



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 دهبند
دی ۱۳۹۷

چکیده

با ظهور داده‌های بزرگ مقیاس و ناتوانی در نگهداری و پردازش این داده‌ها در حافظه، مسئله کاهش بعد اهمیت زیادی پیدا کرده است. یکی از روش‌های کاهش بعد، نگاشت تصادفی است که می‌تواند بر روی مه‌دادهایی که بزرگ مقیاس هستند و همچنین بر روی جریان داده‌ها، اعمال شود. مبنای این روش ضرب ماتریسی داده‌های اولیه در یک ماتریس تصویرگر است که بعد داده‌های اولیه را کاهش داده ولی اطلاعات آماری مورد نیاز در داده‌های اولیه را با دقت مورد نیاز نگه می‌دارد.

داده‌های بزرگ مقیاس داده‌هایی هستند که تعداد پارامترهای مدل از تعداد مشاهدات بیشتر است. روش تصویر تصادفی برای کاهش بعد داده‌های بزرگ مقیاس مزایای متعددی نسبت به روش‌های دیگر کاهش بعد دارد. از جمله سرعت بالا در پردازش، نیاز به حافظه محدود، قابل اعمال بر روی جریان داده و قابل اعمال در شرایطی که تعداد پارامترها از داده‌ها بیشتر است. در این پایان‌نامه این روش برای داده‌های بزرگ مقیاس با دیگر روش‌های کاهش بعد مقایسه شده است. همچنین توانایی این روش برای داده‌هایی با توزیع پایدار غیر نرمال با دیگر روش‌های کاهش بعد مقایسه شده است.

واژه‌های کلیدی:

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

فهرست مطالب

عنوان

صفحه

۱	پیشگفتار	۱
۴	کاهش بعد و داده‌های بزرگ مقیاس	۲
۵	۱-۲ کاهش بعد	۵
۵	۱-۱-۲ تحلیل مؤلفه‌های اصلی	۵
۷	۲-۲ خوشه‌بندی و شاخص‌های سنجش	۷
۷	۱-۲-۲ شاخص رند تعدیل شده	۷
۹	۳-۲ داده‌های حجیم	۹
۱۰	۱-۳-۲ داده‌های حجیم وب	۱۰
۱۲	۲-۳-۲ جریان داده‌های حجیم	۱۲
۱۳	۴-۲ چالش‌های نمونه‌گیری از داده‌های حجیم	۱۳
۱۴	۱-۴-۲ مزایای کاهش بعد با انتخاب تصادفی ابعاد	۱۴
۱۴	۲-۴-۲ معایب نمونه‌گیری تصادفی ابعاد	۱۴
۱۵	۵-۲ نگاشت تصادفی پایدار	۱۵
۱۶	۶-۲ کاربردها	۱۶
۱۷	۱-۶-۲ کاوش قوانین وابستگی	۱۷
۱۷	۲-۶-۲ وابستگی جفتی همه (فاصله‌ها)	۱۷
۱۷	۳-۶-۲ برآورد فاصله‌ها به طور آنلاین	۱۷
۱۸	۴-۶-۲ بهینه‌سازی درخواست از پایگاه داده	۱۸
۱۹	۵-۶-۲ جستجوی نزدیکترین همسایه از مرتبه‌ی زیر خطی	۱۹
۲۱	۳ نگاشت تصادفی پایدار	۲۱
۲۲	۱-۳ مسئله‌ی اصلی در نگاشت تصادفی پایدار	۲۲
۲۳	۱-۱-۳ توزیع‌های پایدار	۲۳
۲۳	۲-۱-۳ مسئله برآورد آماری	۲۳
۲۴	۲-۳ نگاشت تصادفی نرمال	۲۴

۲۵	۱-۲-۳ مشخصه‌های اصلی
۲۸	۳-۳ نگاشت تصادفی زیر گاوسی و بسیار پراکنده
۲۸	۱-۳-۳ نگاشت تصادفی زیر گاوسی
۳۱	۴-۳ نگاشت تصادفی کوشی برای l_1
۳۲	۱-۴-۳ خلاصه نتایج اصلی
۳۳	۵-۳ نگاشت تصادفی α -پایدار
۳۴	۱-۵-۳ نتایج اصلی
۳۶	۴ پیاده‌سازی نگاشت تصادفی
۳۷	۱-۴ داده‌های مورد استفاده
۳۷	۱-۱-۴ داده‌های غده تیروئید
۳۸	۲-۱-۴ داده‌های دیابت
۳۹	۳-۱-۴ داده‌های خرچنگ‌ها
۳۹	۴-۱-۴ داده‌های اسکناس سوئیس
۴۰	۵-۱-۴ داده‌های آیریس
۴۱	۶-۱-۴ مجموعه داده‌های دانه‌ها
۴۲	۷-۱-۴ مجموعه داده‌ی بیان پروتئین موش
۴۳	۲-۴ الگوریتم پیاده‌سازی نگاشت تصادفی
۴۴	۳-۴ ارزیابی روش‌های کاهش بعد
۴۶	۵ جمع‌بندی و نتیجه‌گیری و پیشنهادات
۴۷	۱-۵ بررسی کاهش بعد نرمال ($\alpha = 2$) به دو بعد
۴۷	۱-۱-۵ جداول مقایسه عملکرد خوشه‌بندی
۴۷	۲-۱-۵ نمودار فراوانی عملکرد خوشه‌بندی
۵۱	۲-۵ نتایج برای کاهش بعد نرمال به سه بعد
۵۱	۱-۲-۵ جداول مقایسه عملکرد خوشه‌بندی
۵۱	۲-۲-۵ نمودار فراوانی عملکرد خوشه‌بندی
۵۵	۳-۵ نتایج برای کاهش بعد کوشی ($\alpha = 1$) به دو بعد
۵۵	۱-۳-۵ جداول مقایسه عملکرد خوشه‌بندی

۵۵	۲-۳-۵ نمودار فراوانی عملکرد خوشه‌بندی
۵۹	۴-۵ نتایج برای کاهش بعد کوشی به سه بعد
۵۹	۱-۴-۵ جداول مقایسه عملکرد خوشه‌بندی
۵۹	۲-۴-۵ نمودار فراوانی عملکرد خوشه‌بندی
۶۳	۵-۵ نتایج برای کاهش بعد گسسته $s = 2$ به دو بعد
۶۳	۱-۵-۵ جداول مقایسه عملکرد خوشه‌بندی
۶۳	۲-۵-۵ نمودار فراوانی عملکرد خوشه‌بندی
۶۷	۶-۵ نتایج برای کاهش بعد گسسته $s = 2$ به سه بعد
۶۷	۱-۶-۵ جداول مقایسه عملکرد خوشه‌بندی
۶۷	۲-۶-۵ نمودار فراوانی عملکرد خوشه‌بندی
۷۱	۷-۵ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر α برای کاهش بعد به دو بعد
۷۴	۸-۵ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر α برای کاهش بعد به سه بعد
	۹-۵ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر s برای کاهش بعد گسسته به دو بعد
۷۸	
	۱۰-۵ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر s برای کاهش بعد گسسته به سه بعد
۸۱	
۸۵	منابع و مراجع
۹۲	پیوست

فهرست اشکال

صفحه

شکل

۱-۲ نگاشت تصادفی پایدار $B = A \times R$ ، ماتریس اولیه داده‌ها است. ۱۵

فهرست جداول

صفحه

جدول

۸	۱-۲	نشانه‌گذاری برای جدول پیش‌بینی مقایسه دو بخش
۸	۲-۲	مثال ۱ بررسی شاخص رند تعدیل شده در یک حالت ساده
۱۰	۳-۲	تعداد بازدید صفحات برای کلمات با بازخورد بالا و کلمات با بازخورد نادر
	۴-۲	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
۱۱		را نقض می‌کنند.
	۵-۲	بازدید صفحات گزارش شده توسط گوگل برای چهار کلمه و وابستگی‌های دو، سه و
۱۸		چهارتایی آن‌ها
۴۷	۱-۵	عملکرد تصویر تصادفی نرمال برای کاهش بعد به دو بعد
۵۱	۲-۵	عملکرد تصویر تصادفی نرمال برای کاهش بعد به سه بعد
۵۵	۳-۵	عملکرد تصویر تصادفی کوشی برای کاهش بعد به دو بعد
۵۹	۴-۵	عملکرد تصویر تصادفی کوشی برای کاهش بعد به سه بعد
۶۳	۵-۵	عملکرد تصویر تصادفی گسسته با $s = 1$ برای کاهش بعد به دو بعد
۶۷	۶-۵	عملکرد تصویر تصادفی گسسته با $s = 2$ برای کاهش بعد به سه بعد

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 یکه روی (M, g)
$\Omega^p(M)$	مجموعه p -فرمی‌های روی خمینه M
\mathcal{Q}	اپراتور ریچی
\mathcal{R}	تانسور انحنای ریمان
ric	تانسور ریچی
L	مشتق لی
Φ	۲-فرم اساسی خمینه تماسی
∇	التصاق لوی-چویتای
Δ	لاپلاسین ناهموار
∇^*	عملگر خودالحاق صوری القا شده از التصاق لوی-چویتای
g_s	متر ساساکی
∇	التصاق لوی-چویتای وابسته به متر ساساکی
Δ	عملگر لاپلاس-بلترامی روی p -فرم‌ها

فصل اول

پیشگفتار

عمومیت پیدا کردن داده‌های حجیم مانند داده‌های حجیم تحت وب و جریان داده‌های بزرگ در کاربردهای جدید، موجب به وجود آمدن فرصت‌های و چالش‌هایی برای مهندسين و دانشمندان شده است. [۵۲] 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 نگاشت تصادفی پایدار^۱ برای داده‌های با دم سنگین خیلی خوب کار می‌کند.

روش نگاشت تصادفی پایدار، ماتریس داده‌های اولیه $A \in \mathbb{R}^{n \times D}$ را در ماتریس تصادفی $R \in \mathbb{R}^{D \times k}$ ($k \ll D$) ضرب می‌کند و نتیجه ماتریس $B = AR \in \mathbb{R}^{n \times k}$ است. معمولاً درایه‌های ماتریس تصادفی R به صورت i.i.d^۲ از یک توزیع α -پایدار متقارن انتخاب می‌شوند ($0 < \alpha \leq 2$). ما می‌توانیم مشخصه‌های l_α را در A بر اساس B تخمین بزنیم. در مورد حالت l_2 مزیت توزیع تصادفی پایدار توسط لم JL^۳ 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_α در A را بتوان با ضریب $1 \pm \epsilon$ از روی ماتریس B تخمین زد. در رساله پینگ لی^۴ [۵۲] لمی مشابه لم JL برای $0 < \alpha < 2$ اثبات شده است. روش نگاشت تصادفی پایدار به یک مسئله تخمین آماری کاهش می‌یابد برای تخمین پارامتر مقیاس برای یک توضیح پایدار α 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^۵ و برآوردگر حداکثر بزرگنمایی.

در این پایان‌نامه ما به بررسی موارد خاصی از نگاشت تصادفی پایدار می‌پردازیم. برای نرم l_2 ارتقایی را با استفاده از اطلاعات حاشیه‌ای پیشنهاد می‌کنیم. همچنین برای حالت l_2 می‌توان ماتریس نگاشت گر را از یک توزیع زیر گاوسی^۶ بسیار کوچکتر به جای توزیع نرمال انتخاب کرد. با در نظر گرفتن محدودیت‌های معقولی می‌توان، از یک توزیع خاص زیر گاوسی استفاده کرد. این توزیع شامل $[-1, 0, 1]$ با احتمالات $\{\frac{1}{2s}, 1 - \frac{1}{s}, \frac{1}{2s}\}$ با مقادیر بسیار بزرگی برای s (به عبارتی، نگاشت تصادفی خیلی تُنک^۷) می‌تواند به

¹stable random projection

²independent and identically distributed random variables

³Johnson-Lindenstrauss

⁴Ping Li

⁵fractional power

⁶sub-Gaussian

⁷very sparse random projections

خوبی نگاشت تصادفی نرمال عمل کند. برای حالت نرم l_1 به عبارتی دیگر نگاشت تصادفی کوشی^۸ 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^۹ برای مدل سازی دقیق توزیع MLE بیان شده است.

روش نگاشت تصادفی از پراکندگی داده ها استفاده ای نمی کند. در حالی که داده های بزرگ مقیاس معمولاً بسیار پراکنده هستند. از روش نگاشت تصادفی می توان برای حل مسائل بزرگ مقیاس در علوم و مهندسی در موتورهای جستجو و سیستم های اخذ داده، پایگاه های داده، سیستم های جریان داده جدید، جبر خطی عددی و بسیاری از کارهای یادگیری ماشین و داده کاوی که شامل محاسبه حجم فاصله ها است، استفاده کرد.

در فصل بعد مروری خواهیم داشت بر ادبیات مسئله. در ابتدا کاهش بعد و روش های مرسوم برای آن را معرفی می کنیم. سپس به خوشه بندی روش های آن خواهیم پرداخت و در ادامه شاخص هایی را معرفی خواهیم کرد که کارآمدی خوشه بندی را مورد بررسی قرار می دهند. در ادامه همین فصل موضوع داده های حجیم و اهمیت آن را با معرفی مثال هایی شرح خواهیم داد. نگاشت تصادفی را به عنوان راه حلی برای مسئله نمونه گیری از داده های حجیم بررسی می کنیم و در نهایت به کاربردهای آن خواهیم پرداخت. در فصل سوم مسئله نگاشت تصادفی پایدار و انواع آن را بررسی می کنیم و مبانی این روش مزایا و معایب هر یک از انواع آن را شرح خواهیم داد. در فصل چهارم نحوه پیاده سازی و داده های مورد استفاده را شرح می دهیم. در فصل پنج هم نتایج مقایسه ای را مطرح کرده و به طبقه بندی کارهای بعدی ممکن می پردازیم.

^۸Cauchy random projection

^۹inverse Gaussian

فصل دوم

کاهش بعد و داده‌های بزرگ مقیاس

۱-۲ کاهش بعد

۱-۱-۲ تحلیل مؤلفه‌های اصلی

تحلیل چند متغیره معمولاً بر روی داده‌هایی که شامل تعداد زیادی از متغیرهای مرتبط با هم هستند انجام می‌شود. روش تحلیل مؤلفه‌های اصلی (PCA)^۱ یک ابزار کاهش بعد است که می‌توان از آن برای کاهش یک مجموعه‌ی بزرگ از متغیرها به مجموعه‌ی کوچکتري که غالب اطلاعات مجموعه‌ی بزرگ را دارد استفاده کرد.

روش تحلیل مؤلفه‌های اصلی یک تابع ریاضی است که تعدادی متغیر (احتمالاً) همبسته را به تعداد (کمتر یا مساوی) متغیرهای غیرهمبسته به نام «مؤلفه‌های اصلی» تبدیل می‌کند. بیشترین میزان اطلاعات ممکن در داده در اولین مؤلفه اصلی ثبت می‌شود. بیشترین میزان اطلاعات ممکن باقیمانده به ترتیب در مؤلفه‌های بعدی ثبت می‌شوند.

تحلیل مؤلفه‌های اصلی مشابه یک تابع چند متغیره دیگر به نام تحلیل عاملی است. این دو روش در موارد زیادی با یکدیگر اشتباه گرفته می‌شوند، و تفاوت بین این دو، و انواع تحلیل‌هایی که هر یک برایشان مناسب تر هستند به درستی تشخیص داده نمی‌شود. به طور سنتی، تحلیل مؤلفه‌های اصلی بر روی ماتریس‌های متقارن مربعی انجام می‌شود. این ماتریس‌ها می‌تواند یکی از انواع SSCP^۲ (مجموع خالص مربعات و ضرب‌های داخلی)، ماتریس کوواریانس^۳ (مجموع مقیاس شده مربعات و ضرب‌های داخلی)، یا ماتریس همبستگی^۴ (مجموع مربعات و ضرب‌های داخلی داده‌های استاندارد شده) باشد. نتایج تحلیل روی ماتریس‌های از نوع SSCP و کوواریانس تغییری ندارند، چرا که تغییرات آنها فقط در یک ضریب مقیاس قابل مشاهده است.

از ماتریس‌های همبستگی زمانی استفاده می‌شود که واریانس متغیرهای منحصر به فرد تفاوت‌های زیادی داشته باشد، و یا واحدهای اندازه‌گیری این متغیرها متفاوت باشد.

اهداف تحلیل مؤلفه‌های اصلی

تحلیل مؤلفه‌های اصلی فضای مشخصه‌ها را از تعداد زیادی متغیر به تعداد کمتری عامل کاهش می‌دهد، و یک تابع «غیر وابسته» است (یعنی نیازی وجود ندارد که یک متغیر وابسته تعیین شده باشد). تحلیل

^۱ principal components analysis

^۲ pure sums of squares and cross products

^۳ covariance

^۴ correlation

مؤلفه‌های اصلی یک روش کاهش یا فشرده سازی ابعاد است. هدف، کاهش بعد است و تضمینی وجود ندارد که این ابعاد قابل تفسیر باشند. در نهایت، انتظار آن است که زیرمجموعه‌ای از متغیرها از یک مجموعه بزرگتر انتخاب شود، به گونه‌ای که متغیرهای اولیه بیشترین همبستگی را با مؤلفه اصلی داشته باشند.

تحلیل مؤلفه‌های اصلی به دنبال رسیدن به ترکیبی خطی از متغیرها است، به گونه‌ای که بیشینه واریانس از آنها قابل استخراج باشد. پس از آن، این واریانس حذف شده و ترکیب خطی دومی جستجو می‌شود که بیشینه باقی‌مانده واریانس را توصیف می‌کند، و این روند ادامه پیدا می‌کند. به این روش، روش محور اصلی گفته می‌شود و عامل‌های متعامد غیرهمبسته را به دست می‌دهد. تحلیل مؤلفه‌های اصلی، واریانس (مشترک و یکتای) کل را شرح می‌دهند.

ویژه بردارها: مؤلفه‌های اصلی، هر دو واریانس مشترک و یکتای متغیرها را منعکس می‌کنند و بنابراین ممکن است که این روش به عنوان یک روش واریانس محور دیده شود که هم به دنبال بازتولید واریانس متغیر کل با تمام مؤلفه‌ها و هم بازتولید همبستگی‌ها است. مؤلفه‌های اصلی، ترکیب‌های خطی از متغیرهای اولیه هستند که بر اساس میزان سهمشان در به وجود آمدن واریانس در یک بعد متعامد مشخص، وزن دهی می‌شوند. وزن‌های داده شده برای هر یک از مؤلفه‌های اصلی نسبت به داده‌های اولیه، ویژه بردارها هستند.

ویژه مقدارها: ویژه مقدار یک مؤلفه، واریانس همه‌ی متغیرهایی را که به آن عامل مرتبط هستند اندازه‌گیری می‌کند. نسبت ویژه مقدارها، نسبت اهمیت توصیفی عامل‌ها با توجه به متغیرها است. اگر یک عامل دارای ویژه مقدار پایین باشد، نشانگر آن است که اثر کمی روی توصیف واریانس در متغیرها دارد، و ممکن است از آن در مقابل عامل‌های مهمتر چشم‌پوشی شود. در ویژه مقدارها میزان تغییر در نمونه کل حساب شده است.

ویژه مقدار یک عامل ممکن است حاصل جمع مربعات عامل‌های تمامی متغیرها باشد. باید توجه شود که ویژه مقدارهای مرتبط با راه حل‌های دورانی و غیردورانی متفاوت خواهند بود، اگرچه مقدار کل آنها یکسان است.

برای به دست آوردن واریانس همه متغیرها که توسط عامل لحاظ می‌شود، مجموع مربعات بارگذاری‌های عامل برای آن عامل (ستون) را جمع کرده، و بر تعداد متغیرها تقسیم می‌کنیم. (توجه کنید که تعداد متغیرها برابر با مجموع واریانس آنهاست، چرا که واریانس یک متغیر استاندارد شده مساوی با ۱ است). این کار مشابه تقسیم ویژه مقدار عامل بر تعداد متغیرها است.

امتیاز PC: این امتیازها، امتیازهای هر نمونه (ردیف) در هر عامل (ستون) هستند. امتیاز عامل برای

یک نمونه و برای یک عامل داده شده، به صورت مجموع حاصل ضرب امتیاز استاندارد نمونه در هر متغیر با بارگذاری عامل مربوطه برای عامل داده شده محاسبه می‌شود. [۱]

۲-۲ خوشه‌بندی و شاخص‌های سنجش

۱-۲-۲ شاخص رند تعدیل شده

برای مقایسه نتایج خوشه‌بندی در کنار شاخص‌های بیرونی، نیازمند معیارهای مورد توافق هستیم. برای مجموعه‌ی n عضوی $S = \{O_1, \dots, O_n\}$ ، فرض کنید که $U = \{u_1, \dots, u_R\}$ و $V = \{v_1, \dots, v_R\}$ نمایانگر دو افراز متفاوت از عضوهای S هستند به گونه‌ای که $U \cap V = \emptyset$ و $U \cup V = S$. فرض کنید که U شاخص خارجی و V نتیجه خوشه‌بندی است. فرض کنید a تعداد جفت عضوهای U باشد که در یک کلاس یکسان و b تعداد جفت عضوهای V هستند، c تعداد جفت عضوهای U و V هستند، و d تعداد جفت عضوهای V باشد که در یک کلاس متفاوت U و V هستند. می‌توان مقادیر a و d را به عنوان موارد توافق، و مقادیر b و c را به عنوان موارد عدم توافق تعریف کرد. شاخص رند [۶۰] به سادگی به صورت $\frac{a+b}{a+b+c+d}$ تعریف می‌شود. شاخص رند عددی بین ۰ و ۱ است. زمانی که دو خوشه کاملاً در حالت توافق باشند، مقدار شاخص رند برابر با ۱ خواهد بود.

یکی از مشکلات شاخص رند آن است که مقدار مورد انتظار شاخص رند دو خوشه تصادفی یک مقدار ثابت (به عنوان مثال صفر) نیست. مقدار شاخص رند تعدیل شده^۶ پیشنهادی توسط هوبرت و اربی^۷ [۴۲] بر پایه این فرض است که توزیع فوق هندسی^۸ تعمیم یافته برای مدل تصادفی استفاده می‌شود، به عبارت دیگر خوشه‌های U و V به شکلی تصادفی انتخاب می‌شوند که تعداد عضوهای کلاس‌ها و خوشه‌ها ثابت باشد. فرض کنید n_{ij} تعداد عضوهای U باشد که هم در کلاس u_i و هم در خوشه v_j هستند. در نظر بگیرید n_i و $n_{.j}$ به ترتیب تعداد اعضا در کلاس u_i و خوشه‌ی v_j هستند. تمامی این نشانه‌گذاری‌ها در جدول ۱-۲ بیان شده‌اند.

شکل کلی شاخص با در نظر گرفتن مقادیر انتظاری ثابت بدین شکل است که

مقدار مورد انتظار شاخص-شاخص
مقدار مورد انتظار شاخص-حداکثر شاخص

^۵Rand index

^۶adjusted Rand index

^۷Hubert and Arabie

^۸hypergeometric

جدول ۲-۱: نشانه‌گذاری برای جدول پیش‌بینی مقایسه دو بخش

Class \ Cluster	v_1	v_2	\dots	v_C	Sums
u_1	n_{11}	n_{12}	\dots	n_{1C}	$n_{1.}$
u_2	n_{21}	n_{22}	\dots	n_{2C}	$n_{2.}$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
u_R	n_{R1}	n_{R2}	\dots	n_{RC}	$n_{R.}$
Sums	$n_{.1}$	$n_{.2}$	\dots	$n_{.C}$	$n_{..} = n$

که از بالا به یک محدود است و زمانی که شاخص مقدار مورد انتظار را داشته باشد صفر می‌شود. بر اساس مدل جنرال فوق‌هندسی، می‌توان نشان داد [۴۳]:

$$E \left[\sum_{i,j} \binom{n_{ij}}{2} \right] = \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} \right] / \binom{n}{2} \quad (1-2)$$

عبارت $a + b$ می‌تواند به تبدیل خطی $\sum_{i,j} \binom{n_{ij}}{2}$ ساده‌سازی شود. شاخص رند تعدیل شده می‌تواند به شکل زیر ساده‌سازی شود: [۴۳]

$$\frac{\sum_{i,j} \binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2} \right] - \left[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2} \right] / \binom{n}{2}} \quad (2-2)$$

با مثالی به بیان تعدیل انجام شده می‌پردازیم. **جدول ۲-۲** یک جدول پیش‌بینی به شکل جدول پیش‌بین در **جدول ۱-۲** است.

جدول ۲-۲: مثال ۱ بررسی شاخص رند تعدیل شده در یک حالت ساده

Class \ Cluster	v_1	v_2	v_3	Sums
u_1	1	1	0	2
u_2	1	2	1	4
u_3	0	0	4	4
Sums	2	3	5	$n = 10$

a به عنوان تعداد جفت اشیاء در که یک رده در U و یک خوشه در V قرار دارند. بنابراین a می‌تواند به شکل $\sum_{i,j} \binom{n_{ij}}{2}$ نوشته شود. در مثال **جدول ۲-۲**، $a = \binom{2}{2} + \binom{4}{2} = 7$. b به عنوان تعداد جفت اشیاء که هر دو در یک رده هستند در U ولی در V در دو خوشه‌ی متفاوت جای دارند.

با توجه به الگوی نوشتار **جدول ۱-۲** b را می‌تواند به شکل $\sum_i \binom{n_i}{2} - \sum_{i,j} \binom{n_{ij}}{2}$ بازنویسی کرد. در مثال **جدول ۲-۲**، $b = \binom{2}{2} + \binom{4}{2} + \binom{4}{2} - 7 = 6$. همچنین c به طور مشابه تعداد جفت اشیاءی که در یک خوشه در V قرار دارند ولی در یک رده در U قرار ندارند، تعریف می‌شود. پس c به شکل $\sum_j \binom{n_{.j}}{2} - \sum_{i,j} \binom{n_{ij}}{2} = \binom{2}{2} + \binom{3}{2} + \binom{5}{2} - 7 = 7$ و d تعداد جفت اشیاءی است که نه در U در یک رده قرار دارند و نه در V در یک خوشه. از آنجا که $a+b+c+d = \binom{n}{2}$ پس $a+b+c+d = \binom{n}{2} - 7 - 6 - 7 = 25$. $d = \binom{10}{2} - 7 - 6 - 7 = 25$ پس $a+b+c+d = \binom{n}{2}$ است. در حالی که شاخص رند $\frac{7+25}{45} = 0.711$ **جدول ۲-۲** برای مقایسه دو بخش در مثال **جدول ۲-۲** $\frac{7-14 \times 13/45}{(14+13)/2-14 \times 13/45} = 0.313$ (برای تعریف شاخص رند تعدیل شده به رابطه‌ی (۲-۲) مراجعه کنید). شاخص رند بسیار بیشتر از شاخص رند تعدیل شده است، که این موضوع معمولی است. از آنجا که شاخص رند بین 0 و 1 است، مقدار مورد انتظار شاخص رند (که مقداری ثابت نیست) باید بزرگتر مساوی 0 باشد. از طرفی دیگر، مقدار مورد انتظار شاخص رند تعدیل شده 0 است و بیشترین مقدار آن هم 1 می‌باشد. بنابراین، محدوده‌ی بیشتری از مقادیر توسط شاخص رند تعدیل شده بیان می‌شوند و حساسیت شاخص زیاد می‌شود.

