۴۲	۰/۵۳	۰/۴۵	۰/۳۸	۰/۳۳	۰/۵۳	۰/۴۵	۰/۳۸	-	۱/۲۲	۱/۰۹		
GOMP	۰/۰۲	۰/۰۲	۰/۰۲	۰/۰۱	۰/۱۴	۰/۱۳	۰/۱۳	۰/۱۱	۰/۶۷	۱/۶۶	۱/۶۶	۱/۶۶	۱/۶۶	-	-		
EMGMAMP	۳/۸۵	۳/۰۶	۳/۰۴	۲/۶۰	۰/۶۹	۰/۳۸	۰/۳۳	۰/۲۷	۰/۸۶	۰/۸۳	۰/۷۹	۰/۷۳	-	-	-		
SCSA-FIT	-	۰/۰۳	۰/۰۱	۰/۰۱	-	۰/۵۴	۰/۰۴	۰/۰۴	۳/۳۹	۰/۰۸	۰/۰۷	۲/۲۹	۲/۲۹	-	-		
SCAD	۰/۳۴	۰/۳۳	۰/۳۳	۰/۳۳	۰/۴۸	۰/۴۷	۰/۴۷	۰/۳۷	۰/۳۷	۰/۳۷	۰/۳۷	۰/۳۷	۰/۳۷	-	-		

می‌شود، گرفته می‌شود. اگر بردار اندازه‌گیری متناظر را با y_i نشان دهیم، خواهیم داشت $y_i = \Phi x_i$. سپس، هدف نهایی عبارت است از بازسازی تصویر اصلی X تنها با استفاده از بردارهای y_i . برای این منظور، ابتدا بلوک‌های اصلی x_i از روی y_i تخمین زده شده، و در نهایت، تخمین تصویر اصلی با متوسط‌گیری از این بلوک‌ها حاصل می‌شود. برای بازسازی بلوک‌ها، از فرض تُنک بودن آن‌ها در حوزه تبدیل DCT، که آن را با ماتریس Ψ نشان می‌دهیم، استفاده می‌شود. به عبارت دیگر، اگر s_i تُنک‌ترین جواب $y_i = \Phi \Psi s_i$ باشد، در اینصورت، $\hat{x}_i = \Psi s_i$ یک تخمین از x_i خواهد بود.

برای ارزیابی عملکرد الگوریتم‌ها در این آزمایش، سه تصویر طبیعی که در شکل ۵-۲۵ نشان داده شده‌اند، در نظر گرفته شده است. نتایج برای مقادیر مختلف نرخ نمونه‌برداری، که با $m/64 = \delta$ مشخص شده است، در جدول ۵-۴ آورده شده است. برای دیدن تاثیر پارامتر وزن در الگوریتم IPP، دو حالت $w = 0$ و $w = 0/85$ در نظر گرفته شده است. همچنین، شکل ۵-۲۵ یک مقایسه‌ی دیداری را برای تصاویر بازسازی شده با $\delta = 0/4$ با استفاده از الگوریتم IPP و SCSA (که با توجه به جدول ۵-۴ بهترین عملکرد را در بین سایر الگوریتم‌ها دارد)

جدول ۴-۵: مقایسه‌ی عملکرد الگوریتم‌های مختلف در بازیابی تصاویر از روی نسخه‌های فشرده‌ی بلوک‌های آنها و برای مقادیر مختلف نسبت‌های نمونه‌برداری که با δ نشان داده می‌شود. مقادیر گزارش شده، PSNR برحسب دی بی هستند. بهترین نتیجه با فونت برجسته مشخص شده است.

$\delta = 0.4$			$\delta = 0.3$			$\delta = 0.2$			$\delta = 0.1$			
Monarch	Barbara	House	Monarch	Barbara	House	Monarch	Barbara	House	Monarch	Barbara	House	
۲۷٫۶۵	۳۰٫۱۷	۳۴٫۱۵	۲۴٫۹۷	۲۷٫۳۸	۳۱٫۱۷	۲۲٫۷۱	۲۵٫۳۰	۲۸٫۵۶	۱۹٫۸۷	۲۲٫۹۹	۲۵٫۱۳	ℓ_q
۲۷٫۱۰	۲۹٫۶۹	۳۳٫۵۹	۲۴٫۴۳	۲۷٫۱۴	۳۰٫۸۵	۲۱٫۲۳	۲۳٫۹۴	۲۶٫۶۸	۱۹٫۰۳	۲۲٫۳۸	۲۵٫۰۰	GOMP
۲۷٫۷۹	۳۰٫۲۳	۳۴٫۴۹	۲۴٫۹۹	۲۷٫۳۷	۳۱٫۳۵	۲۲٫۶۴	۲۵٫۲۶	۲۸٫۶۲	۱۹٫۸۸	۲۲٫۹۶	۲۵٫۱۱	SCSA-FIT
۲۷٫۰۶	۲۹٫۷۲	۳۳٫۶۴	۲۴٫۶۵	۲۷٫۱۵	۳۰٫۹۱	۲۲٫۲۲	۲۵٫۰۶	۲۷٫۹۶	۱۹٫۶۹	۲۲٫۹۴	۲۵٫۰۲	EMGMAMP
۲۷٫۷۰	۳۰٫۴۱	۳۳٫۵۶	۲۴٫۷۳	۲۷٫۶۱	۳۱٫۱۱	۲۲٫۹۵	۲۵٫۳۶	۲۷٫۷۲	۲۰٫۴۶	۲۳٫۰۸	۲۵٫۳۱	($w = 0$) IPP
۲۸٫۴۷	۳۰٫۹۲	۳۵٫۰۱	۲۵٫۵۵	۲۸٫۱۷	۳۲٫۵۰	۲۳٫۴۵	۲۵٫۶۳	۲۸٫۶۵	۲۰٫۶۷	۲۳٫۳۸	۲۵٫۵۷	($w = 0.85$) IPP

نشان می‌دهد. با بررسی این نتایج مشاهده می‌شود که الگوریتم IPP با وزن غیرصفر نتایج بهتری نسبت به حالت بدون وزن ارائه می‌دهد. بعلاوه، عملکرد الگوریتم پیشنهادی از الگوریتم‌های دیگر نیز بهتر است.

۵-۵ جمع‌بندی

در این فصل، تعدادی الگوریتم برای بازیابی نمایش تُنک ارائه کردیم. ابتدا، الگوریتم‌های ISP معرفی شدند که در واقع تعمیمی از الگوریتم SL0 هستند. سپس، الگوریتم L0Soft معرفی شد که از نظر ساختار با الگوریتم‌های ISP تفاوتی قابل ملاحظه دارد. اساس این الگوریتم تقریب زدن هموار تابع نُرم صفر، اما با شیوه‌ای متفاوت از روش معمول مانند SL0، است. در ادامه، الگوریتم IPP معرفی شد که سعی در حل مسأله‌ی کمینه‌سازی یک تابع تشویق‌کننده‌ی تُنکی ناهموار دارد. اثبات همگرایی این الگوریتم و نیز ارتباط آن با سایر الگوریتم‌ها، از جمله الگوریتم‌های ISP، نیز بحث شد. در مورد هر الگوریتم، شبیه‌سازی‌هایی ارائه شد که حاکی از عملکرد خوب الگوریتم‌های پیشنهادی در مقایسه با الگوریتم‌های معروف موجود داشتند.

فصل بعد به معرفی الگوریتم‌های پیشنهادی برای آموزش دیکشنری اختصاص دارد. در این فصل، الگوریتم‌هایی برای یادگیری دیکشنری‌های دارای ساختار ویژه و نیز الگوریتمی برای کاهش حجم محاسبات در مورد آموزش دیکشنری از روی داده‌های با ابعاد بالا ارائه خواهند شد. بعلاوه، الگوریتمی جدید برای طراحی



شکل ۵-۲۵: مقایسه‌ی دیداری تصاویر بازسازی شده توسط الگوریتم‌های مختلف با نرخ نمونه‌برداری $\delta = 0.4$. از راست به چپ، هر ستون متناظر است با: تصویر اصلی، تصویر بازسازی شده با الگوریتم SCSA-FIT، $IPP(w = 0)$ و $IPP(w = 0.85)$.

ماتریس‌های ناهمبسته، که در مخابرات و حسگری فشرده کاربرد دارند، معرفی می‌شود.

الگوریتم‌های پیشنهادی برای یادگیری دیکشنری و طراحی ماتریس حسگر

۱-۶ مقدمه

در این فصل الگوریتم‌های پیشنهادی برای آموزش دیکشنری ارائه می‌شوند. ابتدا، الگوریتمی برای پردازش داده‌های با ابعاد بالا معرفی می‌شود که مبتنی بر پردازش توزیعی داده‌های آموزشی و نیز استفاده از ایده‌ی sketching برای کاهش بُعد داده‌ها است. نتایج شبیه‌سازی نشان‌دهنده‌ی عملکرد امیدبخش الگوریتم پیشنهادی است. سپس، الگوریتم‌های کارآمد برای یادگیری دیکشنری‌های با همبستگی متقابل کم معرفی می‌شوند. این الگوریتم‌ها به دنبال آموزش دیکشنری‌های با همبستگی متقابل کم هستند، چرا که این نوع از دیکشنری‌ها خواص بازیابی خوبی دارند. برای این منظور، مسأله‌ی مربوطه به دو صورت مقید (یعنی وقتی یک باند بالا روی همبستگی متقابل مطلوب است) و نامقید (یعنی زمانی که همبستگی دیکشنری به فرم یک جریمه به تابع هدف کلی آموزش دیکشنری اضافه می‌شود)، فرمول‌بندی شده و سپس الگوریتم‌هایی کارآمد مبتنی بر روش‌های پراکسیمال ارائه می‌شوند. در ادامه، نتایج شبیه‌سازی برای ارزیابی عملکرد این الگوریتم‌ها و مقایسه‌ی آنها با روش‌های موجود گزارش می‌شود. همچنین، الگوریتمی برای طراحی ماتریس حسگر در کاربرد حسگری فشرده ارائه و بررسی خواهد شد.

۲-۶ یادگیری دیکشنری برای داده‌های بُعد بالا

همانطور که در قسمت ۳-۴ بحث شد، روز به روز بر حجم و ابعاد داده‌های تولید شده در کاربردهای مختلف افزوده می‌شود. پردازش این حجم عظیم از داده‌ها نیازمند راهکارهای کارآمدی است که به تعدادی از آنها در قسمت ۳-۴ اشاره شد. در این بخش، ما به طور خاص، مسأله‌ی یادگیری دیکشنری را در نظر گرفته، و الگوریتمی برای یادگیری دیکشنری از روی داده‌های با ابعاد بالا پیشنهاد می‌دهیم. با توجه به این نکته که هم تعداد داده‌های آموزشی و هم بُعد آنها می‌تواند زیاد باشد، در الگوریتم پیشنهادی هر دوی این موارد در نظر گرفته می‌شود. در واقع، برای غلبه بر بُعد بالای داده‌ها، از روش‌های sketching، و برای غلبه بر تعداد بالای داده‌ها، از روش‌های پردازش توزیعی داده‌ها، که در بخش ۳-۴ مرور شدند، استفاده می‌کنیم. لازم به ذکر است که اگرچه تاکنون روش‌های مختلفی برای یادگیری توزیع‌شده‌ی دیکشنری معرفی شده است [۷۹، ۳۳، ۳۹، ۴۰]، اما تفاوت عمده‌ی الگوریتم پیشنهادی با روش‌های موجود در این است که اولاً کاهش بُعد داده‌ها را نیز در نظر گرفته، و ثانیاً، برای کاهش حجم اطلاعات مبادله شده از یک ساختار تُنک برای دیکشنری استفاده می‌شود.

۱-۲-۶ الگوریتم پیشنهادی

برای یادآوری، مسأله‌ی کلی یادگیری دیکشنری را در نظر بگیرید

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{X} \in \mathcal{X}} f(\mathbf{D}, \mathbf{X}) = \frac{1}{\sqrt{L}} \|\mathbf{Y} - \mathbf{DX}\|_F = \frac{1}{\sqrt{L}} \sum_{i=1}^L \|y_i - \mathbf{D}x_i\|_2, \quad (1-6)$$

که در آن، \mathcal{D} و \mathcal{X} ، به ترتیب، مجموعه‌ی قید روی دیکشنری و ماتریس ضرایب را مشخص می‌کنند. عموماً، مجموعه‌ی \mathcal{D} بصورت ماتریس‌های با ستون‌های نُرم واحد تعریف شده، و مجموعه‌ی \mathcal{X} یک قید تُنکی اعمال می‌کند. همچنین، داریم $\mathbf{Y} \in \mathbb{R}^{n \times N}$ و $\mathbf{D} \in \mathbb{R}^{n \times M}$ که $M \gg n$ و $n \gg 1$. با توجه به این که تعداد داده‌های آموزشی ممکن است بسیار زیاد باشد، بگونه‌ای که یک پردازنده به تنهایی قادر به حل مسأله‌ی فوق نباشد، ما از روش‌های توزیع شده استفاده می‌کنیم، که در آن، داده‌های آموزشی روی تعدادی پردازنده تقسیم می‌شوند. اگر

تعداد پردازنده‌ها را با P نشان دهیم، مسأله‌ی (۱-۶) را می‌توان بصورت زیر بازنویسی کرد

$$\min_{\mathbf{D} \in \mathcal{D}, \{\mathbf{D}_i\}_{i=1}^P, \{\mathbf{X}_i \in \mathcal{X}\}_{i=1}^P} \frac{1}{\sqrt{L}} \sum_{i=1}^P \|\mathbf{Y}_i - \mathbf{D}_i \mathbf{X}_i\|_F \quad \text{s.t.} \quad \forall i: \mathbf{D}_i = \mathbf{D}, \quad (2-6)$$

که در آن، $\mathbf{Y}_i \in \mathbb{R}^{n \times M_i}$ ، در برگرفته‌ی بخشی از داده‌ها بوده و $\sum_{i=1}^P M_i = M$. همچنین، \mathbf{D}_i ، اصطلاحاً یک کپی از دیکشنری سراسری \mathbf{D} نامیده می‌شود، که توسط پردازنده i ام یادگرفته می‌شود [۲۲]. ماتریس \mathbf{X}_i نیز ضرایب

تُنک داده‌های \mathbf{Y}_i روی دیکشنری \mathbf{D}_i را نشان می‌دهد.

با توجه به این که در الگوریتم‌های توزیع شده، پردازنده‌ها مرتب با مرکز ترکیب (یا گره‌های همسایه‌ی خود در روش‌های غیرمتمرکز) تبادل اطلاعات دارند و دیکشنری‌های آموزش داده شده توسط خود را به این مرکز ارسال می‌کنند، این امر ممکن است بار انتقال داده را افزایش دهد. برای حل این مشکل، فرض می‌کنیم که هر دیکشنری عبارت است از حاصلضرب یک ماتریس پایه، مانند ماتریس DCT، در یک ماتریس تُنک. بعبارت دیگر، $\forall i: \mathbf{D}_i = \Psi \mathbf{A}_i$ که در اینجا، Ψ ماتریس DCT با ابعاد $n \times n$ بوده و $\mathbf{A}_i \in \mathbb{R}^{n \times M}$ ماتریسی با ستون‌های تُنک است. در واقع، با این مدل‌سازی، فرض می‌شود که هر اتم حاصل یک ترکیب خطی تُنک از ستون‌های دیکشنری پایه Ψ است. لازم به ذکر است که این ایده، یعنی مدل کردن دیکشنری بصورت حاصلضرب یک دیکشنری پایه و یک ماتریس تُنک، قبلاً برای حل مسأله‌ی عادی یادگیری دیکشنری معرفی شده است [۱۰۱]. به این ترتیب، پردازنده‌ها کافی است تنها ماتریس‌های \mathbf{A}_i را به مرکز ترکیب ارسال کنند، و از آنجایی که این ماتریس‌ها تُنک هستند، حجم انتقال داده کاهش می‌یابد.

همچنین، برای کم کردن پیچیدگی محاسباتی ناشی از بُعد بالای داده‌ها، از روش sketching، که در قسمت ۳-۴-۳ مرور شد، استفاده می‌کنیم. برای این منظور، ماتریس $\Phi \in \mathbb{R}^{m \times n}$ را، که $m < n$ ، بعنوان ماتریس sketching تعریف می‌کنیم. با اعمال همزمان ایده‌های پردازش توزیعی و کاهش بُعد، مسأله‌ی (۶-۲) به فرم زیر تبدیل می‌شود

$$\min_{\mathbf{A} \in \mathcal{A}, \{\mathbf{A}_i\}_{i=1}^P, \{\mathbf{X}_i \in \mathcal{X}\}_{i=1}^P} \frac{1}{P} \sum_{i=1}^P \|\Phi \mathbf{Y}_i - \Phi \Psi \mathbf{A}_i \mathbf{X}_i\|_F + \lambda \|\mathbf{A}\|_1 \quad \text{s.t. } \mathbf{A} = \mathbf{C}, \mathbf{A}_i = \mathbf{A}, i = 1, \dots, P, \quad (۳-۶)$$

که در آن، \mathbf{A} ماتریس نمایش تُنک مربوط به دیکشنری سراسری بوده، و جمله‌ی دوم در تابع هدف فوق، برای اعمال تُنکی این ماتریس اضافه شده است. پارامتر λ نیز میزان این تُنکی را تنظیم می‌کند. همچنین، مجموعه‌ی \mathcal{A} بصورت زیر تعریف می‌شود

$$\mathcal{A} = \{\mathbf{A} \in \mathbb{R}^{n \times M} \mid \|\mathbf{B} \mathbf{a}_i\|_2 = 1, i = 1, \dots, M\}, \quad (۴-۶)$$

که در آن، $\mathbf{B} \triangleq \Phi \Psi$. علاوه بر این‌ها، به دلیل وجود دو مجموعه قید روی ماتریس \mathbf{A} (یکی قید نُرم یک و دیگری قید $\mathbf{A} \in \mathcal{A}$)، یک متغیر کمکی \mathbf{C} نیز تعریف کرده‌ایم تا بتوان این دو مجموعه قید را از هم جدا کرد (در ادامه، دلیل این امر روشن‌تر خواهد شد). با تعریف $\tilde{\mathbf{Y}}_i \triangleq \Phi \mathbf{Y}_i$ ، نهایتاً فرمول‌بندی زیر را خواهیم داشت

$$\min_{\mathbf{C} \in \mathcal{A}, \{\mathbf{A}_i\}_{i=1}^P, \{\mathbf{X}_i \in \mathcal{X}\}_{i=1}^P} \frac{1}{P} \sum_{i=1}^P \|\tilde{\mathbf{Y}}_i - \mathbf{B} \mathbf{A}_i \mathbf{X}_i\|_F + \lambda \|\mathbf{A}\|_1 \quad \text{s.t. } \mathbf{A} = \mathbf{C}, \mathbf{A}_i = \mathbf{A}, i = 1, \dots, P. \quad (۵-۶)$$

دقت کنید که در مسأله‌ی فوق، قید $\mathbf{A} \in \mathcal{A}$ به قید معادل $\mathbf{C} \in \mathcal{A}$ تبدیل شده است. برای حل (۵-۶)، از الگوریتم ADMM استفاده می‌کنیم. برای این منظور، ابتدا ماتریس لاگرانژین به فرم زیر تشکیل می‌شود

$$L(\mathbf{A}, \mathbf{C}, \{\mathbf{A}_i\}_{i=1}^P, \{\mathbf{X}_i \in \mathcal{X}\}_{i=1}^P) = \frac{1}{\nu} \sum_{i=1}^P \left\{ \|\tilde{\mathbf{Y}}_i - \mathbf{B}\mathbf{A}_i\mathbf{X}_i\|_F^2 + \frac{\rho_1}{\nu} \|\mathbf{A}_i - \mathbf{A} + \mathbf{\Lambda}_i\|_F^2 \right\} + \lambda \|\mathbf{A}\|_1 + \frac{\rho_1}{\nu} \|\mathbf{A} - \mathbf{C} + \mathbf{\Lambda}\|_F^2 + \delta_{\mathcal{A}}(\mathbf{C}). \quad (۶-۶)$$

