



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده ریاضی و علوم کامپیوتر

پایان نامه کارشناسی ارشد  
گرایش سیستم‌های کامپیوتری

کاهش بعد داده‌های بزرگ مقیاس با استفاده از نگاشت  
تصادفی

نگارش  
سیامک دهد

استاد راهنما  
دکتر عادل محمدپور

استاد مشاور  
دکتر هادی زارع

دی ۱۳۹۷

## صفحه فرم ارزیابی و تصویب پایان نامه - فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تأیید و تصویب پایان نامه موسوم به فرم کمیته دفاع- موجود در پرونده آموزشی- را قرار دهید.

### نکات مهم:

- نگارش پایان نامه/رساله باید به **زبان فارسی** و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت **پشت و رو(دورو)** بلامانع است و انجام آن توصیه می شود.



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

## تعهدنامه اصالت اثر

تاریخ: دی ۱۳۹۷

اینجانب **سیامک دهبید** متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

**سیامک دهبید**

امضا

# سپاسگزاری

با تشکر از استاد گرامی دکتر محمدپور بابت همراهی و صبر ایشان

سیامک دهب  
دی ۱۳۹۷

## چکیده

روش تصویر تصادفی برای کاهش بعد داده‌های بزرگ مقیاس مزایای متعددی نسبت به روش‌های دیگر کاهش بعد دارد. در این پایان‌نامه این روش برای داده‌های بزرگ مقیاس با دیگر روش‌های کاهش بعد مقایسه شده است. همچنین توانایی این روش برای داده‌های با توزیع پایدار غیر نرمال با دیگر روش‌های کاهش بعد مقایسه شده است.

## واژه‌های کلیدی:

کاهش بعد، تصویر تصادفی، توزیع پایدار، داده‌های بزرگ مقیاس

## فهرست مطالب

صفحه

عنوان

۱	۱ مقدمه
۳	۲ کاهش بعد و داده‌های بزرگ مقیاس
۴	۱-۲ داده‌های حجیم
۴	۱-۱-۲ داده‌های حجیم وب
۶	۲-۱-۲ جریان‌های داده‌ی حجیم
۶	۲-۲ چالش‌های نمونه‌گیری از داده‌های حجیم
۷	۱-۲-۲ مزایای نمونه‌گیری تصادفی مختصات
۷	۲-۲-۲ معایب نمونه‌گیری تصادفی مختصات
۸	۳-۲ تصویر تصادفی پایدار
۹	۴-۲ کاربردها
۹	۱-۴-۲ کاوش قوانین وابستگی
۹	۲-۴-۲ وابستگی جفتی همه (فاصله‌ها)
۱۰	۳-۴-۲ تخمین فاصله‌ها به طور آنلاین
۱۰	۴-۴-۲ بهینه‌سازی درخواست از پایگاه داده
۱۱	۵-۴-۲ جستجوی نزدیکترین همسایه از مرتبه‌ی زیر خطی
۱۲	۳ تصویر تصادفی پایدار
۱۴	منابع و مراجع
۱۹	پیوست
۲۰	واژه‌نامه‌ی فارسی به انگلیسی
۲۲	واژه‌نامه‌ی انگلیسی به فارسی

## فهرست اشکال

صفحه

شکل

- ۱-۲ تصویر تصادفی پایدار  $\mathbf{B} = \mathbf{A} \times \mathbf{R}$  ، ماتریس اولیه داده‌ها است. . . . . ۸
- ۱-۳ روش تصویر تصافی پایدار ماتریس داده‌ی  $\mathbf{A} \in \mathbb{R}^{n \times D}$  را در یک ماتریس تصادفی  $\mathbf{R} \in \mathbb{R}^{D \times k}$  ضرب می‌کند تا ماتریس تصویر شده‌ی  $\mathbf{B} = \mathbf{AR} \in \mathbb{R}^{n \times k}$  حاصل شود. . . . . ۱۳

## فهرست جداول

صفحه

جدول

۴	۱-۲ تعداد بازدید صفحات برای کلمات با بازخورد بالا و کلمات با بازخورد نادر . . . . .
	۲-۲ با افزایش تعداد عبارات در درخواست، باید فرکانس‌های جفت شده کاهش پیدا کنند. ولی
۵	تخمین‌های بیان شده توسط موتورهای جستجو گاهی این موضوع تثبیت شده را نقض می‌کنند.
	۳-۲ بازدید صفحات گزارش شده توسط Google برای چهار کلمه و وابستگی‌های دو، سه و چهارتایی
۱۰	آنها . . . . .



## فهرست نمادها

نماد	مفهوم
$\mathbb{R}^n$	فضای اقلیدسی با بعد $n$
$\mathbb{S}^n$	کره $n$ یکه بعدی
$M^m$	خمینه $m$ -بعدی $M$
$\mathfrak{X}(M)$	جبر میدان‌های برداری هموار روی $M$
$\mathfrak{X}^1(M)$	مجموعه میدان‌های برداری هموار $1$ یکه روی $(M, g)$
$\Omega^p(M)$	مجموعه $p$ -فرمی‌های روی خمینه $M$
$Q$	اپراتور ریچی
$\mathcal{R}$	تانسور انحنای ریمان
$ric$	تانسور ریچی
$L$	مشتق لی
$\Phi$	۲-فرم اساسی خمینه تماسی
$\nabla$	التصاق لوی-چویتای
$\Delta$	لاپلاسین ناهموار
$\nabla^*$	عملگر خودالحاق صوری القا شده از التصاق لوی-چویتای
$g_s$	متر ساساکی
$\nabla$	التصاق لوی-چویتای وابسته به متر ساساکی
$\Delta$	عملگر لاپلاس-بلترامی روی $p$ -فرم‌ها

# فصل اول

## مقدمه

عمومیت پیدا کردن داده‌های حجیم مانند داده‌های حجیم تحت وب و جریان‌های داده بزرگ در کاربردهای جدید، موجب به وجود آمدن فرصت‌های و چالش‌هایی برای مهندسين و دانشمندان شده است. [۲۶] برای مثال، زمانی که ماتریس داده  $A \in \mathbb{R}^{n \times D}$  ابعادی در حد وب داشته باشد، عملیات ساده‌ای مانند محاسبه  $AA^T$  سخت می‌شود. برای ارائه و نگهداری داده‌های حجیم در حافظه‌ای کوچک و برای استخراج اطلاعات آماری اصلی از مجموعه‌ای از بیانی محدود، روش‌های گوناگونی نمونه‌برداری توسعه یافته است. به طور کلی روش تصویر تصادفی پایدار<sup>۱</sup> برای داده‌های با دم سنگین خیلی خوب کار می‌کند.