در [۵۷]، اندیس‌های مختلفی برای تطابق دو تفکیک مختلف در خوشه‌بندی و برای تعداد مختلفی خوشه مورد بررسی قرار گرفته‌اند و پیشنهاد آن‌ها شاخص رند تعدیل شده بود. ما شاخص رند تعدیل شده را به عنوان سنج‌ای برای توافق معیار خارجی و نتایج خوشه‌بندی قرار دادیم. [۶۶]

۳-۲ داده‌های حجیم

عبارات زیر از سایت *Information Week* نقل قول شده‌اند [۲]:

- 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 10^{15} بایت)، اداره می‌شوند.

پدیده نو ظهور مجموعه داده‌های حجیم، چالش‌های محاسباتی در بسیاری کاربردهای علمی و تجاری به وجود آورده است. شامل اخترفیزیک، بیوتکنولوژی، جمعیت شناسی^۹، مالی، سیستم‌های اطلاعات جغرافیایی، دولت، دارو، ارتباطات از راه دور، محیط زیست، اینترنت.

۲-۳-۱ داده‌های حجیم وب

وب چقدر بزرگ است؟ **جدول ۲-۳** 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 ۲-۳** همچنین نشان می‌دهد که حتی کلماتی که به ندرت کاربرد دارند هم تعداد زیادی بازدید دارند.

جدول ۲-۳: تعداد بازدید صفحات برای کلمات با بازخورد بالا و کلمات با بازخورد نادر

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

کلماتی با بازخورد معمولی چه میزان بازدید دارند؟ برای جواب این سوال ما به طور تصادفی ۱۵ صفحه از لغتنامه‌ی آموزشی انتخاب می‌کنیم. [۴۱] (لغتنامه‌ای با ۵۷,۱۰۰ 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^{۱۰} تحت پوشش قرار نمی‌گیرند یا کلماتی که هنوز به لغتنامه‌های منتشر 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 $D \approx 10^{10}$ بزرگ خواهد شد. در اینجا عدد (i, j) در A تعداد ظهور واژه i در سند j

^۹demographics

^{۱۰}Oxford english dictionary

را نشان می‌دهد.

کارکردن با ماتریسی در این ابعاد بزرگ چالش برانگیز است. برای مثال، شاخص LSI^{۱۱} [۳۰] و یک مدل موضوعی فراگیر، از SVD^{۱۲} بر روی ماتریس عبارت به سند استفاده می‌کند. که انجام این عملیات در ابعاد وب قطعا 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 [۵۶، ۶۵]. 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 قطعا 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 ۲-۴ نشان می‌دهد که تخمین‌های بیان شده با موتورهای جستجوی فعلی، همیشه سازگار نیستند.

جدول ۲-۴: 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.

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

با اینکه، تعداد کل واژه‌های انگلیسی (که به‌طور صحیح نوشته شده‌اند) هم اکنون شگفت‌انگیز 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^{۱۳}). معمولا بهتر

¹¹latent semantic indexing

¹²singular value decomposition

¹³bag-of-words

است سند به عنوان یک بردار از لغات به صورت 1 پیوسته ^{۱۴} [۲۱] بیان شود. برای مثال، با استفاده از مدل ۳ پیوسته، جمله‌ی "It is a nice day" به مجموعه‌ی زیر تجزیه می‌شود. "a", "is a nice", "it is 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.

۲-۳-۲ جریان داده‌های حجیم

در بسیاری کاربردهای جدید پردازش داده، جریان‌های داده‌ی حجیم نقش بنیادی دارند. جریان‌های داده‌ای که از روترهای اینترنت، سوئیچ‌های تلفن، رصد اتمسفر، شبکه‌های سنسور، شرایط ترافیکی بزرگراهی، داده‌های مالی و غیره [۸، ۵۸، ۲۹، ۱۴، ۴۴، ۴۹، ۴۰] 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_α . در برخی کاربردها (برای مثال، طبقه‌بندی صدا/محتوا و جدا سازی) نیاز است یک مدل یادگیری آماری برای رده‌بندی ^{۱۵} یا خوشه‌بندی ^{۱۶} جریان داده‌های حجیم تدوین 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. ^{۱۷} 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.

^{۱۴}l-shingles

^{۱۵}classification

^{۱۶}clustering

^{۱۷} 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.

۴-۲ چالش‌های نمونه‌گیری از داده‌های حجیم

در حالی که مسائل جذاب و چالش‌برانگیزی با ورود داده‌های حجیم شکل گرفته‌اند، این پایان‌نامه بر روی توسعه‌ی روش‌های کاهش بعد 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^{۱۸} AA^T در آمار و یادگیری ماشین معمول است. AA^T بیانگر همه‌ی 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_α (که با $d_{(\alpha)}$ نمایش داده می‌شود)^{۱۹} با عبارات زیر تعریف می‌شوند:

$$a = u_1^T u_2 = \sum_{i=1}^D u_{1,i} u_{2,i} \quad (۳-۲)$$

$$d_{(\alpha)} = \sum_{i=1}^D |u_1 - u_2|^\alpha \quad (۴-۲)$$

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 اندازه‌ی بزرگ مقیاس هستند که نمی‌توان به طور کارا آن‌ها را مدیریت کرد، انتخاب تصادفی ابعاد 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 داده‌ی $A \in \mathbb{R}^{n \times D}$ انتخاب تصادفی ابعاد^{۲۰}، k ستون را از ماتریس داده به طور یکنواخت و تصادفی انتخاب می‌کند.

کاهش بعد از این جهت سودمند است که هم دوره‌های 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

¹⁸Gram matrix

^{۱۹} ما فاصله l_α 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، l_2 ، 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 فاصله‌ی l_2 ».

²⁰random coordinate sampling

افزایش می‌یابد [۵۶]. این پدیده به عنوان «دیوار حافظه»^{۲۱} شناخته می‌شود. [۵۶، ۶۵] بنابراین در کاربردهایی که شامل مجموعه داده‌های حجیم می‌شوند، بحرانی‌ترین کار بیان کردن داده‌ها است. برای مثال، از طریق کاهش بعد با فرمی فشرده برای قرارگیری در ابعاد حافظه در دسترس.

۱-۴-۲ مزایای کاهش بعد با انتخاب تصادفی ابعاد

نمونه‌گیری تصادفی ابعاد 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_α (برای هر مقداری از α)

۲-۴-۲ معایب نمونه‌گیری تصادفی ابعاد

با این حال نمونه‌گیری تصادفی ابعاد 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. [۵۹، ۳۲، ۲۷، ۵۱] 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) [۵۹].

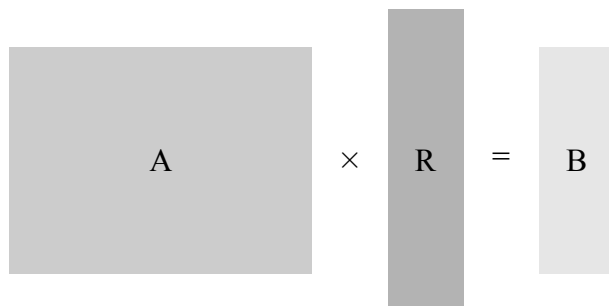
- این روش داده‌های پراکنده را به خوبی مدیریت نمی‌کند. بسیاری از داده‌های بزرگ مقیاس به شدت پراکنده هستند، به عنوان مثال، داده‌های متنی [۳۱] و داده‌های بر اساس بازار [۶۲، ۹]. به جز برخی واژه‌های کاربردی مانند "A" و "The" بیشتر لغات با نسبت بسیار کمی در مستندات ظاهر می‌شوند (1% <) اگر ما داده‌ها را با در نظر گرفتن تعدادی از ستون‌های ثابت، کاهش بعد دهیم. خیلی محتمل است که بیشتر داده‌های (مقادیر غیر صفر) را از دست بدهیم. به خصوص موارد جذابی که در دو نمونه، دو ستون با هم غیر صفر شده‌اند.

²¹memory wall

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

۵-۲ نگاشت تصادفی پایدار

شکل ۱-۲، ایده نگاشت تصادفی را نشان می‌دهد. ایده اصلی نگاشت 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 نگاشت 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: نگاشت تصادفی پایدار $B = A \times R$ ، ماتریس اولیه داده‌ها است.

ماتریس نگاشت‌گر $R \in \mathbb{R}^{D \times k}$ معمولاً از داریه‌های مستقل هم توزیع (i.i.d) یک توزیع متقارن α -پایدار پر شده است. [۶۷] (بنابراین نام این روش «نگاشت تصادفی پایدار» است). بر اساس مشخصات توزیع‌های α -پایدار، داده‌های نگاشت شده هم از توزیع α -پایدار پیروی می‌کنند. که بر اساس آن‌ها شاخص‌های l_α و فاصله دودویی l_α در A تخمین زده می‌شوند و می‌توانیم داده‌های اصلی را دور بریزیم. موفقیت نگاشت تصادفی پایدار توسط لم JL^{۲۲} [۴۷] 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 α کوچکتر از ۲ ($\alpha < 2$) l_α صادق نیست. در صورتی که لازم باشد از برآوردگرهایی استفاده کنیم که متریک باشند (در نامساوی مثلثی صدق کنند). به این نتیجه «عدم امکان»^{۲۳} گفته می‌شود. [۱۹، ۵۰، ۲۲] خوشبختانه شامل برآوردگرهایی که متریک نیستند نمی‌شود. در این پایان‌نامه ما در مورد برآوردگرهای گوناگونی که متریک نیستند صحبت خواهیم کرد.

²²Johnson-Lindenstrauss

²³impossibility

شامل: میانگین هندسی^{۲۴}، میانگین هارمونیک^{۲۵}، توان نسبی^{۲۶} و همچنین حداکثر بزرگنمایی.

۶-۲ کاربردها

علاقه‌ی زیادی به فنون کاهش بعد وجود دارد که در کاربردهای زیادی مورد استفاده قرار می‌گیرند. مانند: قانون وابستگی^{۲۷} [۱۶، ۱۷]، خوشه‌بندی، بهینه‌سازی درخواست^{۲۸} [۵۴، ۲۴]، تشخیص تکراری^{۲۹} [۲۱، ۱۵] و بسیاری موارد دیگر. روش‌های کاهش بعد هر چه بیشتر و بیشتر برای مجموعه‌های بزرگتر اهمیت پیدا می‌کنند.

طرح برودر^{۳۰} [۲۱] در ابتدا برای تشخیص صفحات وب تکراری معرفی شد. URL‌های زیادی به HTML‌های مشابه (یا تقریباً مشابه) اشاره می‌کنند. جواب‌های برآورد 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)^{۳۱} 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. [۱۸] 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^{۳۲} بسازیم و کلیتی از عبارات (برای تخمین ارتباط لغات) یا اسناد (برای تخمین شباهت اسناد)

²⁴geometric mean

²⁵harmonic mean

²⁶fractional power

²⁷association rules

²⁸query optimization

²⁹duplicate detection

³⁰Broder's sketch

³¹information retrieval

³²inverted index

نگهداری کنیم.

۱-۶-۲ کاوش قوانین وابستگی

تحلیل‌های مبتنی بر بازار و قوانین وابستگی [۱۰، ۱۱، ۱۲] ابزارهای مناسبی برای کاوش پایگاه داده‌های تجاری هستند. پایگاه داده‌های تجاری دارند روز به روز بزرگتر و تُنک‌تر می‌شوند. [۹، ۶۲] 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.

۲-۶-۲ وابستگی جفتی همه (فاصله‌ها)

در کاربردهای مختلفی شامل رده‌بندی بر مبنای فاصله یا خوشه‌بندی و مدل‌سازی زبان با بای‌گرام^{۳۳} [۲۶] 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 سطرهاى A است. محاسبه مستقیم می‌تواند به شدت زمان‌بر باشد. همچنین، به طور خاص زمانی که ماتریس داده آنقدر بزرگ است که در حافظه فیزیکی جا نمی‌شود. محاسبه به طور خاص بسیار ناکارآمد خواهد بود.

۳-۶-۲ برآورد 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^{۳۴} همه فاصله‌های جفتی و وابستگی‌ها در A ، $O(n^2)$ فضا مصرف می‌کند. که می‌تواند برای حافظه‌ی فیزیکی بسیار بزرگ 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 برداشت‌ها^{۳۵} در حافظه نگهداری شوند و همه‌ی فاصله‌ها به طور آنلاین، زمانی که مورد نیاز باشد، محاسبه شوند.

³³bi-gram

³⁴materializing

³⁵sketches

۴-۶-۲ بهینه‌سازی درخواست از پایگاه داده

در پایگاه داده‌ها یک وظیفه‌ی بسیار مهم تخمین جوین‌های^{۳۶} چندراهی است، که تاثیر زیادی بر روی کارایی سیستم دارد. [۳۸] بر اساس تخمین دوراهی، سهراهی و حتی جوین‌هایی از مرتبه‌ی بالاتر، بهینه‌گرهای درخواست یک نقشه برای کمینه کردن تابع هزینه می‌سازند (برای مثال، نوشتن‌های میانی^{۳۷}). بهینه بودن اهمیت بسیاری دارد زیرا مثلاً نمی‌خواهیم زمان بیشتری برای بهینه‌سازی نقشه نسبت به زمان اجرای آن تلف کنیم.

ما از مثال «Governator» برای نمایش کاربرد تخمین دو و چند راهه برای بهینه کردن درخواست استفاده می‌کنیم.

جدول ۲-۵: بازدید صفحات گزارش شده توسط گوگل برای چهار کلمه و وابستگی‌های دو، سه و چهارتایی آن‌ها

	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 ۲-۵: 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" \cap)

"Austria" \cap "Governor" \cap "Terminator" این نقشه 579,100 نوشتن میانی بعد از اولین و دومین

جوین دارد. یک بهینه‌سازی می‌تواند ("Terminator" \cap ("Austria" \cap "Schwarzenegger"))

³⁶joins

³⁷intermediate writes

”Governor” باشد که 579,100 را به 136,000 کاهش می‌دهد.

۲-۶-۵ جستجوی نزدیکترین همسایه از مرتبه‌ی زیر خطی

محاسبه‌ی نزدیکترین همسایه در بسیاری کاربردها از اهمیت زیادی برخوردار است. با این حال، به دلیل «نفرین ابعاد»^{۳۸} راه حل فعلی برای پیدا کردن بهینه‌ی نزدیکترین همسایه‌ها (حتی به طور تقریبی) اصلاً رضایت بخش نیست. [۳۵، ۴۶]

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. دوم، پیدا کردن نزدیکترین همسایه‌های یک داده ممکن است $O(nD)$ هزینه‌بر باشد که می‌تواند به شدت زمان‌بر شود.

با این حال، روش 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، $O(nk)$ است 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) [۳۶، ۳۷] و LSH^{۳۹} [۱۳، ۲۸، ۴۶] این الگوریتم‌ها معمولاً با یک فضای متریک کار می‌کنند (که در آن نامساوی مثلثی برقرار است). برای مثال، فضای l_α زمانی که $\alpha \geq 1$ باشد یک متریک است. زمانی که به دنبال نزدیکترین همسایه‌ها در l_α ($\alpha > 1$) 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^{۴۰}، یکی از مشکلات اصلی این است که این الگوریتم‌ها برای کاهش هزینه‌ی محاسباتی به حافظه‌ی ابر خطی^{۴۱} نیاز دارند که می‌تواند

³⁸curse of dimensionality

³⁹locality-sensitive hashing

⁴⁰memory wall

⁴¹super-linear memory

مشکل ساز باشد. [۴۶] به طرح کلی برای LSH توجه کنید که ترکیبی از هش^{۴۲} و نگاشت تصادفی است. متأسفانه این طرح به دلیل هزینه‌ی زیاد پیش پردازش غیر کاربردی است. [۴۶]

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 برآوردهای 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 برآوردهای ما غیر متریک هستند و بنابراین طراحی یک الگوریتم هوشمند و تحلیل‌های تئوری ممکن است سخت باشد، با این حال غیر ممکن نیست.

⁴²hash

فصل سوم

نگاشت تصادفی پایدار

روش نگاشت تصادفی پایدار (زیرنویس ۱) [۴۸، ۴۴، ۴۵، ۱۴، ۶۳، ۵۳] یک روش پرکاربرد در داده‌کاوی و یادگیری ماشین است. با این روش به طور کارا فاصله l_α ($0 < \alpha \leq 2$) در داده‌های حجیم (برای مثال: وب یا جریان داده‌های حجیم) محاسبه می‌شود. در این روش حافظه‌ی کمی استفاده شده و فقط یک بار پایش داده‌ها کافی است.

همانطور که در شکل ۱-۲ مشاهده می‌کنید. ایده نگاشت تصادفی پایدار، ضرب ماتریس داده‌ها $A \in \mathbb{R}^{n \times D}$ در ماتریس تصادفی $R \in \mathbb{R}^{D \times k}$ ($k \ll D$) است که حاصل یک ماتریس نگاشت شده‌ی $B \in \mathbb{R}^{n \times k}$ است. درایه‌های ماتریس تصادفی R به طور i.i.d. (مستقل و هم توزیع) از یک توزیع α -پایدار^۱ حاصل می‌شوند. به همین دلیل به این روش «نگاشت تصادفی پایدار» گفته می‌شود. به این نکته توجه کنید که توزیع ۲-پایدار معادل توزیع نرمال و توزیع پایدار ۱-پایدار معادل کوشی^۲ است. حالت خاص نگاشت تصادفی نرمال (به عبارت دیگر $\alpha = 2$) نسبتاً به خوبی مورد بررسی قرار گرفته است. به رساله [۶۳] مراجعه کنید. بنابراین، بخش اعظم این پایان‌نامه به نگاشت تصادفی پایدار $\alpha < 2$ اختصاص یافته است.

پس از مروری بر حالت کلی نگاشت تصادفی پایدار $0 < \alpha \leq 2$ ، جزئیات بیشتری در خصوص حالت l_2 مورد بررسی قرار می‌گیرد. سپس ارتقاء روش با استفاده از اطلاعات حاشیه‌ای^۳ بررسی می‌شود. در ادامه، نگاشت تصادفی نرمال ساده‌سازی می‌شود. این کار با نمونه‌برداری R از حالت توزیع گسسته‌ی سه‌نقطه‌ای $\{-1, 0, 1\}$ با احتمالات $\{\frac{1}{2s}, 1 - \frac{1}{s}, \frac{1}{2s}\}$ انجام می‌شود. این حالت، یک حالت خاص توزیع‌های زیرگوسی^۴ است. سپس نرم l_1 ^۵ مورد بررسی قرار گرفته و در ادامه حالت کلی $0 < \alpha \leq 2$ مورد بحث قرار می‌گیرد.

۱-۳ مسئله‌ی اصلی در نگاشت تصادفی پایدار

مسئله اصلی نگاشت تصادفی پایدار یک مسئله‌ی برآورد آماری است. همانطور که بیان شد، ماتریس داده‌ی $A \in \mathbb{R}^{n \times D}$ را در ماتریس تصادفی $R \in \mathbb{R}^{D \times k}$ ضرب می‌کنیم تا ماتریس بسیار کوچکتر $B = A \times R \in \mathbb{R}^{n \times k}$ را به دست بیاوریم. هدف این است که مشخصات آماری A بر اساس ماتریس B استنتاج شوند. (شامل نرم و فاصله)

^۱ α -stable distribution

^۲Cauchy

^۳marginal information

^۴sub-Gaussian

^۵Cauchy random projection

بدون از دست دادن کلیت، ما بر ۲ سطر اول A ، $u_1, u_2 \in \mathbb{R}^D$ و دو سطر اول در B ، $v_1, v_2 \in \mathbb{R}^k$ تمرکز می‌کنیم. تعریف می‌کنیم $R = \{r_{ij}\}_{i=1}^D \{j=1}^k$ بنابراین:

$$v_{1,j} = \sum_{i=1}^D r_{ij} u_{1,i}, \quad v_{2,j} = \sum_{i=1}^D r_{ij} u_{2,i}, \quad x_j = v_{1,j} - v_{2,j} = \sum_{i=1}^D r_{ij} (u_{1,i} - u_{2,i}). \quad (1-3)$$

۳-۱-۱ توزیع‌های پایدار

به طور معمول $r_{ij} \sim S(\alpha, 1)$ و به طور i.i.d. استخراج می‌شود. همچنین در ادامه ما حالت‌های ساده‌تری را هم مورد بررسی قرار می‌دهیم. در اینجا $S(\alpha, 1)$ بیانگر یک توزیع متقارن α -پایدار تصادفی است [۶۷] با پارامتر اندیس α و پارامتر مقیاس ۱.

یک متغیر تصادفی z در صورتی متقارن α -پایدار است که تبدیل فوریه آن به شکل زیر باشد.

$$E(\exp(\sqrt{-1}zt)) = \exp(-d|t|^\alpha) \quad (2-3)$$

که $d > 0$ پارامتر مقیاس است. ما می‌نویسیم $z \sim S(\alpha, d)$ که به طور کلی شکل بسته‌ای برای تابع چگالی ندارد. به جز حالت $\alpha = 2$ (نرمال) و $\alpha = 1$ (کوشی).

۳-۱-۲ مسئله برآورد آماری

با توجه به خواص تبدیل فوریه، به راحتی می‌توان نشان داد که داده‌های نگاشت شده هم از توزیع α -پایدار پیروی می‌کنند که در این حالت پارامتر مقیاس مشخصه l_α ی (نرم‌ها، فاصله‌ها) داده‌های اصلی در A است. به طور خاص:

$$v_{1,j} \sim S\left(\alpha, \sum_{i=1}^D |u_{1,i}|^\alpha\right), \quad v_{2,j} \sim S\left(\alpha, \sum_{i=1}^D |u_{2,i}|^\alpha\right), \quad (3-3)$$

$$x_j = v_{1,j} - v_{2,j} \sim S\left(\alpha, d_{(\alpha)} = \sum_{i=1}^D |u_{1,i} - u_{2,i}|^\alpha\right). \quad (4-3)$$

بنابراین، مسئله ما به برآورد پارامتر مقیاس از k نمونه i.i.d. $x_j \sim S(\alpha, d_{(\alpha)})$ ، تقلیل پیدا می‌کند. به این خاطر که هیچ شکل بسته‌ای برای تابع چگالی به جز در حالت $\alpha = 0, 1/2, 1, 2$ وجود ندارد، فرآیند تخمین خود مسئله‌ی جالبی است اگر به دنبال برآوردگرهایی بگردیم که هم به طور آماری دقیق باشند

و هم از لحاظ محاسباتی کارا باشند.

یک موضوع مربوط و نزدیک هم تعیین اندازه نمونه k است. روش استاندارد محدود کردن احتمال دم است $\Pr(|\hat{d}_{(\alpha)} - d_{(\alpha)}| > \epsilon d_{(\alpha)}) \leq 2 \exp\left(-k \frac{\epsilon^2}{G}\right)$ که $\hat{d}_{(\alpha)}$ برآوردگری برای $d_{(\alpha)}$ است و ϵ دقت مورد نظر است (معمولا $0 < \epsilon < 1$). به طور ایده‌آل امیدوار هستیم نشان دهیم ^۶:

$$\Pr(|\hat{d}_{(\alpha)} - d_{(\alpha)}| > \epsilon d_{(\alpha)}) \leq 2 \exp\left(-k \frac{\epsilon^2}{G}\right), \quad (5-3)$$

برای برخی مقادیر ثابت G که می‌تواند تابعی از ϵ هم باشد. برای ماتریس داده‌ی $\mathbf{A} \in \mathbb{R}^{n \times D}$ ، در مجموع $\frac{n(n-1)}{2} < \frac{n^2}{2}$ جفت فاصله وجود دارد. ما معمولا علاقمندیم که احتمالات دم را به طور همزمان برای همه‌ی جفت‌ها محدود کنیم.

۲-۳ نگاشت تصادفی نرمال

برای کاهش بعد در نرم l_2 ، روش نگاشت تصادفی نرمال ماتریس داده‌ی اولیه $\mathbf{A} \in \mathbb{R}^{n \times D}$ را در ماتریس تصادفی $\mathbf{R} \in \mathbb{R}^{D \times k}$ ($k \ll D$) با درایه‌های i.i.d. از $N(0, 1)$ ضرب می‌کنیم، تا ماتریس نگاشت شده‌ی $\mathbf{B} \in \mathbb{R}^{n \times k}$ حاصل شود. تحلیل‌های مربوط به نگاشت تصادفی نرمال نسبتا ساده است. برای مثال، در ادامه به شکل سراسری یک نسخه از لم JL^۷ [۴۷] را برای حالت l_2 استنتاج می‌کنیم.

ما در ابتدا برخی خواص اولیه نگاشت تصادفی نرمال را بیان می‌کنیم و سپس بر روی اطلاعات حاشیه تمرکز می‌کنیم تا تخمین‌ها را بهینه کنیم. حاشیه‌ها (به عبارت دیگر، نرم l_2 برای هر خط در \mathbf{A}) (معمولا در ابتدا در دسترس هستند (برای مثال، از طریق نرمال سازی داده‌ها). ولی حتی در حالتی که در دسترس نیستند، محاسبه‌ی نرم l_2 برای تمام سطرهاى \mathbf{A} فقط نیازمند یکبار مرور داده‌ها است که هزینه‌ای از $O(nD)$ دارد که قابل صرف‌نظر است. ^۸ از آنجا که اعمال نگاشت تصادفی $\mathbf{A} \times \mathbf{R}$ هم اکنون

^۶ بنابر قضیه حدمرکزی برآوردگر $\hat{d}_{(\alpha)}$ بر اساس k نمونه تحت شروط ساده‌ای به حالت نرمال همگرا می‌شود. بنابر محدوده‌ی دم نرمال می‌دانیم که حداقل برای پارامترهای خاصی $\Pr(|\hat{d}_{(\alpha)} - d_{(\alpha)}| \geq \epsilon d_{(\alpha)}) \leq 2 \exp\left(-k \frac{\epsilon^2}{2V}\right)$ باید صادق باشد. در اینجا $\frac{V}{k}$ واریانس مجانبی $\hat{d}_{(\alpha)}$ است. بنابراین، حداقل برای آزمون درستی، می‌توانیم با بررسی این که آیا $\lim_{\epsilon \rightarrow 0+} G = 2V$ چک کنیم که محدوده‌ی دم نسبت مطلوب را دارا باشد.

^۷Johnson-Lindenstrauss

^۸ این وضعیتی برای زمانی که با جریان داده‌های داینامیک سر و کار داریم اندکی متفاوت است. در جریان داده‌های ما معمولا به دنبال اطلاعات آماری یک جریان داده هستیم تا اختلاف میان دو جریان داده را مد نظر داشته باشیم. به عبارت دیگر، محاسبه نرم l_2 حاشیه‌ای گاهی اوقات هدف اصلی است. به دلیل ذات دینامیک جریان داده‌ها (برای مثال، به روز شدن

هزینه‌ای از مرتبه $O(nDk)$ دارد.

در این بخش، ما این قاعده مرسوم تبعیت در ادبیات نگاشت تصادفی [۶۳] پیروی می‌کنیم و تعریف می‌کنیم $B = \frac{1}{\sqrt{k}}AR$.

۱-۲-۳ مشخصه‌های اصلی

ما فرض می‌کنیم یک ماتریس داده $A \in \mathbb{R}^{n \times D}$ و یک ماتریس نگاشت گر $R \in \mathbb{R}^{D \times k}$ که به طور i.i.d. از $N(0, 1)$ تولید شده است. در نظر می‌گیریم $B = \frac{1}{\sqrt{k}}AR$. در نظر بگیرید u_i^T سطر i ام ماتریس A باشد، و سطر متناظر در B ، v_i^T باشد. برای راحتی بر روی دو سطر اول A یعنی u_1 و u_2 همچنین دو سطر اولیه v_1 و v_2 در B تمرکز می‌کنیم. تعریف می‌کنیم:

$$a = u_1^T u_2, \quad m_1 = \|u_1\|^2, \quad m_2 = \|u_2\|^2, \quad d = \|u_1 - u_2\|^2 = m_1 + m_2 - 2a \quad (۶-۳)$$

$\|v_1 - v_2\|$ ، فاصله‌ی l_2 نمونه و $v_1^T v_2$ ضرب داخلی نمونه، برآوردگرهای نااریبی از d و a هستند. لم ۱ واریانس و تابع مشخصه‌ی $v_1^T v_2$ را مشخص می‌کند. اثبات در [۵۲].