در رابطه‌ی فوق، $\rho_1, \rho_2 > 0$ پارامترهای جریمه بوده و $\mathbf{\Lambda}$ و $\mathbf{\Lambda}_i$ ماتریس‌های ضرایب لاگرانژ متناظر با \mathbf{A}_i و \mathbf{A} هستند. سپس، تابع لاگرانژین فوق بصورت پیاپی روی متغیرها کمینه می‌شود. تکرارهای ADMM برای این منظور عبارت خواهند بود از

$$\begin{cases} (\mathbf{A}_i^{k+1}, \mathbf{X}_i^{k+1}) = \operatorname{argmin}_{\mathbf{A}_i, \mathbf{X}_i \in \mathcal{X}} \frac{1}{\nu} \|\tilde{\mathbf{Y}}_i - \mathbf{B}\mathbf{A}_i\mathbf{X}_i\|_F^2 + \frac{\rho_1}{\nu} \|\mathbf{A}_i - \mathbf{A}^k + \mathbf{\Lambda}_i^k\|_F^2, & i = 1, \dots, P, \\ \mathbf{A}^{k+1} = \operatorname{argmin}_{\mathbf{A}} \frac{\rho_1}{\nu} \sum_{i=1}^P \|\mathbf{A} - \mathbf{A}_i^{k+1} - \mathbf{\Lambda}_i^k\|_F^2 + \frac{\rho_2}{\nu} \|\mathbf{A} - \mathbf{C}^k + \mathbf{\Lambda}^k\|_F^2 + \lambda \|\mathbf{A}\|_1, \\ \mathbf{C}^{k+1} = \operatorname{argmin}_{\mathbf{C}} \frac{\rho_2}{\nu} \|\mathbf{C} - \mathbf{A}^{k+1} - \mathbf{\Lambda}^k\|_F^2 + \delta_{\mathcal{A}}(\mathbf{C}), \\ \mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k + (\mathbf{A}^{k+1} - \mathbf{C}^{k+1}), \\ \forall i: \mathbf{\Lambda}_i^{k+1} = \mathbf{\Lambda}_i^k + (\mathbf{A}_i^{k+1} - \mathbf{A}^{k+1}), & i = 1, \dots, P. \end{cases} \quad (۷-۶)$$

دقت کنید که هر یک از مسائل به‌روزرسانی جفت ماتریس‌های $(\mathbf{A}_i, \mathbf{X}_i)$ را می‌توان بصورت مستقل از هم روی یک پردازنده، یا یک گره از شبکه، انجام داد. برای حل هر کدام از این مسائل، همان رهیافت بهینه‌سازی نوبتی استفاده می‌شود. بعبارت دیگر، با شروع از یک نقطه‌ی اولیه، بصورت پیاپی تابع هدف روی یکی از متغیرها، با ثابت گرفتن دیگری، حل می‌شود. برای به‌روزرسانی \mathbf{X}_i از هر الگوریتم بازایی تکنیکی می‌توان استفاده کرد که ما

در اینجا از الگوریتم OMP استفاده می‌کنیم. سپس، برای به‌روزرسانی \mathbf{A}_i مسأله‌ی زیر را خواهیم داشت

$$\mathbf{A}_i^{k+1} = \operatorname{argmin}_{\mathbf{A}_i} F(\mathbf{A}_i) \triangleq \frac{1}{\nu} \|\tilde{\mathbf{Y}}_i - \mathbf{B}\mathbf{A}_i\mathbf{X}_i^{k+1}\|_F^2 + \frac{\rho_1}{\nu} \|\mathbf{A}_i - \mathbf{A}^k + \mathbf{\Lambda}_i^k\|_F^2. \quad (۸-۶)$$

برای حل این مسأله، از الگوریتم گرادیان کاهشی استفاده می‌کنیم. برای این منظور، گرادیان تابع هدف بصورت زیر محاسبه می‌شود (برای جزئیات بیشتر، به [۹۹] مراجعه کنید)

$$\nabla F(\mathbf{A}_i) = \mathbf{B}^T (\mathbf{B}\mathbf{A}_i\mathbf{X}_i^{k+1} - \tilde{\mathbf{Y}}_i) (\mathbf{X}_i^{k+1})^T + \rho_1 (\mathbf{A}_i - \mathbf{A}^k + \mathbf{\Lambda}_i^k). \quad (۹-۶)$$

برای انتخاب طول گام، باید ثابت لیپشیتز گرادیان فوق را محاسبه کنیم. برای این کار، توجه داریم که ثابت لیپشیتز عبارت دوم در رابطه‌ی گرادیان برابر است با ρ_1 . برای محاسبه‌ی ثابت لیپشیتز عبارت اول، می‌نویسیم

$$\begin{aligned} \|\nabla F(\mathbf{A}_2) - \nabla F(\mathbf{A}_1)\|_F &= \|\mathbf{B}^T \mathbf{B}\mathbf{A}_2\mathbf{X}_i^{k+1} (\mathbf{X}_i^{k+1})^T - \mathbf{B}^T \mathbf{B}\mathbf{A}_1\mathbf{X}_i^{k+1} (\mathbf{X}_i^{k+1})^T\|_F = \\ &= \|\mathbf{B}^T \mathbf{B}(\mathbf{A}_2 - \mathbf{A}_1)\mathbf{X}_i^{k+1} (\mathbf{X}_i^{k+1})^T\|_F \leq \|\mathbf{B}^T \mathbf{B}\|_F \cdot \|\mathbf{X}_i^{k+1} (\mathbf{X}_i^{k+1})^T\|_F \cdot \|\mathbf{A}_2 - \mathbf{A}_1\|_F \quad (۱۰-۶) \end{aligned}$$

که در آن، از این نکته استفاده کرده‌ایم که برای هر دو ماتریس \mathbf{A} و \mathbf{B} با ابعاد سازگار، نامساوی زیر برقرار است [۹۱]

$$\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F \cdot \|\mathbf{B}\|_F. \quad (۱۱-۶)$$

بنابراین، می‌توان طول گام گرادیان کاهشی را بصورت زیر انتخاب کرد

$$\mu_A = \frac{1}{\|\mathbf{B}^T \mathbf{B}\|_F \cdot \|\mathbf{X}_i^{k+1} (\mathbf{X}_i^{k+1})^T\|_F}. \quad (۱۲-۶)$$

در نهایت، تکرارهای به‌روزرسانی ماتریس \mathbf{A}_i بصورت زیر خواهد بود

$$\mathbf{A}_i \leftarrow \mathbf{A}_i - \mu_A \nabla F(\mathbf{A}_i). \quad (۱۳-۶)$$

حال، مسأله‌ی به‌روزرسانی \mathbf{A} (سطر دوم از (۷-۶)) را در نظر بگیرید. بعد از محاسباتی سراسری می‌توان نشان داد که این مسأله معادل است با

$$\mathbf{A}^{k+1} = \underset{\mathbf{A}}{\operatorname{argmin}} \frac{\rho_1 P + \rho_2}{2} \|\mathbf{A} - \frac{\rho_1}{\rho_1 P + \rho_2} \sum_{i=1}^P (\mathbf{A}_i^{k+1} + \mathbf{\Lambda}_i^k) - \frac{\rho_2}{\rho_1 P + \rho_2} (\mathbf{C}^k - \mathbf{\Lambda}^k)\|_F^2 + \lambda \|\mathbf{A}\|_1. \quad (۱۴-۶)$$

با توجه به تعریف عملگر پراکسیمال (رابطه‌ی (۴-۱))، جواب مسأله‌ی فوق با پراکسیمال تابع نرم یک، که معادل با آستانه‌گذاری نرم است، داده می‌شود:

$$\mathbf{A}^{k+1} = \operatorname{prox}_{\frac{\lambda}{\rho_1 P + \rho_2} \cdot \|\cdot\|_1} \left(\frac{\rho_1}{\rho_1 P + \rho_2} \sum_{i=1}^P (\mathbf{A}_i^{k+1} + \mathbf{\Lambda}_i^k) + \frac{\rho_2}{\rho_1 P + \rho_2} (\mathbf{C}^k - \mathbf{\Lambda}^k) \right). \quad (۱۵-۶)$$

به‌روزرسانی متغیر \mathbf{C} (سطر سوم از (۷-۶)) نیز، طبق آنچه در فصل ۴ داشتیم، با استفاده از عملگر تصویر روی مجموعه‌ی \mathcal{A} محاسبه می‌شود. بعبارت دیگر،

$$\mathbf{C}^{k+1} = \mathcal{P}_{\mathcal{A}}(\mathbf{A}^{k+1} + \mathbf{\Lambda}^k), \quad (۱۶-۶)$$

که در آن، $\mathcal{P}_{\mathcal{A}}$ عملگر تصویر روی \mathcal{A} بوده و بصورت زیر تعریف می‌شود

$$\mathcal{P}_{\mathcal{A}}(\mathbf{A}) = \tilde{\mathbf{A}}, \quad \forall i: \tilde{\mathbf{a}}_i = \begin{cases} \mathbf{a}_i & \|\mathbf{B}\mathbf{a}_i\|_2 \leq 1 \\ \frac{\mathbf{a}_i}{\|\mathbf{B}\mathbf{a}_i\|_2} & \|\mathbf{B}\mathbf{a}_i\|_2 > 1 \end{cases} \quad (۱۷-۶)$$

الگوریتم نهایی، که آن را HighDim-DL^۱ می‌نامیم، در شکل ۶-۱ خلاصه شده است. در مورد همگرایی این الگوریتم باید گفت که از آنجایی که مسأله‌ی یادگیری دیکشنری غیرمحدب است، تحلیل همگرایی آن دشوار است. در واقع، اگرچه اثبات همگرایی الگوریتم ADMM برای مسائل محدب بطور گسترده‌ای بررسی شده^۲ [۲۲]، اما در مورد مسائل غیرمحدب کارهای کمی انجام شده است (بعنوان مثال، [۱۲۸] و مراجع داخل آن را

^۱High-dimensional dictionary learning

^۲اگر بخواهیم دقیق‌تر باشیم، باید گفت که اثبات همگرایی روش ADMM تنها برای مسائل با حداکثر دو بلوک متغیر تضمین شده است و در حالت کلی، در مورد مسائل با بیش از دو بلوک، تضمینی برای همگرایی این روش وجود ندارد [۳۲].

- هدف: حل مسأله‌ی یادگیری دیکشنری برای داده‌های با ابعاد بالا.
- ورودی: $P, \Phi, \Psi, Y, \{\mathbf{A}_i^0, \mathbf{X}_i^0, \Lambda_i^0\}_{i=1}^P, \Lambda^0, (\rho_1, \rho_2)$.
- شروع الگوریتم: گام‌های زیر را برای $k = 1, 2, \dots$ انجام بده.
 ۱. مسائل یادگیری دیکشنری بیان شده در سطر اول از (۶-۷) را روی پردازنده‌های شماره‌ی یک تا P حل کرده، سپس تخمین‌های بدست آمده برای \mathbf{A}_i ها را به مرکز ترکیب ارسال کن. برای حل این مسائل، از الگوریتم OMP و رابطه‌ی (۶-۱۳) استفاده کن.
 ۲. با تخمین‌های دریافتی از پردازنده‌ها، ماتریس‌های \mathbf{A}, \mathbf{C} و Λ را به ترتیب، با استفاده از روابط (۶-۱۵)، (۶-۱۶) و سطر سوم از (۶-۷) به‌روز کن. سپس، تخمین بدست آمده برای \mathbf{A} را به تمامی پردازنده‌ها ارسال کن.
 ۳. روی هر یک از پردازنده‌ها، ماتریس‌های Λ_i را با استفاده از سطر پنجم رابطه‌ی (۶-۷) به‌روز کن.
- خروجی: \mathbf{A}

شکل ۶-۱: الگوریتم پیشنهادی HighDim-DL برای حل مسأله‌ی یادگیری دیکشنری.

بینید). با این حال، عدم وجود یک اثبات جامع برای حالت غیرمحدب باعث نشده است تا الگوریتم ADMM در کاربردهای با مسائل غیرمحدب استفاده نشود؛ بلکه برعکس، این الگوریتم در عمل نتایج بسیار مطلوبی ارائه داده است [۳۰، ۷۷، ۱۳۵]. شاید بتوان مراجع [۸۰، ۱۲۸] را از معدود تلاش‌ها برای اثبات همگرایی الگوریتم ADMM برای حالت غیرمحدب دانست. با این وجود، بررسی همگرایی الگوریتم پیشنهادی با استفاده از نتایج ارائه شده در [۸۰، ۱۲۸] را بعنوان کارهای آینده در نظر خواهیم گرفت.

۶-۲-۲ نتایج شبیه‌سازی

در این قسمت، عملکرد الگوریتم پیشنهادی بررسی می‌شود. برای این منظور، داده‌های آموزشی، به تعداد $N = 4000$ ، را بلوک‌های 8×8 از تصاویر طبیعی در نظر می‌گیریم که بصورت تصادفی از چند تصویر انتخاب می‌شوند. همچنین، تعداد پردازنده‌ها را برابر با $P = 4$ می‌گیریم. یعنی، هر پردازنده تعداد ۱۰۰۰ داده‌ی آموزشی را پردازش می‌کند. دیکشنری پایه، یعنی Ψ ، یک ماتریس DCT با ابعاد 64×64 انتخاب می‌شود. بعلاوه، ماتریس sketching، یعنی Φ ، نیز با ابعاد $64 \times m$ که m عددی متغیر است، یک‌بار بصورت یک ماتریس تصادفی با توزیع نرمال و بار دیگر بصورت یک تبدیل تصادفی هادامارد زیرنمونه‌برداری شده^۱ (SRHT) [۱۲۹] در نظر گرفته شده است. برای الگوریتم OMP، حداکثر تعداد اتم‌های انتخاب شده برای نمایش t تک داده‌ها را ۵ می‌گیریم.

^۱Subsampled Randomized Hadamard Transform

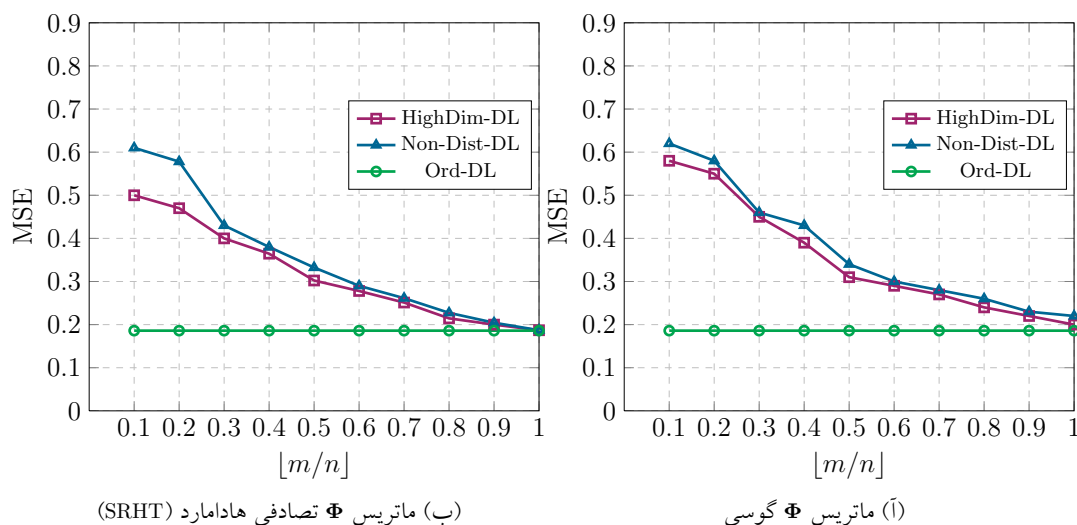
الگوریتم‌های مورد مقایسه عبارت هستند از الگوریتم پیشنهادی، HighDim-DL، الگوریتم غیرتوزیع‌شده‌ی Non-Dist-DL که همان الگوریتم HighDim-DL بوده اما تنها روی یک پردازنده اجرا می‌شود و الگوریتم Ord-DL که همان الگوریتم Non-Dist-DL اما بدون اعمال ماتریس sketching است. برای ارزیابی عملکرد الگوریتم‌های مورد مقایسه، میزان خطای (MSE) نمایش تُنک داده‌ها توسط دیکشنری‌های یادگرفته شده توسط هر یک از این الگوریتم‌ها را معیار قرار می‌دهیم. این معیار بصورت $\|Y - DX\|_F / (n \cdot M)$ تعریف می‌شود، که در آن، دیکشنری D عبارت است از $D = \Psi A$ ، که A ماتریس یادگرفته شده توسط هر یک از الگوریتم‌ها است. تعداد تکرارهای یادگیری دیکشنری‌های محلی (یعنی گام ۱ از شکل ۶-۱) برای الگوریتم HighDim-DL برابر با ۳۰، و برای دو الگوریتم دیگر برابر با ۳۰۰ انتخاب شده است. همچنین، قرار داده‌ایم، $\rho_1 = 10^4$ ، $\rho_2 = 10^3$ و $\lambda = 1000$. تعداد تکرارهای کلی الگوریتم HighDim-DL نیز برابر با ۵۰ در نظر گرفته شده است.

میزان خطای نمایش داده‌ها برای سه الگوریتم مورد مقایسه، برحسب مقادیر مختلف $[m/n]$ ، و برای هر دو حالت ماتریس sketching برابر با ماتریس گوسی و ماتریس SRHT، در شکل ۶-۲ نشان داده شده است. با دقت در این شکل مشخص می‌شود که الگوریتم HighDim-DL عملکرد بهتری نسبت به الگوریتم Non-Dist-DL دارد. همچنین، همانطور که انتظار می‌رفت، اعمال sketching باعث افت عملکرد می‌شود. با این حال، استفاده از ماتریس SRHT، همانطور که در شکل دیده می‌شود، منجر به عملکرد بهتری نسبت به ماتریس گوسی می‌شود. مزیت دیگر ماتریس‌های SRHT، حجم محاسبات کمتر ضرب ماتریسی است [۱۲۹]. لازم به ذکر است که برای این آزمایش، ماتریس A یادگرفته شده توسط الگوریتم‌های مورد مقایسه، درجه تُنکی بالایی دارد (ضریب Gini برای آن حدود ۰/۸۵ است).

۳-۶ یادگیری دیکشنری‌های با همبستگی متقابل کم

همانطور که در فصل ۱ دیدیم، به منظور تضمین عملکرد موفقیت‌آمیز الگوریتم‌های بازیابی نمایش تُنک، دیکشنری موجود باید دارای ویژگی‌های مشخصی باشد. در واقع، نتایج پژوهش‌های متعدد در زمینه‌ی نمایش تُنک نشان داده است که یکتایی و پایداری نمایش تُنک مستقیماً به خواص دیکشنری مربوط است [۵۲]. همبستگی متقابل^۱

^۱Mutual coherence



شکل ۶-۲: مقادیر نهایی خطای نمایش برای الگوریتم‌های مختلف و دو حالت ماتریس sketching.

(MC) و ویژگی همسانی محدود شده^۱ (RIP)، همانطور که در فصل ۲ اشاره شد، دو ویژگی اساسی در مورد یک دیکشنری هستند که کیفیت آن را ارزیابی می‌کنند [۵۲]. برای یادآوری، فرمول همبستگی متقابل برای یک

دیکشنری \mathbf{D} با ستون‌های دارای نرم واحد با عبارت زیر داده می‌شود

$$\mu(\mathbf{D}) \triangleq \max_{i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|. \quad (18-6)$$

همبستگی متقابل، میزان شباهت دوجه‌دوی ستون‌های یک دیکشنری را می‌سنجد. برای یک دیکشنری $n \times N$ نشان

داده شده است که همبستگی متقابل از پائین بصورت زیر محدود است

$$\mu \geq \sqrt{\frac{N-n}{n(N-1)}}, \quad (19-6)$$

که به باند ولج^۲ [۱۱۴] معروف است.

تعریف RIP را در فصل ۲ دیدیم. بسیاری از باندهای عملکرد الگوریتم‌های بازیابی نمایش تنگ مبتنی بر این کمیت هستند. اما ارزیابی RIP برای یک دیکشنری در زمان چندجمله‌ای ممکن نیست و در واقع یک مسأله‌ی NP-hard است [۱۱۷]. بنابراین، پارامتر MC به دلیل این که محاسبه‌ی آن ساده است توجه بیشتری برای طراحی دیکشنری به خود جلب کرده است. علاوه بر این، دو پارامتر MC و δ_s ، یعنی پارامتر RIP برای سطح تنگی s ، که مقادیر کوچک آن مطلوب هستند، به نحوی به هم مربوط هستند. به عنوان مثال، $\delta_s \leq \mu(s-1)$ [۶۰]. همچنین، داریم $\delta_2 = \mu$ [۵۲].

^۱Restricted isometry property

^۲Welch bound

لازم به ذکر است که علاوه بر بازیابی نمایش تُنک، پژوهش‌های اخیر نشان داده‌اند که دو پارامتر MC و RIP نقش مهمی در عملکرد کلی الگوریتم‌های یادگیری دیکشنری دارند. برای مثال، نشان داده شده است که اگر دیکشنری زیربنایی، که در واقع سازنده‌ی داده‌های آموزشی است، به اندازه‌ی کافی پارامتر MC آن کم باشد، در این صورت نقطه‌ی بهینه‌ی محلی مسأله‌ی استاندارد یادگیری دیکشنری خواهد بود [۶۷، ۱۳۰]. در این وضعیت گفته می‌شود که دیکشنری بصورت محلی قابل شناسایی^۱ است. همچنین، نشان داده شده است که یک دیکشنری کامل بصورت محلی قابل شناسایی خواهد بود اگر میزان تُنکی s از مرتبه‌ی $O(\mu^{-2})$ باشد، که μ مقدار همبستگی متقابل دیکشنری است [۱۳۰]. همچنین، پارامتر RIP به عنوان یک شرط تعیین کننده در همگرایی الگوریتم‌های یادگیری دیکشنری مبتنی بر بهینه‌سازی نوبتی استفاده شده است [۲].