روش تصویر تصادفی پایدار، ماتریس داده‌های اولیه  $A \in \mathbb{R}^{n \times D}$  را در ماتریس تصادفی  $R \in \mathbb{R}^{D \times k}$  ضرب می‌کند و نتیجه ماتریس  $B = AR \in \mathbb{R}^{n \times k}$  است. معمولاً درایه‌های ماتریس تصادفی  $R$  به صورت i.i.d<sup>۲</sup> از یک توزیع  $\alpha$ -پایدار متقارن انتخاب می‌شوند ( $0 < \alpha \leq 2$ ). ما می‌توانیم مشخصه‌های  $l_\alpha$  را در  $A$  بر اساس  $B$  تخمین بزنیم. در مورد حالت  $l_2$  مزیت توزیع تصادفی پایدار توسط لم JL<sup>۳</sup> برجسته شده است. لم JL بیان می‌دارد که کافی است  $k = O(\frac{\log n}{\epsilon^2})$  باشد تا هم فاصله دو به دویی با نرم  $l_\alpha$  در  $A$  را بتوان با ضریب  $1 \pm \epsilon$  از روی ماتریس  $B$  تخمین زد. در تز [۳۶] Ping Li<sup>۴</sup> لمی مشابه لم JL برای  $0 < \alpha < 2$  اثبات شده است. روش تصویر تصادفی پایدار به یک مسئله تخمین آماری کاهش می‌یابد برای تخمین پارامتر مقیاس برای یک توضیح پایدار  $\alpha$  متقارن. این مسئله از این جهت مورد توجه قرار می‌گیرد زیرا ما به دنبال برآوردی می‌گردیم که هم از نظر آماری درست باشند و هم از نظر محاسباتی مقرون به صرفه. برآوردگرهای مختلفی را مطالعه و مقایسه کردیم. شامل میانگین حسابی، میانگین هندسی، میانگین هارمونیک، تقسیم توانی<sup>۵</sup> و برآوردگر حداکثر بزرگنمایی.

در این پایان‌نامه ما به بررسی موارد خاصی از تصویر تصادفی پایدار می‌پردازیم. برای نرم  $l_2$  ارتقای را با استفاده از اطلاعات حاشیه‌ای پیشنهاد می‌کنیم. همچنین برای حالت  $l_2$  می‌توان ماتریس تصویرگر را از یک توزیع زیرگوسی<sup>۵</sup> بسیار کوچکتر به جای توزیع نرمال انتخاب کرد. با در نظر گرفتن محدودیت‌های معقولی می‌توان، از یک توزیع خاص زیرگوسی استفاده کرد. این توزیع شامل  $[-1, 0, 1]$  با احتمالات  $\{\frac{1}{s}, 1 - \frac{1}{s}, \frac{1}{s}\}$  با مقادیر بسیار بزرگی برای  $s$  (به عبارتی، تصویر تصادف خیلی گسسته<sup>۶</sup>) می‌تواند به خوبی تصویر تصادفی نرمال عمل کند. برای حالت نرم  $l_1$  به عبارتی دیگر تصویر تصادفی کوچی<sup>۷</sup> انجام تخمین کاری نسبتاً جذاب است. برای مثال، محاسبه برآوردگر بیشینه درست‌نمایی MLE در این حالت از لحاظ محاسباتی ممکن است. و یک توزیع معکوس گاوسی<sup>۸</sup> برای مدل‌سازی دقیق توزیع MLE بیان شده است.

روش تصویر تصادفی از پراکندگی داده‌ها استفاده‌ای نمی‌کند. در حالی که داده‌های بزرگ مقیاس معمولاً بسیار پراکنده هستند. از روش تصویر تصادفی می‌توان برای حل مسائل بزرگ مقیاس در علوم و مهندسی در موتورهای جستجو و سیستم‌های اخذ داده، پایگاه‌های داده، سیستم‌های جریان داده جدید، جبر خطی عددی و بسیاری از کارهای یادگیری ماشین و داده کاوی که شامل محاسبه حجیم فاصله‌ها است، استفاده کرد.

<sup>1</sup>Stable Random Projection

<sup>2</sup>Independent and identically distributed random variables

<sup>3</sup>Johnson-Lindenstrauss

<sup>4</sup>fractional power

<sup>5</sup>sub-Gaussian

<sup>6</sup>Very sparse random projections

<sup>7</sup>Cauchy random projections

<sup>8</sup>inverse Gaussian

## فصل دوم

# کاهش بعد و داده‌های بزرگ مقیاس

## ۱-۲ داده‌های حجیم

عبارات زیر از سایت *Information Week* نقل قول شده‌اند<sup>۱</sup>:

- مقدار داده‌ای که توسط کسب و کارها ذخیره می‌شود تقریباً هر ۱۲ تا ۱۸ ماه دو برابر می‌شود.
  - پایگاه داده‌ها بیشتر هم زمان شده‌اند. فروشگاه‌های زنجیره‌ای Wall-Marat داده‌های فروش را هر ساعت به روز می‌کند.
  - اضافه شدن یک میلیون خط داده اجازه جستجوهای پیچیده‌تری را می‌دهد. شرکت EBay به کارمندان اجازه می‌دهد برای بدست آوردن درکی عمیق‌تر در خصوص رفتار مشتریان در میان داده‌های حراج در بازه‌های زمانی کوتاه جستجو کنند.
  - بزرگترین پایگاه داده‌ها توسط، مرکز شتاب‌دهنده خطی استاندارد، مرکز تحقیقات ناسا، آژانس امنیت ملی و ... در ابعادی در محدوده‌ی پتابایت (هزار ترابایت ۱۰<sup>۱۵</sup> بایت)، اداره می‌شوند.
- پدیده نو ظهور مجموعه داده‌ای حجیم، چالش‌های محاسباتی در بسیاری کاربردهای علمی و تجاری به وجود آورده است. شامل اختریف‌یک، بیوتکنولوژی، جمعیت‌شناسی<sup>۲</sup>، مالی، سیستم‌های اطلاعات جغرافیایی، دولت، دارو، ارتباطات از راه دور، محیط زیست و اینترنت.

## ۱-۱-۲ داده‌های حجیم وب

وب چقدر بزرگ است؟ **جدول ۱-۲** نشان‌دهنده تعداد بازدید صفحات در موتورهای جستجوی امروزی است. به طور تخمینی حدود  $D = 10^{10}$  صفحه‌ی وب را می‌توان بر اساس بازدید دو واژه‌ی بسیار پر کاربرد «A» و «THE» تخمین زد. **جدول ۱-۲** همچنین نشان می‌دهد که حتی کلماتی که به ندرت کاربرد دارند هم تعداد زیادی بازدید دارند.

Query	Google	Bing
A	25,270,000,000	175,000,000
The	25,270,000,000	101,000,000
Kalevala	7,440,000	939,000
Griseofulvin	1,163,000	332,000
Saccade	1,030,000	388,000

جدول ۱-۲: تعداد بازدید صفحات برای کلمات با بازخورد بالا و کلمات با بازخورد نادر

کلماتی با بازخورد معمولی چه میزان بازدید دارند؟ برای جواب این سوال ما به طور تصادفی ۱۵ صفحه از لغتنامه‌ی آموزشی انتخاب می‌کنیم. [۲۷] (لغتنامه‌ای با ۱۰۰,۵۷ کلمه) و اولین کلمه در هر صفحه را مد نظر قرار می‌دهیم. میانه‌ی آماری بر اساس جستجوگر گوگل ۱۰ میلیون صفحه برای کلمه است. زبان انگلیسی چند کلمه دارد؟ در اینجا عبارتی را از AskOxford.com نقل قول می‌کنیم:

«این بیان میدارد که حداقل یک چهارم میلیون واژه‌ی انگلیسی مستقل وجود دارد. به جز افعال صرفی و کلمات فنی و ناحیه‌ای که توسط OED<sup>۳</sup> تحت پوشش قرار نمی‌گیرند یا کلماتی که هنوز به لغتنامه‌های منتشر

<sup>۱</sup> <http://www.informationweek.com/news/showArticle.jhtml?articleID=175801775>

<sup>۲</sup> demographics

<sup>۳</sup> Oxford English Dictionary

شده اضافه نشده‌اند. در صورتی که این موارد هم در نظر گرفته شوند تعداد لغات در حدود سه چهارم میلیون لغت خواهد بود»

بنابراین اگر یک ماتریس «عبارت به سند»  $A \in \mathbb{R}^{n \times D}$  در نظر بگیریم. در ابعاد وب این ماتریس در ابعاد  $n \approx 10^6$  و  $D \approx 10^5$  بزرگ خواهد شد. در اینجا عدد  $(i, j)$  در  $A$  تعداد ظهور واژه  $i$  در سند  $j$  را نشان می‌دهد. کارکردن با ماتریس در این ابعاد بزرگ چالش برانگیز است. برای مثال، شاخص LSI<sup>۴</sup> [۱۹] و یک مدل موضوعی فراگیر، از SVD<sup>۵</sup> بر روی ماتریس عبارت به سند استفاده می‌کند. که انجام این عملیات در ابعاد وب قطعاً غیرممکن است.