لم ۱: $u_1, u_2 \in \mathbb{R}^D$ داده شده‌اند و یک ماتریس تصادفی $R \in \mathbb{R}^{D \times k}$ شامل درایه‌های i.i.d. از نرمال استاندارد $N(0, 1)$. اگر مقادیر $v_1 = \frac{1}{\sqrt{k}}R^T u_1$ و $v_2 = \frac{1}{\sqrt{k}}R^T u_2$ را تعیین کنیم، داریم:

$$E(\|v_1 - v_2\|^2) = d, \quad \text{Var}(\|v_1 - v_2\|^2) = \frac{2}{k}d^2 \quad (۷-۳)$$

$$E(v_1^T v_2) = a, \quad \text{Var}(v_1^T v_2) = \frac{1}{k}(m_1 m_2 + a^2), \quad (۸-۳)$$

سومین گشتاور مرکزی $v_1^T v_2$ عبارت است از:

$$E(v_1^T v_2)^2 = a, \quad \frac{2a}{k^2}(2m_1 m_2 + a^2) \quad (۹-۳)$$

و تابع مولد احتمال برای $v_1^T v_2$ عبارت است از:

مدام، محاسبه‌ی حاشیه‌ها می‌تواند پر هزینه باشد.

$$E(\exp(v_1^T v_2 t)) = \left(1 - \frac{2}{k}at - \frac{1}{k^2}(m_1 m_2 - a^2)t^2\right)^{-\frac{k}{2}} \quad (10-3)$$

که $\frac{-k}{\sqrt{m_1 m_2 - a}} \leq t \leq \frac{-k}{\sqrt{m_1 m_2 + a}}$ است.

بنابراین، برآوردهای ناریبی برای فاصله d و ضرب داخلی a به شکل سر راستی عبارت است از:

$$\hat{d}_{MF} = \|v_1 - v_2\|^2, \quad Var(\hat{d}_{MF}) = \frac{d^2}{k}, \quad (11-3)$$

$$\hat{a}_{MF} = v_1^T v_2, \quad Var(\hat{a}_{MF}) = \frac{1}{k}(m_1 m_2 + a^2), \quad (12-3)$$

که اندیس « MF » به معنی «بدون حاشیه»^۹ نشان دهنده این است که برآوردها از اطلاعات حاشیه‌ای $m_2 = \|u_2\|^2$ و $m_1 = \|u_1\|^2$ استفاده نمی‌کنند.

به این نکته توجه کنید که، $k\hat{d}_{MF}/d$ از توزیع χ^2 با k درجه آزادی، پیروی می‌کند، χ_k^2 . بنابراین، به راحتی می‌توان می‌توانیم این محدوده‌های دم را برای لم ۲ اثبات کنیم.

لم ۲:

$$\Pr(\hat{d}_{MF} - d > \epsilon d) \leq \exp\left(-\frac{k}{2}(\epsilon - \log(1 + \epsilon))\right), \quad \epsilon > 0 \quad (13-3)$$

$$\Pr(\hat{d}_{MF} - d < -\epsilon d) \leq \exp\left(-\frac{k}{2}(-\epsilon - \log(1 - \epsilon))\right), \quad 0 < \epsilon < 1 \quad (14-3)$$

اثبات:

از آنجا که $k\hat{d}_{MF}/d \sim \chi_k^2$ ، بر اساس نام مساوی چرنوف^{۱۰} [۲۵]، برای هر $t > 0$ داریم:

^۹margin-free

^{۱۰}Chernoff inequality

$$\begin{aligned} \Pr(\hat{d}_{MF} - d > \epsilon d) &= \Pr(k\hat{d}_{MF}/d > k(1 + \epsilon)) \\ &\leq \frac{E\left(\exp(k\hat{d}_{MF}/dt)\right)}{\exp((1 + \epsilon)kt)} = \exp\left(-\frac{k}{2}(\log(1 - 2t) + 2(1 + \epsilon)t)\right) \end{aligned} \quad (۱۵-۳)$$

که در $t = t_{NR} = \frac{\epsilon}{2(1+\epsilon)}$ و بنابراین برای هر $\epsilon > 0$ داریم:

$$\Pr(\hat{d}_{MF} - d > \epsilon d) \leq \exp\left(-\frac{k}{2}(\epsilon - \log(1 + \epsilon))\right) \quad (۱۶-۳)$$

ما می‌توانیم به طور مشابه برای دیگر محدوده‌ی دم $\Pr(\hat{d}_{MF} - d < -\epsilon d)$ هم اثبات کنیم. ■

برای راحتی مرسوم است که محدوده دم را در لم ۲ به صورت متقارن $\Pr(|\hat{d}_{MF} - d| > \epsilon d)$ نوشته شود. نامساوی‌های ساده‌ای برای $\log(1 + \epsilon)$ و $\log(1 - \epsilon)$ نتیجه می‌دهد:

$$\Pr(|\hat{d}_{MF} - d| \geq \epsilon d) \leq 2 \exp\left(-\frac{k}{4}\epsilon^2 + \frac{k}{6}\epsilon^3\right), \quad 0 < \epsilon < 1 \quad (۱۷-۳)$$

از آنجا که $\mathbf{A} \in \mathbb{R}^{n \times D}$ تعداد n سطر دارد. به عبارت دیگر $\frac{n(n-1)}{2}$ جفت. ما باید احتمال دم را به طور همزمان برای همه‌ی جفت‌ها محدود کنیم. با استفاده از محدوده اجتماع بنفرونی^{۱۱} کافی است که:

$$\frac{n^2}{2} \Pr(|\hat{d}_{MF} - d| \geq \epsilon d) \leq \delta \quad (۱۸-۳)$$

به عبارت دیگر کافی است اگر:

$$\frac{n^2}{2} 2 \exp\left(-\frac{k}{4}\epsilon^2 + \frac{k}{6}\epsilon^3\right) \leq \delta \Rightarrow k \geq \frac{2 \log n - \log \delta}{\epsilon^2/4 - \epsilon^3/6} \quad (۱۹-۳)$$

^{۱۱} Benferroni union bound

بنابراین ما یک نسخه‌ای از JL را نشان داده‌ایم.

لم ۳: اگر $k \geq \frac{2 \log n - \log \delta}{\epsilon^2/4 - \epsilon^3/6}$ پس با حداقل احتمال $1 - \delta$ ، فاصله l_2 بین هر جفت از داده‌ها (میان n نقطه) می‌تواند با ضریب اطمینان $1 \pm \epsilon$ با استفاده فاصله‌ی l_2 در داده‌های نگاشت شده بعد از نگاشت تصافی نرمال، تخمین زده شود. $0 < \epsilon < 1, 0 < \delta < 1$. ■

۳-۳ نگاشت تصادفی زیر گاوسی و بسیار پراکنده

در بخش قبل ما به بررسی نگاشت تصادفی نرمال پرداختیم، که در آن ماتریس نگاشت گر R از روی توزیع $N(0, 1)$ به طور i.i.d. نمونه‌گیری می‌شود. این انتخاب خاص برای R ، صرفاً برای سهولت تحلیل تئوری است. در واقع می‌توان R را از هر توزیعی با میانگین صفر و واریانس محدود برای کاهش بعد در نرم l_2 نمونه‌گیری کرد.

نمونه‌گیری R از یک توزیع زیر گاوسی هم از نظر تئوری قابل قبول و هم از جنبه‌ی محاسباتی تسهیل کننده است. برای مثال، محدوده‌ی دم زیر گاوسی به سادگی به نسخه‌ای از JL منتهی می‌شود.

ما بر روی یک انتخاب معمول از توزیع زیر گاوسی تمرکز خواهیم کرد، که درایه‌ها ماتریس R از مجموعه‌ی $\{-1, 0, 1\}$ با احتمالات $\{\frac{1}{2s}, 1 - \frac{1}{s}, \frac{1}{2s}\}$ ، که $s \geq 1$ ، به این ترتیب فرآیند نمونه‌گیری ساده‌تر شده و محاسبات سریعتر انجام می‌شوند. در واقع، زمانی که $s < 3$ باشد، واریانس‌های صرحاً کوچکتری نسبت به استفاده از نگاشت تصادفی نرمال به‌دست می‌آید.

با در نظر گرفتن قواعد معقول، برای مثال، داده‌های اولیه ممان سوم محدود داشته باشند، می‌توانیم $s \gg 3$ در نظر بگیریم (حتی $s = \sqrt{D}$). تا نتایج s برابر سریعتر به‌دست بیاوریم؛ و بنابراین، این رویه را نگاشت تصادفی بسیار پراکنده می‌نامیم.

۱-۳-۳ نگاشت تصادفی زیر گاوسی

مشابه **قسمت ۲-۳** ماتریس داده را $A \in \mathbb{R}^{n \times D}$ در نظر می‌گیریم. ماتریس نگاشت تصادفی $R \in \mathbb{R}^{D \times k}$ را تولید کرده و آن را در A ضرب می‌کنیم تا به یک ماتریس نگاشت شده‌ی $B = \frac{1}{\sqrt{k}} AR \in \mathbb{R}^{n \times k}$ برسیم. دوباره رو دو ردیف ابتدایی تمرکز می‌کنیم، که یعنی u_1 و u_2 در A ، و دو ردیف ابتدایی v_1 و v_2 در B و همچنین تساوی‌های زیر را تعریف می‌کنیم:

$$a = u_1^T u_2, \quad m_1 = \|u_1\|^2, \quad m_2 = \|u_2\|^2, \quad d = \|u_1 - u_2\|^2 = m_1 + m_2 - 2a \quad (۲۰-۳)$$

\mathbf{R} را به طور i.i.d از یک توزیع زیر گاوسی مشخصا پر کاربرد تولید می کنیم: ($S \geq 1$)

$$r_{ij} = \sqrt{s} \times \begin{cases} 1 & \text{با احتمال } \frac{1}{2s} \\ 0 & \text{با احتمال } 1 - \frac{1}{s} \\ -1 & \text{با احتمال } \frac{1}{2s} \end{cases} \quad (۲۱-۳)$$

- نمونه گیری از رابطه ی (۲۱-۳) ساده تر از نمونه گیری از $N(0, 1)$ است.
- می تواند از s برابر افزایش سرعت در ضرب ماتریسی $\mathbf{A} \times \mathbf{R}$ بهره برد، زیرا فقط $\frac{1}{s}$ داده های نیازمند پردازش هستند.
- نیازی به عملیات محاسباتی با ممیز شناور نیست و تمامی بار محاسباتی بر روی عملیات تجمیع پایگاه داده است که به خوبی بهینه شده.
- وقتی $s < 3$ باشد می توان به تخمین هایی با دقت بیشتر (واریانس کمتر) دست پیدا کرد.
- هزینه نگهداری ماتریس \mathbf{R} از $O(Dk)$ به $O(Dk/s)$ کاهش می یابد.

[۶، ۷] نشان می دهند زمانی که $s = 1$ و $s = 3$ باشد، می توان به همان محدوده ی JL ای دست پیدا کرد که در نگاشت تصادفی نرمال وجود دارد. ما در ادامه به بررسی خواص توزیع زیر گاوسی می پردازیم، که برای تحلیل محدوده ی دم مناسب است. در واقع، آنالیز زیر گاوسی نشان می دهد که می توان حتی در بدترین شرایط از مقادیری اندکی بیشتر از ۳ برای s استفاده کرد.

توزیع زیر گاوسی

ما در اینجا مقدمه ای کوتاه بر توزیع های زیر گاوسی بیان می کنیم. برای جزئیات و منابع بیشتر می توانید به [۲۳] مراجعه کنید. تئوری توزیع های زیر گاوسی در حدود ۱۹۶۰ آغاز شد.

متغیر تصادفی x ^{۱۲} زیر گاوسی است اگر ثابت $g > 0$ وجود داشته باشد به شکلی که:

$$\mathbb{E}(\exp(xt)) \leq \exp\left(\frac{g^2 t^2}{2}\right), \forall t \in \mathbb{R} \quad (۲۲-۳)$$

می توان مقدار بهینه ی g^2 را از تعریف $T^2(x)$ با استفاده از فرمول زیر به دست آورد.

$$T^2(x) = \sup_{t \neq 0} \frac{2 \log \mathbb{E}(\exp(xt))}{t^2} \quad (۲۳-۳)$$

توجه کنید که $T^2(x)$ فقط یک نمادگذاری برای مقدار ثابت بهینه ی زیر گاوسی یک متغیر تصادفی x است (و نه یک نمونه مشخص از x).
برخی از ویژگی های اولیه ی توزیع های زیر گاوسی:

- اگر x زیر گاوسی باشد آنگاه $\mathbb{E}(x) = 0$ و $\mathbb{E}(x^2) \leq T^2(x)$. برای هر مقدار ثابت c ، $T^2(cx) = c^2 T^2(x)$ و

$$\Pr(|x| > t) \leq 2 \exp\left(-\frac{t^2}{2T^2(x)}\right) \quad (۲۴-۳)$$

- اگر x_1, x_2, \dots, x_D زیر گاوسی مستقل باشند، آنگاه $\sum_{i=1}^D x_i$ زیر گاوسی است.

$$T^2\left(\sum_{i=1}^D x_i\right) \leq \sum_{i=1}^D T^2(x_i) \quad (۲۵-۳)$$

- اگر x زیر گاوسی باشد، آنگاه برای همه ی $t \in [0, 1]$

$$\mathbb{E}\left(\exp\left(\frac{x^2 t}{2T^2(x)}\right)\right) \leq (1-t)^{-\frac{1}{2}} \quad (۲۶-۳)$$

[۶، ۷] همچنین رابطه ی (۲۶-۳) را برای توزیع ویژه ی رابطه ی (۲۱-۳) به دست آورده اند. یک

^{۱۲} در این پایان نامه در تمامی موارد متغیرهای تصادفی با حروف کوچک نشان داده شده اند.

متغیر تصادفی زیر گاوسی x صریحا زیر گاوسی است اگر $E(x^2) = T^2(x)$

• اگر x صریحا زیر گاوسی باشد، آنگاه $E(x^3) = 0$ و کشیدگی^{۱۳} غیر مثبت خواهد بود، به عبارت

$$\text{دیگر } 0 \leq \frac{E(x^4)}{E^2(x^2)} - 3$$

• اگر x_1, x_2, \dots, x_D صریحا زیر گاوسی مستقل باشند، آنگاه $\sum_{i=1}^D x_i$ صریحا زیر گاوسی است.

$$T^2\left(\sum_{i=1}^D x_i\right) = \sum_{i=1}^D T^2(x_i) = \sum_{i=1}^D E(x_i^2) \quad (27-3)$$

■

۴-۳ نگاشت تصادفی کوشی برای l_1

در بخش‌های قبلی به نگاشت تصادفی برای کاهش بعد در نرم l_2 پرداخته شد. در این بخش به کاهش بعد در نرم l_1 پرداخته خواهد شد.

در اینجا هم با یک ماتریس داده‌ی $A \in \mathbb{R}^{n \times D}$ کار خواهیم کرد. و یک ماتریس نگاشت‌گر تصادفی $R \in \mathbb{R}^{D \times k}$ که به طور i.i.d. از توزیع کوشی استاندارد $C(0, 1)$ نمونه‌گیری شده است، تولید خواهیم کرد. ما اجازه خواهیم داد که ماتریس نگاشت شده $B = A \times R \in \mathbb{R}^{n \times k}$ باشد. بدون آنکه ضریب نرمال‌سازی $\frac{1}{\sqrt{k}}$ که در بخش‌های قبلی مشاهده کردیم، حضور داشته باشد. ضمن آنکه این کار به یک تخمین آماری منجر خواهد شد که پارامتر مقیاس‌دهی را از تعداد k متغیر تصادفی کوشی به طور i.i.d. برآورد می‌کند.

از آنجا که کوشی میانگین محدود ندارد. نمی‌توانیم از یک برآوردگر خطی آنطور که در نگاشت تصادفی نرمال استفاده کردیم، استفاده کنیم. علاوه بر این، نتیجه‌ی عدم امکان بیان شده در [۲۰، ۵۰، ۱۹] اثبات کرده است که وقتی از یک نگاشت‌گر خطی استفاده شود، نمی‌توان از برآوردهای خطی بدون رخ دادن خطاهای بزرگ استفاده کرد. به عبارت دیگر، لم JL برای l_1 صدق نمی‌کند.

در این بخش سه برآوردگر غیرخطی ارائه و یک معادل برای لم JL برای l_1 استنتاج می‌شود. از آنجا که برآوردهای ما، متریک نیستند، این معادل لم JL از حالت کلاسیک لم JL برای l_2 ضعیفتر است.

¹³kurtosis

۳-۴-۱ خلاصه نتایج اصلی

ما دوباره مانند بخش‌های قبلی دو سطر اول A ، u_1 و u_2 و دو سطر اول B ، v_1 و v_2 را در نظر می‌گیریم. فاصله‌ی l_1 را با $d = \sum_{i=1}^D |u_{1,i} - u_{2,i}|$ تعریف می‌کنیم.

در نگاشت تصادفی کوشی، فعالیت اصلی آن است که پارامتر مقیاس‌دهی کوشی از k نمونه‌ی $x_j \sim C(0, d)$ به طور i.i.d. استخراج شود. برخلاف نگاشت تصادفی نرمال، نمی‌توان d را از میانگین نمونه برآورد کرد (به عبارت دیگر، $\frac{1}{k} \sum_{j=1}^k |x_j|$ زیرا $E(x_j) = \infty$).

سه نوع برآوردگر غیر خطی مورد بررسی قرار خواهند گرفت: برآوردگرهای میانه‌ی نمونه، برآوردگرهای میانگین هندسی و برآوردگرهای حداکثر درست‌نمایی.

• برآوردگرهای میانه نمونه

برآوردگر میانه‌ی نمونه \hat{d}_{me} و نسخه‌ی بدون انحراف $\hat{d}_{me,c}$ به شکل زیر هستند.

$$\hat{d}_{me} = \text{median}(|x_j|, j = 1, 2, \dots, k) \quad (28-3)$$

$$\hat{d}_{me,c} = \frac{\hat{d}_{me}}{b_{me}} \quad (29-3)$$

$$b_{me} = \int_0^1 \frac{(2m+1)!}{(m!)^2} \tan\left(\frac{\pi}{2}t\right) (t-t^2)^2 dt, \quad k = 2m+1 \quad (30-3)$$

برای سهولت، ما فقط $k = 2m+1, m = 1, 2, \dots$ را در نظر می‌گیریم.

در بین تمامی برآوردگرهای چندکی، \hat{d}_{me} (و $\hat{d}_{me,c}$) کوچکترین مقدار واریانس مجانبی را به دست می‌دهد.

• برآوردگرهای میانگین هندسی

برآوردگر میانگین هندسی، \hat{d}_{gm} و نسخه‌ی بدون انحراف $\hat{d}_{gm,c}$ به شکل زیر هستند:

$$\hat{d}_{gm} = \prod_{j=1}^k |x_j|^{1/k} \quad (31-3)$$

$$\hat{d}_{gm,c} = \cos^k\left(\frac{\pi}{2k}\right) \prod_{j=1}^k |x_j|^{1/k} \quad (32-3)$$

از نظر واریانس‌های مجانبی، برآوردگرهای میانگین هندسی به صورت مجانبی متناظر با برآوردگرهای میانه‌ی نمونه هستند. اگر چه از نظر محدوده‌ی دم، برآوردگرهای میانه‌ی نمونه ممکن است نیازمند نمونه‌ای به اندازه‌ی تا دو برابر بزرگتر باشند.

• برآوردگر حداکثر درست‌نمایی

این برآوردگر که به صورت $\hat{d}_{MLE,c}$ تعریف می‌شود. برآوردگر بدون انحراف حداکثر درست‌نمایی (MLE) عبارت است از:

$$\hat{d}_{MLE,c} = \hat{d}_{MLE} \left(1 - \frac{1}{k} \right) \quad (3-33)$$

که \hat{d}_{MLE} یک معادله‌ی غیر خطی MLE را حل می‌کند.

$$-\frac{k}{\hat{d}_{MLE}} + \sum_{j=1}^k \frac{2\hat{d}_{MLE}}{x_j^2 + \hat{d}_{MLE}^2} = 0 \quad (3-34)$$

برآوردگرهای میانه‌ی نمونه و میانگین هندسی از نظر واریانس مجانبی، دقتی معادل 80% MLE دارند. در حالی که استنتاج محدوده‌های دمی فرم-بسته دشوار است. نشان خواهیم داد که توزیع $\hat{d}_{MLE,c}$ را می‌توان به وسیله‌ی یک معکوس گاوسی^{۱۴} تخمین زد.

۵-۳ نگاشت تصادفی α -پایدار

توضیحات در بخش‌های قبلی، در مورد نگاشت تصادفی نرم l_2 و نگاشت تصادفی نرم l_1 صحبت کردیم. در این بخش، کاهش بعد در نرم l_α ، برای $0 < \alpha \leq 2$ مورد بررسی قرار خواهد گرفت. و نرم‌های l_1 و l_2 به عنوان حالت خاص بررسی می‌شوند.

مسئله اساسی در نگاشت تصادفی پایدار، انجام برآورد آماری است. به عبارت دیگر، برآورد پارامتر مقیاس‌دهی توزیع پایدار متقارن. از آنجا که چگالی احتمال توزیع پایدار جز برای $\alpha = 1, 2$ فرم بسته ندارد. تولید برآوردگرهایی که از نظر آماری دقیق و از نظر محاسباتی بهینه هستند، جذاب است.

¹⁴Inverse Gaussian

برآوردگرهایی که بر اساس میانه‌های نمونه (با به طور کلی بر اساس چندک‌های نمونه) تولید شده‌اند، در علم آمار شناخته شده‌اند، اما خیلی دقیق نیستند به خصوص در مورد نمونه‌های کوچک، و برای تحلیل نظری از جمله محدوده‌های دم زمانی که $\alpha \neq 1, 2$ راحت نیستند. ما در اینجا برآوردگرهای مختلفی را بر اساس میانگین هندسی، میانگین هارمونیک و توان نسبی بررسی خواهیم کرد.

۳-۵-۱ نتایج اصلی

گفته شد که، اگر دو بردار $u_1, u_2 \in \mathbb{R}^D$ (برای مثال، u_1 و u_2 دو ردیف اول در ماتریس داده‌ی A باشند)، اگر $v_1 = R^T u_1$ و $v_2 = R^T u_2$ باشند که $R \in \mathbb{R}^{D \times k}$ شامل نمونه‌های i.i.d. در $S(\alpha, 1)$ باشد، آنگاه $x_j = v_{1,j} - v_{2,j}, j = 1, 2, \dots, k$ ، همگی $S(\alpha, d_{(\alpha)})$ های i.i.d. هستند. که $d_{(\alpha)} = \sum_{i=1}^D |u_{1,i} - u_{2,i}|^\alpha$. بنابراین مسئله اصلی برآورد پارامتر مقیاس‌دهی $d_{(\alpha)}$ از تعداد k نمونه‌ی i.i.d. توزیع $S(\alpha, d_{(\alpha)})$ است. در بخش‌های قبلی به طور خلاصه توزیع‌های پایدار را مرور کردیم. برآوردگری پرکاربرد در آمار بر اساس نمونه میان چندکی^{۱۵} [۳۳، ۳۴، ۵۵] است که به دلیل تقارن $S(\alpha, d_{(\alpha)})$ می‌توان آن را به صورت برآوردگر میانه‌ی نمونه، ساده‌سازی کرد.

$$\hat{d}_{(\alpha),me} = \frac{\text{median} \{|x_j|^\alpha, j = 1, 2, \dots, k\}}{\text{median} \{S(\alpha, 1)\}^\alpha} \quad (35-3)$$

علی‌رغم سادگی، مسائل بسیاری براساس برآوردگر میانه‌ی نمونه $\hat{d}_{(\alpha),me}$ وجود دارند. این برآوردگر، علی‌الخصوص برای نمونه‌های کوچک یا α ی کوچک دقیق نیست. همچنین برای تحلیل نظری دقیق از جمله تحلیل محدوده‌ی دم دشوار است. ما برآوردگرهای زیادی را بر اساس میانگین هندسی، میانگین هارمونیک و توان کسری ارائه خواهیم کرد.

• برآوردگر میانگین هندسی (بدون انحراف) $\hat{d}_{(\alpha),gm}$:

• برآوردگر میانگین هندسی (دارای انحراف) $\hat{d}_{(\alpha),gm,b}$:

¹⁵inter-quantiles

این معادله به طور مجانبی معادل $\hat{d}_{(\alpha),gm}$ است. اگرچه برای $0.25 \leq \alpha \leq 1$ ، توزیع $\hat{d}_{(\alpha),gm,b}$ خطای میانگین توان دوم خطای کوچکتری در مقایسه با $\hat{d}_{(\alpha),gm}$ دارد.

• برآوردگر میانگین هارمونیک $\hat{d}_{(\alpha),hm}$: این معادله زمانی که $\alpha \rightarrow 0+$ به صورت مجانبی بهینه است و در مقایسه با برآوردگرهای میانگین هندسی، برای $\alpha \leq 0.344$ واریانس مجانبی کوچکتری دارد.

• برآوردگر میانگین ریاضی

برای $\alpha = 2$ بهترین روش استفاده از برآوردگر میانگین ریاضی $\frac{1}{k} \sum_{j=1}^k |x_j|^2$ است. این برآوردگر را می‌توان با استفاده از برآوردگر بیشینه درست‌نمایی، به شکلی که در بخش نگاشت تصادفی نرمال توضیح داده شد، با استفاده از اطلاعات حاشیه‌ای ارتقاء داد.

• برآوردگر توان کسری $\hat{d}_{(\alpha),fp}$:

برای $\alpha = 2$ ، $\hat{d}_{(\alpha),fp}$ معادل برآوردگر میانگین ریاضی و برای $\alpha \rightarrow 0+$ معدل برآوردگر میانگین هارمونیک است. علاوه بر این، برای $\alpha \rightarrow 1$ ، $\hat{d}_{(\alpha),fp}$ دارای واریانس مجانبی برابر با برآوردگر میانگین هندسی است.

فصل چهارم

پیاده‌سازی نگاشت تصادفی

در این فصل به بررسی نحوه پیاده‌سازی نگاشت تصادفی و نحوه بررسی عملکرد آن می‌پردازیم همان‌طور که در فصل دوم اشاره شد تصویر تصادفی به عنوان یک روش کاهش بعد باید با دیگر روش‌های کاهش بعد مقایسه شود. در همین راستا مهمترین موردی که باید بدان توجه کرد مقایسه با روش‌های مرسوم کاهش بعد مانند PCA است.

برای پیاده‌سازی روش نگاشت تصادفی مجموعه داده‌های مختلفی تهیه و بر روی آن‌ها این روش برای کاهش بعد به دو بعد و سه بعد اعمال شده است. نتایج پس از خوشه‌بندی مورد مقایسه با نتایج خوشه‌بندی بدون کاهش بعد شده است. این فرآیند درکی از حوزه کاربرد نگاشت تصادفی و عملکرد آن بر روی داده‌های واقعی به ما می‌دهد.