اهمیت کمیت همبستگی متقابل برای یک دیکشنری، همانطور که در بالا توضیح داده شد، باعث شده است تا به عنوان یک معیار کیفیت در الگوریتم‌های یادگیری دیکشنری به کار گرفته شود. الگوریتم‌های موجود در این ارتباط به دو دسته‌ی کلی تقسیم می‌شوند. دسته‌ی اول، الگوریتم‌های مقید هستند که حد بالایی به عنوان مقدار مطلوب همبستگی متقابل دیکشنری در مسأله‌ی یادگیری دیکشنری بصورت یک قید در نظر می‌گیرند. این مسأله در حالت کلی به فرم زیر تعریف می‌شود

$$\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{\sqrt{p}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \mu(\mathbf{D}) \leq \mu_0, \quad (20-6)$$

که $\mu_0 > 0$ یک مقدار ثابت و مطلوب است. از الگوریتم‌های معروف برای حل این مسأله می‌توان به الگوریتم Incoherent K-SVD (INK-SVD) [۸۲] و الگوریتم IPR-DL [۱۴] اشاره کرد. الگوریتم INK-SVD مسأله‌ی (20-6) را اینگونه حل می‌کند که ابتدا حداقل‌کننده‌ی مسأله را بدون توجه به قید مسأله بدست آورده و سپس نتیجه‌ی حاصل را برای رسیدن به حد مطلوب همبستگی متقابل پردازش می‌کند. در واقع، اگر نتیجه‌ی حاصل از

$$\text{حل مسأله نامقید را با } \bar{\mathbf{D}} \text{ نشان دهیم، این الگوریتم مسأله‌ی زیر را حل می‌کند}$$

$$\min_{\mathbf{D}} \frac{1}{\sqrt{p}} \|\mathbf{D} - \bar{\mathbf{D}}\|_F^2 \quad \text{s.t.} \quad \mu(\mathbf{D}) \leq \mu_0. \quad (21-6)$$

برای رسیدن به حد مطلوب همبستگی متقابل، الگوریتم INK-SVD بصورت تکراری، جفت اتم‌هایی را که همبستگی آنها بیشتر از μ_0 شناسایی کرده و اصطلاحاً آنها را ناهمبسته می‌کند. یک ایراد مهم که به این الگوریتم می‌توان گرفت این است که حین فرایند ناهمبسته‌سازی، خطای نمایش داده‌ها، یعنی $\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F$ ، در نظر گرفته نمی‌شود

^۱Locally identifiable^۲Iterative projection-rotation dictionary learning

[۱۴]. برای حل این مشکل، در [۱۴] یک الگوریتم جدید ارائه شده است که شامل دو گام است. در گام اول، که گام ناهمبسته‌سازی نامیده می‌شود، یک عملیات تصویرسازی بصورت تکراری انجام می‌شود تا تضمین کند که همبستگی متقابل \bar{D} کمتر از حد مطلوب $\mu_0 > 0$ است. در گام دوم، که گام چرخش دیکشنری^۱ نامیده می‌شود، نتیجه‌ی حاصل از گام اول به نحوی «چرخانده» می‌شود که خطای نمایش داده‌ها حداقل شود. واضح است که چرخاندن دیکشنری، یعنی ضرب آن در یک ماتریس متعامد یکه، مقدار همبستگی متقابل را تغییر نمی‌دهد اما خطای نمایش را می‌تواند تا حد چشمگیری کاهش دهد.

گروه دوم الگوریتم‌های یادگیری دیکشنری با همبستگی متقابل پائین، مسأله‌ی نامقید زیر را حل می‌کنند

$$[۱۱، ۱۲، ۱، ۱۱۱]$$

$$\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{\gamma} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \lambda \mathcal{R}(\mathbf{D}), \quad (۲۲-۶)$$

که $\lambda \geq 0$ و

$$\mathcal{R}(\mathbf{D}) \triangleq \|\mathbf{D}^T \mathbf{D} - \mathbf{I}\|_F^2. \quad (۲۳-۶)$$

استفاده از \mathcal{R} بعنوان تابعی برای کم کردن همبستگی متقابل با رابطه‌ی زیر توجیه می‌شود

$$\mathcal{R}(\mathbf{D}) = \sum_{i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|^2 + \sum_i (\langle \mathbf{d}_i, \mathbf{d}_i \rangle - 1)^2. \quad (۲۴-۶)$$

در عبارت فوق، جمله‌ی اول در سمت راست مسئول کمینه‌سازی مربع همبستگی متقابل دوبه‌دوی اتم‌ها است و جمله‌ی دوم قید یک بودن نرم اتم‌ها را اعمال می‌کند. برای حل مسأله‌ی (۲۲-۶) الگوریتم‌های متنوعی ارائه شده‌اند. برای جزئیات این روش‌ها به مراجع نمونه که در بالا اشاره شد مراجعه شود.

۴-۶ الگوریتم‌های پیشنهادی

در این قسمت، الگوریتم‌هایی جدید و کارآ برای یادگیری دیکشنری‌های با همبستگی متقابل کم، هم برای حالت مقید (۲۰-۶) و هم حالت نامقید (۲۲-۶)، ارائه می‌شوند. در واقع، انگیزه‌ی اصلی برای ارائه‌ی این الگوریتم‌ها این است که روش‌های قبلی قادر نیستند به نحوی بهینه قید کم بودن همبستگی متقابل دیکشنری و بصورت همزمان داشتن خطای نمایش کم را برآورده کنند. این مشکل یا به دلیل ناکارآ بودن الگوریتم‌های موجود در حل مسائل

^۱Dictionary rotation

مربوطه است، مانند روش پیشنهاد شده در [۱۴] برای حل (۶-۲۰)، و یا اساساً به دلیل استفاده از معیار نه چندان کارآ برای همبستگی متقابل، مانند معیار تعریف شده در (۶-۲۴).

برای توضیح بیشتر، تابع \mathcal{R} تعریف شده در (۶-۲۴) را در نظر بگیرید. این تابع، طبق تعریف، تنها متوسط مربعات همبستگی‌های متقابل بین اتم‌ها را به حساب آورده و در نتیجه، از تعریف واقعی همبستگی متقابل، داده شده در (۶-۱۸)، که ماکسیمم اندازه‌ی همبستگی‌های متقابل است، فاصله دارد. در واقع، اگر جمله‌ی خطای نمایش

را در تابع هدف (۶-۲۲) در نظر نگیریم و بخواهیم صرفاً تابع \mathcal{R} را کمینه کنیم، در اینصورت خواهیم داشت

$$\min_{\mathbf{D} \in \mathcal{D}} \left\{ \mathcal{R}(\mathbf{D}) \triangleq \sum_{i,j:i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|^2 \right\}. \quad (۶-۲۵)$$

از طرفی، مسأله‌ی کمینه‌سازی همبستگی متقابل به فرم زیر است

$$\min_{\mathbf{D} \in \mathcal{D}} \left\{ \mu(\mathbf{D}) \triangleq \max_{i,j:i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle| \right\}. \quad (۶-۲۶)$$

حال با بررسی دو مسأله‌ی فوق مشخص است که مسأله‌ی (۶-۲۵) تنها تضمین می‌کند که متوسط $|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|^2$ (به توان دو توجه کنید) کمینه می‌شود؛ نه تک تک جملات $|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|$. مسأله‌ی (۶-۲۶)، از دیگر سو، تک تک جملات $|\langle \mathbf{d}_i, \mathbf{d}_j \rangle|$ را کمینه کرده که منجر به کمینه‌سازی همبستگی متقابل دیکشنری می‌شود. به بیان دیگر، اگر درایه‌های $\{\langle \mathbf{d}_i, \mathbf{d}_j \rangle\}_{i \neq j}$ را در بردار \mathbf{g} قرار دهیم، در این صورت داریم $\mathcal{R}(\mathbf{D}) = \|\mathbf{g}\|_2^2$ و $\mu(\mathbf{D}) = \|\mathbf{g}\|_\infty$. این موضوع به وضوح مشخص می‌سازد که کمینه‌سازی این دو تابع به جواب‌های به کلی متفاوتی منجر می‌شود؛ اولی به جواب حداقل نرْم l_2 ، و دومی به جواب حداقل نرْم l_∞ .

از طرف دیگر، الگوریتم IPR-DL برای حل (۶-۲۰)، این مسأله را بصورت مستقیم حل نمی‌کند و در نتیجه زیربهبینه است. علاوه بر این، نیازمند انجام عملیات SVD و EVD^۱ در هر تکرار است که حجم محاسبات بالایی را، به خصوص در مورد داده‌های با ابعاد بالا، می‌طلبد. راه حل ما برای این مشکل ارائه‌ی الگوریتمی است که بصورت مستقیم مسأله‌ی (۶-۲۰) را حل می‌کند و به علاوه، هیچگونه عملیات SVD و EVD در ساختار آن وجود ندارد [۱۰۶، ۱۰۹]. بنابراین، گزینه‌ی بهتری برای داده‌های با ابعاد بالا است. تا آنجا که ما اطلاع داریم، الگوریتم‌های پیشنهادی اولین روش‌هایی هستند که بصورت مستقیم تابع همبستگی متقابل را وارد مسأله‌ی یادگیری دیکشنری می‌کنند. در ادامه، این الگوریتم‌ها را با جزئیات بیشتر توضیح می‌دهیم.

^۱Eigenvalue decomposition

۱-۴-۶ حالت نامقید

در این قسمت، الگوریتم پیشنهادی برای حل مسأله‌ی نامقید یادگیری دیکشنری‌های با همبستگی متقابل کم ارائه می‌شود. برای این منظور، مسأله‌ی زیر را پیشنهاد می‌دهیم

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{X} \in \mathcal{X}} \frac{1}{\alpha} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \lambda \mu(\mathbf{D}). \quad (27-6)$$

در مسأله‌ی فوق، برخلاف مسأله‌ی (۲۲-۶)، از تابع همبستگی متقابل به عنوان تابع تشویق‌کننده‌ی همبستگی متقابل پائین استفاده شده است. برای حل این مسأله، ابتدا دقت داریم که تابع همبستگی متقابل را برای دیکشنری‌های با ستون‌های نرم واحد می‌توان بصورت معادل زیر نوشت

$$\mu(\mathbf{D}) = \|\mathbf{D}^T \mathbf{D} - \mathbf{I}\|_\infty, \quad (28-6)$$

که در آن، $\|\mathbf{X}\|_\infty \triangleq \max_{i,j} |x_{ij}|$. بنابراین، مسأله‌ی (۲۷-۶) را می‌توان بصورت زیر بازنویسی کرد

$$\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{\alpha} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \lambda \|\mathbf{D}^T \mathbf{D} - \mathbf{I}\|_\infty. \quad (29-6)$$

با در نظر گرفتن (۲۸-۶) و (۲۳-۶) می‌توان نتیجه گرفت که تابع \mathcal{R} ، که در کارهای قبلی استفاده می‌شده است، تنها تقریبی از تابع همبستگی متقابل است که در آن، با توجه به ناهموار بودن و در نتیجه دشوار بودن کمینه‌سازی نرم ℓ_∞ ، این نرم با نرم فروبینیوس جایگزین شده است.

مشابه (۲۲-۶)، مسأله‌ی (۲۹-۶) نیز غیرمحدب است. با این حال، به دلیل ناهموار بودن تابع ℓ_∞ ، حل (۲۹-۶) دشوارتر است. برای حل این مشکل، یک متغیر جدید $\mathbf{G} \triangleq \mathbf{D}^T \mathbf{D}$ تعریف می‌کنیم. با این کار، مسأله‌ی (۲۹-۶) به فرم زیر بازنویسی می‌شود

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{G}} \frac{1}{\alpha} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \lambda \|\mathbf{G} - \mathbf{I}\|_\infty \quad \text{s.t.} \quad \mathbf{G} = \mathbf{D}^T \mathbf{D}. \quad (30-6)$$

به این ترتیب، جمله‌ی $\mathbf{D}^T \mathbf{D}$ از درون تابع ناهموار ℓ_∞ بیرون آورده می‌شود و راه برای اعمال روش‌های پراکسیمال باز می‌شود. برای حل مسأله‌ی فوق، از روش جریمه کمک می‌گیریم. برای این منظور، مسأله‌ی زیر را باید حل

کنیم

$$\min_{\mathbf{D}, \mathbf{G}} \left\{ H(\mathbf{D}, \mathbf{G}) \triangleq F(\mathbf{D}, \mathbf{G}) + r_d(\mathbf{D}) + r_g(\mathbf{G}) \right\}, \quad (31-6)$$

که در آن

$$F(\mathbf{D}, \mathbf{G}) \triangleq \frac{1}{\alpha} \|\mathbf{Y} - \mathbf{DX}\|_F^2 + \frac{1}{\alpha} \|\mathbf{G} - \mathbf{D}^T \mathbf{D}\|_F. \quad (32-6)$$

همچنین، $r_d \triangleq \delta_{\mathcal{D}}$ تابع مشخصه‌ی مجموعه‌ی $\mathcal{D} \triangleq \{\mathbf{D} \in \mathbb{R}^{n \times N} \mid \forall i, \|\mathbf{d}_i\|_2 = 1\}$ بوده، $r_g(\mathbf{G}) \triangleq \lambda \|\mathbf{G} - \mathbf{I}\|_{\infty}$ و در نهایت، $\alpha > 0$ ضریب جریمه است. برای $\alpha \rightarrow 0$ ، جریمه‌ی تخطی از قید تساوی در (۳۰-۶) بیشتر و بیشتر شده و مجموعه جواب‌های (۳۱-۶) به جواب‌های (۳۰-۶) میل می‌کنند. در نتیجه، مطابق با روش استاندارد جریمه [۹۴]، ما از یک دنباله‌ی کاهش‌ی $\{\alpha_k\}$ استفاده کرده، جواب (۳۱-۶) را برای هر k بدست می‌آوریم. جواب‌های بدست آمده، که متناظر با α_k هستند، بعنوان نقاط شروع برای حل مسأله‌ی متناظر با α_{k+1} استفاده می‌شوند.

به منظور حل (۳۱-۶) به ازای یک مقدار ثابت α_k ، از بهینه‌سازی نوبتی استفاده می‌شود که در آن، بصورت تکراری تابع هدف روی هر یک از متغیرها و با ثابت گرفتن دیگری کمینه می‌شود. در ادامه، مسائل به‌روزرسانی \mathbf{G} و \mathbf{D} را با جزئیات بررسی می‌کنیم.

به‌روزرسانی \mathbf{G}

بعد از محاسباتی ساده، مسأله‌ی به‌روزرسانی \mathbf{G} تبدیل می‌شود به

$$\mathbf{G}_{k+1} = \mathbf{I} + \operatorname{argmin}_{\mathbf{G}} \left\{ \frac{1}{\gamma} \|\mathbf{G} - \mathbf{M}_k\|_F^2 + \eta_k \|\mathbf{G}\|_{\infty} \right\}, \quad (۳۳-۶)$$

که $\mathbf{M}_k \triangleq \mathbf{D}_k^T \mathbf{D}_k - \mathbf{I}$ و $\eta_k \triangleq \lambda \cdot \alpha_k$ جمله‌ی دوم در سمت راست عبارت (۳۳-۶) طبق تعریف، عملگر پراکسیمال تابع ℓ_{∞} است که طبق لم زیر بدست می‌آید.

لم ۱-۶ فرض کنید g تابع $\mathbb{R} \rightarrow \mathbb{R}^{N \times N}$ را نشان دهد. عملگر پراکسیمال g عبارت است از

$$\operatorname{prox}_g(\mathbf{U}) = \mathbf{U} - P_{\mathcal{B}^{\eta}}(\mathbf{U}), \quad (۳۴-۶)$$

که $P_{\mathcal{B}^{\eta}}(\cdot) : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{N \times N}$ عملگر تصویر روی گوی ℓ_1 به شعاع η است.

اثبات لم فوق با استفاده از تجزیه‌ی مورو، معادله‌ی (۴-۸)، و استفاده از خواص ابتدایی عملگر پراکسیمال قابل انجام است. بنابراین، مسأله‌ی به‌روزرسانی \mathbf{G} عبارت خواهد بود از

$$\mathbf{G}_{k+1} = \mathbf{I} + \mathbf{M}_k - P_{\mathcal{B}^{\eta_k}}(\mathbf{M}_k) \quad (۳۵-۶)$$

یا بصورت معادل،

$$\mathbf{G}_{k+1} = \mathbf{D}_k^T \mathbf{D}_k - P_{\mathcal{B}^{\eta_k}}(\mathbf{D}_k^T \mathbf{D}_k - \mathbf{I}). \quad (۳۶-۶)$$

محاسبه‌ی عملگر پراکسیمال در (۳۶-۶) نیازمند تصویر روی توپ ℓ_1 است که برای آن الگوریتم‌های بهینه‌ای وجود دارد؛ [۴۸] را ببینید.

به‌روزرسانی \mathbf{D}

مسئله‌ی به‌روزرسانی \mathbf{D} بصورت زیر است

$$\min_{\mathbf{D} \in \mathcal{D}} \{f(\mathbf{D}) + r_d(\mathbf{D})\}, \quad (۳۷-۶)$$

که در آن، $f(\mathbf{D}) \triangleq F(\mathbf{D}, \mathbf{G}_{k+1})$. این مسئله شباهت زیادی به مسئله‌ی (۲۲-۶) دارد و بنابراین، در حالت کلی جواب فرم بسته‌ای ندارد. با این حال، ما برای حل این مسئله از الگوریتم‌های پراکسیمال کمک می‌گیریم. برای این منظور، تابع f را با تقریب درجه‌ی دوم آن حول تخمین قبلی دیکشنری جایگزین کرده و حداقل‌کننده‌ی آن را به عنوان تخمین جدید دیکشنری در نظر می‌گیریم. به عبارت دقیق‌تر، داریم

$$\mathbf{D}^{k+1} = \operatorname{argmin}_{\mathbf{D}} \left\{ f(\mathbf{D}_k) + \nabla^T f(\mathbf{D}_k)(\mathbf{D} - \mathbf{D}_k) + \frac{1}{2\mu_d} \|\mathbf{D} - \mathbf{D}_k\|_F^2 + r_d(\mathbf{D}) \right\}, \quad (۳۸-۶)$$

که در آن

$$\nabla f(\mathbf{D}) = (\mathbf{D}\mathbf{X} - \mathbf{Y})\mathbf{X}^T + \frac{2}{\alpha_k} \mathbf{D}(\mathbf{D}^T \mathbf{D} - \mathbf{G}_{k+1}) \quad (۳۹-۶)$$

گرادیان f بوده و $\mu_d > 0$. مسئله‌ی (۳۸-۶) را می‌توان بصورت معادل زیر نوشت

$$\mathbf{D}^{k+1} = \operatorname{argmin}_{\mathbf{D}} \left\{ \frac{1}{2} \|\mathbf{D} - \bar{\mathbf{D}}_k\|_F^2 + r_d(\mathbf{D}) \right\} = P_{\mathcal{D}}(\bar{\mathbf{D}}_k), \quad (۴۰-۶)$$

که $\bar{\mathbf{D}}_k \triangleq \mathbf{D}_k - \mu_d \nabla f(\mathbf{D}_k)$. به طور خلاصه، با این روش، دیکشنری بصورت تکراری با یک گام گرادیان کاهش‌ی و سپس با تصویر کردن روی \mathcal{D} به‌روزرسانی می‌شود. پارامتر طول گام μ_d با توجه به ثابت لپشیتز ∇f ، که آن را با L نشان می‌دهیم، تعیین می‌شود. ابتدا نشان می‌دهیم که تابع ∇f روی \mathcal{D} لپشیتز است. برای این منظور، تعریف می‌کنیم

$$f_1(\mathbf{D}) \triangleq \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2, \quad f_2(\mathbf{D}) \triangleq \frac{1}{2} \|\mathbf{D}^T \mathbf{D} - \mathbf{G}_{k+1}\|_F^2. \quad (۴۱-۶)$$

بنابراین، $\nabla f(\mathbf{D}) = \nabla f_1(\mathbf{D}) + \frac{1}{\alpha_k} \nabla f_2(\mathbf{D})$. می‌دانیم تابع ∇f_1 با ثابت $L_1 = \|\mathbf{X}^T \mathbf{X}\|$ گرادیان-لپشیتز است