یک مشکل اصلی در قبال مجموعه داده‌های سنگین، حافظه کامپیوتر است. به این دلیل که ابعاد و سرعت حافظه فیزیکی بسیار رشد کمتری در مقایسه با پردازنده‌ها (CPU) دارد. این پدیده به عنوان دیوار حافظه شناخته می‌شود [۳۹، ۴۴]. برای مثال، هر چند ممکن است تمامی رخدادهای همزمان دوتایی از پیش محاسبه شوند، ولی نگهداری این حجم از داده در حافظه غیر ممکن است. علاوه بر این، گاهی اوقات تخصیص‌هایی با بیش از دو عامل هم اهمیت پیدا می‌کنند زیرا درخواست‌ها ممکن است شامل بیش از دو واژه هم باشند. یک راه حل ممکن این است که یک «نمونه» از  $A$  نگهداری شود و همزمانی‌ها بر اساس این نمونه در حین کار تخمین زده شوند. ما حدس می‌زنیم که این روش توسط موتورهای جستجوی امروزی مورد استفاده قرار می‌گیرد، هر چند که روش واقعی قطعاً جزو اسرار تجاری آن‌ها است.

هر چند که انتظار می‌رود تخمین‌ها سازگار باشند و فرکانس‌های جفت شده باید با افزایش عبارت به درخواست، کاهش پیدا کنند. جدول ۲-۲ نشان می‌دهد که تخمین‌های بیان شده با موتورهای جستجوی فعلی، همیشه سازگار نیستند.

Query	Hits(Bing)	Hits(Google)
America	150,731,182	393,000,000
America & China	15,240,116	66,000,000
America & China & Britain	235,111	6,090,000
America & CHina & Britain & Japan	154,444	23,300,000

جدول ۲-۲: با افزایش تعداد عبارات در درخواست، باید فرکانس‌های جفت شده کاهش پیدا کنند. ولی تخمین‌های بیان شده توسط موتورهای جستجو گاهی این موضوع تثبیت شده را نقض می‌کنند.

با اینکه، تعداد کل واژه‌های انگلیسی (که به‌طور صحیح نوشته شده‌اند) هم اکنون شگفت‌آور است، در بسیاری کاربردهای متن کاوی، ما باید با ابعاد بسیار بزرگتری سر و کار داشته باشیم. در حالی که یک سند ممکن است بیانگر برداری از تک واژه‌ها باشد (به عبارت دیگر، مدل کیسه لغات<sup>۶</sup>). معمولاً بهتر است سند به عنوان یک بردار از لغات به صورت  $l$  پیوسته<sup>۷</sup> [۱۳] بیان شود. برای مثال، با استفاده از مدل ۳ پیوسته، جمله  $l$  "It is a nice day" به مجموعه‌ی زیر تجزیه می‌شود.  $\{ "it is a", "is a nice", "a nice day" \}$  این مدل به طور چشمگیری ابعاد داده‌ها را افزایش می‌دهد. به خاطر اینکه، اگر مجموعه‌ی  $10^6$  تک لغت انگلیسی موجود داشته باشد. مدل ۳ پیوسته تعداد ابعاد را از  $10^6$  به  $10^{18}$  افزایش می‌دهد.

<sup>4</sup>latent semantic indexing

<sup>5</sup>singular value decomposition

<sup>6</sup>bag-of-words

<sup>7</sup>l-shingles

## ۲-۱-۲ جریان‌های داده‌ی حجیم

در بسیاری کاربردهای جدید پردازش داده، جریان‌های داده‌ی حجیم نقش بنیادی دارند. جریان‌های داده‌ای که از روترهای اینترنت، سوئیچ‌های تلفن، رصد امسفر، شبکه‌های سنسور، شرایط ترافیکی بزرگراهی، داده‌های مالی و غیره [۱، ۴۰، ۱۸، ۷، ۲۸، ۳۳، ۲۶] حاصل می‌شوند.

برخلاف پایگاه داده‌های سنتی، معمول نیست که جریان‌های داده‌ی حجیم (که با سرعت زیادی منتقل می‌شوند) در جای نگهداری شوند. بنابراین پردازش معمولاً به طور همزمان انجام می‌شوند. برای مثال، گاهی اوقات «رصد تصویری» داده‌ها با رصد تغییرات زمانی برخی آماره‌ها کفایت می‌کند. برای مثال آماره‌های نظیر: مجموع، تعداد آیتم‌های مجزا، برخی نرم‌های  $l_\alpha$ . در برخی کاربردها (برای مثال، طبقه‌بندی صدا/محتوا و جدا سازی) نیاز است یک مدل یادگیری آماری برای کلاسه‌بندی<sup>۸</sup> یا خوشه‌بندی<sup>۹</sup> جریان داده‌های حجیم توین شود. ولی معمولاً فقط می‌توانیم یک‌بار داده‌ها را مورد بررسی قرار دهیم.

یک خاصیت مهم جریان‌های داده‌ای این است که دینامیک هستند. به عنوان یک مدل محبوب، جریان  $u$  شامل ورودی‌های  $(i, u_i)$  است که  $i = 1 \text{ to } D$  برای مثال،  $D = 2^64$  زمانی که جریان بیان‌گر IP آدرس‌ها است.<sup>۱۰</sup> ورودی‌ها ممکن است به هر ترتیبی باشند و ممکن است مرتباً به روز شوند. ذات دینامیک جریان داده‌های حجیم فرآیند نمونه‌گیری را بسیار چالش‌برانگیزتر از زمانی می‌کند که با داده‌های ایستا سر و کار داریم.