۱-۴ داده‌های مورد استفاده

در این پایان‌نامه از ۷ مجموعه داده‌ی استاندارد و پرکاربرد، آیریس، اسکناس، دیابت، تیروئید، خرچنگ، بذر، پروتئین استفاده شده است.

دادگان‌های آیریس، بذر و پروتئین از سایت <https://archive.ics.uci.edu/ml/index.php> گردآوری شده و سایر مجموعه داده‌ها در کتابخانه‌های نرم‌افزار R وجود دارد. یکی از ویژگی‌های داده‌های دم‌سنگین این است که شامل نقاط دورافتاده باشند. نقاط دور افتاده داده‌ها را بررسی کردیم. داده‌های اسکناس، آیریس، بذر، پروتئین، تیروئید، دیابت، خرچنگ دارای به ترتیب ۲۰، ۵۰، ۷۷، ۱۰۳، ۵۵، ۴۴، ۱۴ نقطه دور افتاده هستند. متأسفانه آزمونی برای برازش توزیع پایدار چند متغیره وجود ندارد و فقط می‌توانیم آزمون برازش یک متغیره توزیع پایدار را بررسی کنیم و مشاهده کردیم هیچ کدام از متغیرهای دادگان‌های خرچنگ، اسکناس و بذر به توزیع پایدار برازش نشدند اما برخی از متغیرهای دادگان‌های تیروئید، آیریس، دیابت و پروتئین به توزیع پایدار برازش شدند. همچنین هیچ کدام از دادگان‌ها به توزیع نرمال چند متغیره برازش نشد و آزمون نرمال بودن رد شد. توزیع این داده‌ها را بررسی کردیم و با استفاده از چهار آزمون ماردیا، هنس-زیرکلر، شاپیرو و رویستون، نرمال بودن این دادگان‌ها رد شده است.

۱-۱-۴ داده‌های غده تیروئید

این مجموعه داده که با نام اختصاری Thyroid در جداول مشخص شده است، از کتابخانه `melust` [۶۱] در R استخراج شده. داده‌ها شامل پنج تست آزمایشگاهی انجام شده بر روی ۲۱۵ بیمار است. داده‌های بدین منظور مورد استفاده قرار گرفته‌اند تا بتوانیم پیش‌بینی کنیم که آیا غده تیروئید بیمار را می‌توان در

دسته‌ی «مواردی که غده تیروئید به شکل نرمال کار می‌کند»^۱، «غده تیروئید کم کار است و هورمون کافی تولید نمی‌کند»^۲ یا «غده تیروئید بیش فعال است و مقدار زیادتری هورمون T3 و یا T4^۳ ترشح می‌کند»^۴. درمان عملکرد تیروئید بر اساس اطلاعات کامل پزشکی صورت گرفته است. شامل، سابقه‌ی درمانی، اسکن، ...

داده به شکل یک قاب داده‌ها^۵ شامل متغیرهای زیر در R قرار می‌گیرند:

• **Diagnosis** درمانی که برای تیروئید تجویز شده است. شامل: Hyper, Normal, Hypo

• **RT3U** تست میزان جذب **T3-resin** (به شکل درصد)

• **T4** کل سرم تیروکسین که با روش جایگزاری ایزوتوپ^۶ اندازه‌گیری شده است.

• **T3** کل سرم تیرودوتیروئین^۷ که با روش رادیو ایمنو^۸ اندازه‌گیری شده است.

• **TSH** هورمون اصلی محرک تیروئید که با روش رادیوایمونو اندازه‌گیری شده است. **DTSH** مقدار

حداکثر اختلاف **TSH** بعد از تزریق $200\mu gr$ هورمون رها ساز تیروتروپین^۹ در مقایسه با مقدار

پایه.

۲-۱-۴ داده‌های دیابت

برای این مجموعه داده که به اختصار Diabetes در جداول ذکر شده است، از کتابخانه **mcclust** [۶۱] نرم‌افزار R استفاده شده است.

این مجموعه داده شامل سه اندازه‌گیری است که بر روی ۱۴۵ بیمار غیر چاق صورت گرفته است. این بیماران به سه گروه تقسیم شده‌اند.

داده‌ها به شکل یک قاب داده‌ها شامل متغیرهای زیر در R قرار دارند:

• **class** نوع دیابت را مشخص می‌کند و به رده‌ی Normal, Overt, Chemical طبقه‌بندی می‌شود.

¹euthyroidism

²hypothyroidism

³thyroxine

⁴hyperthyroidism

⁵data.frame

⁶isotopic displacement method

⁷triiodothyronine

⁸radioimmuno

⁹thyrotropin

• **glucose** مساحت زیر نمودار گلوکوز در پلاسما پس از سه ساعت آزمون OGTT^{۱۰}

• **insulin** مساحت زیر نمودار انسولین در پلاسما پس از سه ساعت آزمون OGTT

• **sspg** میزان گلوکوز پلاسما در حالت پایدار

۳-۱-۴ داده‌های خرچنگ‌ها

این مجموعه داده‌ها اندازه‌گیری مورفولوژیکی^{۱۱} بر روی خرچنگ لپتوکراپسوس که به اختصار Crabs در جداول نشان داده شده‌اند، از کتابخانه‌ی MASS [۶۴] در نرم‌افزار R استخراج شده‌اند.

قالب داده‌های خرچنگ‌ها شامل ۲۰۰ سطر و هشت ستون است که بیانگر ۵ اندازه‌ی مورفولوژیک برای ۵۰ خرچنگ از دو رنگ و دو جنسیت است. قالب داده‌ی مربوط شامل این ستون‌ها است:

• **sp** گونه‌ی «B» برای رنگ آبی و گونه‌ی «O» برای رنگ نارنجی

• **sex** جنسیت

• **index** اندیس از ۱ تا ۵۰ در داخل هر گروه

• **FL** لوب پیشانی بر حسب میلی‌متر

• **RW** عرض عقب بر حسب میلی‌متر

• **CL** طول لاک بر حسب میلی‌متر

• **CW** عرض لاک بر حسب میلی‌متر

• **BD** عمق بدن بر حسب میلی‌متر

۴-۱-۴ داده‌های اسکناس سوئیس

این داده‌ها که به اختصار «Swiss banknotes» در جداول بیان شده است از کتابخانه‌ی mclust [۶۱] در R استخراج شده‌اند.

¹⁰oral glucose tolerance test

¹¹morphological

این داده‌ها شامل مجموعه‌ی ۶ اندازه‌گیری مرتبط به ۱۰۰ اسکناس اصل و ۱۰۰ اسکناس تقلبی ۱۰۰۰ فرانکی قدیمی است. قالب داده‌های مربوط شامل موارد زیر است:

- **Status** وضعیت اسکناس که شامل دو حالت اصل (genuine) یا تقلبی (fake) می‌شود.

- **Length** طول اسکناس بر حسب میلی‌متر

- **Left** عرض لبه‌ی چپ بر حسب میلی‌متر

- **Right** عرض لبه‌ی راست بر حسب میلی‌متر

- **Bottom** عرض حاشیه پایین بر حسب میلی‌متر

- **Top** عرض حاشیه بالا بر حسب میلی‌متر

- **Diagonal** طول قطر بر حسب میلی‌متر

۴-۱-۵ داده‌های آیریس

این مجموعه داده که در جداول به اختصار Iris نامیده می‌شود از سایت UCI [۳] تهیه شده است. این مجموعه داده‌ها احتمالاً شناخته‌شده‌ترین مجموعه داده در ادبیات شناخت الگور^{۱۲} است. مقاله فیشر که یک مقاله کلاسیک در این حوزه است و بارها مورد ارجاع قرار گرفته از این مجموعه داده استفاده می‌کند. این مجموعه داده شامل سه رده‌ی ۵۰ تایی است. که هر رده نمونه‌ای از یک نوع گیاه آیریس است. یک رده که از دو گروه دیگر به طور خطی قابل تفکیک است ولی دو گروه بعدی به طور خطی قابل تفکیک نیستند.

داده‌ها در قالب یک فایل متنی با ستون‌های زیر از سایت UCI دانلود شده‌اند:

- **Sepal length** طول کاسبرگ که به سانتی‌متر بیان شده است.

- **Sepal Width** عرض کاسبرگ که به سانتی‌متر بیان شده است.

- **Petal length** طول گلبرگ که به سانتی‌متر بیان شده است.

¹²pattern recognition

• **Petal width** عرض گلبرگ که به سانتی‌متر بیان شده است.

• **Class** نوع گل آیریس که می‌تواند یکی از سه مقدار *setosa*, *versicolor*, *virginica* را داشته باشد.

۴-۱-۶ مجموعه داده‌های دانه‌ها

این مجموعه داده که با مخفف «Seeds» در جداول و نمودارها آورده شده است، از سایت UCI [۵] جمع‌آوری شده است.

گروه مورد آزمایش شامل سه گروه از گونه‌ای مختلف گندم است. گونه‌ی کاما، رز و کانادایی^{۱۳}. ۷۰ مورد از هر گونه به طور تصادفی برای آزمایش انتخاب شده‌اند. تصاویری با کیفیت بالا از ساختار دورنی دانه‌گندم با استفاده از تکنیک پرتو ایکس نرم^{۱۴} تهیه شده است. این روش یک روش غیر مخرب است و نسبت به دیگر روش‌های پیچیده‌ی تصویربرداری مانند میکروسکوپ روبشی و تکنولوژی لیزر، ارزانتر است. تصاویر بر روی فیلم پرتو ایکس کدک ۱۳ در ۱۸ سانتی‌متر ضبط شده است. مطالعه براساس دانه‌ی جمع‌آوری شده از مزارع آزمایشی به طور ترکیبی انجام شده است و در انستیتو آکروفیزیکی آکادمی لهستان در لوبین مورد بررسی قرار گرفته است.^{۱۵} این داده‌ها می‌تواند برای رده‌بندی و خوشه‌بندی مورد استفاده قرار گیرد.

این مجموعه داده در قالب یک فایل متنی شامل ۸ ستون تهیه شده است. توضیح ستون‌ها شامل موارد زیر است:

• **A** مساحت

• **P** محیط

• **C** فشردگی $4\pi A/P^2$

• **length of kernel** طول هسته

• **width of kernel** عرض هسته

• **symmetry coefficient** ضریب تقارن

¹³Kama, Rose, Canadian

¹⁴soft x-ray

¹⁵institute of agrophysics of the Polish academy of science in Lubin

• **length of kernel groove** طول رشد هسته

• **Class** نوع دانه

۷-۱-۴ مجموعه داده‌ی بیان پروتئین موش

این مجموعه داده که با نام اختصاری MPE^{۱۶} در جداول و نمودارها آورده شده است، از وب سایت UCI [۴] جمع‌آوری شده است. مجموعه داده شامل سطوح بیان ۷۷ تغییرات پروتئین/پروتئین‌ها است که سیگنال قابل اندازه‌گیری در شکست پوسته‌ی هسته تولید می‌کنند.

۳۳ موش کنترلی و ۳۴ موش تیروزومی^{۱۷} (دارای سندرم داون) مورد مطالعه قرار گرفته‌اند. که در کل شامل ۷۲ موش می‌شود. در این آزمایش، ۱۵ اندازه‌گیری مربوط به هر پروتئین برای هر موش اندازه‌گیری شده است. بنابراین برای موش‌های کنترلی، $38 \times 15 = 570$ اندازه‌گیری و برای موش‌های تیروزومی $34 \times 15 = 510$ اندازه‌گیری انجام شده است. این مجموعه داده شامل مجموعاً ۱۰۸۰ اندازه‌گیری برای هر پروتئین است. هر اندازه‌گیری می‌تواند به عنوان یک نمونه/موش مستقل در نظر گرفته شود.

۸ رده‌ی موش‌ها بر اساس مشخصاتی مانند ژنوتیپ^{۱۸}، رفتار و درمان طبقه‌بندی شده‌اند. از نظر ژنوتیپ موش‌ها می‌توانند کنترل با تیروزومی باشند. از نظر رفتاری، به یادگیری تحریک شده‌اند (زمینه-شوک)^{۱۹} و بقیه تحریک نشده‌اند (شوک-زمینه) و برای سنجش تأثیر دارو، داروی ممانتین^{۲۰} قابلیت یادگیری در موش‌های تیروزومی را افزایش می‌دهد. به برخی دارو تزریق شده است و به بقیه خیر.

رده‌ها:

- **c-CS-s**: موش کنترلی، تحریک شده به آموزش، تزریق با محلول آب و نمک (۹ موش)
- **c-CS-m**: موش کنترلی، تحریک شده به آموزش، تزریق با محلول ممانتین (۱۰ موش)
- **c-SC-s**: موش کنترلی، تحریک نشده به آموزش، تزریق با محلول آب و نمک (۹ موش)
- **c-SC-m**: موش کنترلی، تحریک نشده به آموزش، تزریق با محلول ممانتین (۱۰ موش)
- **t-CS-s**: موش تیروزومی، تحریک شده به آموزش، تزریق با محلول آب و نمک (۷ موش)

¹⁶mice protein expression

¹⁷trisomic

¹⁸genotype

¹⁹context-shock

²⁰memantine

- **t-CS-m**: موش تیروزومی، تحریک شده به آموزش، تزریق با محلول ممانتین (۹ موش)
- **t-SC-s**: موش تیروزومی، تحریک نشده به آموزش، تزریق با محلول آب و نمک (۹ موش)
- **t-SC-m**: موش تیروزومی، تحریک نشده به آموزش، تزریق با محلول ممانتین (۹ موش)

هدف تعیین زیرگروهی از پروتئین‌ها است که بین رده‌ها تمایز ایجاد می‌کنند.
داده‌ها در قالب یک فایل متنی شامل ستون‌های زیر تهیه شده‌اند:

- ۱: شناسه‌ی موش
- ۲ تا ۷۸: میزان بیان پروتئین برای ۷۷ پروتئین
- ۸۰: نحوه‌ی درمان: ممانتین (m) یا نمک (s)
- ۸۱: رفتار (CS یا SC)
- ۸۲: رده‌ها

۲-۴ الگوریتم پیاده‌سازی نگاشت تصادفی

پیاده‌سازی نگاشت تصادفی نسبتاً ساده‌است. چون در همه حالات درایه‌های ماتریس تصادفی $R \in \mathbb{R}^{D \times k}$ به طور i.i.d انتخاب می‌شوند. چالشی از جهت وابستگی متغیرهای تصادفی در طول فرآیند تولید نداریم. به طور کلی از دو نوع توزیع جهت تولید ماتریس تصادفی R استفاده کردیم. یکی توزیع پایدار و دیگری توزیع زیرگاوسی بسیار تُنک. توزیع‌های نرمال و کوچی حالت خاصی از توزیع نرمال قلمداد شده‌اند. کد مربوط به پیاده‌سازی تصویر تصادفی در حالت پایدار به سادگی زیر است:

```
randomProjection <- function(d, targetNumDimention, alpha = 2) {
  pm <- matrix(rstable(ncol(d) * targetNumDimention, alpha = alpha, beta = 0),
               nrow = ncol(d))

  d %*% pm
}
```

این تابع که دو یا سه ورودی می‌تواند داشته باشد ماتریس داده $A \in \mathbb{R}^{n \times D}$ را در قالب متغیر d ، تعداد ابعاد مد نظر برای کاهش بعد k در قالب متغیر `targetNumDimention` و پارامتر α توزیع پایدار

با متغیر α بیان شده است. پارامتر α می‌تواند در وردی‌های تابع ظاهر نشود و مقدار پیش‌فرض ۲ را دارد.

برای حالت تُنک تابع `sRandomProjection` به شکل زیر پیاده‌سازی شده است.

```
sRandomProjection <- function(d, targetNumDimention, s = 2) {
  r <- runif(ncol(d) * targetNumDimention, min = 0, max = 2*s)
  r[r <= 1] = -1
  r[r >= (2*s-1)] = 1
  r[r != 1 & r != -1] <- 0
  pm <- matrix(r,
               nrow = ncol(d))

  d %%% pm
}
```

دو متغیر ورودی `d` و `targetNumDimention` مانند تابع قبلی بیانگر داده‌هایی ورودی و تعداد ابعاد هدف است. ورودی `s` ولی نشان دهنده پارامتر `s` در توزیع ورودی مربوط به این تابع است. در این حالت از یک توزیع گسسته با مقادیر $\{-1, 0, 1\}$ با احتمالات $\{\frac{1}{2s}, 1 - \frac{1}{s}, \frac{1}{2s}\}$ استفاده می‌کنیم. هرچه مقدار پارامتر `s` در توزیع بیشتر باشد بهتر است.

۳-۴ ارزیابی روش‌های کاهش بعد

برای ارزیابی روش‌های کاهش‌بعد، معیارهای مختلفی معرفی شده است. [۳۹] یکی از روش‌های ارزیابی عملکرد کاهش‌بعد، خوشه‌بندی است. یکی دیگر از کاربردهای خوشه‌بندی علاوه بر کاهش‌بعد، ارزیابی کاهش‌بعد نیز است. برای ارزیابی روش‌های کاهش‌بعد با استفاده از خوشه‌بندی، ابتدا خوشه‌بندی را روی داده‌ها با تمام متغیرها انجام می‌دهیم و با استفاده از یکی از معیار ارزیابی خوشه‌بندی شاخص‌رند تعدیل شده که در بخش ۲-۲-۱ بیان شده است، عملکرد خوشه‌بندی را محاسبه می‌کنیم. سپس روش کاهش‌بعد را روی داده‌ها اعمال کرده و مجدد روی مجموعه داده‌های کاهش‌بعد یافته، خوشه‌بندی را پیاده‌سازی و عملکرد خوشه‌بندی را روی داده‌ها با ابعاد کمتر محاسبه می‌کنیم. سرانجام با مقایسه عملکرد خوشه‌بندی قبل و بعد از کاهش‌بعد، روش کاهش‌بعد را ارزیابی خواهیم کرد.

برای مقایسه روش‌های کاهش‌بعد بر اساس نگاشت تصادفی‌های متفاوت، از معیار نرخ خطار رده‌بندی استفاده می‌کنیم. معیار نرخ خطای رده‌بندی به صورت زیر تعریف می‌شود:

$$C_e = 100(\text{ARI}_d - \text{ARI}_p) \quad (۱-۴)$$

که در آن ARI_d و ARI_p به ترتیب شاخص رند تعدیل شده در فضای \mathbb{R}^d و \mathbb{R}^p می‌باشند ($d < p$) . ARI_d ، شاخص رند تعدیل شده خوشه‌بندی با متغیرهای کمتر و ARI_p ، شاخص رند تعدیل شده خوشه‌بندی با متغیرهای کامل است. هر چقدر معیار C_e بزرگتر باشد، بیانگر این است که کاهش بعد بهتری صورت گرفته است. در جداول فصل بعد کاهش بعد به دو و سه بعد انجام شده است. و عملکرد کاهش بعد با معیار C_e سنجیده شده است.

فصل پنجم

جمع‌بندی و نتیجه‌گیری و پیشنهادات

۱-۵ بررسی کاهش بعد نرمال ($\alpha = 2$) به دو بعد

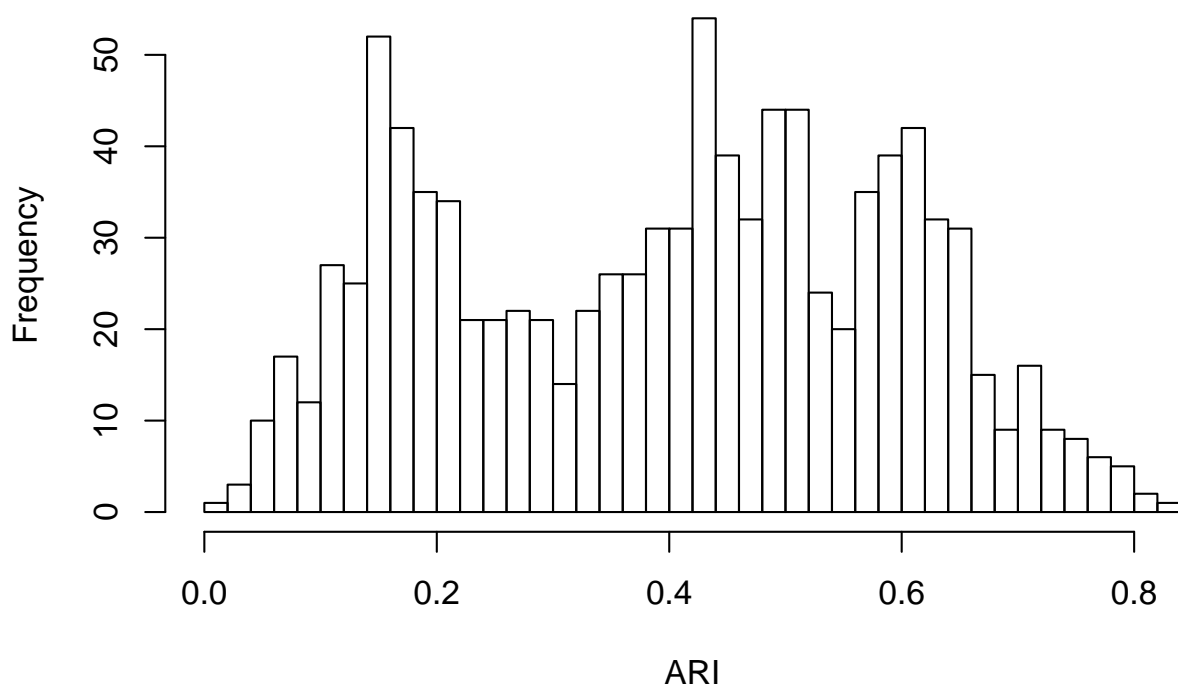
۱-۱-۵ جداول مقایسه عملکرد خوشه‌بندی

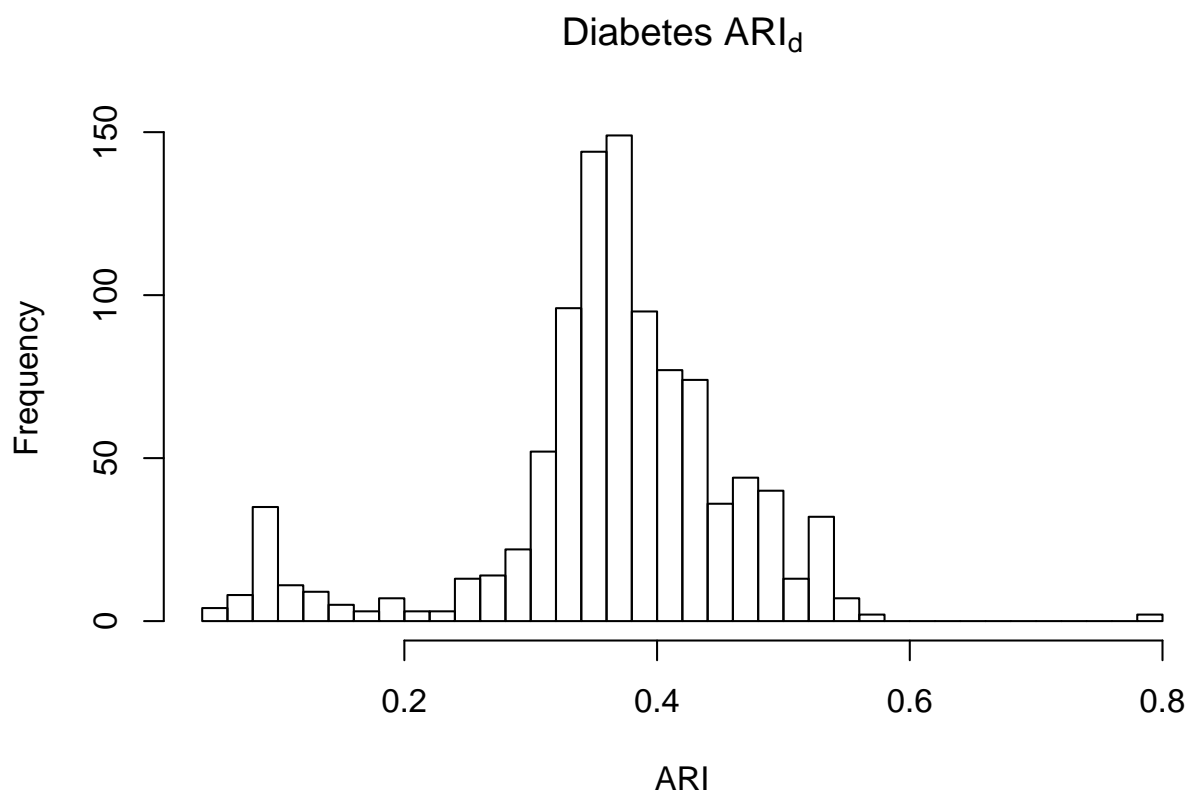
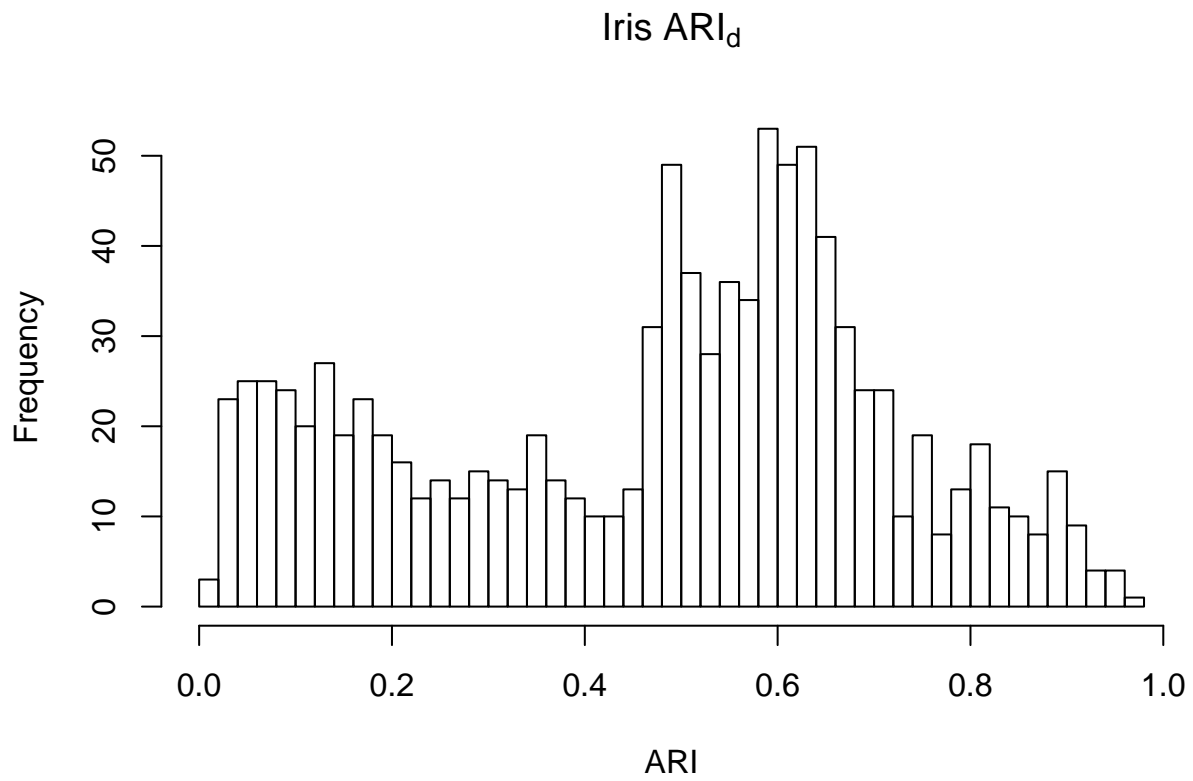
جدول ۱-۵: عملکرد تصویر تصادفی نرمال برای کاهش بعد به دو بعد