[۱۳۴]، که $\|\cdot\|$ ماکسیمم مقدار تکین یک ماتریس را نشان می‌دهد. برای تابع ∇f_2 لم زیر را داریم:

لم ۲-۶ گرادیان تابع f_2 که در (۴۱-۶) تعریف شده است، روی مجموعه‌ی \mathcal{D} لپشیتز است. یعنی، عددی مانند

$L_2 > 0$ وجود دارد که برای همه‌ی $\mathbf{D}_1, \mathbf{D}_2 \in \mathcal{D}$

$$\|\nabla f_2(\mathbf{D}_2) - \nabla f_2(\mathbf{D}_1)\|_F \leq L_2 \|\mathbf{D}_2 - \mathbf{D}_1\|_F. \quad (۴۲-۶)$$

از رابطه‌ی (۳۹-۶) برای گرادیان تابع f_2 داریم

$$\nabla f_2(\mathbf{D}) = 2\mathbf{D}(\mathbf{D}^T \mathbf{D} - \mathbf{G}_{k+1}). \quad (۴۳-۶)$$

با استفاده از اطلاعات فوق، به راحتی می‌توان نشان داد که

$$\begin{aligned} \nabla f_{\gamma}(\mathbf{D}_1) - \nabla f_{\gamma}(\mathbf{D}_2) &= \gamma \left(\mathbf{D}_1 \mathbf{D}_1^T \mathbf{D}_1 - \mathbf{D}_1 \mathbf{G}_{k+1} - \mathbf{D}_2 \mathbf{D}_2^T \mathbf{D}_2 + \mathbf{D}_2 \mathbf{G}_{k+1} \right) = \\ &= \gamma \left((\mathbf{D}_1 - \mathbf{D}_2) \mathbf{D}_1^T \mathbf{D}_1 + \mathbf{D}_2 (\mathbf{D}_1 - \mathbf{D}_2)^T \mathbf{D}_1 + \mathbf{D}_2 \mathbf{D}_2^T (\mathbf{D}_1 - \mathbf{D}_2) - (\mathbf{D}_1 - \mathbf{D}_2) \mathbf{G}_{k+1} \right). \end{aligned} \quad (44-6)$$

سپس، با بکارگیری نامساوی مثلث برای نُرم به همراه ویژگی (۱۱-۶) خواهیم داشت

$$\|\nabla f_{\gamma}(\mathbf{D}_1) - \nabla f_{\gamma}(\mathbf{D}_2)\|_F \leq \gamma \left(\|\mathbf{D}_1\|_F^2 + \|\mathbf{D}_1\|_F \|\mathbf{D}_2\|_F + \|\mathbf{D}_2\|_F^2 + \|\mathbf{G}_{k+1}\|_F \right) \cdot \|\mathbf{D}_1 - \mathbf{D}_2\|_F \quad (45-6)$$

از طرف دیگر، به راحتی می‌توان نشان داد که

$$\forall \mathbf{D} \in \mathcal{D}: \|\mathbf{D}\|_F = \sqrt{N}. \quad (46-6)$$

در نهایت، با توجه به (۴۵-۶) و (۴۶-۶)، نامساوی رابطه‌ی (۴۲-۶) نتیجه می‌دهد که

$$L_{\gamma} = \epsilon N + \gamma \|\mathbf{G}_{k+1}\|_F. \quad (47-6)$$

الگوریتم نهایی برای حل (۳۱-۶) عبارت است از تکرار بین (۳۶-۶) و (۴۰-۶). قضیه‌ی زیر، همگرایی دنباله‌ی تولید شده را بیان می‌کند.

قضیه ۶-۱ فرض کنید $\{\mathbf{D}_k, \mathbf{G}_k\}$ دنباله‌ی تولید شده توسط (۴۰-۶) و (۳۶-۶) بوده و $\mu_d \in (0, 1/L]$ در اینصورت، هر نقطه‌ی حدی $\{\mathbf{D}_k, \mathbf{G}_k\}$ به یک نقطه‌ی بحرانی از $H(\mathbf{D}, \mathbf{G})$ که در (۳۱-۶) تعریف شده است، همگرا می‌شود.

اثبات این قضیه با استفاده از اثبات همگرایی کلی الگوریتم PALM، که در قسمت ۴-۶ مرور شد، امکان‌پذیر است. یک توصیف کلی از الگوریتم پیشنهادی، که آن را RINC-DL^۱ نام نهاده‌ایم، در شکل ۶-۳ آورده شده است. در این شکل، $SA(\mathbf{Y}, \mathbf{D}, \tau)$ به معنای بدست آوردن نمایش تُنک داده‌های \mathbf{Y} روی دیکشنری \mathbf{D} و با پارامتر تُنکی τ است.

۶-۴-۲ حالت مقید

در این قسمت، الگوریتم پیشنهادی برای یادگیری دیکشنری‌های مقید به یک حد بالا برای همبستگی متقابل ارائه می‌شود. این الگوریتم سعی در حل مستقیم مسأله‌ی (۲۰-۶) دارد. همانطور که قبلاً گفته شد، این موضوع بر

^۱Regularized incoherent dictionary learning

• هدف: حل مسأله‌ی (۶-۲۹) با استفاده از PALM

• ورودی: $J, I, c, L_1, c, \lambda, \tau, \mathbf{D}, \mathbf{Y}$

• مقداردهی اولیه: $\mathbf{G} = \mathbf{0}, \mathbf{D} = \mathbf{D}_0$

• شروع الگوریتم: گام‌های زیر را برای $k = 0, 1, \dots$ انجام بده

۱. نمایش تُنک: $\mathbf{X} = \text{SA}(\mathbf{Y}, \mathbf{D}, \tau)$
۲. به‌روزرسانی دیکشنری.
۳. $L_1 = \|\mathbf{X}^T \mathbf{X}\|$
۴. $\alpha = 3 \cdot \|\mathbf{D}^T \mathbf{D} - \mathbf{I}\|_\infty$
۵. $i = 1$
۶. تا زمانی که $\|\mathbf{G} - \mathbf{D}^T \mathbf{D}\|_F > \epsilon$ و $i \leq I$ انجام بده:
 - $\mu_d = 1 / (L_1 + \alpha^{-1} L_2)$ □
 - برای $j = 1, 2, \dots, J$ انجام بده:
 - * $\mathbf{G} = \mathbf{D}^T \mathbf{D} - P_{\mathcal{B}_{\lambda, \alpha}}(\mathbf{D}^T \mathbf{D} - \mathbf{I})$ *
 - * $\mathbf{D} = P_{\mathcal{D}}(\mathbf{D} - \mu_d \nabla f(\mathbf{D}))$ *
 ۷. $\alpha \leftarrow c \cdot \alpha$
 ۸. $i \leftarrow i + 1$

• خروجی: \mathbf{X} و \mathbf{D}

شکل ۶-۳: الگوریتم پیشنهادی، RINC-DL، برای حل مسأله‌ی (۶-۲۹). در این شکل، منظور از SA یافتن نمایش تُنک داده‌ها است.

خلاف الگوریتم IPR-DL است، که ابتدا بدون در نظر گرفتن قید همبستگی متقابل، دیکشنری را به‌روز می‌کند و سپس نتیجه‌ی حاصل را برای داشتن همبستگی متقابل مطلوب بهینه می‌کند. با جایگزینی $\mu(\mathbf{D})$ با تعریف معادل آن در (۶-۲۸)، مسأله‌ی (۶-۲۰) به فرم زیر تبدیل می‌شود

$$\min_{\mathbf{D} \in \mathcal{D}} \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{D}^T \mathbf{D} - \mathbf{I}\|_\infty \leq \mu_0. \quad (۶-۴۸)$$

استراتژی پیشنهادی برای حل مسأله‌ی فوق همانند چیزی است که برای بدست آوردن الگوریتم RINC-DL انجام شد. برای این منظور، مشابه قبل، متغیر کمکی $\mathbf{G} \triangleq \mathbf{D}^T \mathbf{D}$ را تعریف می‌کنیم. با این کار، مسأله‌ی زیر را خواهیم داشت

$$\min_{\mathbf{D} \in \mathcal{D}, \mathbf{G}} \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{G} - \mathbf{I}\|_\infty \leq \mu_0, \quad \mathbf{G} = \mathbf{D}^T \mathbf{D}. \quad (۶-۴۹)$$

با استفاده از ایده‌ی روش جریمه، مسأله‌ای که باید حل شود مشابه (۶-۳۱) است، با این تفاوت که در اینجا،

$$r_g(\mathbf{G}) \triangleq \delta_{\hat{\mathbf{G}}_{\mu_0}} \quad \text{که در آن}$$

$$\hat{\mathbf{G}}_{\mu_0} \triangleq \{\mathbf{G} \mid \|\mathbf{G} - \mathbf{I}\|_{\infty} \leq \mu_0\}. \quad (50-6)$$

ابتدا مسأله‌ی به‌روزرسانی \mathbf{G} را در نظر بگیرید. این مسأله بصورت زیر است

$$\mathbf{G}_{k+1} = \mathbf{I} + \operatorname{argmin}_{\mathbf{G} \in \mathcal{B}_{\infty}^{\mu_0}} \frac{1}{\sqrt{2}} \|\mathbf{G} - \tilde{\mathbf{G}}_k\|_F^2, \quad (51-6)$$

که در آن، $\tilde{\mathbf{G}}_k \triangleq \mathbf{D}_k^T \mathbf{D}_k - \mathbf{I}$. عبارت دوم در سمت راست تساوی فوق عبارت است از تصویر $\tilde{\mathbf{G}}_k$ روی توپ ℓ_{∞} به شعاع μ_0 . با توجه به این که مجموعه‌ی $\mathcal{B}_{\infty}^{\mu_0}$ محدب و بسته است، تصویر روی این مجموعه وجود داشته و یکتا است [۲۳]. برای بدست آوردن این تصویر، از لم زیر استفاده می‌کنیم

لم ۳-۶ تصویر ماتریس \mathbf{U}^0 روی توپ ℓ_{∞} ، یعنی \mathcal{B}_{∞}^r ، را که با $\mathbf{U}^p \triangleq P_{\mathcal{B}_{\infty}^r}(\mathbf{U}^0)$ نشان می‌دهیم با عبارت زیر داده

می‌شود:

$$u_{ij}^p = \begin{cases} \operatorname{sgn}(u_{ij}^0) \cdot r & |u_{ij}^0| > r \\ u_{ij}^0 & \text{oth.} \end{cases} \quad (52-6)$$

برای اثبات این لم، دقت داریم که با توجه به تعریف نرم بی‌نهایت برای یک ماتریس (ماکسیمم اندازه‌ی درایه‌های

آن)، داریم

$$\{\mathbf{U} \in \mathcal{B}_{\infty}^r\} \equiv \{\forall i, j : |u_{ij}| \leq r\}. \quad (53-6)$$

سپس، با توجه به تعریف $P_{\mathcal{B}_{\infty}^r}(\cdot)$ داریم

$$\mathbf{U}^p = \operatorname{argmin}_{\mathbf{U} \in \mathcal{B}_{\infty}^r} \frac{1}{\sqrt{2}} \|\mathbf{U} - \mathbf{U}^0\|_F^2, \quad (54-6)$$

و یا

$$\mathbf{U}^p = \operatorname{argmin}_{|u_{ij}| \leq r} \frac{1}{\sqrt{2}} \sum_{i,j} (u_{ij} - u_{ij}^0)^2. \quad (55-6)$$

مسأله‌ی فوق روی درایه‌های u_{ij} جداپذیر است. این موضوع منجر به رابطه‌ی زیر می‌شود

$$\forall i, j : u_{ij}^p = \operatorname{argmin}_{|u| \leq r} \frac{1}{\sqrt{2}} (u - u_{ij}^0)^2. \quad (56-6)$$

این مسأله عبارت است از تصویر روی مجموعه قیدهایی به فرم بازه. جواب این مسأله به فرم بسته است و خود

مقدار u_{ij}^p را برمی‌گرداند اگر دامنه‌ی آن کمتر از r باشد و در غیر اینصورت یکی از دو انتهای بازه (که به u_{ij}^0

نزدیک‌تر است) [۲۳]. این، اثبات لم فوق را کامل می‌کند. بنابراین، فرمول نهایی برای به‌روزرسانی \mathbf{G}_{k+1} با عبارت

زیر داده می‌شود

$$\mathbf{G}_{k+1} = \mathbf{I} + P_{\mathcal{B}_{\infty}^{\mu_0}}(\mathbf{D}_k^T \mathbf{D}_k - \mathbf{I}). \quad (57-6)$$

• هدف: حل مسأله‌ی (۴۸-۶) با استفاده از PALM

• ورودی: $J, I, \epsilon, L_1, c, \lambda, \tau, \mathbf{D}, \mathbf{Y}$

• مقداردهی اولیه: $\mathbf{G} = \mathbf{0}, \mathbf{D} = \mathbf{D}_0$

• شروع الگوریتم: گام‌های زیر را برای $k = 0, 1, \dots$ انجام بده

۱. نمایش تُنک. $\mathbf{X} = \text{SA}(\mathbf{Y}, \mathbf{D}, \tau)$
۲. به‌روزرسانی دیکشنری.
۳. $L_1 = \|\mathbf{X}^T \mathbf{X}\|$
۴. $\alpha = 3 \cdot \|\mathbf{D}^T \mathbf{D} - \mathbf{I}\|_\infty$
۵. $i = 1$
۶. تا زمانی که $i \leq I$ و $\|\mathbf{G} - \mathbf{D}^T \mathbf{D}\|_F > \epsilon$ انجام بده:
 - $\mu_d = 1 / (L_1 + \alpha^{-1} L_1)$ □
 - برای $j = 1, 2, \dots, J$ انجام بده:
 - $\mathbf{G} = \mathbf{I} + P_{B_\infty^{\mu_d}}(\mathbf{D}^T \mathbf{D} - \mathbf{I})$ *
 - $\mathbf{D} = P_{\mathcal{D}}(\mathbf{D} - \mu_d \nabla f(\mathbf{D}))$ *
 ۷. $\alpha \leftarrow c \cdot \alpha$
 ۸. $i \leftarrow i + 1$

• خروجی: \mathbf{X} و \mathbf{D}

شکل ۴-۶: الگوریتم پیشنهادی، CINC-DL، برای حل مسأله‌ی (۲۹-۶). در این شکل، منظور از SA یافتن نمایش تُنک داده‌ها است.

مسأله‌ی به‌روزرسانی \mathbf{D}_{k+1} مشابه (۳۷-۶) و به همین دلیل جزئیات آن را دوباره ذکر نمی‌کنیم. انجام تکرار بین (۵۷-۶) و (۴۰-۶) جوابی تقریبی برای مسأله‌ی (۴۹-۶) بدست می‌دهد. همچنین، قضیه‌ی زیر، که مشابه قضیه‌ی ۱-۶ است، همگرایی الگوریتم حالت مقید را، که آن را CINC-DL^۱ می‌نامیم، نشان می‌دهد. جزئیات این الگوریتم در شکل ۴-۶ تشریح شده است.

قضیه ۲-۶ فرض کنید دنباله‌ی تولید شده توسط (۴۰-۶) و (۵۷-۶) بوده و $\mu_d \in (0, 1/L]$. در اینصورت، هر نقطه‌ی حدی $\{\mathbf{D}_k, \mathbf{G}_k\}$ به یک نقطه‌ی بحرانی از $H(\mathbf{D}, \mathbf{G})$ که در (۳۱-۶) تعریف شده است، همگرا می‌شود.

اثبات این قضیه نیز از روی اثبات الگوریتم PALM (بخش ۴-۶) دنبال می‌شود.

^۱Constrained incoherent dictionary learning

بحث

مشابه IPR-DL، الگوریتم CINC-DL در کاربردهایی استفاده می‌شود که یک حد بالا روی $\mu(\mathbf{D})$ مطلوب است. از طرف دیگر، الگوریتم RINC-DL در کاربردهایی که یک مصالحه بین خطای نمایش و میزان $\mu(\mathbf{D})$ مورد نیاز است، محل استفاده پیدا می‌کند. در مقایسه با دو الگوریتم IPR-DL و CINC-DL، الگوریتم RINC-DL به راحتی و با قراردادن مقدار ضریب λ برابر با صفر، به یک الگوریتم عادی یادگیری دیکشنری تبدیل شود. یک مزیت الگوریتم CINC-DL (و نیز RINC-DL) نسبت به IPR-DL این است که با حل مستقیم مسأله‌ی مقید یادگیری دیکشنری با همبستگی متقابل کم و بصورت تکراری، این الگوریتم از مزیت شروع گرم (که در قسمت ۲-۴-۳ نیز به آن اشاره شد) سود می‌برد. به عبارت دیگر، در هر تکرار از الگوریتم، تخمین قبلی \mathbf{D} به عنوان نقطه‌ی شروع برای تکرار بعدی مورد استفاده قرار می‌گیرد. به این ترتیب، مقدار همبستگی متقابل به مرور در حین تکرارهای الگوریتم کم می‌شود. این موضوع برخلاف الگوریتم IPR-DL است که ماهیت تکراری نداشته و در نتیجه نمی‌تواند از شروع گرم بهره‌بردار.

انتخاب پارامتر

در هر تکرار از دو الگوریتم پیشنهادی، پارامتر α بصورت یک ضریب ثابت از $\|\mathbf{D}^T \mathbf{D} - \mathbf{I}\|_\infty$ مقداردهی می‌شود. این انتخاب با انجام شبیه‌سازی‌های متعدد حاصل شده است. علاوه بر پارامتر جریمه، پارامترهای دیگری وجود دارند که عبارتند از ضریب کاهش پارامتر جریمه، یعنی c ، تعداد تکرارهای بهینه‌سازی نوبتی برای به‌روزرسانی \mathbf{G} و \mathbf{D} ، که آن را با J نشان داده‌ایم، شرط توقف، ϵ ، و ثابت لیبشیتز ∇f_2 ، یعنی L_2 . تاثیر این پارامترها روی عملکرد الگوریتم‌های پیشنهادی در ادامه توضیح داده می‌شود و علاوه بر آن، در قسمت شبیه‌سازی هم مورد بررسی قرار می‌گیرند.

- c : یک مقدار کوچک برای c متناظر با کاهش سریع پارامتر جریمه، یعنی α ، است. بنابراین، انتظار می‌رود که گام‌های به‌روزرسانی دیکشنری در الگوریتم‌های پیشنهادی با سرعت بیشتری همگرا شوند. از طرف دیگر، یک مقدار کوچک برای c پیوستگی دنباله‌ی α و تکرارهای الگوریتم را به میزان بیشتری حفظ می‌کند، که می‌تواند به عملکرد بهتری منجر شود اما زمان اجرا را افزایش می‌دهد.
- J : روش شروع گرم که در تکرارهای داخلی و خارجی الگوریتم‌های پیشنهادی انجام می‌شود کمک می‌کند

تا به روزرسانی متغیرها با تعداد تکرارهای کمتری انجام شود. بنابراین، نیازی وجود ندارد که تا همگرایی کامل حلقه‌های مربوط به J صبر کنیم، بلکه تنها انجام چند تکرار مختصر کفایت می‌کند.

- ϵ : شرط توقف در گام‌های به‌روزرسانی الگوریتم‌های پیشنهادی بر عملکرد کلی الگوریتم‌ها تاثیر دارند. یک مقدار کوچک برای ϵ کیفیت نهایی جواب‌ها را بهتر می‌کند، اما این امر به قیمت حجم محاسبات بیشتر تمام می‌شود.

- L_2 : ثابت لپشیتز ∇f_2 طول گام را برای به‌روزرسانی دیکشنری در الگوریتم‌های پیشنهادی مشخص می‌کند. این پارامتر اگر به درستی انتخاب نشود می‌تواند عملکرد کلی الگوریتم‌ها را تحت تاثیر قرار دهد. یک مقدار تقریبی برای این پارامتر در رابطه‌ی (۶-۴۷) بدست آمد که می‌تواند نتایج قابل قبولی داشته باشد. برای تنظیم بهتر آن می‌توان آن را در طول الگوریتم و با روش‌هایی متعدد موجود، مثلاً آنچه در [۹۵] پیشنهاد شده است، تنظیم نمود.

پیچیدگی محاسباتی

در این قسمت، تحلیل پیچیدگی محاسباتی الگوریتم‌های پیشنهادی و نیز الگوریتم IPR-DL برحسب تعداد عملیات نقطه-شناور^۱ (flops) ارائه می‌شود. از آنجایی که گام نمایش تُنک برای این الگوریتم‌ها یکسان است، تنها گام به‌روزرسانی دیکشنری مورد بررسی قرار می‌گیرد.