## ۲-۲ چالش‌های نمونه‌گیری از داده‌های حجیم

در حالی که مسائل جذاب و چالش‌برانگیزی با ورود داده‌های حجیم شکل گرفته‌اند، این پایان‌نامه بر روی توسعه‌ی روش‌های نمونه‌گیری برای محاسبه فاصله در داده‌هایی با ابعاد بسیار بالا با استفاده از حافظه محدود تمرکز دارد. در کاربردهای مدل‌سازی آماری و یادگیری ماشین، در اغلب موارد به جای داده‌های اصلی به فاصله، به خصوص فاصله‌ی جفتی نیاز داریم. برای مثال، محاسبه ماتریس گرام  $AA^T$ <sup>۱۱</sup> در آمار و یادگیری ماشین معمول است.  $AA^T$  بیانگر هم‌بستگی ضرب‌های داخلی دوتایی در ماتریس داده‌ی  $A$  است. دو داده‌ی  $u_1, u_2 \in \mathbb{R}^D$  داده شده‌اند. ضرب داخلی آن‌ها (که با  $a$  نمایش داده می‌شود) و  $l_\alpha$  (که با  $d_{(\alpha)}$  نمایش داده می‌شوند با عبارات زیر تعریف می‌شوند):<sup>۱۲</sup>

$$a = u_1^T u_2 = \sum_{i=1}^D u_{1,i} u_{2,i} \quad (1-2)$$

$$d_{(\alpha)} = \sum_{i=1}^D |u_1 - u_2|^\alpha \quad (2-2)$$

<sup>8</sup>Classification<sup>9</sup>Clustering

<sup>۱۰</sup> هرچند ما بیشتر اوقات تعداد دقیق ابعاد ( $D$ ) یک جریان داده را نمی‌دانیم ولی در بیشتر کاربردها کافی است حد بالایی محافظه‌کارانه‌ای را در نظر بگیریم. برای مثال  $D = 2^64$  زمانی که جریان بیانگر IP های ورودی است. همچنین این یکی از دلایلی است که داده‌ها بسیار پراکنده هستند. به این نکته توجه داشته باشید که ابعاد بسیار بزرگ تأثیری در محاسبه‌ی فاصله‌ها و نمونه‌گیری طی الگوریتم‌های معرفی شده در این پایان‌نامه ندارد.

<sup>۱۱</sup>Gram matrix

<sup>۱۲</sup> ما فاصله  $l_\alpha$  را به صورت  $d_{(\alpha)} = \sum_{i=1}^D |u_1 - u_2|^\alpha$  تعریف کرده‌ایم. به جای اینکه به شکل  $(\sum_{i=1}^D |u_1 - u_2|^\alpha)^{1/\alpha}$  تعریف کنیم. زیرا شکل اول در کاربردهای عملی عمومیت بیشتری دارد. برای مثال، LM، در ادبیات معمولاً به شکل توان دو  $l_2$  بیان می‌شود.  $\sum_{i=1}^D |u_1 - u_2|^2$  به جای  $(\sum_{i=1}^D |u_1 - u_2|^2)^{1/2}$ . در این پایان‌نامه، ما برای سادگی  $\sum_{i=1}^D |u_1 - u_2|^2$  را «فاصله  $l_2$ » بیان می‌کنیم به جای «مربع فاصله  $l_2$ ».

به این نکته توجه داشته باشید که هم ضرب داخلی و هم فاصله به شکل جمع  $D$  جمله تعریف می‌شوند. بنابراین، زمانی که داده‌ها به اندازه‌ای حجیم هستند که نمی‌توان به طور کارا آن‌ها را مدیریت کرد، نمونه‌گیری خیلی عادی به نظر می‌رسد تا بتوان با انتخاب تصادفی  $k$  عضو از  $D$  جمله تخمینی از مجموع به دست آوریم (با ضریب مقیاس  $\frac{D}{k}$ ). در خصوص ماتریس داده‌ای  $A \in \mathbb{R}^{n \times D}$  انتخاب تصادفی مختصات <sup>۱۳</sup>،  $k$  ستون را از ماتریس داده به طور یکنواخت و تصادفی انتخاب می‌کند.

نمونه‌گیری از این جهت سودمند است که هم سایکل‌های کاری CPU را کاهش می‌دهد و هم در حافظه صرفه‌جویی می‌کند. در کاربردهای جدید، در اغلب موارد صرفه‌جویی در حافظه از اهمیت بیشتری برخوردار است. در نیم قرن گذشته گلوگاه محاسباتی حافظه بوده است، نه پردازشگر. سرعت پردازشگرها با نرخ تقریبی ۷۵ درصد در سال رو به افزایش است. در حالی که سرعت حافظه تقریباً سالی ۷ درصد افزایش می‌یابد [۳۹]. این پدیده به عنوان «دیوار حافظه»<sup>۱۴</sup> شناخته می‌شود [۳۹، ۴۴]. بنابراین در کاربردهایی که شامل مجموعه داده‌های حجیم می‌شوند، بحرانی‌ترین کار بیان کردن داده‌ها است. برای مثال، از طریق نمونه‌گیری با فرمی فشرده برای قرارگیری در ابعاد حافظه در دسترس.

## ۱-۲-۲ مزایای نمونه‌گیری تصادفی مختصات

نمونه‌گیری تصادفی مختصات به دو دلیلی معمولاً انتخاب پیش‌فرض است.

• **سادگی** این روش از لحاظ زمانی تنها از مرتبه  $O(nk)$  برای نمونه‌گیری  $k$  ستون از  $A \in \mathbb{R}^{n \times D}$  طول می‌کشد.

• **انعطاف پذیری** یک مجموعه نمونه را می‌توان برای تخمین بسیاری از شاخص‌های آماری استفاده کرد. شامل: ضرب داخلی، فاصله  $l_\alpha$  (برای هر مقداری از  $\alpha$ )

## ۲-۲-۲ معایب نمونه‌گیری تصادفی مختصات

با این حال نمونه‌گیری تصادفی مختصات دو ایراد اساسی دارد.

• معمولاً دقیق نیست زیرا مقادیری با مقدار زیاد محتمل است که گم شوند. مخصوصاً زمانی که داده‌ها دم سنگینی داشته باشند. داده‌های بزرگ مقیاس دنیای واقعی (مخصوصاً داده‌های مربوط به اینترنت) همیشه دم سنگین هستند و از قاعده توانی پیروی می‌کنند. [۴۱، ۲۱، ۱۶، ۳۵] زمانی که فاصله  $l_2$  یا ضرب داخلی را تخمین می‌زنیم. واریانس تخمین‌ها بر اساس ممان چهارم داده‌ها تعیین می‌شود. در حالی که در داده‌های دم سنگین، گاهی اوقات حتی ممان اول هم معنی‌دار نیست (محدود نیست) [۴۱].

• این روش داده‌های پراکنده را به خوبی مدیریت نمی‌کند. بسیاری از داده‌های بزرگ مقیاس به شدت پراکنده هستند، به عنوان مثال، داده‌های متنی [۲۰] و داده‌های بر اساس بازار [۲، ۴۲]. به جز برخی واژه‌های کاربردی مانند "A" و "The" بیشتر لغات با نسبت بسیار کمی در مستندات ظاهر می‌شوند ( $\square$ ) اگر ما داده‌ها را با در نظر گرفتن تعدادی از ستون‌های ثابت نمونه‌گیری کنیم. خیلی محتمل است که بیشتر داده‌های (مقادیر غیر صفر) را از دست بدهیم. به خصوص موارد جذابی که دو مقدار با هم غیر صفر شده‌اند.

<sup>13</sup>Random coordinate sampling

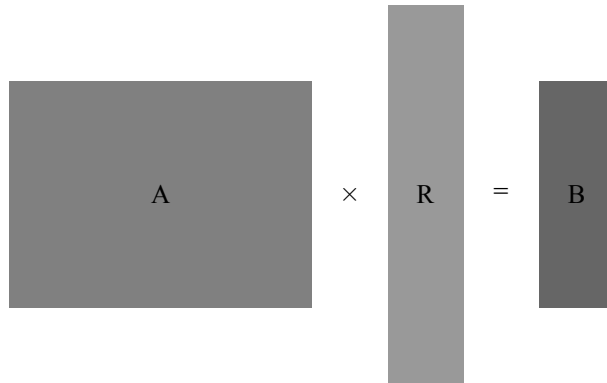
<sup>14</sup>Memory wall



در این پایان‌نامه ما روش تصویر تصادفی را مورد بررسی قرار می‌دهیم و نشان خواهیم داد که این روش به خوبی قابلیت مدیریت داده‌های دم‌سنگین را دارد.