Dataset	ARI_p	ARI_d	C_e
Thyroid	0.5831656	0.3989981	-18
Iris	0.6201352	0.4710315	-15
Diabetes	0.3801662	0.3647537	-2
Swiss Banknotes	0.8456292	0.3880871	-46
Seeds	0.7732937	0.4482112	-33
Crabs	0.0481402	0.0439549	0
Mice Protein Expression	0.1316117	0.0657659	-7

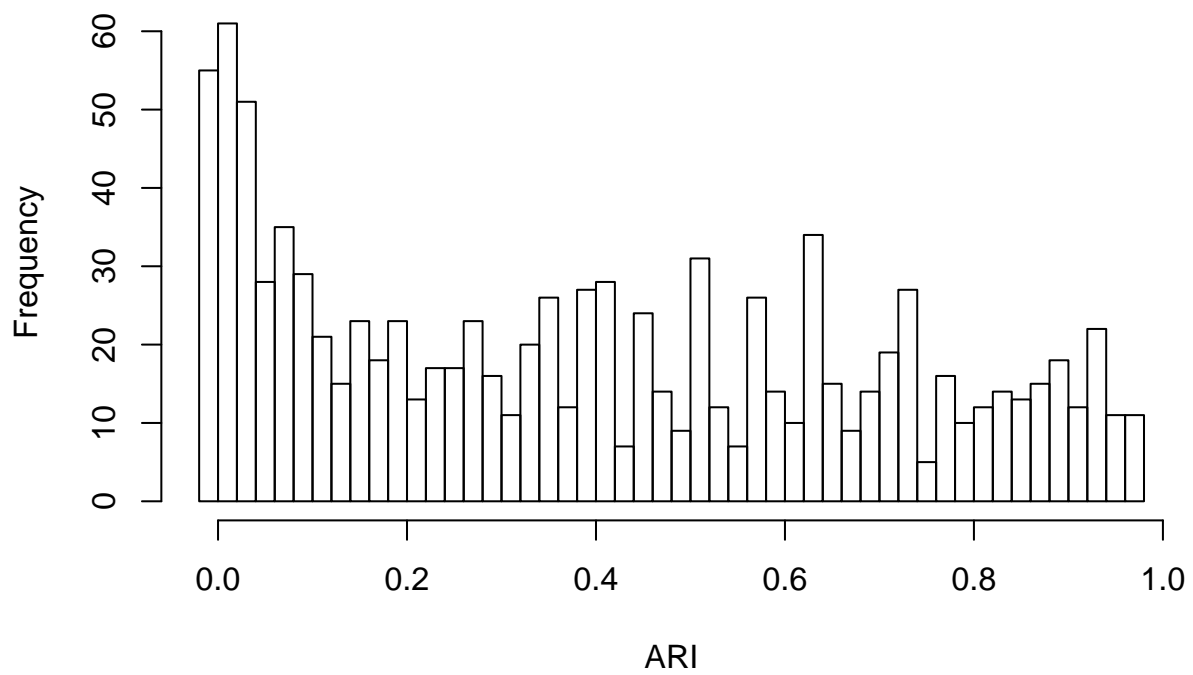
۲-۱-۵ نمودار فراوانی عملکرد خوشه‌بندی

Thyroid ARI_d

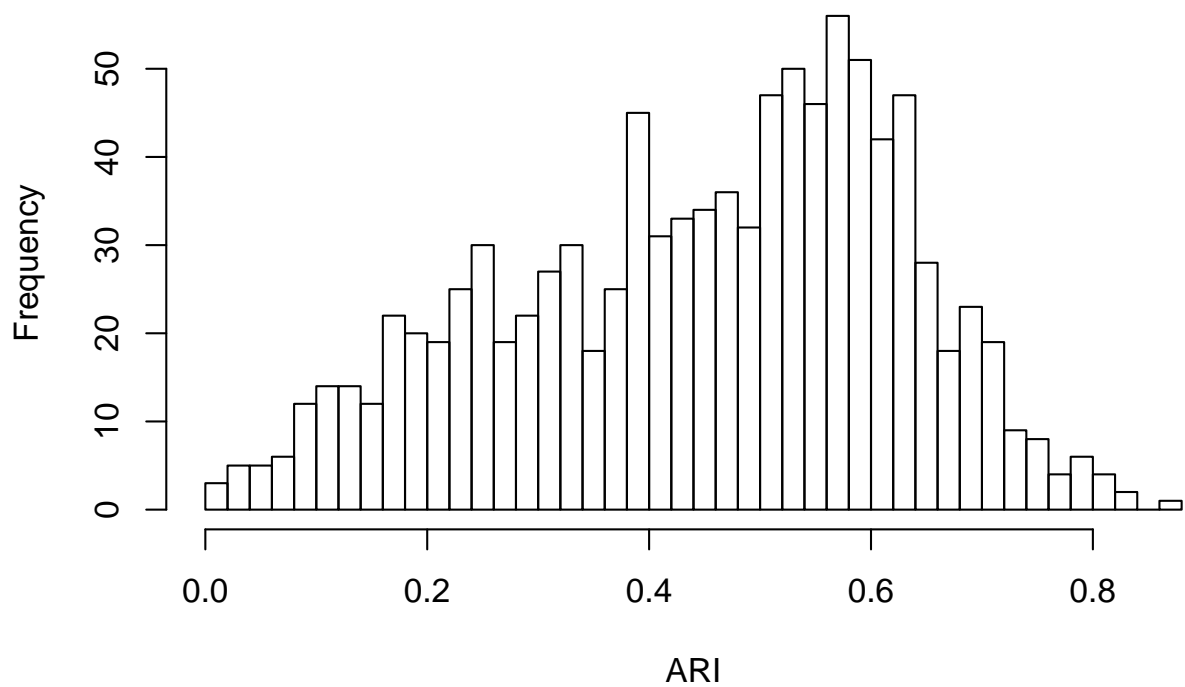




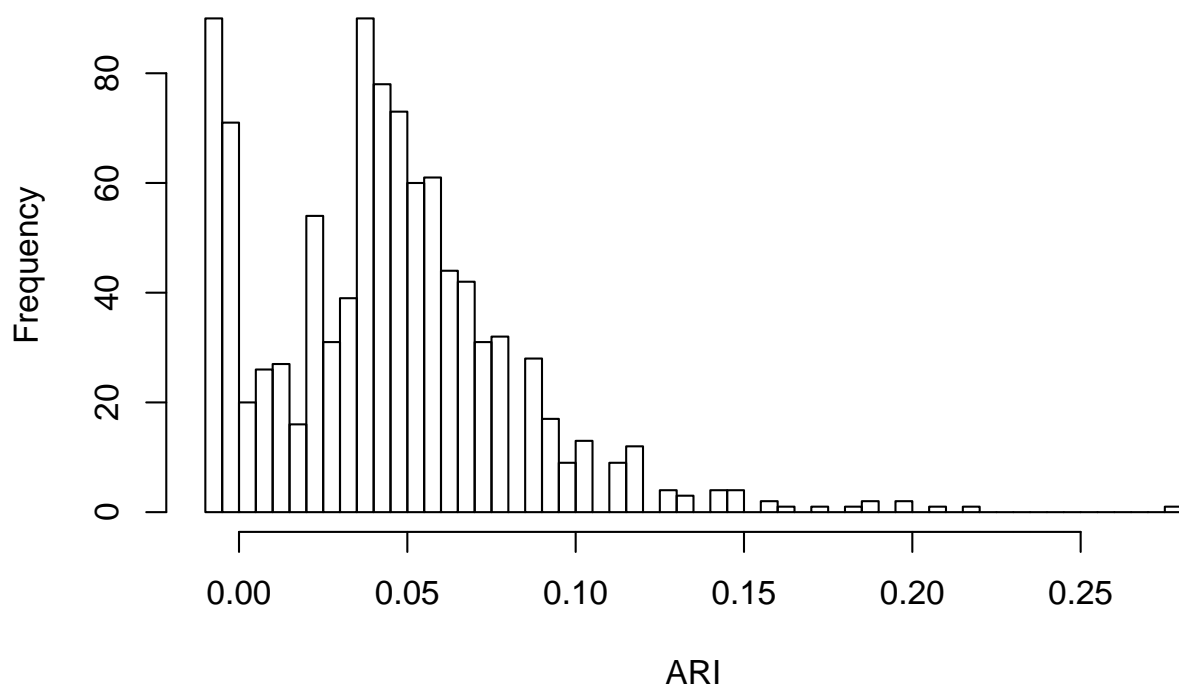
Swiss Banknotes ARI_d



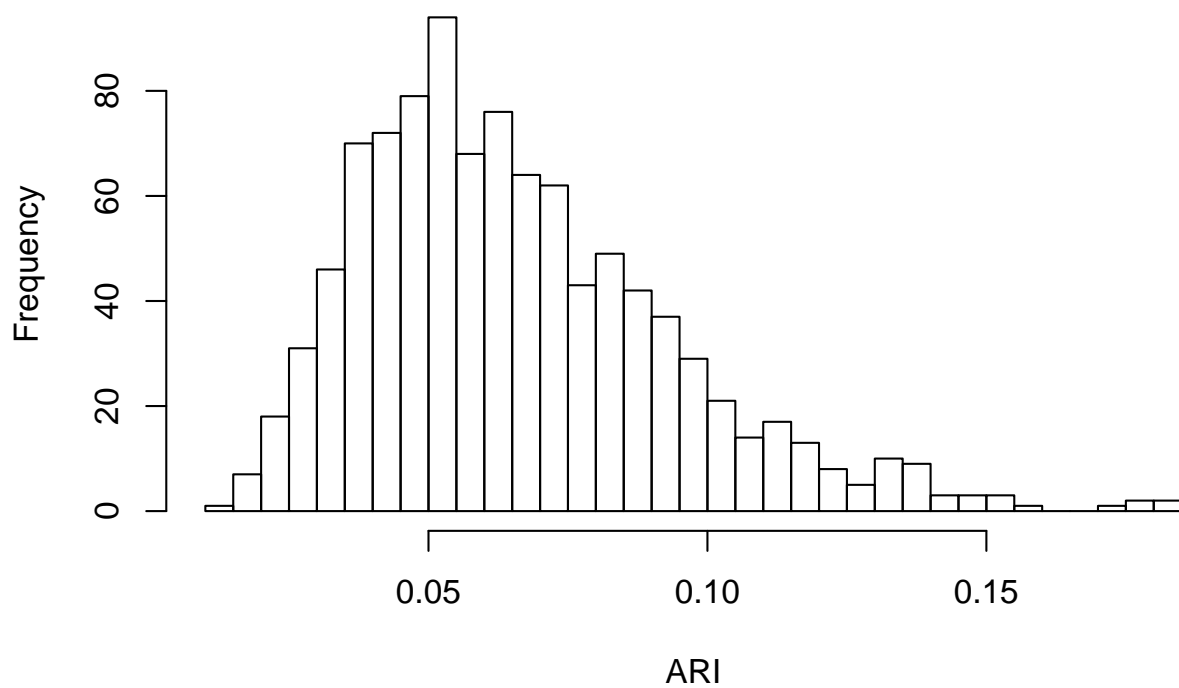
Seeds ARI_d



Crabs ARI_d



Mice Protein Expression ARI_d



۲-۵ نتایج برای کاهش بعد نرمال به سه بعد

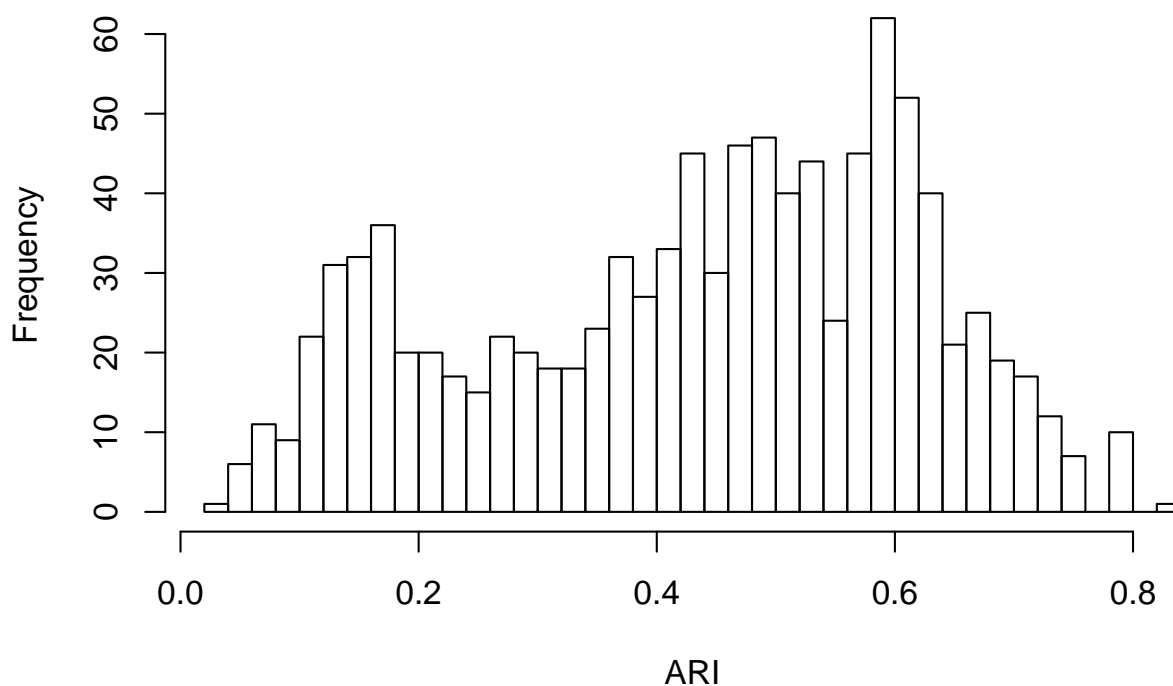
۱-۲-۵ جداول مقایسه عملکرد خوشه‌بندی

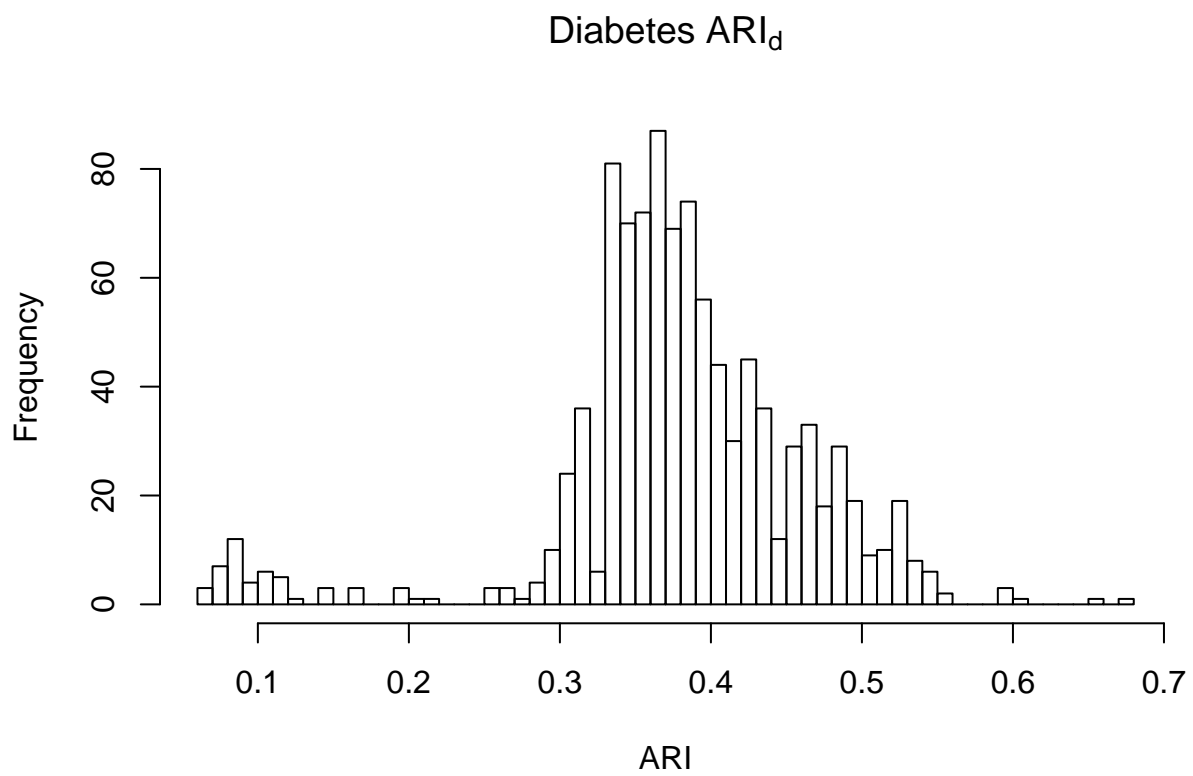
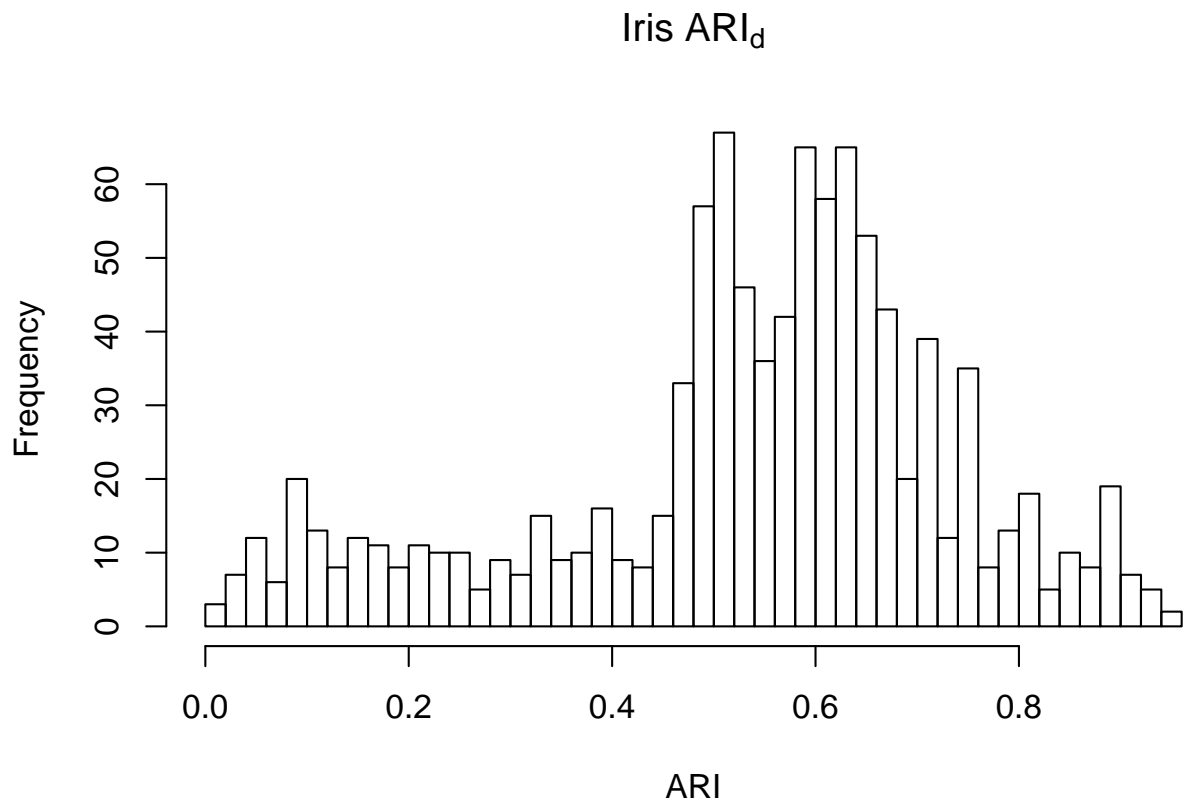
جدول ۲-۵: عملکرد تصویر تصادفی نرمال برای کاهش بعد به سه بعد

Dataset	ARI_p	ARI_d	C_e
Thyroid	0.5831656	0.4344288	-15
Iris	0.6201352	0.5359746	-8
Diabetes	0.3801662	0.3805076	0
Swiss Banknotes	0.8456292	0.4714675	-37
Seeds	0.7732937	0.5299329	-24
Mice Protein Expression	0.1316575	0.0814613	-5
Crabs	0.0481402	0.0485252	0