همانطور که در [۱۴] بحث شده است، پیچیدگی محاسباتی الگوریتم IPR-DL بیشتر مربوط به محاسبه‌ی EVD ماتریس $D^T D$ است که نیازمند $O(N^3)$ عملیات بوده، محاسبه ماتریس کواریانس $C = (DX)Y^T$ که نیاز به $O(n^2 M)$ عملیات دارد، و در نهایت، محاسبه‌ی SVD ماتریس کواریانس که برابر با $O(n^3)$ است. برای الگوریتم پیشنهادی، پیچیدگی محاسباتی با گام‌های به‌روزرسانی G و حلقه‌های داخلی به‌روزرسانی دیکشنری تعیین می‌شود. هر تصویر کردن ستونی روی توپ ℓ_1 هزینه‌ای برابر با $O(N)$ دارد [۴۸]. بنابراین، کل تصویر کردن (برای همه‌ی ستون‌ها) در الگوریتم RINC-DL نیازمند $O(N^2)$ عملیات است. گام به‌روزرسانی G در الگوریتم CINC-DL یک آستانه‌گذاری ساده بوده که $O(N^2)$ عملیات لازم دارد. حلقه‌های داخلی در هر دو الگوریتم پیشنهادی به اندازه‌ی $O(nNM + nN^2)$ محاسبات نیاز دارد. تحلیل کامل پیچیدگی برای الگوریتم‌های مختلف در جدول ۶-۱

^۱Floating-point operations

جدول ۶-۱: پیچیدگی‌های محاسباتی برای الگوریتم‌های مختلف. ابعاد دیکشنری $n \times N$ و تعداد داده‌های آموزشی برابر با M است.

الگوریتم	پیچیدگی
IPR-DL	$O(nNM + MN^2 + 3n^2N + 2n^3 + 2N^3)$
CINC-DL	$O(nNM + nN^2)$
RINC-DL	$O(nNM + nN^2)$

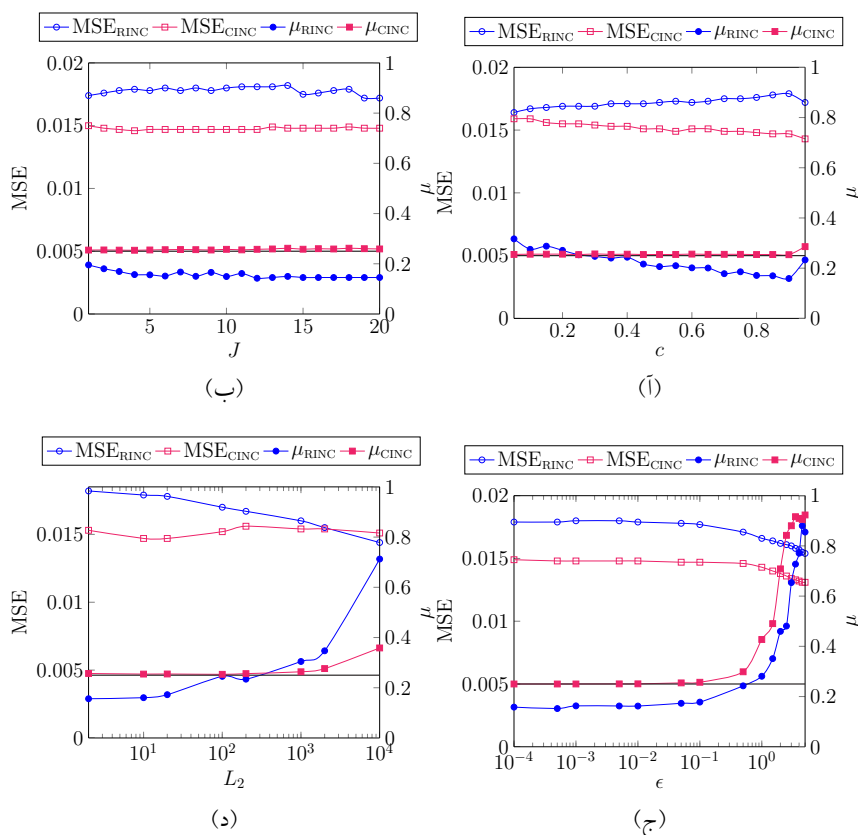
آمده است. با توجه به این جدول، برای تمامی الگوریتم‌ها پیچیدگی محاسباتی نسبت به تعداد داده‌های آموزشی بصورت خطی تغییر می‌کند. همچنین، دو الگوریتم پیشنهادی پیچیدگی مشابهی دارند. مهم‌تر از همه، پیچیدگی IPR-DL بصورت درجه سه با ابعاد داده‌ها و نیز تعداد اتم‌ها تغییر می‌کند، حال آن‌که الگوریتم‌های پیشنهادی رفتاری خطی نسبت به این دو پارامتر دارند. در نتیجه، این الگوریتم‌ها برای داده‌های با ابعاد بالا مناسب‌تر هستند.

۶-۴-۳ شبیه‌سازی

در این قسمت، عملکرد الگوریتم‌های پیشنهادی با الگوریتم‌های قبلی برای یادگیری دیکشنری‌های با همبستگی متقابل کم مقایسه می‌شود. از میان الگوریتم‌های نامقید، الگوریتم BSC-DL^۱ [۱۱۱] برای مقایسه انتخاب شده است. از میان الگوریتم‌های حالت مقید نیز الگوریتم IPR-DL را در نظر گرفته‌ایم.

شیوه‌ی انجام آزمایش این گونه است که ابتدا تعداد $M = 50,000$ بلوک 8×8 از چند تصویر طبیعی معروف را بصورت تصادفی استخراج کرده و آنها را تبدیل به بردارهایی با طول معادل ۶۴ می‌کنیم. این بلوک‌ها سپس ماتریس داده‌های آموزشی \mathbf{Y} را تشکیل می‌دهند. برای همه‌ی الگوریتم‌ها، دیکشنری اولیه را DCT با ابعاد 256×64 انتخاب کردیم. همچنین، از الگوریتم OMP به عنوان گام نمایش تُنک استفاده شده است. حداکثر تعداد اتم‌ها برای نمایش تُنک را $s = 10$ گرفته‌ایم. تعداد تکرارهای یادگیری دیکشنری نیز ۳۰۰ در نظر گرفته شده است. برای ارزیابی کیفیت خروجی الگوریتم‌ها، از معیار MSE که بصورت $\|\mathbf{Y} - \mathbf{DX}\|_F / (n \cdot M)$ تعریف می‌شود، استفاده شده است. پارامترهای الگوریتم‌ها به شیوه‌ی زیر تنظیم شده‌اند. برای الگوریتم BSC-DL، از پارامترهای پیشنهاد شده در [۱۱۱] استفاده شده است. برای الگوریتم IPR-DL، حداکثر تعداد تکرارها در گام ناهمبسته‌سازی آن برابر با ۱۰۰ قرار داده شده است. برای الگوریتم‌های پیشنهادی نیز قرار داده‌ایم $I = 100$ ، $J = 3$ ، $L_2 = 20$.

^۱Bounded self-coherence DL

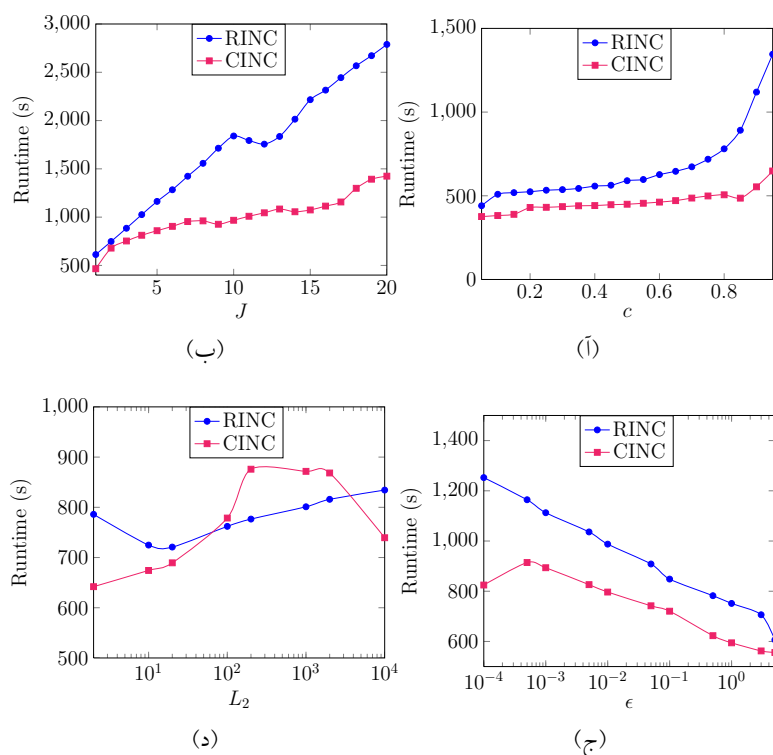


شکل ۶-۵: نمودارهای مقادیر نهایی خطا و همبستگی متقابل برای الگوریتم‌های پیشنهادی و مقادیر مختلف پارامترهای آنها. مقادیر پیش فرض برای پارامترها برابر است با $c = 0.85$, $J = 3$, $\epsilon = 0.05$, $L_2 = 20$. برای بررسی تاثیر هر کدام از پارامترها، بقیه پارامترها به مقادیر پیش فرض خود تنظیم شده‌اند.

$$c = 0.85 \text{ و } \epsilon = 0.05.$$

ابتدا تاثیر پارامترهای مختلف الگوریتم‌های پیشنهادی را بر عملکرد آنها بررسی می‌کنیم. برای دیدن نقش هر پارامتر، بقیه پارامترهای به مقادیر پیش فرضی تنظیم می‌شوند. بعلاوه، پارامتر λ در الگوریتم RINC-DL برابر با ۵۰ (که متناظر با یک همبستگی متقابل پائین است) و پارامتر μ در الگوریتم CINC-DL برابر با ۰.۲۵ قرار داده شده است. برای همه پارامترها، مقادیر نهایی MSE به همراه مقادیر همبستگی متقابل دیکشنری‌های یادگیری شده در شکل ۶-۵ و زمان‌های اجرای متناظر در شکل ۶-۶ نشان داده شده‌اند.

با دقت در این شکل‌ها، بحث‌هایی را که قبلاً در مورد نقش پارامترها بر روی عملکرد الگوریتم‌ها شده بود، می‌توان تصدیق نمود. به طور خاص، همانطور که بحث شد، یک مقدار کم برای J کافی است تا نتیجه‌ای مطلوب حاصل شود. همچنین، در مورد پارامتر c ، به نظر می‌رسد که مقادیر نزدیک‌تر به یک منجر به عملکرد متوسط

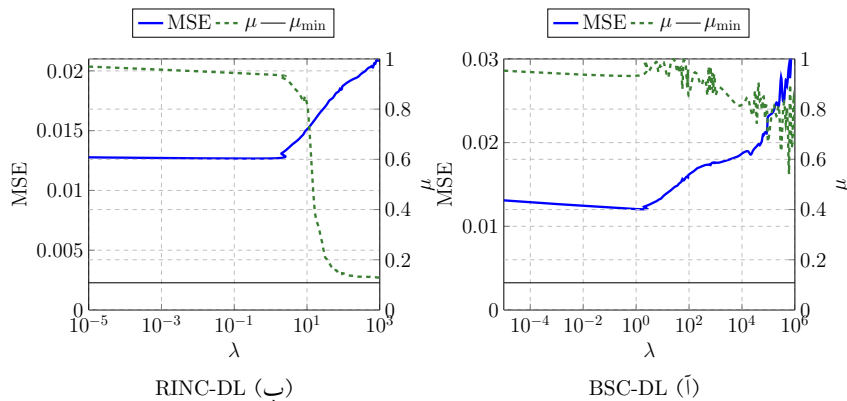


شکل ۶-۶: زمان اجرای الگوریتم‌های پیشنهادی برحسب مقادیر مختلف پارامترهای آنها. بقیه‌ی تنظیمات شبیه شکل ۶-۵ است.

بهتری می‌شوند.

در قدم بعدی، عملکرد الگوریتم‌های BSC-DL و RINC-DL را بعنوان نمونه‌هایی از الگوریتم‌های مقید بررسی می‌کنیم. نتایج شبیه‌سازی مربوط به این الگوریتم‌ها در شکل ۶-۷ خلاصه شده است. با توجه به این شکل، می‌توان نتیجه گرفت که:

- همانطور که انتظار می‌رفت، برای هر دو الگوریتم، با بالا بردن پارامتر λ مقدار همبستگی متقابل دیکشنری نهایی کاهش می‌یابد. بعلاوه، الگوریتم BSC-DL رفتار متغیری در این زمینه نشان می‌دهد که از نوسانات نمودار مربوط به آن آشکار است. الگوریتم پیشنهادی اما رفتاری با ثبات‌تر دارد. همچنین، مقادیر خطای نمایش برای الگوریتم پیشنهادی به مراتب کمتر از الگوریتم BSC-DL است.
- حداقل مقدار همبستگی متقابل دیکشنری آموزش دیده شده توسط الگوریتم BSC-DL برابر با 0.54 است، حال آن که الگوریتم پیشنهادی می‌تواند دیکشنری‌ای با همبستگی متقابل 0.1296 بدست بدهد که به مقدار باند ولج، یعنی $\mu_{\min} = 0.1085$ نزدیک است.

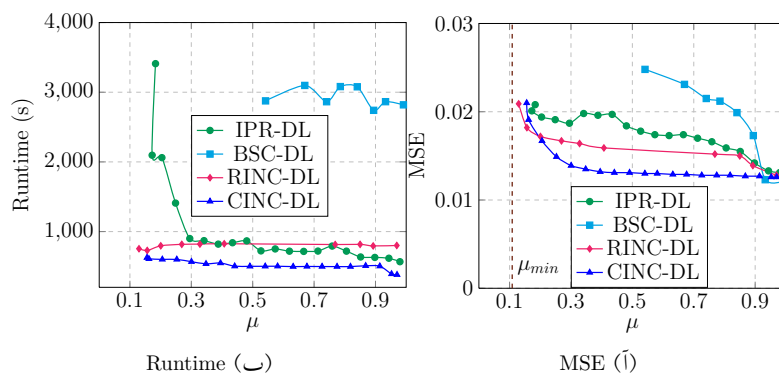


شکل ۶-۷: مقادیر نهایی خطای نمایش و همبستگی متقابل دیکشنری‌های یادگیری‌شده، برای دو الگوریتم BSC-DL و RINC-DL و بعنوان توابعی از پارمتر λ . مقدار باند ولج هم با μ_{min} مشخص شده است.

در ادامه، عملکرد دو الگوریتم مقید CINC-DL و IPR-DL مقایسه می‌شود. مقادیر نهایی خطای نمایش برحسب مقادیر مختلف همبستگی متقابل و برای تمامی الگوریتم‌ها، که شامل BSC-DL و RINC-DL هستند، در شکل ۶-۸ و نیز زمان اجرای آنها در شکل ۶-۸ رسم شده است. با بررسی نتایج این شکل‌ها نتیجه‌گیری‌های متعددی می‌توان داشت. اول، همانطور که دیده می‌شود، الگوریتم CINC-DL بهترین عملکرد را از منظر خطای نمایش و نیز زمان اجرا دارد. با این حال، RINC-DL عملکرد بهتری در کاهش همبستگی متقابل دیکشنری از خود نشان می‌دهد. از طرف دیگر، الگوریتم IPR-DL عملکرد نسبتاً ضعیفی در مقابل الگوریتم‌های پیشنهاد دارد. زمان اجرای آن نیز به میزان همبستگی متقابل وابسته است، به نحوی که یادگیری دیکشنری‌های با همبستگی متقابل نزدیک به باند ولج زمان اجرای بیشتری دارد. در نهایت، الگوریتم BSC-DL بدترین عملکرد را هم از نظر زمان اجرا و هم خطای نمایش دارد. بعلاوه، حداقل میزان همبستگی متقابلی که به آن دست می‌یابد از مقدار متناظر در الگوریتم‌های دیگر بالاتر است.

۵-۶ طراحی ماتریس حسگر

در قسمت قبل، یادگیری دیکشنری‌های با همبستگی متقابل کم از روی داده‌های آموزشی را بررسی کردیم. در این قسمت، طراحی ماتریس‌های با همبستگی متقابل کم را در نظر می‌گیریم. این ماتریس‌ها کاربرد گوناگونی در زمینه‌های مختلف پردازش سیگنال از جمله حسگری فشرده [۲۶] و مخابرات [۷۵] دارند. در این کاربردها، برخلاف یادگیری دیکشنری، داده‌ی آموزشی نداریم و هدف صرفاً طراحی ماتریس‌هایی است که همبستگی متقابل



شکل ۶-۸: مقایسه‌ی الگوریتم‌های مختلف برحسب (آ) مقدار نهایی خطای نمایش و (ب) زمان اجرا، و بازای مقادیر همبستگی‌های متقابل.

کمی دارند. برای ادامه‌ی بحث، نیاز به یادآوری مفهوم فریم^۱ داریم [۱۱۵]. همانطور که در بخش ۳-۲ مرور شد،

مجموعه‌ای از m بردار $\{\mathbf{f}_i\}_{i=1}^m$ تشکیل یک فریم برای \mathbb{R}^n می‌دهند اگر برای هر $\mathbf{v} \in \mathbb{R}^n$ داشته باشیم

$$\alpha \|\mathbf{v}\|^2 \leq \sum_{i=1}^m |\mathbf{f}_i^T \mathbf{v}|^2 \leq \beta \|\mathbf{v}\|^2, \quad (58-6)$$

که در آن، α و β با $0 < \alpha \leq \beta < \infty$ ، به ترتیب باندهای پائین و بالایی فریم نامیده می‌شوند. عملگر ترکیب فریم^۲ بعنوان ماتریسی تعریف می‌شود که ستون‌های آن عبارتند از بردارهای فریم، یعنی $\mathbf{F} = [\mathbf{f}_1, \dots, \mathbf{f}_m]$. هنگام ارجاع به یک فریم، عموماً منظور ماتریس \mathbf{F} است. یک فریم α -تایت واحد \mathbf{F} فریمی است که برای آن α تمام بردارهای فریم واحد است. بعلاوه، اگر $\alpha = \beta$ ، فریم α -تایت^۳ نامیده می‌شود. ماتریس گرام^۴ برای یک فریم \mathbf{F} بصورت $\mathbf{G} = \mathbf{F}^T \mathbf{F}$ تعریف می‌شود. فریم‌های با همبستگی متقابل خیلی کم، فریم‌های ناهمبسته^۵ نامیده می‌شوند.

همانطور که از بحث‌های گذشته می‌توان فهمید، در بسیاری از کاربردها، بخصوص در بازیابی نمایش \mathbf{t} ، فریم‌های با همبستگی متقابل کم مطلوب هستند. قبلاً دیدیم که کمترین مقدار همبستگی متقابل برای یک فریم با باند ولج مشخص می‌شود. فریم‌هایی که همبستگی متقابل آنها برابر با باند ولج است، فریم‌های گراسمانی (بهینه)^۶ یا فریم‌های تایت هم‌زاویه^۷ (ETF) نامیده می‌شوند. بنابراین، فریم‌های ETF از اهمیت ویژه‌ای برخوردار

^۱Frame

^۲Frame synthesis operator

^۳N-tight frame

^۴Gram matrix

^۵Incoherent frames

^۶Grassmannian frames

^۷Equiangular tight frame

هستند. با این وجود، طراحی این فریم‌ها، بخصوص برای ابعاد بالا، کاری ساده نیست. بعلاوه، این فریم‌ها برای هر ابعادی وجود ندارند [۱۲۲، ۱۱۵]. برخی از رهیافت‌های طراحی فریم‌های ناهمبسته مبتنی بر روش‌های جبری و تحلیلی هستند، مانند [۱۳۱، ۱۲۶، ۱۰، ۵۶]. این روش‌ها ولی محدودیت‌هایی روی ابعاد و مشخصات فریم‌ها می‌گذارند و لذا برای طراحی هر نوع فریمی قابل استفاده نیستند. دسته‌ی دیگری از روش‌ها مبتنی بر فرمول‌بندی مسائل محدب یا غیرمحدب برای طراحی فریم هستند، مانند [۱۲۰، ۵۱، ۱۰۲، ۷۸، ۱۲۲، ۱۰۳]. برخلاف روش‌های قبلی، روش‌های مبتنی بر بهینه‌سازی هیچ قیدی روی مشخصات فریم نگذاشته و در نتیجه برای طراحی فریم‌های ناهمبسته با هر بُعدی مناسب هستند.