## ۳-۲ تصویر تصادفی پایدار

تصویر شکل ۱-۲، ایده تصویر تصادفی را نشان می‌دهد. ایده اصلی تصویر تصادفی ضرب ماتریس داده‌ی  $A \in \mathbb{R}^{n \times D}$  در ماتریس تصادفی  $R \in \mathbb{R}^{D \times k}$  ( $k \ll D$ ) است. که حاصل ماتریس تصویر شده‌ی  $B = A \times R \in \mathbb{R}^{n \times k}$  است.  $B$  بسیار کوچکتر از  $A$  است و بنابراین به راحتی قابل ذخیره‌سازی است. (برای مثال: برای حافظه‌های فیزیکی به اندازه‌ی کافی کوچک است)



شکل ۱-۲: تصویر تصادفی پایدار  $B = A \times R$ ، ماتریس اولیه داده‌ها است.

ماتریس تصویرگر  $R \in \mathbb{R}^{D \times k}$  معمولاً از داریه‌های مستقل هم توزیع (i.i.d) یک توزیع متقارن  $\alpha$ -پایدار پر شده است. [۴۵] (بنابراین نام این روش «تصویر تصادفی پایدار» است.) بر اساس مشخصات توزیع‌های  $\alpha$ -پایدار، داده‌های تصویر شده هم از توزیع  $\alpha$ -پایدار پیروی می‌کنند. که بر اساس آن‌ها شاخص‌های  $l_\alpha$  و فاصله دودویی  $l_\alpha$  در  $A$  تخمین زده می‌شوند و می‌توانیم داده‌های اصلی را دور بریزیم. موفقیت تصویر پایدار تصادفی توسط لم Johnson-Lindenstrauss (JL) [۳۱] برای کاهش بعد در  $l_2$  نشان داده شده است. لم JL بیان می‌کند: رعایت  $k = O\left(\frac{\log n}{\epsilon^2}\right)$  تضمین می‌کند هر فاصله  $l_2$  میان  $n$  نقطه در هر تعداد بعدی با دقت  $1 \pm \epsilon$  با احتمال بالایی تخمین زده شود. ( $k$  در اینجا بیانگر تعداد ابعاد کاهش یافته است) با این حال لم JL برای نرم‌های فاصله با  $\alpha$  کوچکتر از ۲ ( $\alpha < 2$ ) صادق نیست. در صورتی که لازم باشد از برآوردگرهایی استفاده کنیم که متریک باشند (در نامساوی مثلثی صدق کنند). به این نتیجه «عدم امکان»<sup>۱۵</sup> گفته می‌شود. [۱۲، ۳۴، ۱۴] خوشبختانه شامل برآوردگرهایی که متریک نیستند نمی‌شود. در این پایان‌نامه ما در مورد برآوردگرهای کوناگونی که متریک نیستند صحبت خواهیم کرد. شامل: میانگین هندسی<sup>۱۶</sup>، میانگین هارمونیک<sup>۱۷</sup>، نسبت توانی<sup>۱۸</sup> و همچنین حداکثر بزرگنمایی.

<sup>15</sup>Impossibility

<sup>16</sup>Geometric mean

<sup>17</sup>Harmonic mean

<sup>18</sup>Fractional power

## ۴-۲ کاربردها

علاقه‌ی زیادی به تکنیک‌های نمونه برداری وجود دارد که در کاربردهای زیادی مورد استفاده قرار می‌گیرند. مانند: قانون وابستگی<sup>۱۹</sup> [۹، ۱۰]، خوشه‌بندی، بهینه‌سازی درخواست<sup>۲۰</sup> [۳۸، ۱۵]، تشخیص تکراری<sup>۲۱</sup> [۱۳، ۸] و بسیاری موارد دیگر. روش‌های نمونه بردار هر چه بیشتر و بیشتر برای مجموعه‌های بزرگتر اهمیت پیدا می‌کنند. طرح برودر<sup>۲۲</sup> [۱۳] در ابتدا برای تشخیص صفحات وب تکراری معرفی شد. URL‌های زیادی به HTML‌های مشابه (یا تقریباً مشابه) اشاره می‌کنند. جواب‌های تخمین زده شده به اندازه‌ی کافی خوب بودند. نیازی نبود تا همه تکراری‌ها پیدا شوند ولی کاربردی بود که تعداد زیادی از آن‌ها پیدا شوند، بدون اینکه بیش از ارزش آن از توان محاسباتی استفاده شود.

در کاربردهای بازیابی اطلاعات (IR)<sup>۲۳</sup> معمولاً گلوگاه حافظه‌ی فیزیکی است. زیرا مجموعه‌ی وب برای حافظه (RAM) بسیار بزرگ است و از طرفی ما می‌خواهیم زمان گشتن به دنبال داده‌ها بر روی دیسک را کمینه کنیم. زیرا زمان پاسخ به یک درخواست کلیدی است [۱۱]. به عنوان یک وسیله صرفه‌جویی در فضا، کاهش بعد یک ارائه فشرده از داده‌ها فراهم می‌کند که برای تولید جواب‌های تخمینی در حافظه فیزیکی مورد استفاده قرار می‌گیرند.

ما به بازدید صفحات وب اشاره کردیم. اگر ما یک عبارت جستجوی دو کلمه‌ای داشته باشیم، می‌خواهیم بدانیم چه تعداد از صفحات هر دو کلمه را دارند. فرض می‌کنیم محاسبه‌ی از قبل و نگهداری بازدید صفحات غیر ممکن باشد. حداقل نه برای کلماتی که تکرار زیادی ندارند و سری‌های چند کلمه‌ای.

مرسوم است که در بازیابی اطلاعات با یک ماتریس بزرگ عبارت به ازای سند شروع کنیم که در آن مقادیر ورودی نشان‌دهنده‌ی وجود عبارت در متن است. بنا به کاربردهای خاص می‌توانیم یک اندیس معکوس<sup>۲۴</sup> بسازیم و کلیتی از عبارات (برای تخمین ارتباط لغات) یا اسناد (برای تخمین شباهت اسناد) نگهداری کنیم.

## ۱-۴-۲ کاوش قوانین وابستگی

تحلیل‌های مبتنی بر بازار و قوانین وابستگی [۳، ۴، ۵] ابزارهای مناسبی برای کاوش پایگاه داده‌های تجاری هستند. پایگاه داده‌های تجاری دارند روز به روز بزرگتر و گسسته‌تر می‌شوند. [۲، ۴۲] الگوریتم‌های مختلف نمونه‌برداری پیشنهاد شده است. نمونه برداری این امکان را فراهم می‌کند تا قواعد تخصیص را به صورت آنلاین برآورد کنیم. که می‌تواند مزایایی در کاربردهای خاص داشته باشد.