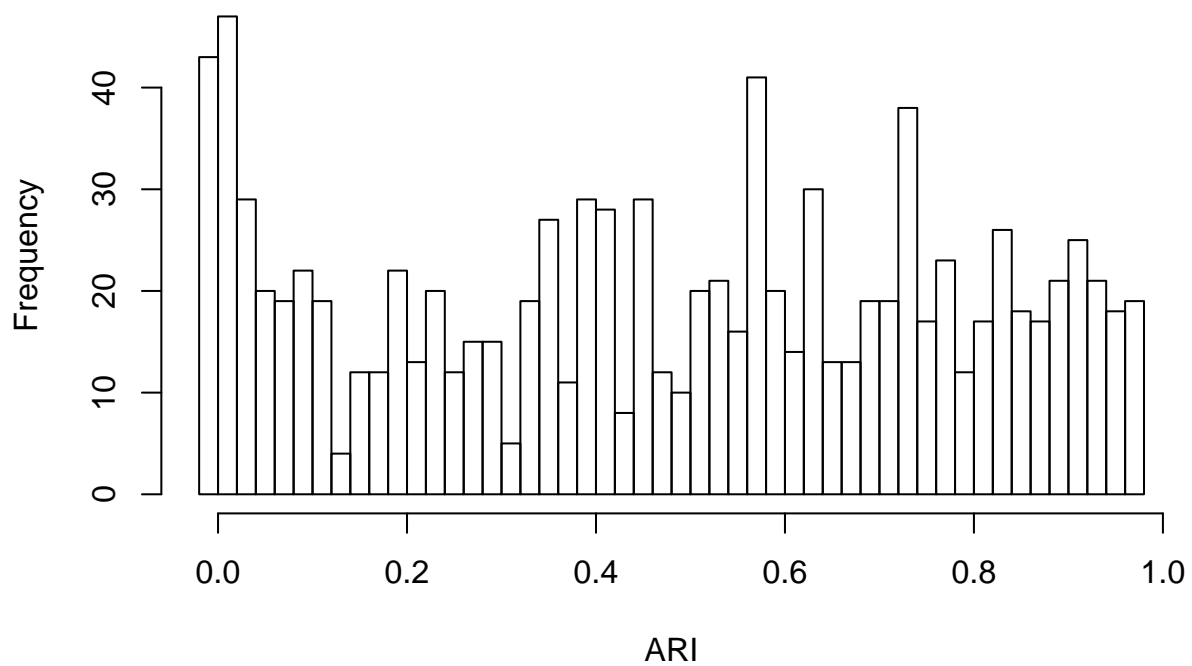
۲-۲-۵ نمودار فراوانی عملکرد خوشه‌بندی

Thyroid ARI_d

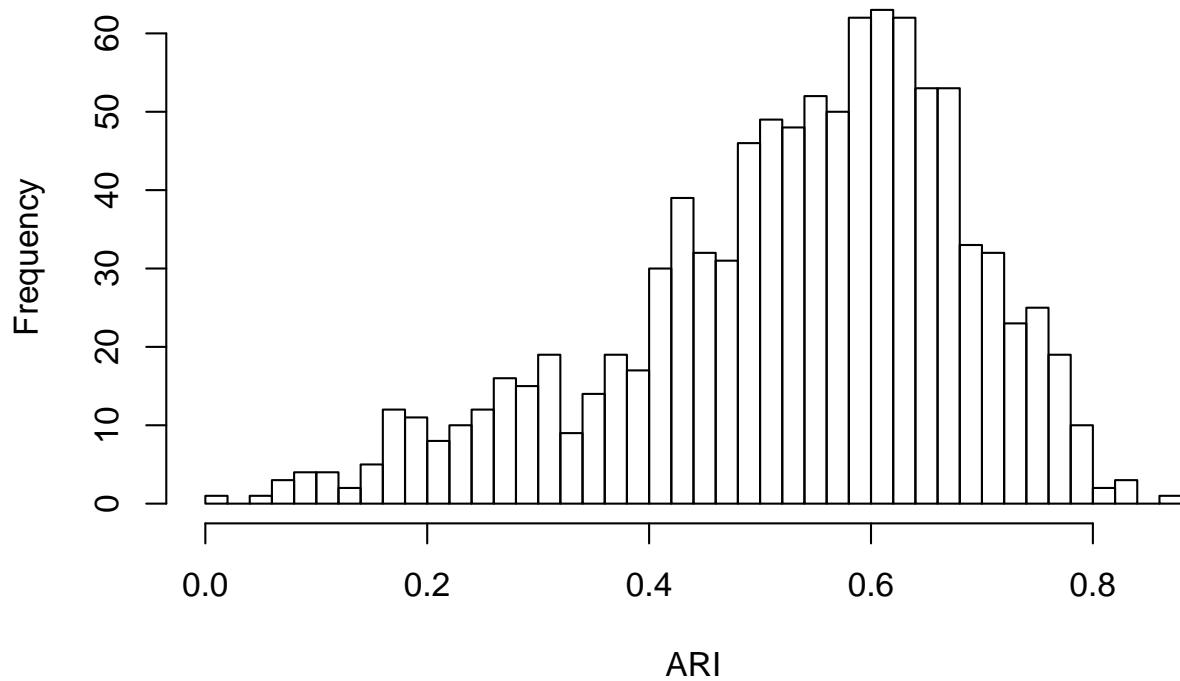




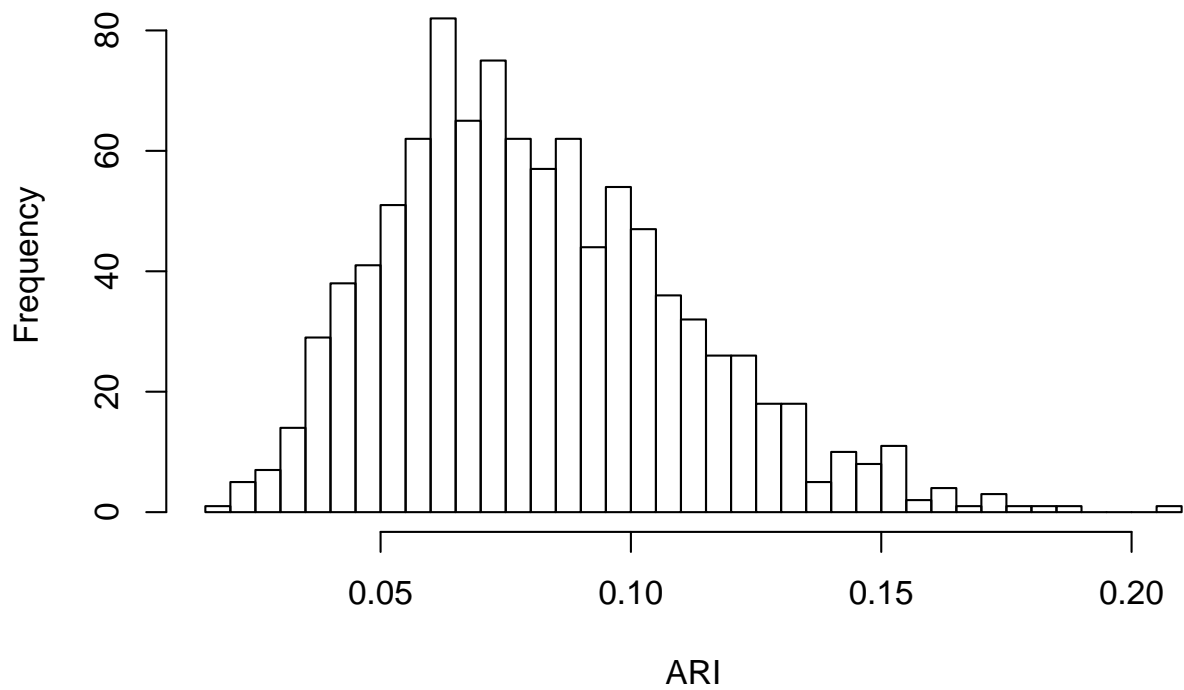
Swiss Banknotes ARI_d



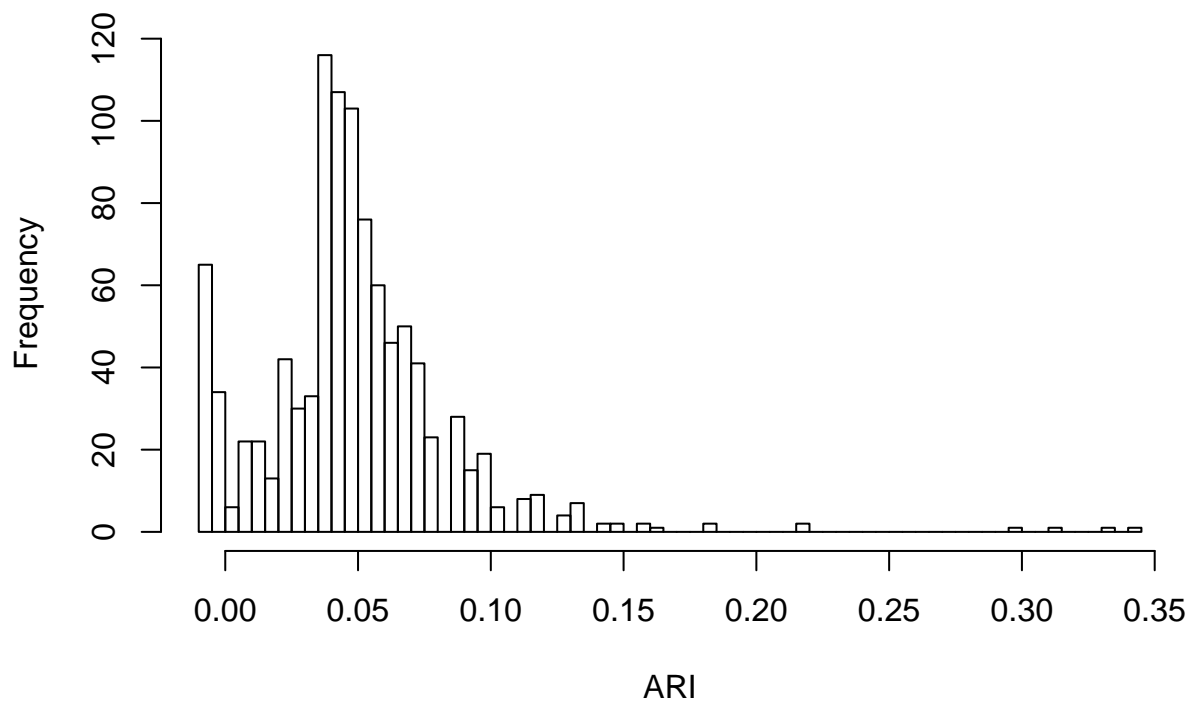
Seeds ARI_d



Mice Protein Expression ARI_d



Crabs ARI_d



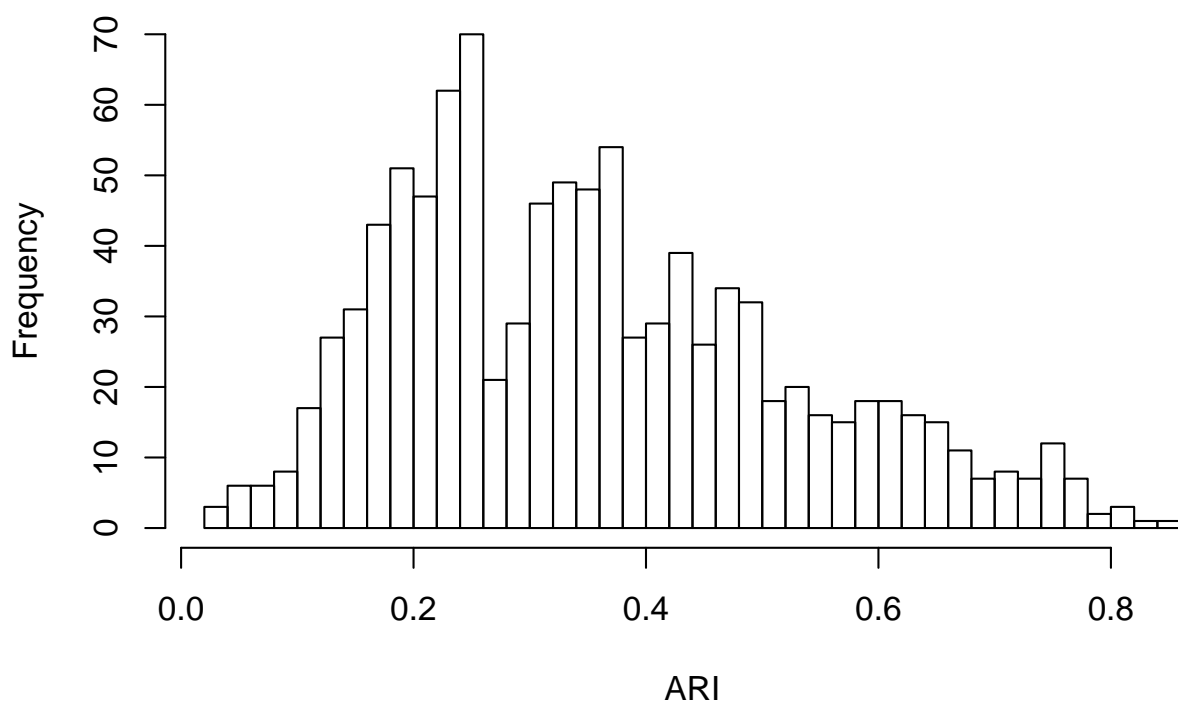
۳-۵ نتایج برای کاهش بعد کوشی ($\alpha = 1$) به دو بعد

۱-۳-۵ جداول مقایسه عملکرد خوشه‌بندی

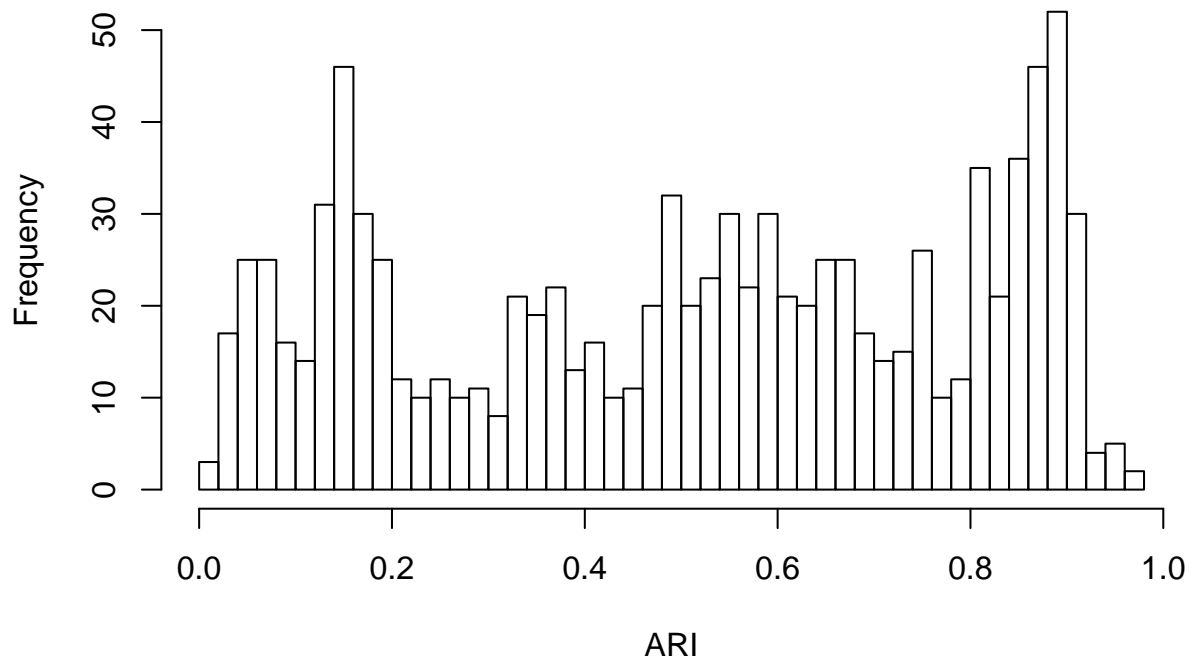
جدول ۳-۵: عملکرد تصاویر تصادفی کوشی برای کاهش بعد به دو بعد

Dataset	ARI_p	ARI_d	C_e
Thyroid	0.5831656	0.3559301	-23
Iris	0.6201352	0.5078172	-11
Diabetes	0.3801662	0.3341399	-5
Swiss Banknotes	0.8456292	0.4011119	-44
Seeds	0.7732937	0.4488349	-32
Mice Protein Expression	0.1317342	0.0592468	-7
Crabs	0.0481402	0.0469365	0

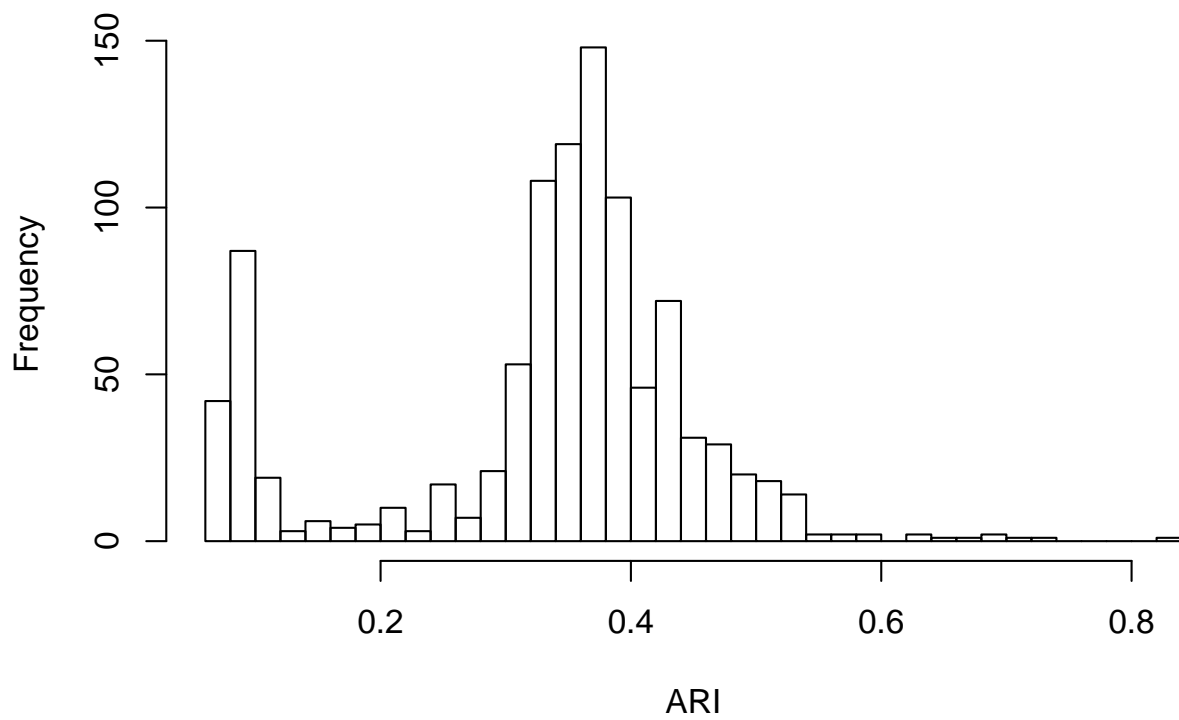
۲-۳-۵ نمودار فراوانی عملکرد خوشه‌بندی

Thyroid ARI_d 

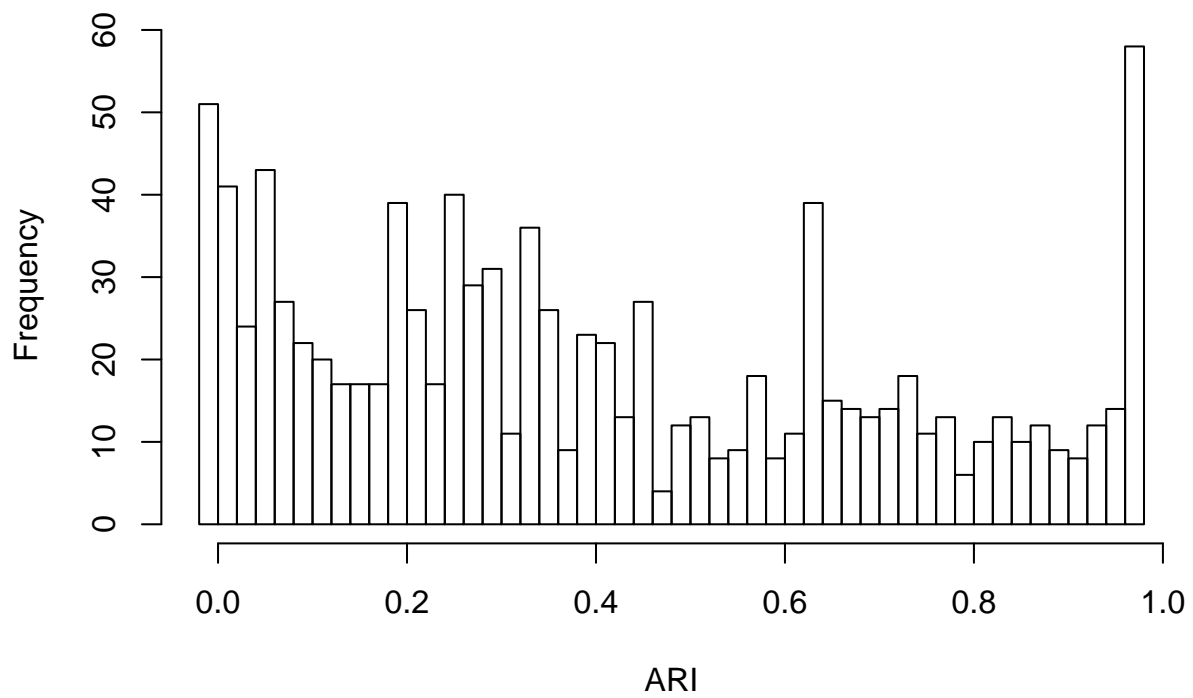
Iris ARI_d



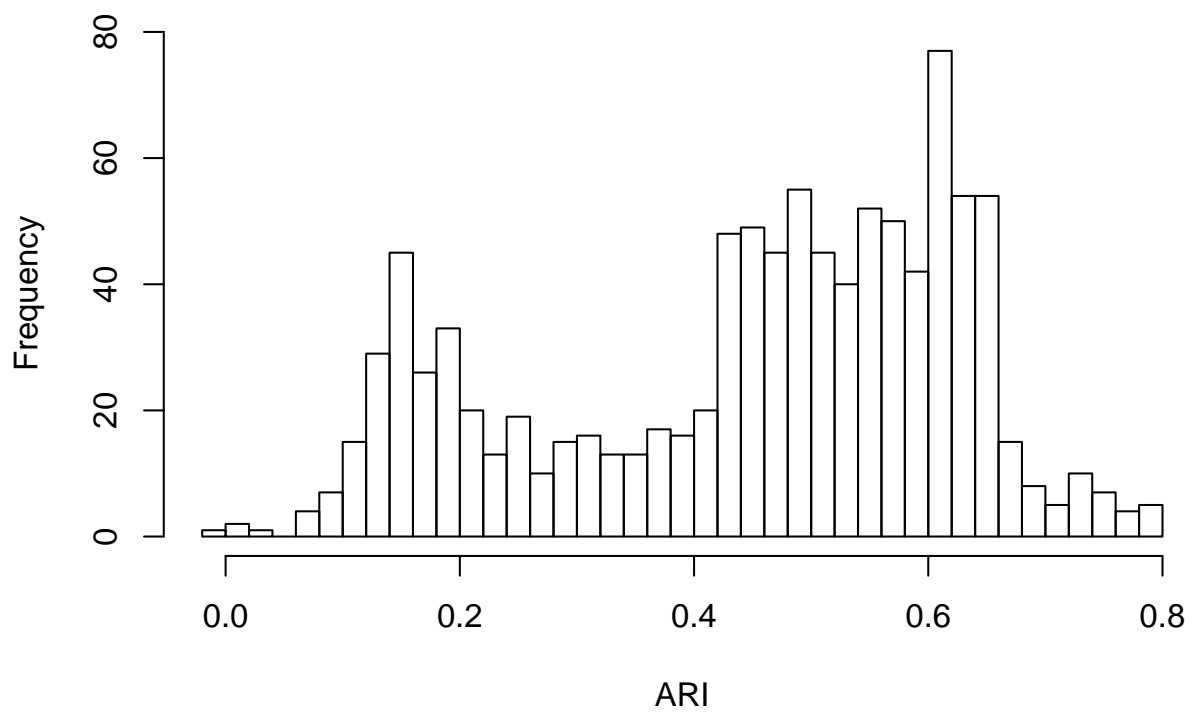
Diabetes ARI_d



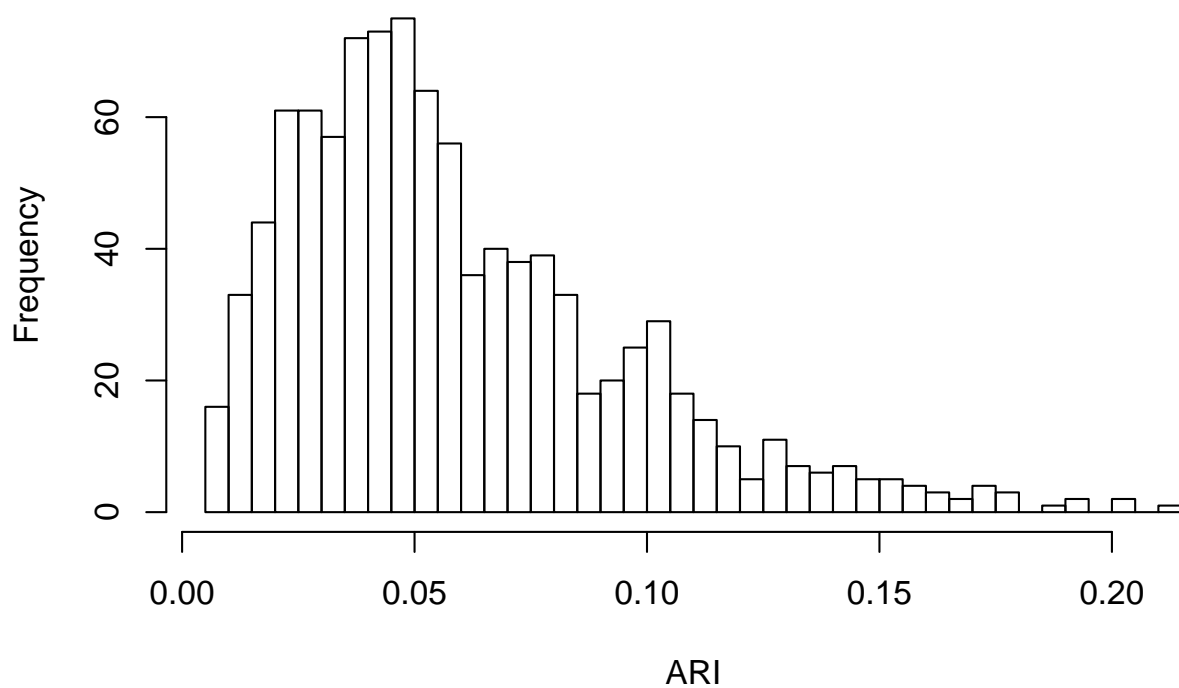
Swiss Banknotes ARI_d



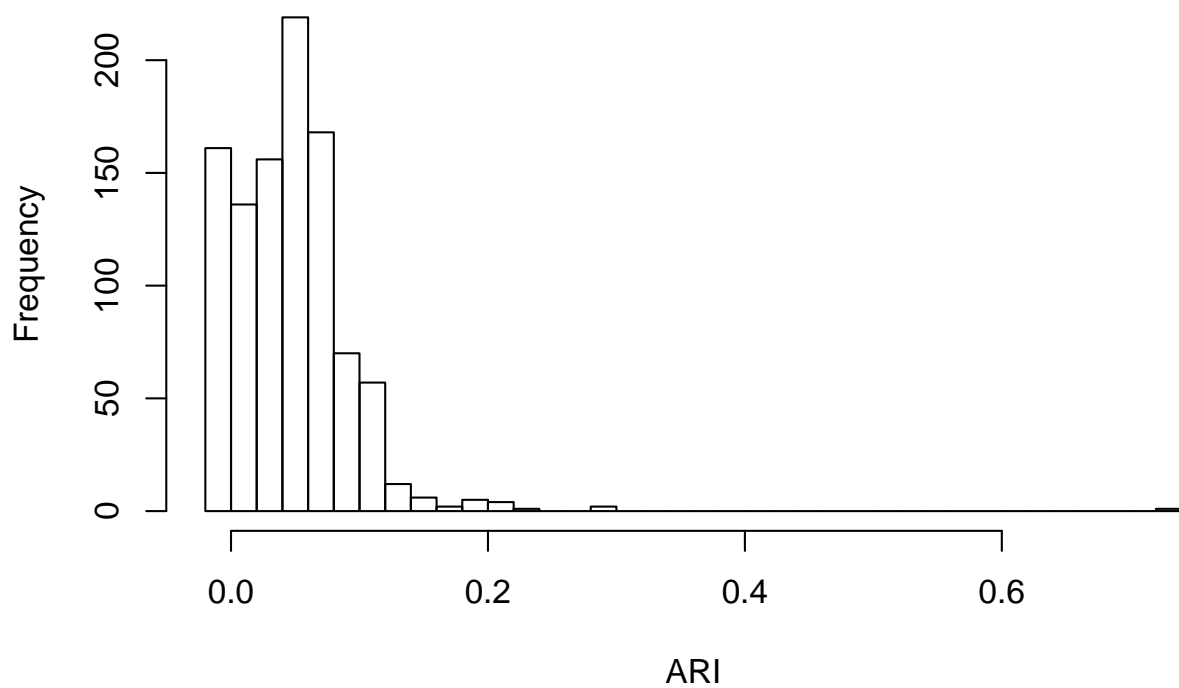
Seeds ARI_d



Mice Protein Expression ARI_d



Crabs ARI_d



۴-۵ نتایج برای کاهش بعد کوشی به سه بعد

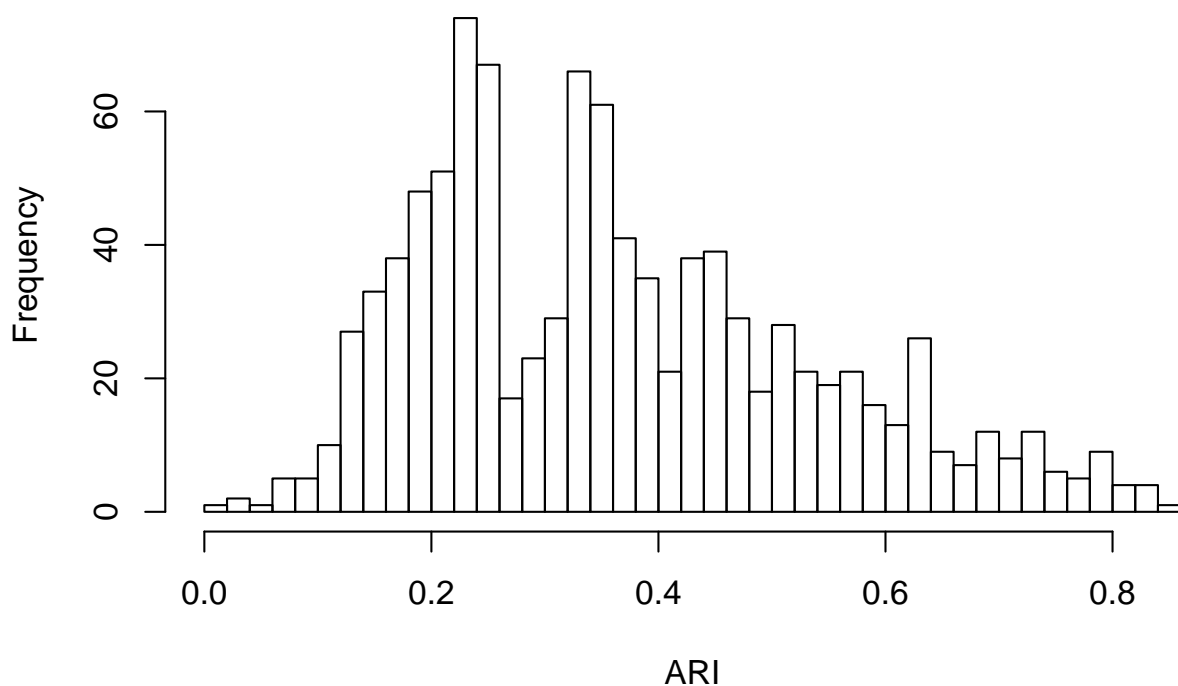
۱-۴-۵ جداول مقایسه عملکرد خوشه‌بندی

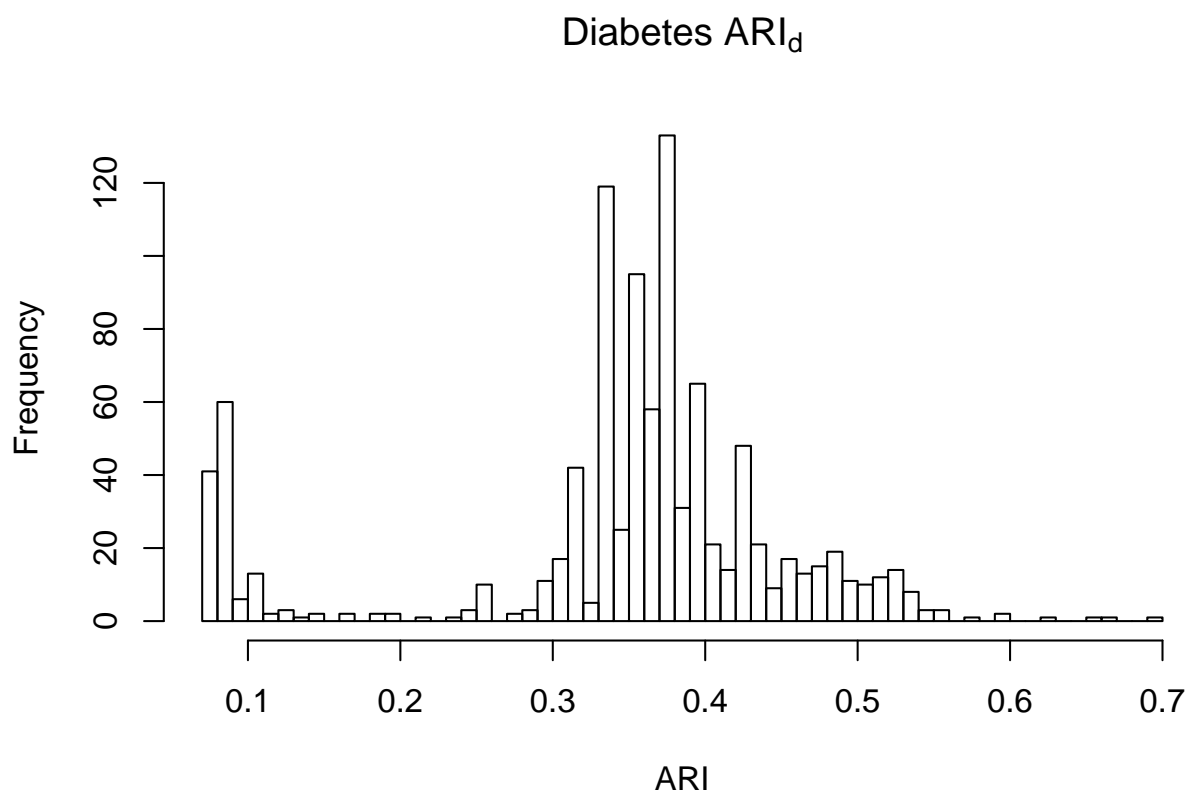
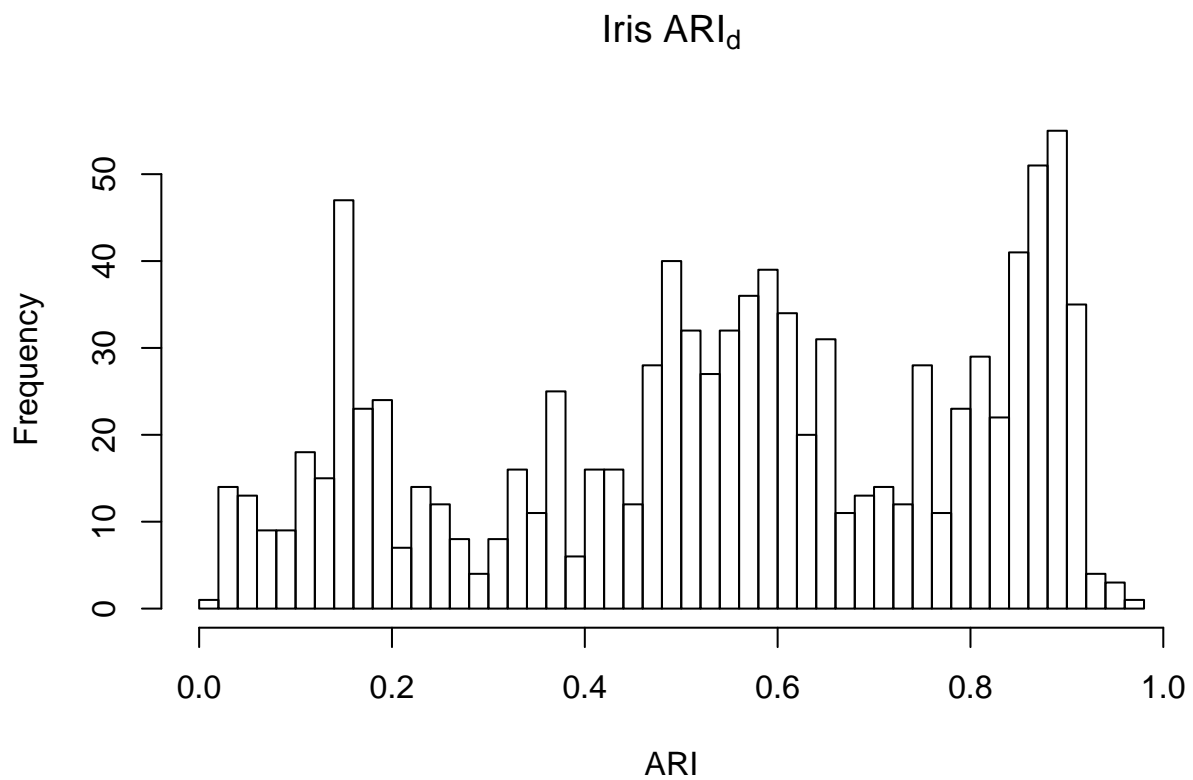
جدول ۴-۵: عملکرد تصویر تصادفی کوشی برای کاهش بعد به سه بعد