در این قسمت، تمرکز ما بر روی الگوریتم‌های طراحی فریم‌های ناهمبسته مبتنی بر بهینه‌سازی است. قبل از ارائه الگوریتم پیشنهادی [۱۰۵]، ابتدا مروری داریم بر یک الگوریتم نسبتاً جدید که در [۱۰۲] معرفی شده است. این الگوریتم، مسأله‌ی زیر را حل می‌کند

$$\min_{\mathbf{F} \in \mathcal{F}} \max_{i,j: i \neq j} |\mathbf{f}_i^T \mathbf{f}_j|, \quad (59-6)$$

که در آن، $\mathcal{F} \triangleq \{\mathbf{F} \in \mathbb{R}^{n \times m} \mid \forall i: \|\mathbf{f}_i\|_2 = 1\}$. برای این منظور، ایده‌ای که استفاده شده است به این ترتیب است که بردارهای فریم بصورت یکی یکی و با ثابت گرفتن بقیه به‌روزرسانی می‌شوند. برای به‌روزرسانی بردار فریم شماره i ، یعنی \mathbf{f}_i ، مسأله‌ی زیر حل می‌شود

$$\min_{\mathbf{f}_i} \max_{j: j \neq i} |\mathbf{h}_j^T \mathbf{f}_i| \text{ s.t. } \forall i: \|\mathbf{f}_i - \mathbf{h}_i\|_2 \leq T_i, \quad (60-6)$$

که \mathbf{h}_i ها تخمین‌های قبلی از بردارهای فریم هستند و T_i ثابت‌هایی هستند که ناحیه جستجو را مشخص می‌کنند. مسأله‌ی فوق برای هر کدام از بردارهای فریم با استفاده از بسته‌ی بهینه‌سازی CVX [۶۶] حل می‌شود. این روش در مقاله‌ی [۱۰۳] بهبود داده شده است. الگوریتم جدید، که SIDCO^۱ نام دارد عملکرد بهتری نسبت به روش‌های قبلی از خود نشان داده است.

۶-۵-۱ الگوریتم پیشنهادی

در این قسمت، روش پیشنهادی برای طراحی فریم‌های ناهمبسته ارائه می‌شود [۱۰۵]. برای این منظور، مشابه الگوریتم SIDCO، مسأله‌ی اصلی پیشنهادی برای حل، کمینه‌سازی همبستگی متقابل است. این مسأله را به فرم

^۱Sequential iterative decorrelation by convex optimization

زیر بیان می‌کنیم

$$\min_{\mathbf{F} \in \mathcal{F}} \|\mathbf{F}^T \mathbf{F} - \mathbf{I}\|_{\infty}. \quad (۶۱-۶)$$

برای حل این مسأله، از روش جریمه استفاده کرده و متغیر جدید $\mathbf{Q} \triangleq \mathbf{F}^T \mathbf{F} - \mathbf{I}$ را معرفی می‌کنیم. به این ترتیب، مسأله‌ی زیر را داریم

$$\min_{\mathbf{F} \in \mathcal{F}, \mathbf{Q}} \|\mathbf{Q}\|_{\infty} \quad \text{s.t.} \quad \mathbf{Q} = \mathbf{F}^T \mathbf{F} - \mathbf{I} \quad (۶۲-۶)$$

که با استفاده از روش جریمه، به مسأله‌ی زیر تبدیل می‌شود

$$\{\mathbf{Q}^{\alpha}, \mathbf{F}^{\alpha}\} = \operatorname{argmin}_{\mathbf{F} \in \mathcal{F}, \mathbf{Q}} \left\{ \|\mathbf{Q}\|_{\infty} + \frac{1}{\sqrt{\alpha}} \|\mathbf{Q} - \mathbf{F}^T \mathbf{F} + \mathbf{I}\|_F^2 \right\}, \quad (۶۳-۶)$$

که مشابه قبل، $\alpha > 0$ ضریب جریمه است. مسأله‌ی فوق در ادامه با استفاده از بهینه‌سازی نوبتی حل می‌شود. به عبارت دیگر، با شروع از یک مقدار اولیه برای \mathbf{F}^{α} ، کل فرایند حل عبارت است از حل متوالی دو مسأله‌ی زیر

$$\mathbf{Q}_{k+1}^{\alpha} = \operatorname{argmin}_{\mathbf{Q}} \left\{ \|\mathbf{Q}\|_{\infty} + \frac{1}{\sqrt{\alpha}} \|\mathbf{Q} - \mathbf{F}_k^{\alpha T} \mathbf{F}_k^{\alpha} + \mathbf{I}\|_F^2 \right\}, \quad (۶۴-۶)$$

و

$$\mathbf{F}_{k+1}^{\alpha} = \operatorname{argmin}_{\mathbf{F} \in \mathcal{F}} \frac{1}{\sqrt{\alpha}} \|\mathbf{Q}_{k+1}^{\alpha} - \mathbf{F}^T \mathbf{F} + \mathbf{I}\|_F^2. \quad (۶۵-۶)$$

ابتدا گام به‌روزرسانی \mathbf{Q} را در نظر می‌گیریم. عملگر $\operatorname{vec}(\cdot) : \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m^2}$ یک ماتریس را بعنوان ورودی گرفته و آن را با روی هم قرار دادن ستون‌هایش، تبدیل به یک بردار می‌کند. عملگر $\operatorname{vec}^{-1}(\cdot)$ نیز عکس این عملیات، یعنی تبدیل یک بردار به ماتریس متناظر را انجام می‌دهد. با این تعریف، مسأله‌ی (۶۴-۶) را می‌توان به فرم معادل

زیر نوشت

$$\mathbf{q}_{k+1}^{\alpha} = \operatorname{argmin}_{\mathbf{q}} \left\{ \|\mathbf{q}\|_{\infty} + \frac{1}{\sqrt{\alpha}} \|\mathbf{q} - \mathbf{p}_k^{\alpha}\|_2^2 \right\}, \quad (۶۶-۶)$$

که $\mathbf{p}_k^{\alpha} \triangleq \operatorname{vec}(\mathbf{P}_k^{\alpha})$ ، $\mathbf{q}_{k+1}^{\alpha} \triangleq \operatorname{vec}(\mathbf{Q}_{k+1}^{\alpha})$ و $\mathbf{P}_k^{\alpha} \triangleq \mathbf{F}_k^{\alpha T} \mathbf{F}_k^{\alpha} - \mathbf{I}$ طبق تعریف، عبارت (۶۶-۶) برابر است با عملگر پراکسیمال $\alpha \|\cdot\|_{\infty}$ در نقطه‌ی \mathbf{p}_k^{α} . مشابه قبل و با استفاده از ویژگی مورو، عبارت زیر را خواهیم داشت

$$\mathbf{q}_{k+1}^{\alpha} = \operatorname{prox}_{\alpha f}(\mathbf{p}_k^{\alpha}) = \mathbf{p}_k^{\alpha} - \alpha \cdot \operatorname{prox}_{f^*/\alpha}(\mathbf{x}/\alpha). \quad (۶۷-۶)$$

که $f(\mathbf{x}) = \|\mathbf{x}\|_{\infty}$ در نهایت، عبارت $\mathbf{Q}_{k+1}^{\alpha}$ در (۶۴-۶) بصورت زیر بدست می‌آید

$$\mathbf{Q}_{k+1}^{\alpha} = \mathbf{P}_k^{\alpha} - \operatorname{vec}^{-1}(\mathcal{P}_{B_f^{\alpha}}(\operatorname{vec}(\mathbf{P}_k^{\alpha}))). \quad (۶۸-۶)$$

برای سرعت‌بخشی به الگوریتم، از یک ایده‌ی وزن‌دهی بصورت زیر استفاده می‌شود [۹۵]

$$\begin{cases} \mathbf{P}_k^{\alpha} = \mathbf{F}_k^{\alpha T} \mathbf{F}_k^{\alpha} - \mathbf{I} \\ \mathbf{R}_k^{\alpha} = \mathbf{P}_k^{\alpha} + w_1 (\mathbf{P}_k^{\alpha} - \mathbf{P}_{k-1}^{\alpha}) \\ \mathbf{Q}_{k+1}^{\alpha} = \mathbf{R}_k^{\alpha} - \operatorname{vec}^{-1}(\mathcal{P}_{B_f^{\alpha}}(\operatorname{vec}(\mathbf{R}_k^{\alpha}))) \end{cases} \quad (۶۹-۶)$$

- هدف: طراحی یک فریم \mathbf{F} با همبستگی متقابل کم
- ورودی: \mathbf{F}_0 , $\eta > 0$, $w_1 \geq 0$, $w_2 \geq 0$, $c \in [0.5, 1)$, N_o و N_i و T
- مقداردهی اولیه: $\mathbf{F}^{\alpha_0} = \mathbf{F}_0$, $\alpha = \alpha_0$, $i = 0$
- تا زمانی که $i < N_o$ گام‌های زیر را انجام بده:
 ۱. برای $k = 0, \dots, N_i - 1$ گام‌های زیر را انجام بده:
 - به‌روزرسانی $\mathbf{Q}_{k+1}^{\alpha_i}$ از طریق (۶۹-۶)
 - به‌روزرسانی $\mathbf{F}_{k+1}^{\alpha_i}$ با انجام T تکرار از (۷۰-۶)
 ۲. $\alpha_{i+1} = c \cdot \alpha_i$
 ۳. $i \leftarrow i + 1$
- خروجی: \mathbf{F}

شکل ۶-۹: الگوریتم پیشنهادی IFD-AMPM برای طراحی فریم‌های ناهمبسته.

که $w_1 \geq 0$ یک ثابت داده شده است. در ادامه، مسأله‌ی به‌روزرسانی \mathbf{F} (۶۵-۶) را در نظر می‌گیریم. این مسأله با

استفاده از روش‌های پراکسیمال به شیوه‌ی زیر حل می‌شود [۹۵]

$$\begin{cases} \mathbf{S}(t) = \mathbf{F}(t) - \eta \mathbf{F}(t) (\mathbf{F}(t)^T \mathbf{F}(t) - \mathbf{Q}_{k+1}^{\alpha} - \mathbf{I}) \\ \mathbf{F}(t+1) = \mathcal{P}_{\mathcal{F}} \{ \mathbf{S}(t) \} + w_2 (\mathbf{F}(t) - \mathbf{F}(t-1)) \end{cases} \quad (70-6)$$

که t نشان‌دهنده‌ی شماره‌ی تکرار بوده، $\eta > 0$ طول گام است، $w_2 \geq 0$ پارامتر وزن بوده و در نهایت $\mathcal{P}_{\mathcal{F}} \{ \cdot \}$

عملگر تصویر روی \mathcal{F} است. با انتخاب مناسب طول گام η الگوریتم تکراری (۷۰-۶) به یک نقطه‌ی بحرانی از

(۶۵-۶) همگرا می‌شود [۹۵]. الگوریتم نهایی برای حل (۶۳-۶) بین دو گام به‌روزرسانی $\mathbf{Q}_{k+1}^{\alpha}$ از طریق (۶۹-۶)

و انجام چند تکرار از (۷۰-۶) برای به‌روزرسانی $\mathbf{F}_{k+1}^{\alpha}$ تکرار انجام می‌دهد.

الگوریتم کلی پیشنهادی، که آن را IFD-AMPM^۱ می‌نامیم، در شکل ۶-۹ خلاصه شده است. پارامترهای

موجود در این الگوریتم در ادامه بررسی می‌شوند. برای مقداردهی اولیه‌ی الگوریتم، از فریم‌های تایت نرم واحد

استفاده می‌شود. این فریم‌ها را می‌توان با الگوریتم‌هایی ساده طراحی کرد [۱۶، ۲۷]. طول گام η باید مقدار کوچکی

باشد که برای آزمایش‌های ما، مقدار $\eta = 0.05$ عملکرد خوبی داشت. همچنین، $w_1 = w_2 = 0.85$ نتایج قابل

قبولی در پی دارد. مقدار اولیه‌ی ضریب جریمه، یعنی α_0 را بصورت $\alpha_0 = 500 \times \|\mathbf{P}^{\alpha_0}\|_{\infty}$ انتخاب می‌کنیم که

در شبیه‌سازی‌ها به نتایج خوبی منجر شد. ضریب کاهشی این پارامتر را نیز $c = 0.9$ می‌گیریم. پارامترهای N_o و

N_i که تعداد تکرارهای داخلی و خارجی الگوریتم را مشخص می‌کنند بستگی به کاربرد مورد نظر و ابعاد مسأله

^۱Incoherent frame design via alternating minimization penalty method

می‌توانند تنظیم شوند. همچنین، برای به‌روزرسانی F تعداد اندکی تکرار، مثلاً $T = 3$ کفایت می‌کند.

۶-۵-۲ شبیه‌سازی

این قسمت عملکرد الگوریتم پیشنهادی را با الگوریتم‌های SIDCO [۱۰۳] و LZYCB [۷۸]، که یک روشی برای طراحی فریم‌های ناهمبسته از طریق حل مسأله‌ی بهینه‌سازی است، مقایسه می‌کند. تعداد تکرارهای هر سه الگوریتم ۲۰۰ قرار داده شده است. بعلاوه، برای الگوریتم IFD-AMPM قرار داده‌ایم $N_i = 15$.

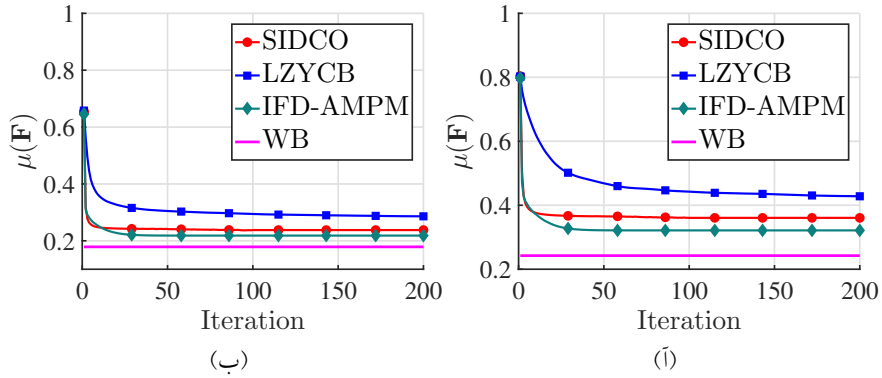
ابتدا، الگوریتم‌ها در طراحی فریم‌های ناهمبسته با ابعاد $120 \times n$ و برای مقادیر مختلف n مقایسه می‌شوند. حداقل مقدار همبستگی متقابل بدست آمده توسط این الگوریتم‌ها روی ۱۰۰ بار انجام مختلف هر آزمایش در جدول ۱ آمده است. همانطور که مشاهده می‌شود، مقادیر نهایی همبستگی‌های متقابل برای الگوریتم پیشنهادی کمتر از مقادیر متناظر در دو الگوریتم دیگر است.

شکل ۶-۱۰ رفتار $\mu(F)$ را در طول تکرار الگوریتم‌ها (که روی ۳۰ آزمایش مختلف میانگین‌گیری شده است) و برای ابعاد 120×15 و 120×25 نشان می‌دهد. با دقت در این شکل می‌توان نتیجه گرفت که الگوریتم پیشنهادی و SIDCO همگرایی سریع‌تری نسبت به الگوریتم LZYCB دارند. بعلاوه، الگوریتم پیشنهادی نهایتاً منجر به همبستگی متقابل کمتری نسبت به دو الگوریتم دیگر می‌شود. از منظر زمان اجرا، هر تکرار الگوریتم SIDCO به طور متوسط ۱۸ ثانیه، الگوریتم LZYCB ۰/۰۲۳ و الگوریتم پیشنهادی ۰/۰۴۱ طول می‌کشد. همانطور که مشخص است، الگوریتم SIDCO زمان اجرای به مراتب بالاتری نسبت به الگوریتم‌های دیگر دارد.

در آزمایش بعدی، هدف مقایسه عملکرد الگوریتم‌ها با نتایج حاصل از طراحی عددی است [۳۸]. برای این منظور، الگوریتم‌های مورد مقایسه برای طراحی فریم‌های $F \in \mathbb{R}^{3 \times m}$ با مقادیر افزایشی m اجرا شدند. نتایج حاصل در جدول ۶-۳ خلاصه شده است. همانطور که از این جدول مشخص است، نتایج الگوریتم پیشنهادی بسیار به نتایج عددی نزدیک است. همچنین، الگوریتم SIDCO نیز عملکرد مشابهی دارد ولی برای برخی مقادیر m نتایج ضعیف‌تر از الگوریتم پیشنهادی ارائه داده است. الگوریتم LZYCB نیز برای مقادیر $m \geq 14$ قادر به طراحی فریم ناهمبسته نیست.

جدول ۶-۲: مقادیر همبستگی متقابل برای الگوریتم‌های SIDCO [۱۰۳]، LZYCB [۷۸] و IFD-AMPM و فریم‌های $\mathbf{F} \in \mathbb{R}^{n \times 120}$ با مقادیر مختلف برای n .

WB	IFD-AMPM	LZYCB	SIDCO	n
۰٫۲۴۲۵	۰٫۳۲۲۵	۰٫۴۱۴۶	۰٫۳۵۰۲	۱۵
۰٫۲۰۵۰	۰٫۲۶۰۵	۰٫۳۴۰۹	۰٫۲۷۷۵	۲۰
۰٫۱۷۸۷	۰٫۲۱۸۳	۰٫۲۷۶۶	۰٫۲۳۰۳	۲۵
۰٫۱۵۸۸	۰٫۱۸۷۹	۰٫۲۳۵۸	۰٫۱۹۷۷	۳۰
۰٫۱۴۲۹	۰٫۱۶۴۹	۰٫۲۰۳۷	۰٫۱۷۲۷	۳۵
۰٫۱۲۹۶	۰٫۱۴۶۸	۰٫۱۸۱۵	۰٫۱۵۲۹	۴۰
۰٫۱۱۸۳	۰٫۱۳۱۹	۰٫۱۶۱۰	۰٫۱۳۷۱	۴۵
۰٫۱۰۸۵	۰٫۱۱۹۳	۰٫۱۴۶۷	۰٫۱۲۳۶	۵۰
۰٫۰۹۹۷	۰٫۱۰۸۶	۰٫۱۳۳۷	۰٫۱۱۲۰	۵۵
۰٫۰۹۱۷	۰٫۰۹۹۱	۰٫۱۲۰۹	۰٫۱۰۱۹	۶۰
۰٫۰۴۱۰	۰٫۰۴۴۹	۰٫۰۶۶۶	۰٫۱۴۲۱	۱۰۰
۰٫۰۲۷۶	۰٫۰۳۱۵	۰٫۰۹۸۶	۰٫۰۸۴۰	۱۱۰
۰٫۰۱۹۱	۰٫۰۲۱۹	۰٫۰۴۶۴	۰٫۰۴۵۲۰	۱۱۵
۰٫۰۰۸۴	۰٫۰۰۸۴	۰٫۰۲۳۰	۰٫۰۰۸۴	۱۱۹



شکل ۶-۱۰: مقادیر همبستگی متقابل برحسب تکرار الگوریتم‌های SIDCO [۱۰۳]، LZYCB [۷۸] و IFD-AMPM برای طراحی فریم‌های با ابعاد (آ) 15×120 و (ب) 25×120 .

۶-۶ جمع‌بندی

در این فصل، الگوریتم‌های جدیدی برای یادگیری دیکشنری برای داده‌های با ابعاد بالا، یادگیری دیکشنری‌های با همبستگی متقابل کم و نیز طراحی ماتریس‌های ناهمبسته ارائه کردیم. الگوریتم‌های پیشنهادی برای یادگیری دیکشنری‌های ناهمبسته، بر خلاف الگوریتم‌های فعلی، مستقیماً از تابع همبستگی متقابل استفاده می‌کنند. علاوه بر این، برای داده‌های با ابعاد بالا پیچیدگی محاسباتی کمتری نیز دارند. شبیه‌سازی‌های انجام شده حاکی از برتری روش‌های پیشنهادی دارد. در زمینه‌ی طراحی ماتریس‌های ناهمبسته، که کاربرد زیادی در حسگری فشرده و نیز مخابرات دارند، الگوریتم پیشنهادی نسبت به الگوریتم‌های رقیب زمان اجرای کمتری داشته و همزمان قادر به طراحی ماتریس‌های با همبستگی متقابل کمتر است.

در فصل آینده، مطالب بیان شده در این رساله را جمع‌بندی کرده، پیشنهادهای برای کارهای آینده ذکر

می‌کنیم.

جدول ۳-۶: مقادیر همبستگی متقابل برای الگوریتم‌های SIDCO [۱۰۳]، LZYCB [۷۸] و IFD-AMPM و فریم‌های $\mathbf{F} \in \mathbb{R}^{3 \times m}$ با مقادیر مختلف برای m . همچنین، مقادیر بدست آمده از طریق عددی، که در [۳۸] گزارش شده است نیز در این جدول ذکر شده است.