## ۲-۴-۲ وابستگی جفتی همه (فاصله‌ها)

در کاربردهای مختلفی شامل کلاسه‌بندی بر مبنای فاصله یا خوشه‌بندی و مدل‌سازی زبان با bi-gram<sup>۲۵</sup> ما نیازمند محاسبه‌ی همه‌ی جفت تخصیص‌ها (یا فاصله‌ها) هستیم. ماتریس داده‌ی  $A$  شامل  $n$  سطر و  $D$  ستون داده شده است. محاسبه‌ی مستقیم  $AA^T$ ،  $O(n^2D)$  هزینه بر است. یا به طور بهینه‌تر  $O(n^2\bar{f})$  که  $\bar{f}$  تعداد میانگین مقادیر غیر صفر میان تمام سطرهای  $A$  است. محاسبه مستقیم می‌تواند به شدت زمان‌بر باشد. همچنین، به طور

<sup>19</sup>Association rules<sup>20</sup>Query optimization<sup>21</sup>Duplicate detection<sup>22</sup>Broder's sketch<sup>23</sup>information retrieval<sup>24</sup>inverted index<sup>25</sup>lite48

خاص زمانی که ماتریس داده آنقدر بزرگ است که در حافظه فیزیکی جا نمی‌شود. محاسبه به طور خاص بسیار ناکارآمد خواهد بود.

### ۳-۴-۲ تخمین فاصله‌ها به طور آنلاین

در حالی که ماتریس داده‌ی اولیه  $A \in \mathbb{R}^{n \times D}$  ممکن است برای حافظه‌ی فیزیکی بسیار بزرگ باشد، نگهداری <sup>۲۶</sup> همه فاصله‌های جفتی و وابستگی‌ها در  $A$ ،  $O(n^2)$  فضا مصرف می‌کند. که می‌تواند برای حافظه‌ی فیزیکی بسیار بزرگتر باشد. در این میان وابستگی‌های چندتایی را کنار می‌گذاریم. در بسیاری از کاربردها نظیر یادگیری برخط، سیستم‌های توصیه آنلاین، تحلیل‌های بازار برخط و موتورهای جستجو، بهتر است که نمونه‌ها (sketches) در حافظه نگهداری شوند و همه‌ی فاصله‌ها به طور آنلاین، زمانی که مورد نیاز باشد، محاسبه شوند.

### ۴-۴-۲ بهینه‌سازی درخواست از پایگاه داده

در پایگاه داده‌ها یک وظیفه‌ی بسیار مهم تخمین join‌های چندراهی است، که تاثیر زیادی بر روی کارایی سیستم دارد [۲۵]. بر اساس تخمین دوراهی، سه‌راهی و حتی join‌هایی از مرتبه‌ی بالاتر، بهینه‌گرهای درخواست یک نقشه برای کمینه کردن تابع هزینه می‌سازند (برای مثال، نوشتن‌های میانی <sup>۲۷</sup>). بهینه بودن اهمیت بسیاری دارد زیرا مثلاً نمی‌خواهیم زمان بیشتری برای بهینه‌سازی نقشه نسبت به زمان اجرای آن تلف کنیم. ما از مثال Governor برای نمایش کاربرد تخمین دو و چند راهه برای بهینه کردن درخواست استفاده می‌کنیم.

	Query	Hits(Google)
One-way	Austria	88,200,000
	Governor	37,300,000
	Schwarzenegger	4,030,000
	Terminator	3,480,000
Two-way	Governor & Schwarzenegger	1,220,000
	Governor & Austria	708,000
	Schwarzenegger & Terminator	504,000
	Terminator & Austria	171,000
	Governor & Terminator	132,000
	Schwarzenegger & Austria	120,000
Tree-way	Governor & Schwarzenegger & Terminator	75,100
	Governor & Schwarzenegger & Austria	46,100
	Schwarzenegger & Terminator & Austria	16,000
	Governor & Terminator & Austria	11,500
Four-way	Governor & Schwarzenegger & Terminator & Austria	6,930

جدول ۳-۲: بازدید صفحات گزارش شده توسط Google برای چهار کلمه و وابستگی‌های دو، سه و چهارتایی آن‌ها

**جدول ۳-۲** بازدید صفحات را برای چهار کلمه و ترکیبات دو، سه، چهارتایی آن‌ها نشان می‌دهد. فرض کنیم بهینه‌ساز قصد استخراج نقشه برای درخواست: "Governor, Schwarzenegger, Terminator, Austria" را داشته باشد. راه حل استاندارد این است که با عبارات با کمترین فراوانی شروع کند: ("Schwarzenegger") ("Governor") ("Terminator") ("Austria") این نقشه ۵۷۹,۱۰۰ نوشتن میانی بعد از اولین و دومین join

<sup>26</sup>Materializing

<sup>27</sup>Intermediate writes

ها دارد. یک بهینه‌سازی می‌تواند "Governor"  $\cap$  "Terminator"  $\cap$  "Austria"  $\cap$  "Schwarzenegger" باشد که  $579,100$  را به  $136,000$  کاهش می‌دهد.

## ۵-۴-۲ جستجوی نزدیکترین همسایه از مرتبه‌ی زیر خطی

محاسبه‌ی نزدیکترین همسایه در بسیاری کاربردها از اهمیت زیادی برخوردار است. با این حال، به دلیل «نفرتین ابعاد»<sup>۲۸</sup> راه حل فعلی برای پیدا کردن بهینه‌ی نزدیکترین همسایه‌ها (حتی به طور تقریبی) اصلاً رضایت بخش نیست. [۲۲، ۳۰]

به دلیل ملاحظات محاسباتی، دو شکل اصلی در جستجوی نزدیکترین همسایه‌ها وجود دارد. اول اینکه ماتریس اصلی داده‌ها  $A \in \mathbb{R}^{n \times D}$  ممکن است برای حافظه فیزیکی بسیار بزرگ باشد ولی اسکن کردن دیسک‌های سخت برای پیدا کردن نزدیکترین همسایه‌ها می‌تواند خیلی کند باشد. دوماً، پیدا کردن نزدیکترین همسایه‌های یک داده ممکن است  $O(nD)$  هزینه بر باشد که می‌تواند به شدت زمان بر شود.

با این حال، روس کاهش ابعادی در این پایان‌نامه می‌تواند در حافظه صرفه‌جویی کند و سرعت محاسبات را افزایش دهد. برای مثال: وقتی ماتریس داده‌ی اولیه  $A$  به ماتریس داده‌ی  $B \in \mathbb{R}^{n \times k}$  کاهش می‌یابد. با این حال،  $O(nk)$  و معمولاً این درخواست وجود دارد که هزینه‌ی محاسباتی از  $O(n)$  به  $O(n^\gamma)$  برای  $\gamma < 1$  کاهش پیدا کند، حداقل برای کاربردهای خاص.

دو گروه اصلی الگوریتم‌های زیر خطی برای محاسبه عبارتند از KD-Trees (و انواع آن) [۲۳، ۲۴] و Locality-Sensitive Hashing (LSH) [۶، ۱۷، ۳۰] این الگوریتم‌ها معمولاً با یک فضای متریک کار می‌کنند (که در آن نامساوی مثلثی برقرار است). برای مثال، فضای  $l_\alpha$  زمانی که  $\alpha \geq 1$  باشد یک متریک است. زمانی که به دنبال نزدیکترین همسایه‌ها در  $l_\alpha$  ( $\alpha > 1$ ) می‌گردیم، می‌توانیم (نسبتاً به سادگی) فضای جستجو را به طور کاملاً اساسی با استفاده از نامساوی مثلثی کاهش دهیم. به عبارت دیگر، نیازی نیست که همه  $n$  نقطه داده‌ها را مورد بررسی قرار دهیم.