Dataset	ARI_p	ARI_d	C_e
Thyroid	0.5831656	0.3658900	-22
Iris	0.6201352	0.5441285	-8
Diabetes	0.3801662	0.3460760	-3
Swiss Banknotes	0.8456292	0.4330544	-41
Seeds	0.7732937	0.4697687	-30
Mice Protein Expression	0.1317362	0.0659775	-7
Crabs	0.0481402	0.0452313	0

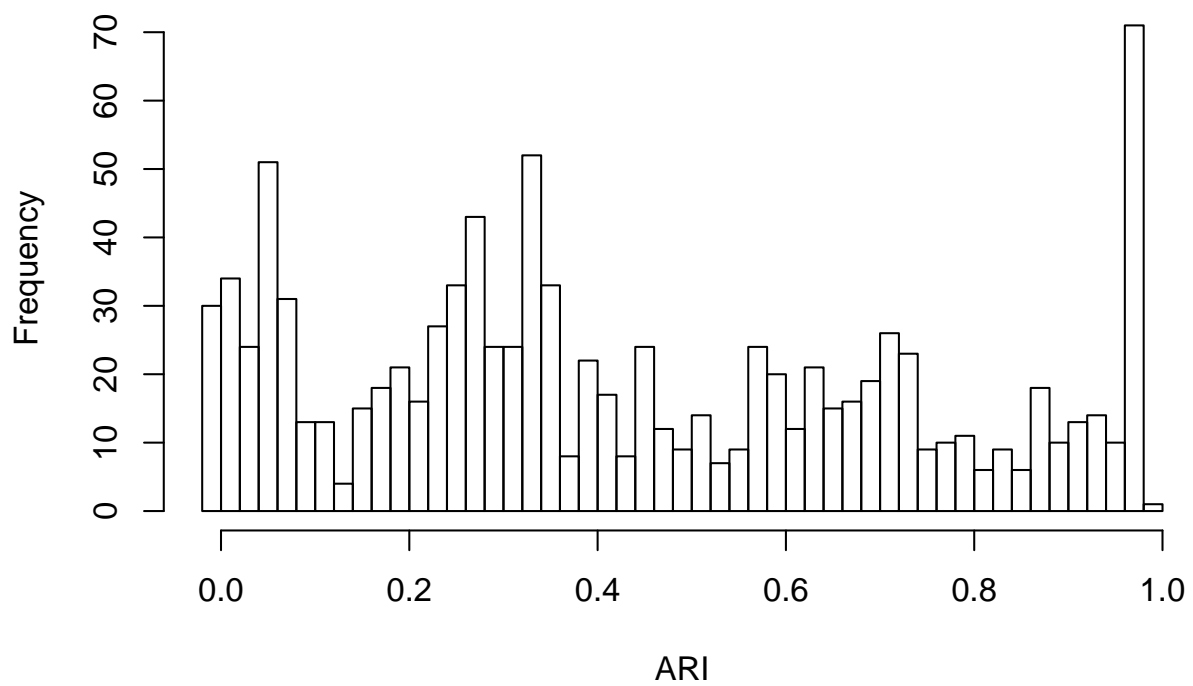
۲-۴-۵ نمودار فراوانی عملکرد خوشه‌بندی

Thyroid ARI_d

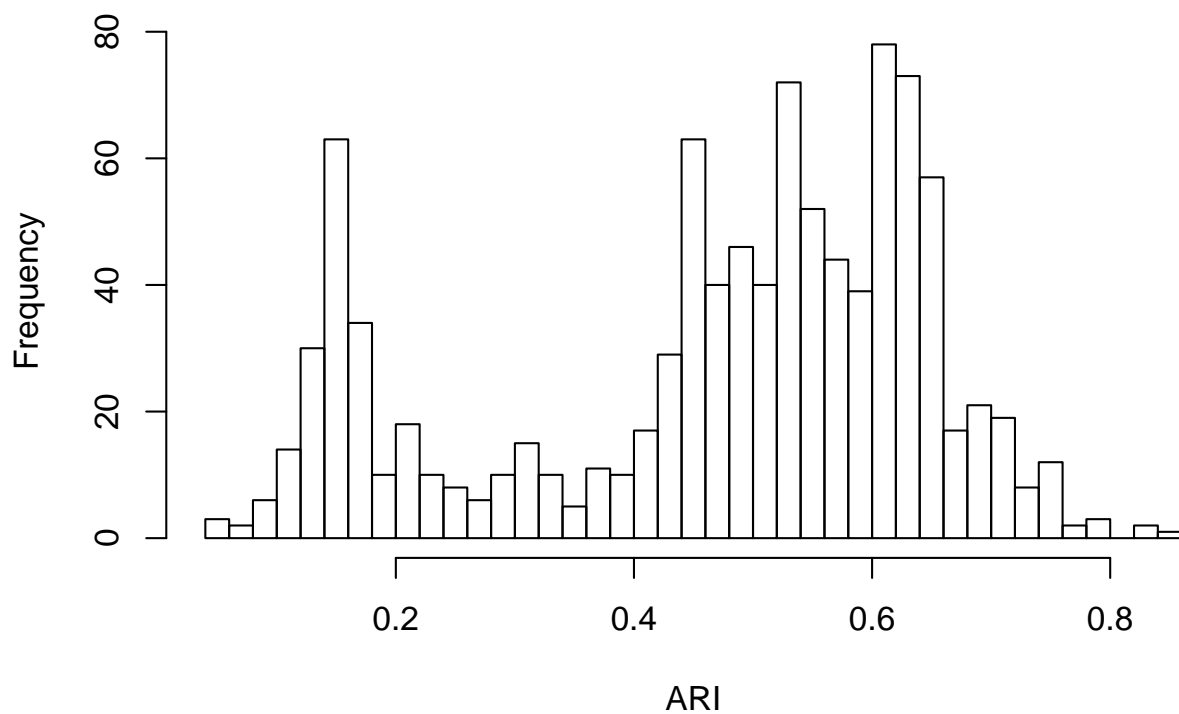




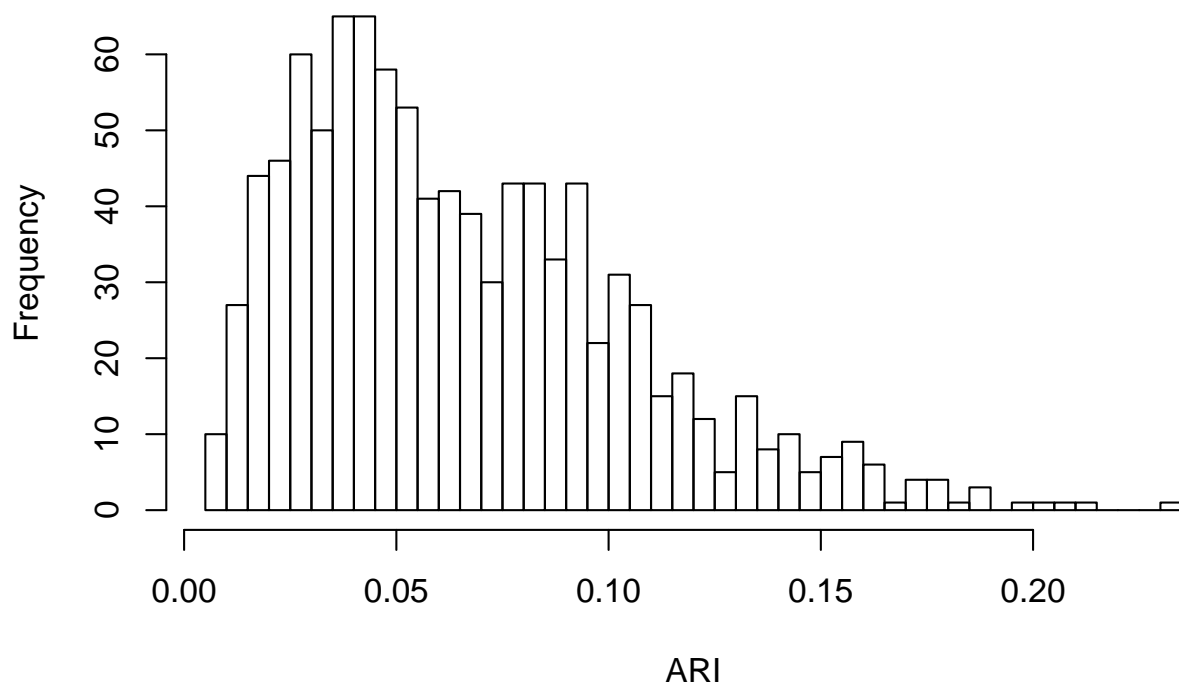
Swiss Banknotes ARI_d



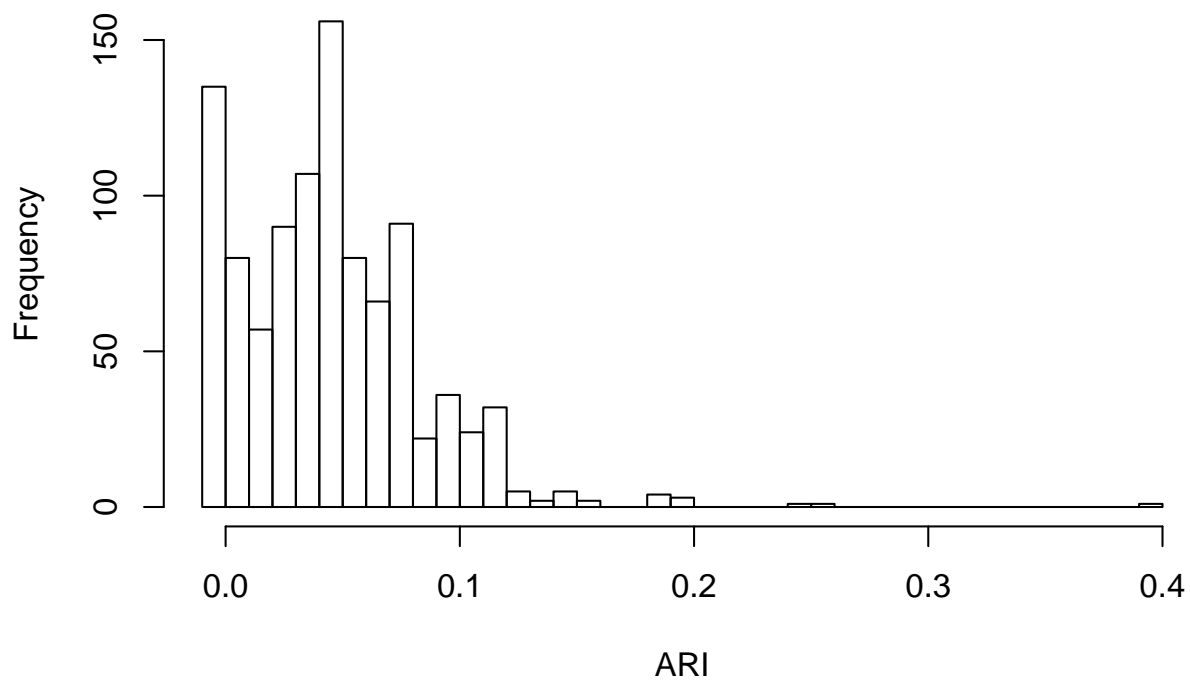
Seeds ARI_d



Mice Protein Expression ARI_d



Crabs ARI_d



۵-۵ نتایج برای کاهش بعد گسسته $s = 2$ به دو بعد

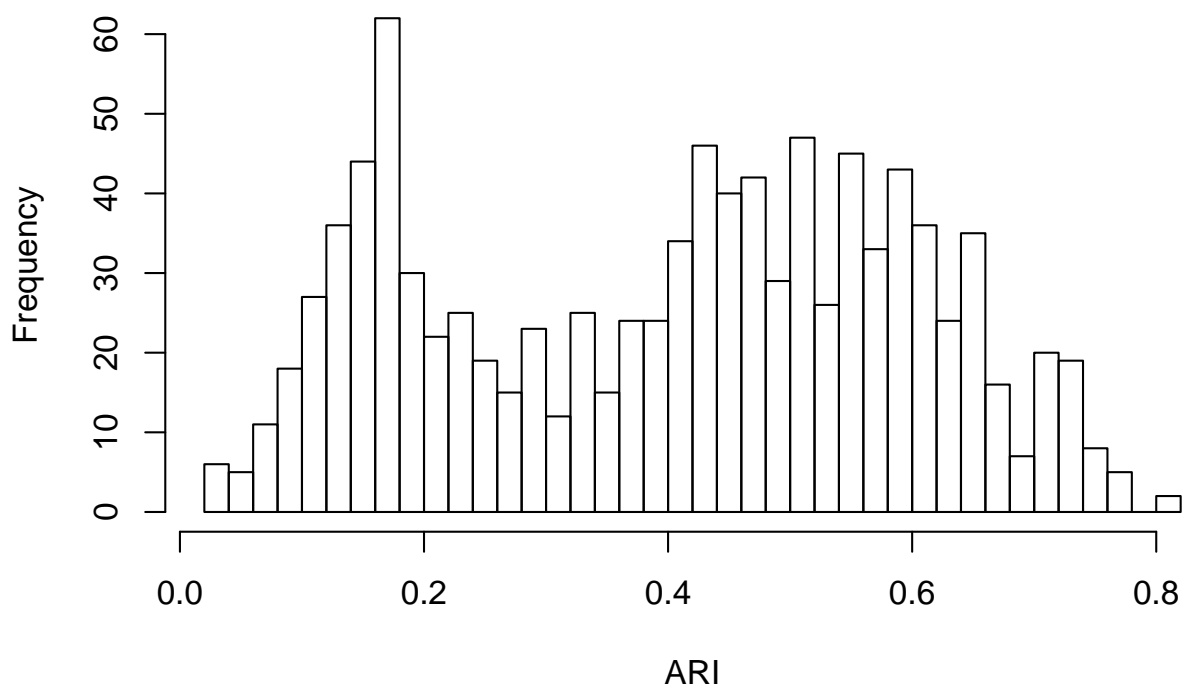
۱-۵-۵ جداول مقایسه عملکرد خوشه‌بندی

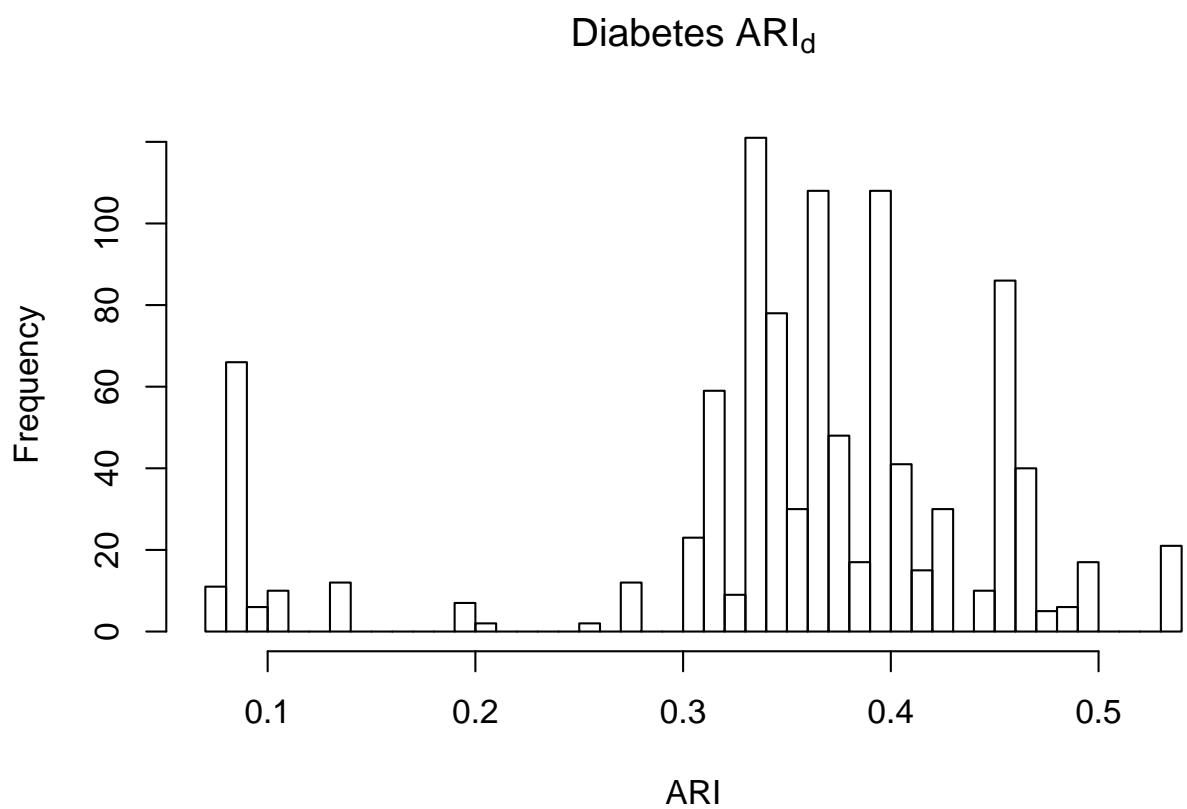
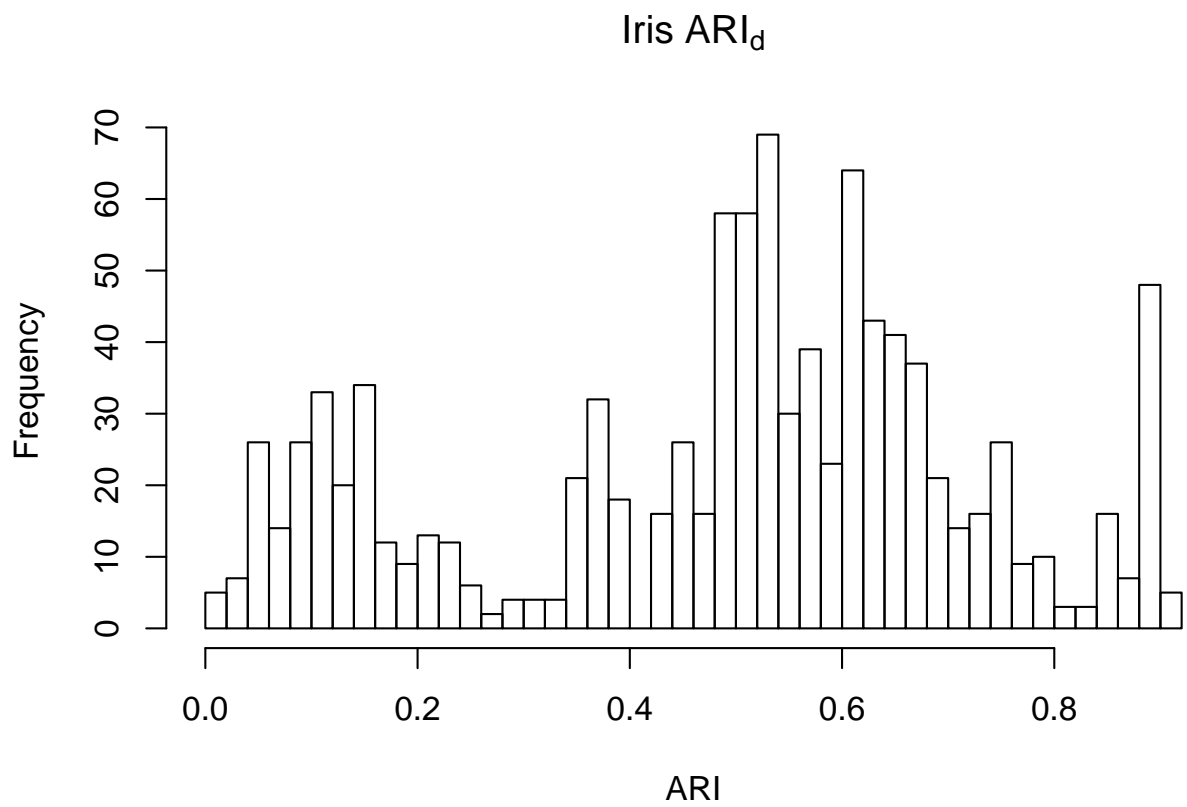
جدول ۵-۵: عملکرد تصویر تصادفی گسسته با $s = 1$ برای کاهش بعد به دو بعد

Dataset	ARI_p	ARI_d	C_e
Thyroid	0.5831656	0.4016237	-18
Iris	0.6201352	0.4890477	-13
Diabetes	0.3801662	0.3515435	-3
Swiss Banknotes	0.8456292	0.4012454	-44
Seeds	0.7732937	0.4459104	-33
Mice Protein Expression	0.1314435	0.0647088	-7
Crabs	0.0481402	0.0467305	0