WB	Num. Opt.	IFD-AMPM	LZYCB	SIDCO	m
۰	۰	۰	۰	۰	۳
۰/۳۳۳۳	۰/۳۳۳۳	۰/۳۳۳۳	۰/۳۳۳۳	۰/۳۳۳۳	۴
۰/۴۰۸۲	۰/۴۴۷۲	۰/۴۴۷۲	۰/۵۳۹۳	۰/۴۴۷۲	۵
۰/۴۴۷۲	۰/۴۴۷۲	۰/۴۴۷۲	۰/۴۴۷۲	۰/۴۴۷۲	۶
۰/۴۷۱۴	۰/۵۷۷۴	۰/۵۷۷۴	۰/۵۷۷۷	۰/۵۸۹۳	۷
۰/۴۸۸۰	۰/۶۴۷۶	۰/۶۴۷۸	۰/۷۴۶۶	۰/۶۴۸۰	۸
۰/۵۰۰۰	۰/۶۶۹۴	۰/۶۶۹۴	۰/۷۴۷۱	۰/۶۶۹۴	۹
۰/۵۰۹۲	۰/۶۸۶۱	۰/۶۸۶۱	۰/۶۹۷۹	۰/۶۸۷۰	۱۰
۰/۵۱۶۴	۰/۷۱۴۴	۰/۷۱۴۴	۰/۷۵۶۵	۰/۷۱۴۹	۱۱
۰/۵۲۲۲	۰/۷۴۴۵	۰/۷۴۴۵	۰/۹۵۶۸	۰/۷۵۵۹	۱۲
۰/۵۲۷۰	۰/۷۶۸۱	۰/۷۶۸۱	۰/۹۶۳۱	۰/۷۷۲۱	۱۳
۰/۵۳۱۱	۰/۷۸۰۶	۰/۷۸۰۶	۱	۰/۷۸۲۸	۱۴
۰/۵۳۴۵	۰/۷۸۶۶	۰/۷۸۶۷	۱	۰/۷۸۸۸	۱۵
۰/۵۳۷۵	۰/۷۹۴۷	۰/۷۹۴۷	۱	۰/۷۹۶۹	۱۶
۰/۵۴۰۱	۰/۸۱۶۸	۰/۸۱۶۸	۱	۰/۸۲۰۴	۱۷
۰/۵۴۲۳	۰/۸۲۵۰	۰/۸۲۵۰	۱	۰/۸۲۸۳	۱۸
۰/۵۴۴۳	۰/۸۳۶۷	۰/۸۳۶۷	۱	۰/۸۳۸۸	۱۹
۰/۵۴۶۱	۰/۸۴۱۴	۰/۸۴۱۴	۱	۰/۸۴۴۷	۲۰
۰/۵۴۷۷	۰/۸۴۶۰	۰/۸۴۷۱	۱	۰/۸۴۸۸	۲۱
۰/۵۴۹۲	۰/۸۴۹۰	۰/۸۴۹۱	۱	۰/۸۶۰۹	۲۲
۰/۵۵۰۵	۰/۸۶۱۶	۰/۸۶۱۶	۱	۰/۸۶۳۲	۲۳
۰/۵۵۱۷	۰/۸۶۴۶	۰/۸۶۴۶	۱	۰/۸۷۲۴	۲۴
۰/۵۵۲۸	۰/۸۷۲۵	۰/۸۷۲۸	۱	۰/۸۷۷۹	۲۵
۰/۵۵۳۸	۰/۸۷۷۰	۰/۸۷۸۴	۱	۰/۸۸۱۸	۲۶
۰/۵۵۴۷	۰/۸۸۰۹	۰/۸۸۱۰	۱	۰/۸۸۵۱	۲۷
۰/۵۵۵۶	۰/۸۸۴۲	۰/۸۸۴۵	۱	۰/۸۸۶۹	۲۸
۰/۵۵۶۳	۰/۸۸۶۸	۰/۸۸۷۱	۱	۰/۸۹۴۴	۲۹
۰/۵۵۷۱	۰/۸۹۱۰	۰/۸۹۱۲	۱	۰/۸۹۵۶	۳۰

جمع بندی و پیشنهادات

۱-۷ جمع بندی

نمایش تُنک سیگنال‌ها طی دهه‌ی اخیر کانون توجه تحقیقات زیادی بوده است. کارآئی این زمینه از پردازش سیگنال در کاربردهای گوناگونی از جمله بهبود و فشرده‌سازی تصاویر، تشخیص الگو، تصویربرداری پزشکی و جداسازی کور منابع مورد بررسی قرار گرفته و حکایت از توانائی بالای آن در این کاربردها دارد. هدف از نمایش تُنک، غلبه بر کمبودها و ناتوانائی‌های تبدیل‌های کلاسیک در پردازش و توصیف مناسب سیگنال‌ها است. برای این منظور، به جای استفاده از یک تبدیل کامل، از یک تبدیل فوق کامل، به امید رسیدن به یک نمایش ساده‌تر استفاده می‌شود. این کار معادل بسط دادن سیگنال روی یک مجموعه شامل تعداد زیادی (معمولاً چندین برابر طول سیگنال) بُردار پایه است که «اتم» خوانده می‌شوند. مجموعه‌ی متشکل از این اتم‌ها یک «دیکشنری» نامیده می‌شود. این اتم‌ها باید نمایانگر ویژگی‌های بارز سیگنال بوده و به تعبیری هم‌خانواده‌ی با آن باشند. در نتیجه، ابتدا باید یک دیکشنری مناسب برای یک کلاس مشخص از سیگنال‌ها پیدا کنیم. این موضوع منجر به پیدایش شاخه‌ای از بحث نمایش تُنک، موسوم به «آموزش دیکشنری» شده است.

در این رساله، هدف اصلی، ارائه الگوریتم‌هایی کارآمدتر برای بازیابی نمایش تُنک و نیز یادگیری دیکشنری، با تأکید بر داده‌های با ابعاد بالا، بود. برای این منظور، بعد از مروری بر نمایش تُنک و یادگیری دیکشنری، الگوریتم‌های پراکسیمال را مرور کردیم. این الگوریتم‌ها، که اساس عمده‌ی روش‌های پیشنهادی این رساله را

تشکیل می‌دهند، به دلیل حجم محاسبات پائین و عملکرد موفق، در بسیاری از کاربردها، از جمله یادگیری ماشین و پردازش سیگنال مورد استفاده قرار گرفته‌اند. سپس، در فصل پنجم، الگوریتم‌هایی برای بازیابی نمایش تُنک معرفی شد. الگوریتم اول، که ما آن را ISP نام نهادیم، مبتنی بر تعمیم الگوریتم نُرم صفر هموار شده، SL0، به حالتی است که تابع تشویق‌کننده‌ی تُنکی می‌تواند ناهموار نیز باشد. در واقع، ساختار کلی الگوریتم ISP همان ساختار الگوریتم SL0 را دنبال می‌کند، با این تفاوت که گام گرادیان کاهش را با یک گام آستانه‌گذاری، که در حالت خاص می‌تواند همان گرادیان کاهش باشد، جایگزین می‌کند. بعلاوه، در مسیر توسعه‌ی این الگوریتم، نکاتی جالب توجه در مورد الگوریتم SL0 نیز روشن شد. همچنین، با استفاده از پیشرفت‌هایی که در زمینه‌ی روش‌های پراکسیمال برای مسائل غیرمحدب در چند سال اخیر صورت گرفته، اثبات همگرایی این الگوریتم ارائه شد. بعنوان یک نوآوری دیگر، حالت مقاوم این الگوریتم مورد بررسی قرار گرفته، و الگوریتمی برای انجام گام تصویر کردن، که مستلزم حل یک مساله‌ی محدب است، ارائه گردید. شبیه‌سازی‌های متعدد نشان از برتری الگوریتم ISP نسبت به الگوریتم‌های رقیب موجود داشتند. در ادامه، الگوریتم جدید دیگری برای حل مساله‌ی کمینه‌سازی نُرم صفر پیشنهاد شد. این الگوریتم، مشابه الگوریتم‌هایی مانند SL0، تابع ناهموار نُرم صفر را با یک تابع خوش‌رفتارتر جایگزین می‌کند. تفاوت عمده‌ای که الگوریتم پیشنهادی، که آن را L0Soft نامیده‌ایم، با الگوریتم‌های مشابه دارد در نحوه‌ی استخراج تقریب خوش‌رفتار تابع نُرم صفر نهفته است. به بیان دقیق‌تر، در الگوریتم پیشنهادی، از یک تعریف خاص از تابع نُرم صفر استفاده می‌شود که نُرم صفر یک بردار را بصورت جمع قدرمطلق علامت درایه‌های آن بیان می‌کند. به این ترتیب، با تقریب تابع قدر مطلق و بازنویسی مساله، به یک مساله‌ی کمینه‌سازی نُرم صفر رسیدیم که با روش‌های پراکسیمال قابل حل بود. شبیه‌سازی‌های انجام شده روی داده‌های مصنوعی و طبیعی حکایت از عملکرد مطلوب این الگوریتم دارند. سپس، الگوریتم IPP را معرفی کردیم. این الگوریتم، سعی در کمینه‌سازی یک تابع ناهموار تشویق‌کننده‌ی تُنکی دارد. برای توسعه‌ی این الگوریتم، روش‌های پراکسیمال به همراه ایده‌های سرعت‌بخشی مورد استفاده قرار گرفت. همچنین، رابطه‌ی این الگوریتم با الگوریتم‌های ISP مورد بررسی قرار گرفته، و عملکرد برتر آن با شبیه‌سازی‌های متعدد نشان داده شد.

در فصل ششم، الگوریتم‌های پیشنهادی برای یادگیری دیکشنری معرفی شدند. ابتدا الگوریتمی برای یادگیری دیکشنری از روی داده‌های با ابعاد بالا ارائه شد. این الگوریتم، که HighDim-DL نامیده شد، با استفاده از روش‌های sketching، بعد داده‌ها را پائین آورده و داده‌های جدید را بین چند پردازنده توزیع می‌کند. هر پردازنده

با داشتن بخش کوچکی از کل داده‌ها قادر خواهد بود یک دیکشنری محلی آموزش بدهد. بعلاوه، برای رسیدن به یک دیکشنری سراسری، پردازنده‌ها با یک مرکز پردازش در تماس بوده، و تخمین‌های خود از دیکشنری‌های محلی را به آن ارسال می‌کنند. برای کاهش حجم اطلاعات مبادله شده، یک ساختار t_{nk} روی دیکشنری فرض شد. شبیه‌سازی‌های انجام شده عملکرد امیدبخش این الگوریتم را نشان داد. در ادامه، الگوریتم‌هایی برای یادگیری دیکشنری‌های با همبستگی متقابل پائین معرفی شد. هدف از این الگوریتم‌ها، یادگیری دیکشنری‌های با همبستگی متقابل کم است. چنین دیکشنری‌هایی خواص بازیابی بسیار خوبی دارند. در الگوریتم‌هایی که معرفی شد، برخلاف روش‌های موجود، مستقیماً از تابع همبستگی متقابل، که تابعی ناهموار است، برای تشویق کردن همبستگی کم حین یادگیری دیکشنری استفاده شد. ایده‌ی اصلی روش‌های پیشنهادی برای حل مسائل مربوطه، فرمول‌بندی آنها به شیوه‌ای بود که بتوان از مفاهیم پراکسیمال و خواص عملگر پراکسیمال استفاده کرد. با توسعه‌ای این الگوریتم‌ها، که یکی برای حالت مقید (یعنی وقتی حد بالایی برای همبستگی متقابل دیکشنری مطلوب است)، و دیگری برای حالت نامقید (یعنی وقتی تابع همبستگی متقابل با ضریبی با تابع خطای نمایش t_{nk} جمع می‌شود)، نشان داده شد که نسبت به الگوریتم‌های موجود پیچیدگی محاسباتی کمتری داشته، و در نتیجه برای داده‌های با ابعاد بالا مناسب‌تر هستند. بعلاوه، این الگوریتم‌ها توانایی بهتری در کم کردن همبستگی متقابل دیکشنری‌های یادگرفته شده دارند. در واقع، تا آنجا که ما اطلاع داریم، این الگوریتم‌ها اولین روش‌هایی هستند که بصورت مستقیم تابع همبستگی متقابل را وارد مسأله‌ی یادگیری دیکشنری می‌کنند. در ادامه‌ی این فصل، ایده‌ای مشابه برای طراحی ماتریس‌های با همبستگی متقابل کم ارائه شد. این ماتریس‌ها کاربرد زیادی در پردازش سیگنال، بعنوان مثال برای طراحی ماتریس حسگر در حسگری فشرده، و نیز مخابرات، برای انتقال داده و کم کردن احتمال خطا دارند. شبیه‌سازی‌های انجام شده نشان داد که روش‌های پیشنهادی نسبت به الگوریتم‌های موجود به همبستگی متقابل کمتری دست می‌یابند.

۷-۲ پیشنهاداتی برای کارهای آینده

در این قسمت، چند جهت برای کارهای آینده بیان می‌کنیم.

- انتخاب ماتریس sketching مناسب برای الگوریتم HighDim-DL از اهمیت بالایی برخوردار است. این ماتریس علاوه بر این که باید محاسبات سریعی داشته باشد، باید قادر باشد تا اطلاعات موجود در داده‌های اصلی را تا حد ممکن حفظ کند. در نتیجه، تحقیق روی طراحی چنین ماتریسی، بعنوان مثال از طریق

فرمول‌بندی یک مسأله‌ی یادگیری برای آن، می‌تواند یک ایده‌ی کاربردی باشد. بعنوان مثال، می‌توان از الگوریتم معرفی شده در [۷۱] الهام گرفت. علاوه بر این، تلاش برای اثبات همگرایی این الگوریتم و نیز نحوه‌ی مقداردهی پارامترهای ρ_1 و ρ_2 ، که مربوط به روش ADMM است، نیز بعنوان کارهای آینده مطرح است.

۲. با توجه به این که مسأله‌ی یادگیری دیکشنری، یک مسأله‌ی غیرمحدب است، می‌توان از روش‌های تقریب محدب^۱ [۹۷، ۷۲] برای حل آن استفاده کرد. این روش‌ها اخیراً مورد توجه زیادی قرار گرفته و عملکرد مطلوبی از خود نشان داده‌اند. در مورد مسأله‌ی یادگیری دیکشنری، استفاده از تقریب محدب علاوه بر این که ممکن است به جواب‌های بهتری منتهی شود، احتمالاً حجم محاسبات را نیز کاهش خواهد داد. بعنوان یکی از روش‌های نمونه و البته پرکاربرد برای تقریب محدب، که در مورد یادگیری دیکشنری نیز قابل اعمال است، می‌توان به روش کمینه‌سازی محدب متوالی اشاره کرد^۲ (SCP) [۲۱]. برای مرور مختصر این روش، مسأله‌ی غیرمحدب زیر را در نظر بگیرید:

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \quad (1-7)$$

سپس، ایده‌ی روش SCP برای حل تقریبی مسأله‌ی فوق عبارت است از حل پیاپی مسائل محدبی به فرم زیر

$$\mathbf{x}^{k+1} = \underset{\mathbf{x} \in \mathcal{T}}{\operatorname{argmin}} \tilde{f}(\mathbf{x}) \quad \text{s.t.} \quad \tilde{g}_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \quad (2-7)$$

که در آن، $\mathcal{T} \triangleq \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}_k\|_2 \leq \delta_k\}$ ناحیه‌ی قابل اعتماد^۳ نامیده می‌شود [۲۱] و \tilde{f} و \tilde{g}_i به ترتیب، تقریب‌هایی محدب برای توابع f و g_i بوده که روی مجموعه‌ی \mathcal{T} اعتبار دارند. با حل دنباله‌ی مسائل محدب فوق، می‌توان جوابی تقریبی برای مسأله‌ی اولیه‌ی غیرمحدب (۱-۷) بدست آورد. در مورد مسأله‌ی یادگیری دیکشنری، برای بدست آوردن تقریبی محدب برای تابع هدف، می‌توان از روش پیشنهاد شده در [۱۰۷] استفاده کرد.

۳. در مورد مسأله‌ی طراحی فریم‌های با همبستگی متقابل کم، که آن هم غیرمحدب است، رسیدن به جوابی با کیفیت بالا، یعنی با همبستگی متقابل تا حد ممکن پائین، از اهمیت زیادی برخوردار است. در واقع،

^۱Convex approximation/relaxation

^۲Sequential convex programming

^۳Trust region

تلاش‌های زیادی که در سالیان گذشته برای طراحی فریم‌های ناهمبسته صورت گرفته، با توجه به دشواری حل مسأله‌ی متناظر، همگی در جهت بهبود کیفیت و ارائه‌ی روش‌هایی برای غلبه بر حداقل‌کننده‌های محلی و نزدیک شدن به حداقل‌کننده سراسری هستند. ایده‌ای که برای حل سراسری مسأله‌ی کمینه‌سازی همبستگی متقابل می‌توان استفاده کرد، استفاده از روش‌های کمینه‌سازی سراسری^۱ [۶۴، ۵۸] است. به طور خاص، روش شاخه و حد^۲ [۶] رهیافتی کارآمد برای این منظور خواهد بود. این روش اگر چه حجم محاسبات بالایی دارد، اما می‌تواند دید خوبی در مورد جواب بهینه‌ی مسأله‌ی کمینه‌سازی همبستگی متقابل بدست بدهد.

۴. ارائه‌ی الگوریتمی برای آموزش دیکشنری که در آن اتم‌ها، یا همان ویژگی‌های برجسته، یکی یکی از داده‌های آموزشی استخراج می‌شوند. چنین الگوریتمی دیدی شهودی‌تر نسبت به سایر الگوریتم‌ها داشته و در آن لازم نیست تعداد اتم‌ها از ابتدا معلوم فرض شود. الگوریتم‌های به‌روزرسانی اتم به اتم، که در [۱۰۸] معرفی شده‌اند، می‌توانند نقطه‌ی شروعی برای این منظور باشد. در واقع، یک ایده‌ی اولیه برای این کار می‌تواند به اینصورت باشد که با تعریف یک ماتریس باقی‌مانده و مقداردهی آن با ماتریس داده‌های آموزشی به فرم

$$\mathbf{R}_k = \mathbf{Y} - \mathbf{D}\mathbf{X}, \text{ مراحل زیر برای استخراج متوالی اتم‌ها انجام شود (} k \geq 0, \mathbf{D} = [], \mathbf{X} = [])$$

$$\left\{ \begin{array}{l} \{\mathbf{d}_k, \mathbf{x}_{[k]}\} = \operatorname{argmin}_{\mathbf{d}, \mathbf{x}} \|\mathbf{R}_k - \mathbf{d}\mathbf{x}^T\|_F + \lambda\|\mathbf{x}\|_1 \\ \mathbf{D} = [\mathbf{D}, \mathbf{d}_k] \\ \mathbf{X} = [\mathbf{X}, \mathbf{x}_{[k]}] \\ \mathbf{D} = \operatorname{argmin}_{\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \\ k \leftarrow k + 1 \\ \mathbf{R}_k = \mathbf{Y} - \mathbf{D}\mathbf{X} \end{array} \right. \quad (3-7)$$

در رابطه‌ی فوق، \mathbf{d}_k عبارت است از k امین اتم استخراج شده و $\mathbf{x}_{[k]}$ سطر متناظر با آن در ماتریس ضرایب، یعنی \mathbf{X} ، را نشان می‌دهد. تکرارهای فوق تا زمانی که نُرم ماتریس خطا به کمتر از یک سطح مشخص برسد، انجام می‌شود. با حل مسأله‌ی استخراج اتم و بردار ضرایب آن در (۳-۷)، در حقیقت بصورت تکراری ماتریس‌هایی رُتبه یک به ماتریس خطای نمایش برآزش می‌شود. حل این مسأله با الگوریتم پیشنهاد شده در [۱۰۸] به شیوه‌ای کارآ قابل انجام است. نکته‌ی جالب اینک، ساختار الگوریتم فرمول‌بندی شده در (۳-۷) بسیار شبیه ساختار الگوریتم OMP، برای بازیابی نمایش تُنک است، که در فصل دوم مرور شد.

^۱Global optimization^۲Branch and bound method

۵. در انتها، پیاده‌سازی الگوریتم‌های پیشنهادی برای بازیابی نمایش تُنگ با استفاده از ساختار شبکه‌های عصبی مصنوعی [۶۵] می‌تواند جهت کاری مناسبی برای آینده باشد. در واقع، می‌توان تکرارهای الگوریتم‌های پیشنهادی را بصورت لایه‌های مختلف یک شبکه‌ی عصبی در نظر گرفته، و با استفاده از تعدادی داده‌ی آموزشی پارامترهای آن را آموزش داد [۹۰]. این کار علاوه بر این که حجم محاسبات پردازش‌های بعدی، یعنی برای داده‌های جدید، را کاهش می‌دهد، می‌تواند به عملکرد احتمالاً خیلی بهتری نسبت به حالت عادی منجر شود. دلیل عملکرد بهتر، آموزش دادن پارامترهای الگوریتم‌ها، از جمله توابع آستانه‌گذاری، و وفق دادن آن به ساختار داده‌ها است.