در داده‌هایی با ابعاد بسیار بزرگ، الگوریتم‌های زیر خطی موجود شامل KD-trees و LSH، عملکرد رضایت بخشی ندارند. وقتی حافظه‌ی فیزیکی (به جای CPU) گلوگاه باشد<sup>۲۹</sup>، یکی از مشکلات اصلی این است که این الگوریتم‌ها برای کاهش هزینه‌ی محاسباتی به حافظه‌ی ابر خطی<sup>۳۰</sup> نیاز دارند که می‌تواند مشکل ساز باشد. [۳۰] به طرح کلی برای LSH توجه کنید که ترکیبی از هاش<sup>۳۱</sup> و تصویر تصادفی است. متأسفانه این طرح به دلیل هزینه‌ی زیاد پیش پردازش غیر کاربردی است. [۳۰]

در این پایان‌نامه، موفقیت اصلی کاهش بعد داده  $A \in \mathbb{R}^{n \times D}$  به  $B \in \mathbb{R}^{n \times k}$  و تامین برآوردگرهای دقیق برای استخراج فاصله‌های اولیه در  $A$  بر اساس  $B$  است. در حالی که سناریوهای مهمی وجود دارند که در آن‌ها نتایج ما رضایت بخش هستند، توسعه‌ی یک الگوریتم زیر-خطی برای تخمین نزدیکترین همسایه‌ها، بر اساس الگوریتم ما یک ایده جذاب برای تحقیقات آینده است. یک مانع اصلی در این راه این است که بیشتر برآوردگرهای ما غیر متریک هستند و بنابراین طراحی یک الگوریتم هوشمند و تحلیل‌های تئوری ممکن است سخت باشد، با این حال غیر ممکن نیست.

<sup>28</sup>Curse of dimensionality

<sup>29</sup>Memory wall

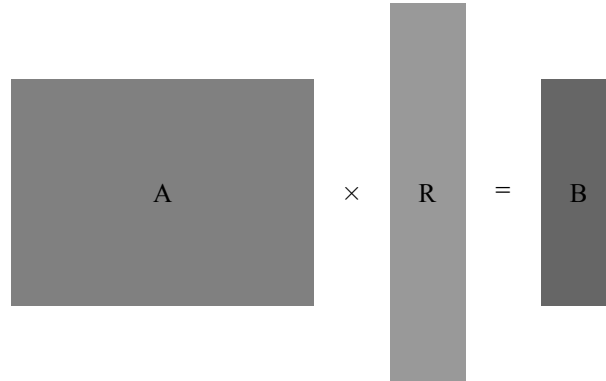
<sup>30</sup>Super-linear memory

<sup>31</sup>Hash

## فصل سوم

# تصویر تصادفی پایدار

روش تصویر تصادفی پایدار<sup>۱</sup> [۳۷، ۴۳، ۷، ۲۹، ۲۸، ۳۲] یک روش پرکاربرد در داده کاوی و یادگیری ماشین است. با این روش به طور کار  $(0 < \alpha \leq 2)$  فاصله در داده‌های حجیم (برای مثال: وب یا جریان‌های داده‌ای حجیم) محاسبه می‌شود. در این روش حافظه‌ی کمی استفاده شده و فقط یک بار پایش داده‌ها کافی است.



شکل ۱-۳: روش تصویر تصادفی پایدار ماتریس داده‌ی  $A \in \mathbb{R}^{n \times D}$  را در یک ماتریس تصادفی  $R \in \mathbb{R}^{D \times k}$  ضرب می‌کند تا ماتریس تصویر شده‌ی  $B = AR \in \mathbb{R}^{n \times k}$  حاصل شود.

همانطور که در شکل ۱-۳ می‌بینید.

<sup>1</sup>Stable Random Projections

## منابع و مراجع

- [1] Aggarwal, Charu C. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.
- [2] Aggarwal, Charu C, Wolf, Joel L, and Yu, Philip S. *A new method for similarity indexing of market basket data*. ACM, 1999.
- [3] Agrawal, Rakesh, Imieliński, Tomasz, and Swami, Arun. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.
- [4] Agrawal, Rakesh, Mannila, Heikki, Srikant, Ramakrishnan, Toivonen, Hannu, Verkamo, A Inkeri, et al. Fast discovery of association rules. *Advances in knowledge discovery and data mining*, 12(1):307–328, 1996.
- [5] Agrawal, Rakesh, Srikant, Ramakrishnan, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.
- [6] Andoni, Alexandr and Indyk, Piotr. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.
- [7] Babcock, Brian, Babu, Shivnath, Datar, Mayur, Motwani, Rajeev, and Widom, Jennifer. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16. ACM, 2002.
- [8] Brin, Sergey, Davis, James, and Garcia-Molina, Hector. Copy detection mechanisms for digital documents. In *ACM SIGMOD Record*, volume 24, pages 398–409. ACM, 1995.

- [9] Brin, Sergey, Motwani, Rajeev, and Silverstein, Craig. Beyond market baskets: Generalizing association rules to correlations. In *Acm Sigmod Record*, volume 26, pages 265–276. ACM, 1997.
- [10] Brin, Sergey, Motwani, Rajeev, Ullman, Jeffrey D, and Tsur, Shalom. Dynamic itemset counting and implication rules for market basket data. *Acm Sigmod Record*, 26(2):255–264, 1997.
- [11] Brin, Sergey and Page, Lawrence. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.
- [12] Brinkman, Bo and Charikar, Moses. On the impossibility of dimension reduction in  $l_1$ . *Journal of the ACM (JACM)*, 52(5):766–788, 2005.
- [13] Broder, Andrei Z. On the resemblance and containment of documents. In *Compression and complexity of sequences 1997. proceedings*, pages 21–29. IEEE, 1997.
- [14] Buhler, Jeremy and Tompa, Martin. Finding motifs using random projections. *Journal of computational biology*, 9(2):225–242, 2002.
- [15] Chaudhuri, Surajit, Motwani, Rajeev, and Narasayya, Vivek. On random sampling over joins. In *ACM SIGMOD Record*, volume 28, pages 263–274. ACM, 1999.
- [16] Crovella, Mark E and Bestavros, Azer. Self-similarity in world wide web traffic: evidence and possible causes. *IEEE/ACM Transactions on networking*, 5(6):835–846, 1997.
- [17] Datar, Mayur, Immorlica, Nicole, Indyk, Piotr, and Mirrokni, Vahab S. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.
- [18] Datar, Mayur and Indyk, Piotr. Comparing data streams using hamming norms. In *Proceedings 2002 VLDB Conference: 28th International Conference on Very Large Databases (VLDB)*, page 335. Elsevier, 2002.
- [19] Deerwester, Scott, Dumais, Susan T, Furnas, George W, Landauer, Thomas K, and Harshman, Richard. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.
- [20] Dhillon, Inderjit S and Modha, Dharmendra S. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1-2):143–175, 2001.