مراجع

- [1] V. Abolghasemi, S. Ferdowsi, and S. Sanei, “Fast and incoherent dictionary learning algorithms with application to fMRI,” *Signal, Image and Video Processing (Springer)*, vol. 9, no. 1, pp. 147–158, 2015.
- [2] A. Agarwal, A. Anandkumar, P. Jain, and P. Netrapalli, “Learning sparsely used overcomplete dictionaries via alternating minimization,” *SIAM J. Optim.*, vol. 26, no. 4, pp. 2775–2799, 2016.
- [3] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [4] A. Ali Amini, M. Babaie-Zadeh, and C. Jutten, “Fast sparse decomposition by iterative detection-estimation,” [Available Online] <https://arxiv.org/pdf/1009.3890>.
- [5] A. Ali Amini, M. Babaie-Zadeh, and C. Jutten, “A fast method for sparse component analysis based on iterative detection-estimation,” in *American Institute of Physics (AIP) Conference Proceeding (MaxEnt2006)*, 2006, vol. 872, pp. 123–130.
- [6] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, “ α BB: A global optimization method for general constrained nonconvex problems,” *Journal of Global Optimization*, vol. 7, no. 4, pp. 337–363, Dec 1995.
- [7] S. Arora, R. Ge, T. Ma, and A. Moitra, “Simple, efficient, and neural algorithms for sparse coding,” in *Conference On Learning Theory (COLT)*, 2015.
- [8] H. Attouch and J. Bolte, “On the convergence of the proximal algorithm for nonsmooth functions involving analytic features,” *Math. Program.*, vol. 116, pp. 5–16, 2009.
- [9] M. Azghani, P. Kosmas, and F. Marvasti, “Microwave medical imaging based on sparsity and an iterative method with adaptive thresholding,” *IEEE Trans. on Medical Imaging*, vol. 34, no. 2, pp. 357–365, 2015.
- [10] W. U. Bajwa, R. Calderbank, and D. G. Mixon, “Two are better than one: Fundamental parameters of frame coherence,” *Appl. Comput. Harmon. Anal.*, vol. 33, no. 1, pp. 58–78, 2012.

-
- [11] C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning for sparse coding: Algorithms and convergence analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, to be published.
- [12] C. Bao, Y. Quan, and H. Ji, "A convergent incoherent dictionary learning algorithm for sparse coding," in *Computer Vision – ECCV*, vol. 8694 of *Lecture Notes in Computer Science*, pp. 302–316. Springer, 2014.
- [13] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Proc. Magazine*, vol. 24, no. 4, pp. 118–121, 2007.
- [14] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Trans. on Signal Proc.*, vol. 61, no. 8, pp. 2055–2065, 2013.
- [15] J. Becker, S. Bobin and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imaging Sci.*, vol. 4, no. 1, pp. 1–39, 2011.
- [16] J. J. Benedetto and M. Fickus, "Finite normalized tight frames," *Adv. Comput. Math.*, vol. 18, pp. 357–385, 2003.
- [17] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, 1987.
- [18] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.
- [19] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for non-convex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.
- [20] L. Bottou, *Online Algorithms and Stochastic Approximations*, On-line learning in neural networks. Cambridge University Press, 1998.
- [21] S. Boyd, "Sequential convex programming," 2008, Lecture Notes, Stanford University.
- [22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [23] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [24] O. Bryt and M. Elad, "Compression of facial images using the K-SVD algorithm," *Journal of Visual Communication and Image Representation*, vol. 19, no. 4, pp. 270–283, 2008.
- [25] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [26] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [27] P. G. Casazza, M. Fickus, and D. G. Mixon, "Auto-tuning unit norm frames," *Appl. Comput. Harmon. Anal.*, vol. 32, no. 1, pp. 1–15, 2012.

- [28] V. Cevher, "Learning with compressible priors," in *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, 2009.
- [29] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 32–43, Sept. 2014.
- [30] R. Chartrand and B. Wohlberg, "A nonconvex ADMM algorithm for group sparsity with sparse groups," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6009–6013.
- [31] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008.
- [32] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, vol. 155, no. 1–2, 2016.
- [33] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2015.
- [34] S. S. Chen, D. D. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, pp. 129–159, 2001.
- [35] W. Chen, M. R. D. Rodrigues, and I. J. Wassell, "Projection design for statistical compressive sensing: A tight frame based approach," *IEEE Trans. on Signal Proc.*, vol. 61, no. 8, pp. 2016–2029, 2013.
- [36] P. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212, 2011.
- [37] P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation*, Elsevier, 2010.
- [38] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: Packings in grassmannian spaces," *Exper. Math.*, vol. 5, no. 2, pp. 139–159, 1996.
- [39] A. Daneshmand, G. Scutari, and F. Facchinei, "Distributed dictionary learning," in *50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 1001–1005.
- [40] A. Daneshmand, Y. Sun, G. Scutari, and F. Facchinei, "D2L: Decentralized dictionary learning over dynamic networks," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4084–4088.
- [41] I. Daubechies, M. Defrise, and C. De-Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [42] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.
- [43] D. L. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

- [44] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proc. Nat. Aca. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [45] D. L. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Info. Theory*, vol. 52, no. 1, pp. 6–18, 2006.
- [46] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.
- [47] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, 2009.
- [48] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l_1 -ball for learning in high dimensions," in *International Conference on Machine Learning (ICML)*, 2008.
- [49] B. Dumitrescu and P. Irofti, *Dictionary Learning Algorithms and Applications*, Springer International Publishing, 2018.
- [50] A. Eftekhari, M. Babaie-Zadeh, C. Jutten, and H. Abrishami-Moghaddam, "Robust-SL0 for stable sparse representation in noisy settings," in *Proceedings of ICASSP2009*, 2009, pp. 3433–3436.
- [51] M. Elad, "Optimized projections for compressed sensing," *IEEE Trans. on S*, vol. 55, no. 12, pp. 5695–5702, 2007.
- [52] M. Elad, *Sparse and Redundant Representations*, Springer, 2010.
- [53] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, "A wide-angle view at iterated shrinkage algorithms," in *Proc. SPIE (Wavelet XII)*, 2007, pp. 26–29.
- [54] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proceedings of IEEE ICASSP*, 1999, vol. 5.
- [55] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of American Statistical Association*, vol. 96, pp. 1348–1360, 2001.
- [56] M. Fickus, D. G. Mixon, and J. C. Tremain, "Steiner equiangular tight frames," *Linear Algebra Appl.*, vol. 436, no. 5, pp. 1014–1027, 2012.
- [57] M. A.T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.
- [58] C. A. Floudas, *Deterministic Global Optimization: Theory, Methods and Applications*, Non-convex Optimization and Its Applications. Springer, 2000.
- [59] S. Foucart and M.-J. Lai, "Sparsest solutions of underdetermined linear systems via ℓ_q -minimization for $0 < q \leq 1$," *Appl. Comput. Harmonic Anal.*, vol. 26, no. 3, pp. 395–407, 2009.
- [60] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Applied and Numerical Harmonic Analysis. Birkhäuser Basel, 2013.

- [61] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and dc programming," *IEEE Trans. on Signal Process.*, vol. 57, no. 12, pp. 4686–4698, 2009.
- [62] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Springer, 1992.
- [63] F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, "Sparse signal recovery using iterative proximal projection," *IEEE Trans. on Signal Proc.*, 2018, Accepted.
- [64] A. M. Gleixner, T. Berthold, B. Müller, and S. Weltge, "Three enhancements for optimization-based bound tightening," *Journal of Global Optimization*, vol. 67, no. 4, pp. 731–757, 2017.
- [65] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
- [66] M. Grant and S. Boyd, "Matlab software for disciplined convex programming," [Online], 2011, Available: <http://cvxr.com/>.
- [67] R. Gribonval and K. Schnass, "Dictionary identification sparse matrix-factorization via ℓ_1 - minimization," *IEEE Trans. Info. Theory*, vol. 56, no. 7, pp. 3523–3539, 2010.
- [68] H. H. Sohrab, *Basic Real Analysis*, Birkhäuser Basel, 2014.
- [69] W. W. Hager, "Updating the inverse of a matrix," *SIAM Review*, vol. 31, no. 2, pp. 221–239, 1989.
- [70] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.
- [71] C. Hegde, A. C. Sankaranarayanan, W. Yin, and R. G. Baraniuk, "NuMax: A convex approach for learning near-isometric linear embeddings," *IEEE Trans. On Signal Processing*, vol. 63, no. 22, pp. 6109–6121, 2015.
- [72] K. Huang and N. D. Sidiropoulos, "Consensus-ADMM for general quadratically constrained quadratic programming," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5297–5310, 2016.
- [73] N. Hurley and S. Rickard, "Comparing measures of sparsity," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4723–4741, 2009.
- [74] J. Kovacevic and A. Chebira, "Life beyond bases: The advent of frames (Part I)," *IEEE Signal Proc. Magazine*, vol. 24, no. 4, pp. 86–104, 2007.
- [75] J. Kovacevic and A. Chebira, "Life beyond bases: The advent of frames (Part II)," *IEEE Signal Proc. Magazine*, vol. 24, no. 5, pp. 115–125, 2007.
- [76] K. Labusch, E. Barth, and T. Martinetz, "Robust and fast learning of sparse codes with stochastic gradient descent," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 1048–1060, 2011.
- [77] R. Lai and S. Osher, "A splitting method for orthogonality constrained problems," *Journal of Scientific Computing*, vol. 58, no. 2, pp. 431–449, 2014.

- [78] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai, "On projection matrix optimization for compressive sensing systems," *IEEE Trans. on Signal Proc.*, vol. 61, no. 11, pp. 2887–2898, 2013.
- [79] J. Liang, M. Zhang, X. Zeng, and G. Yu, "Distributed dictionary learning for sparse representation in sensor networks," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2528–2541, 2014.
- [80] Q. Liu, X. Shen, and Y. Gu, "Linearized ADMM for non-convex non-smooth optimization with convergence analysis," *arXiv preprint*, 2017.
- [81] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *IEEE Signal Proc. Magazine*, vol. 25, no. 2, pp. 72–82, 2008.
- [82] B. Maill e, D. Barchiesi, and M. D. Plumbley, "INK-SVD: Learning incoherent dictionaries for sparse representations," in *IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2012, pp. 3573–3576.
- [83] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.
- [84] M. Malek-Mohammadi, A. Koochakzadeh, M. Babaie-Zadeh, M. Jansson, and C. R. Rojas, "Successive concave sparsity approximation for compressed sensing," *IEEE Trans. on Signal Proc.*, vol. 64, no. 21, pp. 5657–5671, 2016.
- [85] A. Maleki and D. L. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 330–341, 2010.
- [86] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Proc.*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [87] F. Marvasti, A. Amini, F. Haddadi, M. Soltanolkotabi, B.H. Khalaj, A. Aldroubi, S. Sanei, and J. Chambers, "A unified approach to sparse signal processing," *EURASIP Journal on Advances in Signal Processing*, vol. 44, 2012.
- [88] F. Marvasti, M. Azghani, P. Imani, P. Pakrouh, S.J. Heydari, A. Golmohammadi, A. Kazerouni, and M.M. Khalili, "Sparse signal processing using iterative method with adaptive thresholding (IMAT)," *19th International Conference on Telecommunications (ICT'12)*, 2012.
- [89] F. Marvasti and M. Boloursaz, "Wideband analog to digital conversion by random or level crossing sampling," Aug. 2017.
- [90] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.
- [91] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics (SIAM), 2000.
- [92] H. Mohimani, M. Babaie-Zadeh, and Ch. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^o norm," *IEEE Trans. on Signal Processing*, vol. 57, pp. 289–301, 2009.

- [93] D. Needell and J. A. Tropp, “Cosamp: Iterative signal recovery from in-complete and inaccurate samples,” *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.
- [94] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 1999.
- [95] P. Ochs, Y. Chen, T. Brox, and T. Pock, “iPiano: Inertial proximal algorithm for nonconvex optimization,” *SIAM J. Imag. Sci.*, vol. 7, no. 2, pp. 388–1419, 2014.
- [96] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.
- [97] J. Park and S. Boyd, “General heuristics for nonconvex quadratically constrained quadratic programming,” *arXiv preprint*, 2017.
- [98] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” in *In Proc. Asilomar Conf. Signal Syst. Comput.*, 1993.
- [99] K. B. Petersen and M.S. Pedersen, “The matrix cookbook,” 2008.
- [100] R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [101] R. Rubinstein, M. Zibulevsky, and M. Elad, “Double sparsity: Learning sparse dictionaries for sparse signal approximation,” *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1553–1564, 2010.
- [102] C. Rusu, “Design of incoherent frames via convex optimization,” *IEEE Signal Proc. Letters*, vol. 20, no. 7, pp. 673–676, 2013.
- [103] C. Rusu and N. González-Prelcic, “Designing incoherent frames through convex techniques for optimized compressed sensing,” *IEEE Trans. on Signal Proc.*, vol. 64, no. 9, pp. 2334–2344, 2016.
- [104] M. Sadeghi and M. Babaie-Zadeh, “Iterative sparsification-projection: Fast and robust sparse signal approximation,” *IEEE Trans. on Signal Proc.*, vol. 64, no. 21, pp. 5536–5548, 2016.
- [105] M. Sadeghi and M. Babaie-Zadeh, “Incoherent unit-norm frame design via an alternating minimization penalty method,” *IEEE Signal Proc. Letters*, vol. 24, no. 1, pp. 32–36, 2017.
- [106] M. Sadeghi and M. Babaie-Zadeh, “Learning low-coherence dictionaries for sparse signal approximation,” *Signal Processing*, 2018 (submitted).
- [107] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, “Dictionary learning for sparse representation: A novel approach,” *IEEE Signal Proc. Letters*, vol. 20, no. 12, pp. 1195–1198, 2013.
- [108] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, “Learning over-complete dictionaries based on atom-by-atom updating,” *IEEE Trans. on Signal Proc.*, vol. 62, no. 4, pp. 883–891, 2014.
- [109] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, “Regularized low-coherence overcomplete dictionary learning for sparse signal decomposition,” in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, 2016.

- [110] M. Sadeghi, F. Ghayem, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund, and C. Jutten, “L0Soft: ℓ_0 minimization via soft thresholding,” *Signal Processing*, 2018 (submitted).
- [111] C. D. Sigg, T. Dikk, and J. M. Buhmann, “Learning dictionaries with bounded self-coherence,” *IEEE Signal Proc. Letters*, vol. 19, no. 12, pp. 861–864, 2012.
- [112] K. Skretting and K. Engan, “Recursive least squares dictionary learning algorithm,” *IEEE Trans. on Signal Processing*, vol. 58, pp. 2121 – 2130, 2010.
- [113] K. Slavakis, G.B. Giannakis, and G. Mateos, “Modeling and optimization for big data analytics,” *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 18–31, Sept. 2014.
- [114] T. Strohmer and Heath R. W., “Grassmannian frames with applications to coding and communication,” *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.
- [115] T. Strohmer and Heath R. W., “Grassmannian frames with applications to coding and communication,” *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.
- [116] A. Taimori and F. Marvasti, “Measurement-adaptive sparse image sampling and recovery,” *IEEE Trans. Computational Imaging*, vol. 4, no. 3, pp. 311–325, 2018.
- [117] A. M. Tillmann and M. E. Pfetsch, “The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1248–1259, 2014.
- [118] I. Tasic and P. Frossard, “Dictionary learning,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.
- [119] J. A. Tropp, “An introduction to matrix concentration inequalities,” *Foundations and Trends in Machine Learning*, vol. 8, no. 1–2, pp. 1–230, 2015.
- [120] J. A. Tropp, I. S. Dhillon, R. W. Heath, and T. Strohmer, “Designing structured tight frames via an alternating projection method,” *IEEE Trans. Info. Theory*, vol. 51, no. 1, pp. 188–209, 2005.
- [121] J. A. Tropp and S. J. Wright, “Computational methods for sparse solution of linear inverse problems,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.
- [122] E. V. Tsiliogianni, L. P. Kondi, and A. K. Katsaggelos, “Construction of incoherent unit norm tight frames with application to compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2319–2330, 2014.
- [123] R. Tyrrell Rockafellar and R. J-B Wets, *Variational Analysis*, Springer, 1998.
- [124] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborova, “Adaptive damping and mean removal for the generalized approximate message passing algorithm,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 2021–2025.
- [125] J. P. Vila and P. Schniter, “Expectation-maximization gaussian-mixture approximate message passing,” *IEEE Trans. on Signal Proc.*, vol. 61, no. 19, pp. 4658–4672, 2013.
- [126] S. Waldron, “On the construction of equiangular frames from graphs,” *Linear Algebra Appl.*, vol. 431, no. 11, pp. 2228–2242, 2009.

-
- [127] J. Wang, S. Kwon, and B. Shim, “Generalized orthogonal matching pursuit,” *IEEE Trans. on Signal Proc.*, vol. 60, no. 12, pp. 6202–6216, 2012.
- [128] Y. Wang, W. Yin, and J. Zeng, “Global convergence of ADMM in nonconvex nonsmooth optimization,” Tech. Rep., UCLA CAM Report 15-62, 2017.
- [129] D. P. Woodruff, “Sketching as a tool for numerical linear algebra,” *Foundations and Trends in Theoretical Computer Science*, vol. 10, no. 1-2, pp. 1–157, Oct. 2014.
- [130] S. Wu and B. Yu, “Local identifiability of l_1 -minimization dictionary learning: a sufficient and almost necessary condition,” arXiv:1505.04363, 2015.
- [131] P. Xia, S. Zhou, and G. B. Giannakis, “Achieving the Welch bound with difference sets,” *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1900–1907, 2005.
- [132] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.
- [133] Y. Xu and W. Yin, “A globally convergent algorithm for nonconvex optimization based on block coordinate update,” *Journal of Scientific Computing*, pp. 1–35, 2017.
- [134] M. Yaghoobi, T. Blumensath, and M. E. Davies, “Dictionary learning for sparse approximations with the majorization method,” *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2178 – 2191, 2009.
- [135] L. Yang, T. K. Pong, and X. Chen, “Alternating direction method of multipliers for nonconvex background/foreground extraction,” *SIAM Journal on Imaging Sciences*, vol. 10, no. 1, pp. 74–110, 2017.
- [136] W. H. Young, “On classes of summable functions and their fourier series,” *Proceedings of the Royal Society A*, vol. 87, no. 594, pp. 225–229, 1912.

ABSTRACT

Sparse representation has attracted much attention over the past decade. The main idea is that natural signals have information contents much lower than their ambient dimensions, and as such, they can be represented by using only a few basis signals (also called atoms). In other words, a natural signal of length n , which in general needs n atoms to be represented, can be written as a linear combination of s atoms, where $s \ll n$. To achieve a sparser representation, i.e., a smaller s , the number of atoms is chosen much larger than n . In this way, there are more choices to represent a signal and we can choose the sparsest possible combination. The set of atoms is called a *dictionary*. Here, two questions arise. One is basically how to choose the collection of atoms, and the other one is how to select the best set of atoms to represent a given signal. The best answer for the first question is to *learn* a set of atoms from some training signals. This is called *dictionary learning*, which has gained a lot of interest. The second question is answered by a large number of existing sparse recovery (or sparse coding) algorithms. In this thesis, we are going to study the sparse representation and especially the dictionary learning problem with more details. Moreover, we will also focus on high-dimensional data. To this end, new algorithms for both dictionary learning and sparse coding are proposed. The main features of the proposed algorithms are their simple structures and high quality performance. These algorithms are based on *proximal algorithms*. Proximal algorithms are first order tools for solving a broad range of optimization problems in signal processing and machine learning that have low computational complexity but a very good performance. This is a reason why our proposed algorithms suit high-dimensional data. Specifically, inspired by the smoothed ℓ_1 norm (SLO) algorithm, new algorithms with improved performance and more robustness to noise are proposed. Convergence analysis is provided for these algorithms, and our extensive simulations confirm their superior performance over existing methods. In addition, new algorithms are proposed for learning low-mutual coherence dictionaries, which in contrast to previous methods, directly use the mutual coherence function. Besides, a new algorithm is proposed to learn dictionary from high-dimensional data. This algorithm is based on reducing the dimension of data and distributing them over multiple processors. Simulation results on synthetic as well as real data demonstrate the promising performance of the proposed algorithms.

KEYWORDS

1. Sparse Representation.
2. Dictionary Learning.
3. Proximal Algorithms.
4. High-dimensional Data.



SHARIF UNIVERSITY OF TECHNOLOGY
ELECTRICAL ENGINEERING DEPARTMENT

Ph.D. Dissertation

Title:

**Sparse Recovery and Dictionary Learning
based on Proximal Methods in
Optimization**

by:

Mostafa Sadeghi

Supervisor:

Prof. Massoud Babaie-Zadeh

April 2018