- [21] Faloutsos, Michalis, Faloutsos, Petros, and Faloutsos, Christos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, pages 251–262. ACM, 1999.
- [22] Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. *The elements of statistical learning*, volume 10. Springer series in statistics New York, NY, USA:, 2001.
- [23] Friedman, Jerome H, Baskett, Forest, and Shustek, Leonard J. An algorithm for finding nearest neighbors. *IEEE Transactions on computers*, 100(10):1000–1006, 1975.
- [24] Friedman, Jerome H, Bentley, Jon Louis, and Finkel, Raphael Ari. An algorithm for finding best matches in logarithmic time. *ACM Trans. Math. Software*, 3(SLAC-PUB-1549-REV. 2):209–226, 1976.
- [25] Garcia-Molina, Hector. Database systems: the complete book/hector garcia, molina jeffrey d. ullman, jennifer widom, 2002.
- [26] Henzinger, Monika Rauch, Raghavan, Prabhakar, and Rajagopalan, Sridhar. Computing on data streams. *External memory algorithms*, 50:107–118, 1998.
- [27] Hornby, Albert Sydney, editor. *Oxford Advanced Learner’s Dictionary of Current English*. Oxford University Press, Oxford, UK, fourth edition, 1989.
- [28] Indyk, Piotr. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *focs*, page 189. IEEE, 2000.
- [29] Indyk, Piotr. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM)*, 53(3):307–323, 2006.
- [30] Indyk, Piotr and Motwani, Rajeev. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.
- [31] Johnson, William B and Lindenstrauss, Joram. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
- [32] Johnson, William B and Schechtman, Gideon. Embedding  $l_p$  into  $l_1$ . *Acta Mathematica*, 149(1):71–85, 1982.

- [33] Kannan, J Feigenbaum S, Strauss, M, and Viswanathan, M. An approximate  $l_1$ -difference algorithm for massive data streams. *Unknown*, Unknown.
- [34] Lee, James R and Naor, Assaf. Embedding the diamond graph in  $l_p$  and dimension reduction in  $l_1$ . *Geometric & Functional Analysis GAFA*, 14(4):745–747, 2004.
- [35] Leland, Will E, Willinger, Walter, Taqqu, Murad S, and Wilson, Daniel V. On the self-similar nature of ethernet traffic. *ACM SIGCOMM Computer Communication Review*, 25(1):202–213, 1995.
- [36] Li, Ping. *Stable random projections and conditional random sampling, two sampling techniques for modern massive datasets*. Stanford, 2007.
- [37] Li, Ping. Estimators and tail bounds for dimension reduction in  $l_\alpha$  ( $0 < \alpha \leq 2$ ) using stable random projections. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 10–19. Society for Industrial and Applied Mathematics, 2008.
- [38] Matias, Yossi, Vitter, Jeffrey Scott, and Wang, Min. Wavelet-based histograms for selectivity estimation. In *ACM SIGMOD Record*, volume 27, pages 448–459. ACM, 1998.
- [39] McKee, Sally A. Reflections on the memory wall. In *CF'04: Proceedings of the 1st conference on Computing frontiers*, page 162, 2004.
- [40] Muthukrishnan, S. Data streams: Algorithms and applications (foundations and trends in theoretical computer science). *Hanover, MA: Now Publishers Inc*, 2005.
- [41] Newman, Mark EJ. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.
- [42] Strehl, Alexander and Ghosh, Joydeep. A scalable approach to balanced, high-dimensional clustering of market-baskets. In *International Conference on High-Performance Computing*, pages 525–536. Springer, 2000.
- [43] Vempala, Santosh S. *The random projection method*, volume 65. American Mathematical Soc., 2005.
- [44] Wulf, Wm A and McKee, Sally A. Hitting the memory wall: implications of the obvious. *ACM SIGARCH computer architecture news*, 23(1):20–24, 1995.

- [45] Zolotarev, VM. One-dimensional stable distributions. translated from the russian by hh mcfaden. translation edited by ben silver. translations of mathematical monographs, 65. *American Mathematical Society, Providence, RI*, 1986.

## پیوست

موضوعات مرتبط با متن گزارش پایان نامه که در یکی از گروه‌های زیر قرار می‌گیرد، در بخش پیوست‌ها آورده شوند:

۱. اثبات‌های ریاضی یا عملیات ریاضی طولانی.
۲. داده و اطلاعات نمونه (های) مورد مطالعه (Case Study) چنانچه طولانی باشد.
۳. نتایج کارهای دیگران چنانچه نیاز به تفصیل باشد.
۴. مجموعه تعاریف متغیرها و پارامترها، چنانچه طولانی بوده و در متن به انجام نرسیده باشد.

## کد میپل

```
with(DifferentialGeometry):  
with(Tensor):  
DGsetup([x, y, z], M)  
frame name: M  
a := evalDG(D_x)  
D_x  
b := evalDG(-2 y z D_x+2 x D_y/z^3-D_z/z^2)
```

## واژه‌نامه‌ی فارسی به انگلیسی

Automorphism . . . . . خودریختی	آ
د	
Degree . . . . . درجه	اسکالر . . . . . Scalar
ر	ب
microprocessor . . . . . ریزپردازنده	بالابر . . . . . Lift
ز	پ
Submodule . . . . . زیرمدول	پایا . . . . . Invariant
س	ت
Character . . . . . سرشت	تناظر . . . . . Correspondence
ص	ث
Faithful . . . . . صادقانه	ثابت‌ساز . . . . . Stabilizer
ض	ج
Inner product . . . . . ضرب داخلی	جایگشت . . . . . Permutation
ط	چ
Loop . . . . . طوقه	چند جمله‌ای . . . . . Polynomial
ظ	ح
Valency . . . . . ظرفیت	حاصل ضرب دکارتی . . . . . Cartesian product
ع	خ

Nonadjacency . . . . . عدم مجاورت

ف

Vector space . . . . . فضای برداری

ک

Complete reducibility . . . . . کاملاً تحویل پذیر

گ

Graph . . . . . گراف

م

Permutation matrix . . . . . ماتریس جایگشتی

ن

Disconnected . . . . . ناهمبند

و

Invertible . . . . . وارون پذیر

ه

Connected . . . . . همبند

ی

Edge . . . . . یال

## واژه‌نامه‌ی انگلیسی به فارسی

<b>A</b>	Lift ..... بالابر
Automorphism ..... خودریختی	<b>M</b>
<b>B</b>	Module ..... مدول
Bijection ..... دوسویی	<b>N</b>
<b>C</b>	Natural map ..... نگاشت طبیعی
Cycle group ..... گروه دوری	<b>O</b>
<b>D</b>	One to One ..... یک به یک
Degree ..... درجه	<b>P</b>
<b>E</b>	Permutation group ..... گروه جایگشتی
Edge ..... یال	<b>Q</b>
<b>F</b>	Quotient graph ..... گراف خارج‌قسمتی
Function ..... تابع	<b>R</b>
<b>G</b>	Reducible ..... تحویل پذیر
Group ..... گروه	<b>S</b>
<b>H</b>	Sequence ..... دنباله
Homomorphism ..... همریختی	<b>T</b>
<b>I</b>	Trivial character ..... سرشت بدیهی
Invariant ..... پایا	<b>U</b>
<b>L</b>	

Unique ..... منحصر بفرد

Vector space ..... فضای برداری

**V**



## **Abstract**

This page is accurate translation from Persian abstract into English.

### **Key Words:**

Write a 3 to 5 KeyWords is essential. Example: AUT, M.Sc., Ph. D, ..



**Amirkabir University of Technology**  
**(Tehran Polytechnic)**

**Department of ...**

**MSc Thesis**

**Title of Thesis**

**By**

**Name Surname**

**Supervisor**

**Dr.**

**Advisor**

**Dr.**

**Month & Year**