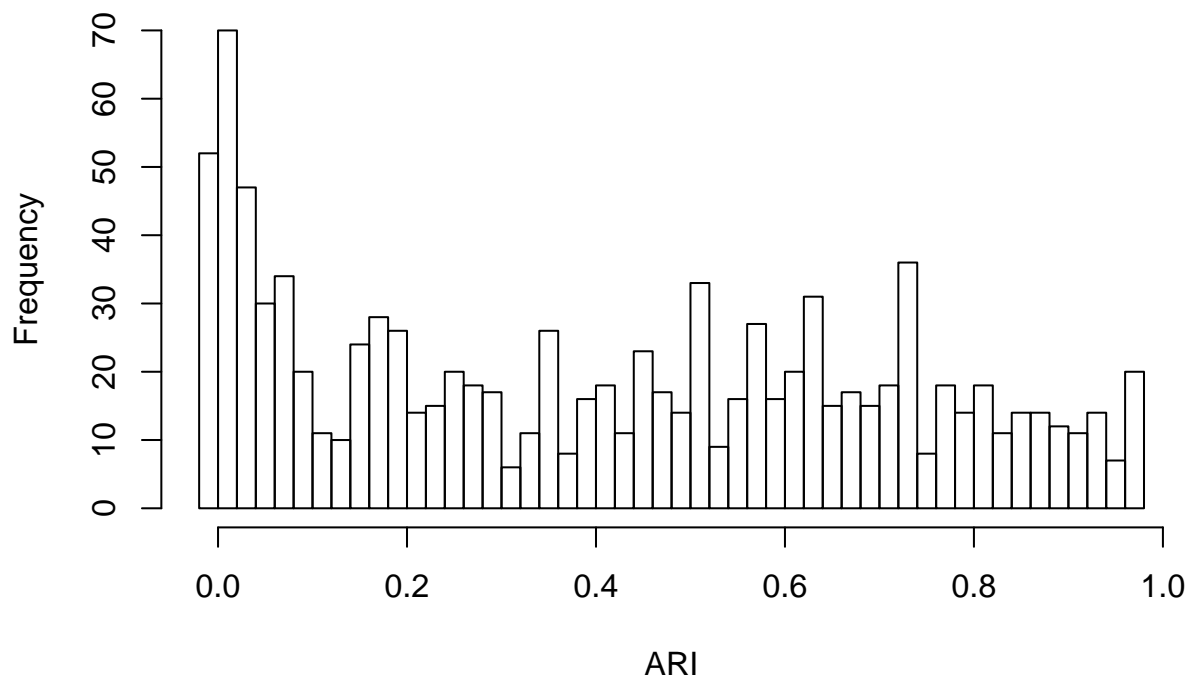
۲-۵-۵ نمودار فراوانی عملکرد خوشه‌بندی

Thyroid ARI_d

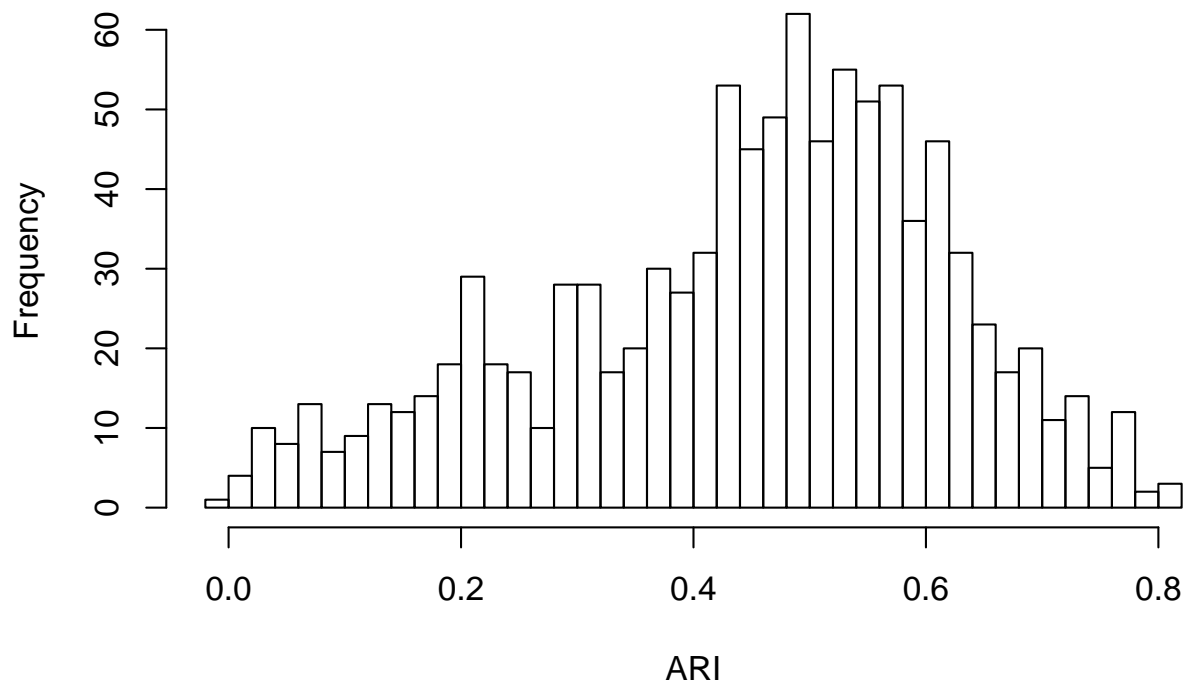




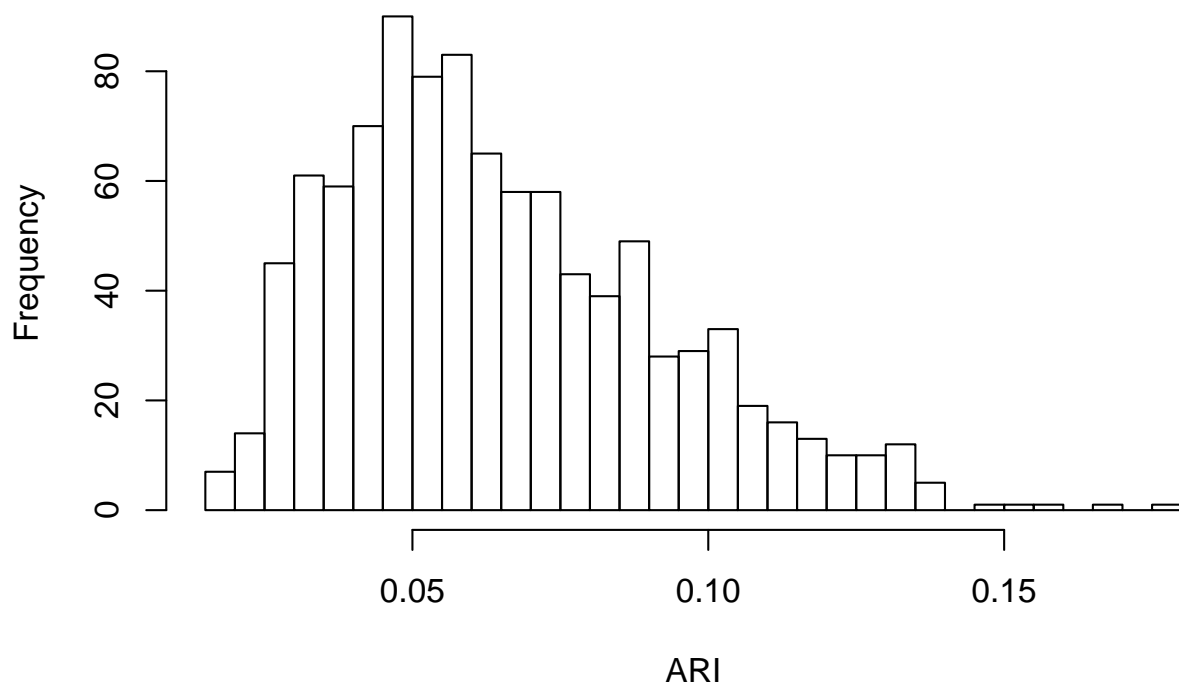
Swiss Banknotes ARI_d



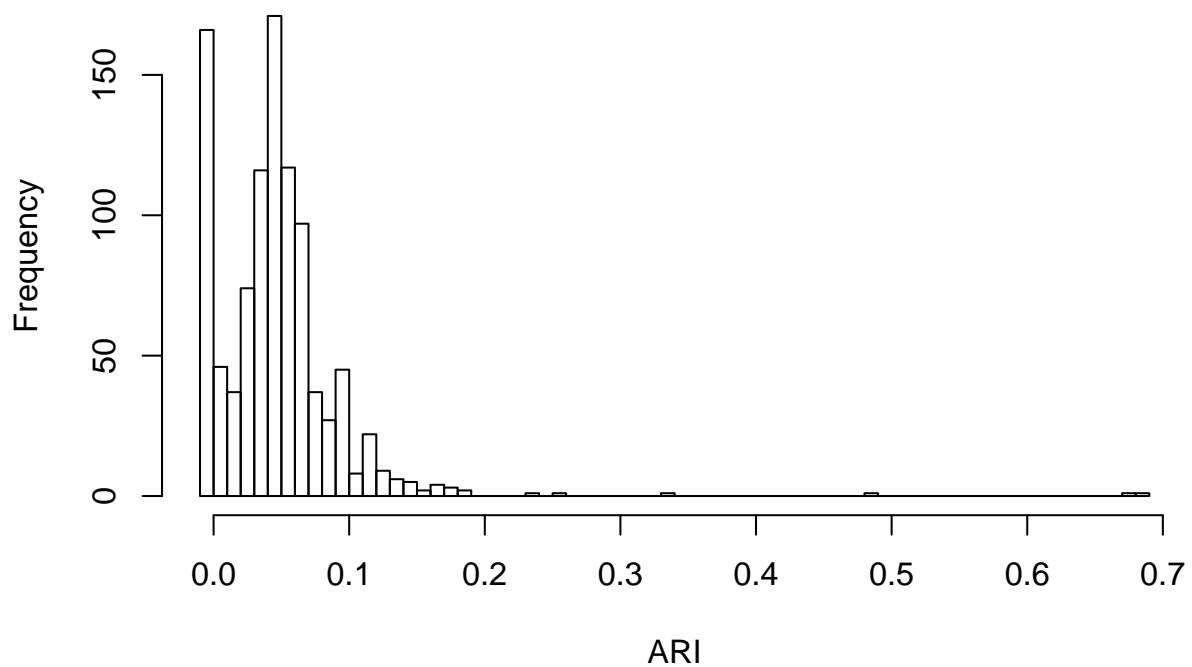
Seeds ARI_d



Mice Protein Expression ARI_d



Crabs ARI_d



۵-۶ نتایج برای کاهش بعد گسسته $s = 2$ به سه بعد

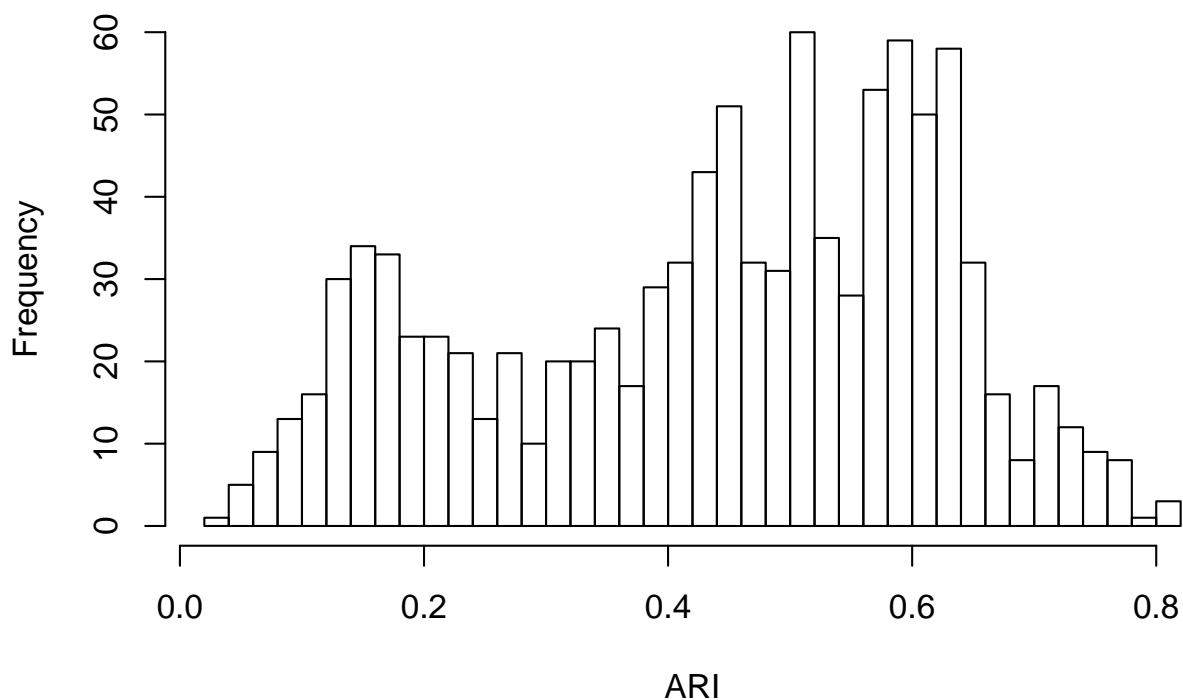
۵-۶-۱ جداول مقایسه عملکرد خوشه‌بندی

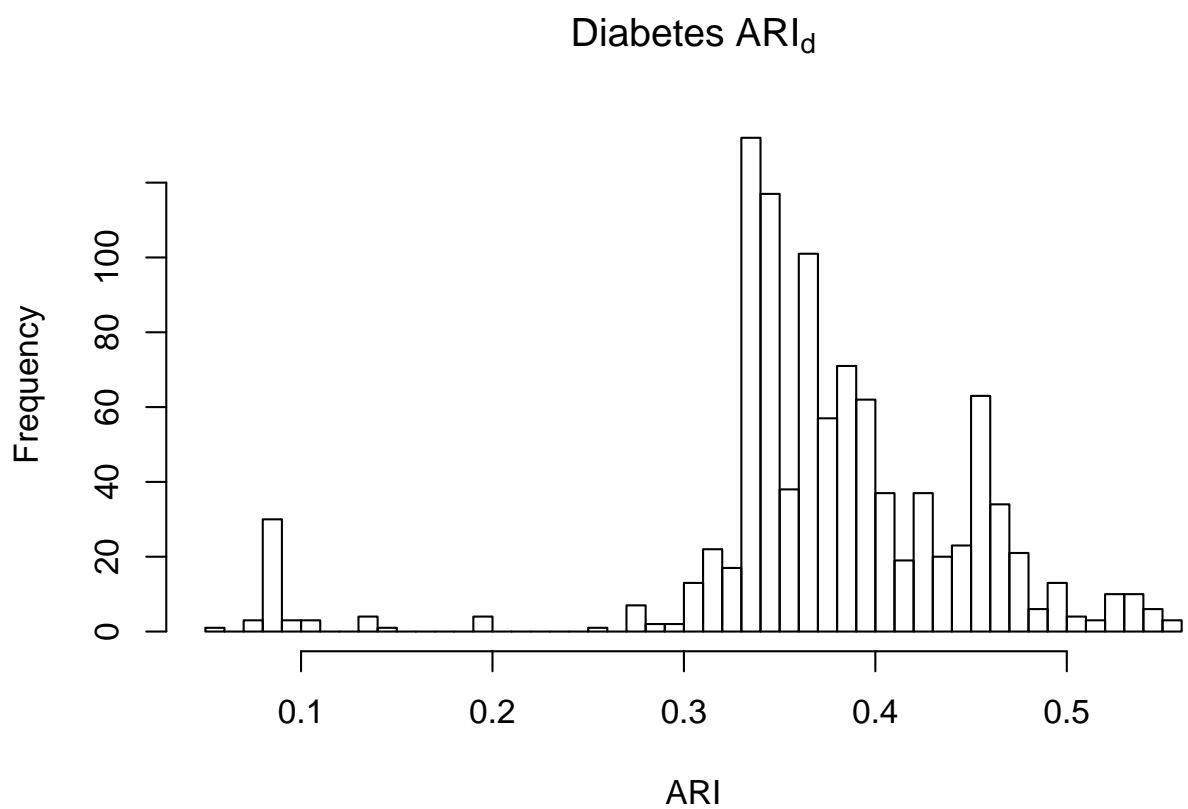
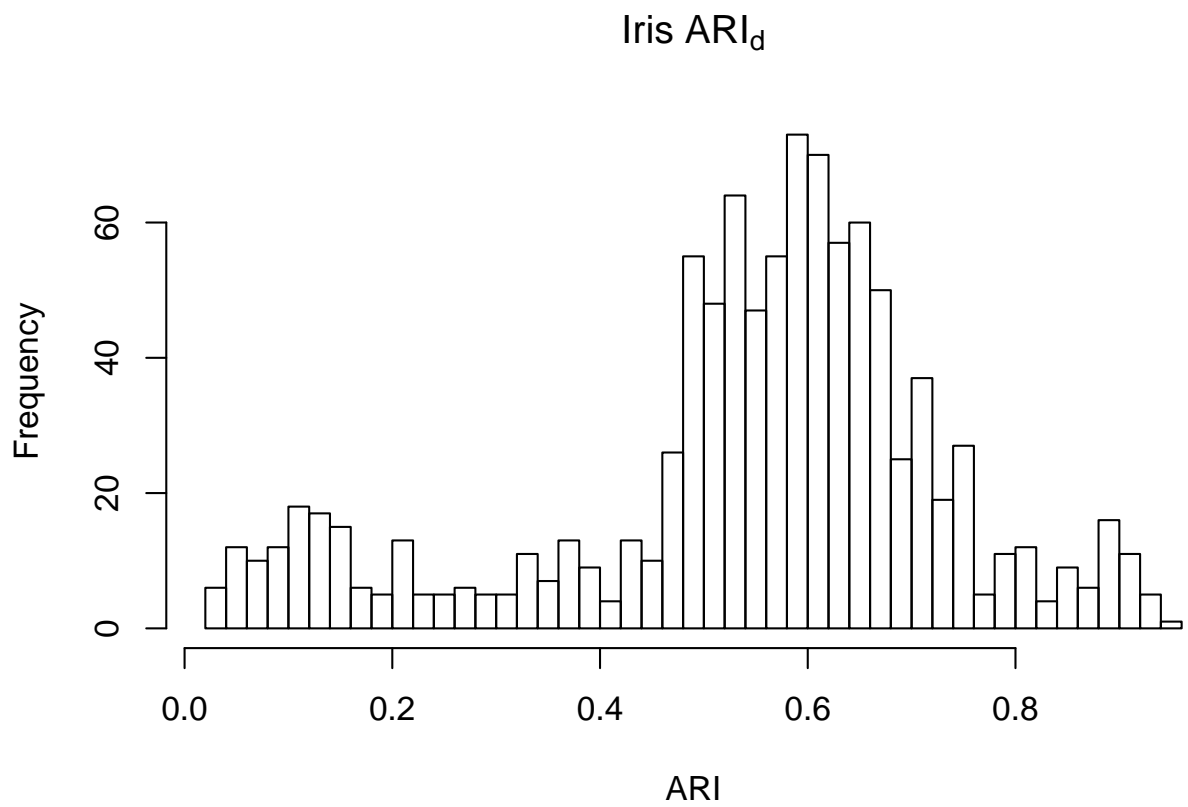
جدول ۵-۶: عملکرد تصویر تصادفی گسسته با $s = 2$ برای کاهش بعد به سه بعد

Dataset	ARI_p	ARI_d	C_e
Thyroid	0.5831656	0.4386178	-14
Iris	0.6201352	0.5410759	-8
Diabetes	0.3801662	0.3735668	-1
Swiss Banknotes	0.8456292	0.4881694	-36
Seeds	0.7732937	0.5289697	-24
Mice Protein Expression	0.1314406	0.0798832	-5
Crabs	0.0481402	0.0463655	0

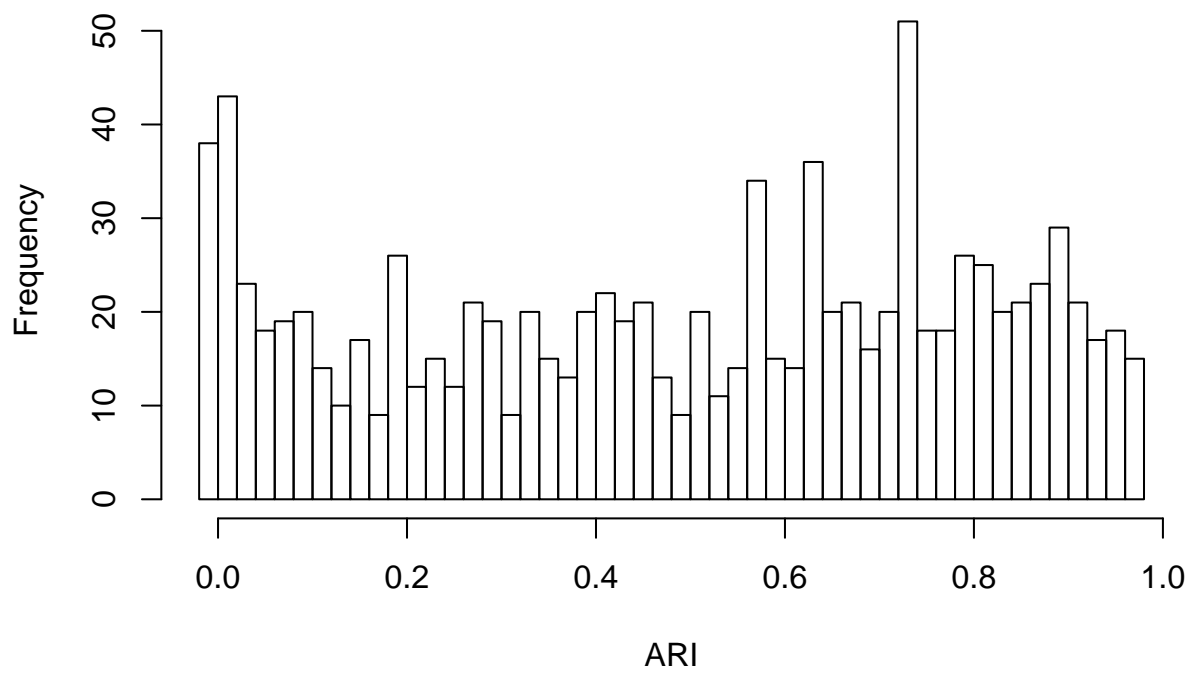
۵-۶-۲ نمودار فراوانی عملکرد خوشه‌بندی

Thyroid ARI_d

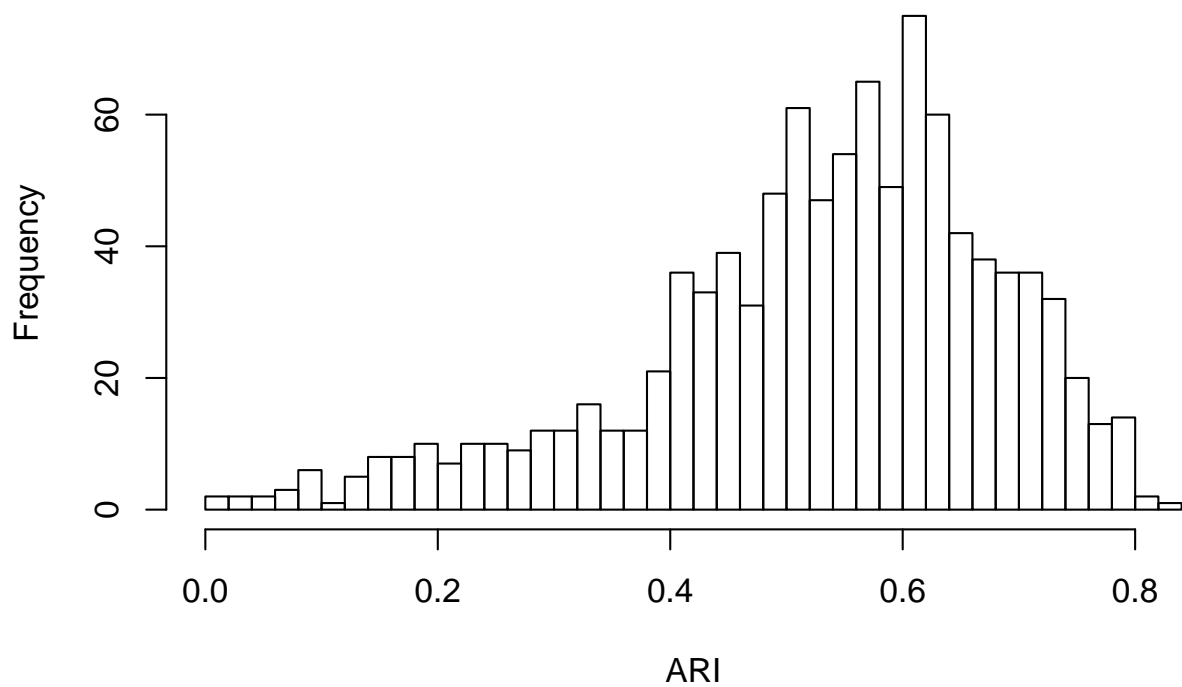




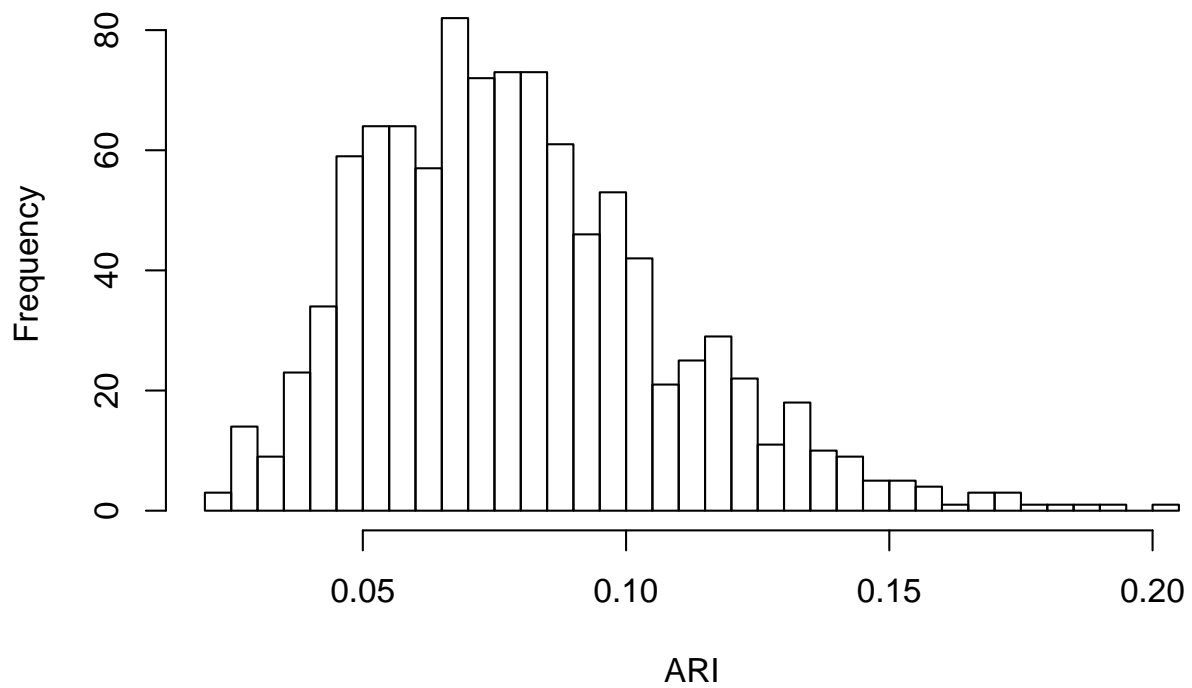
Swiss Banknotes ARI_d



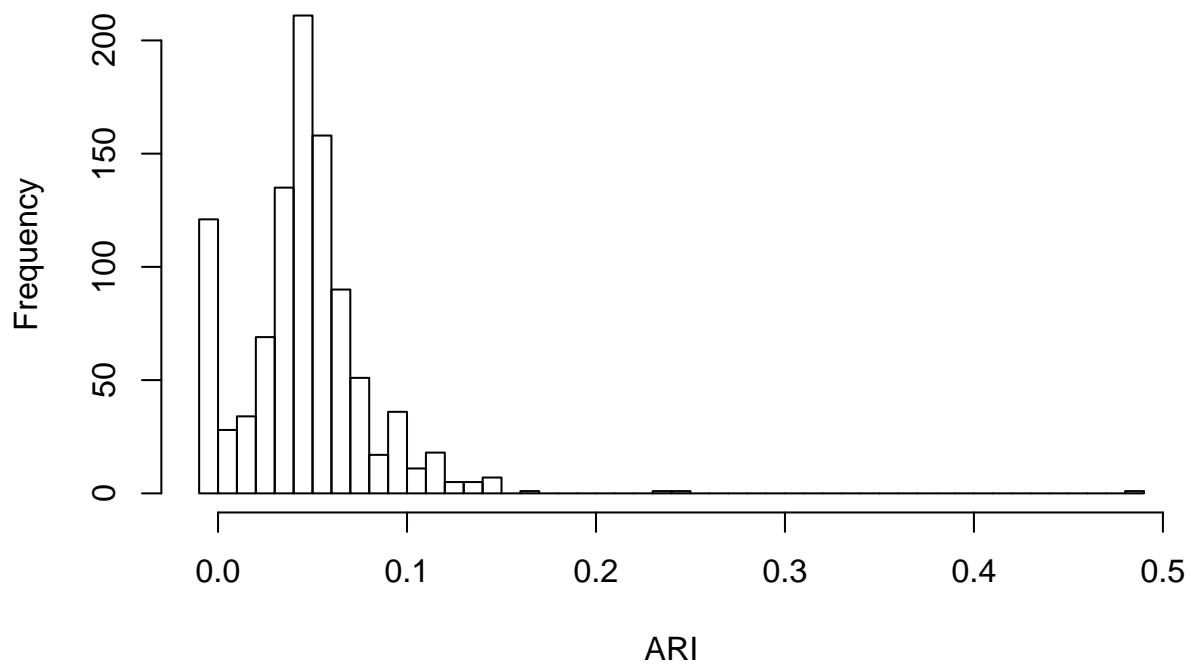
Seeds ARI_d



Mice Protein Expression ARI_d



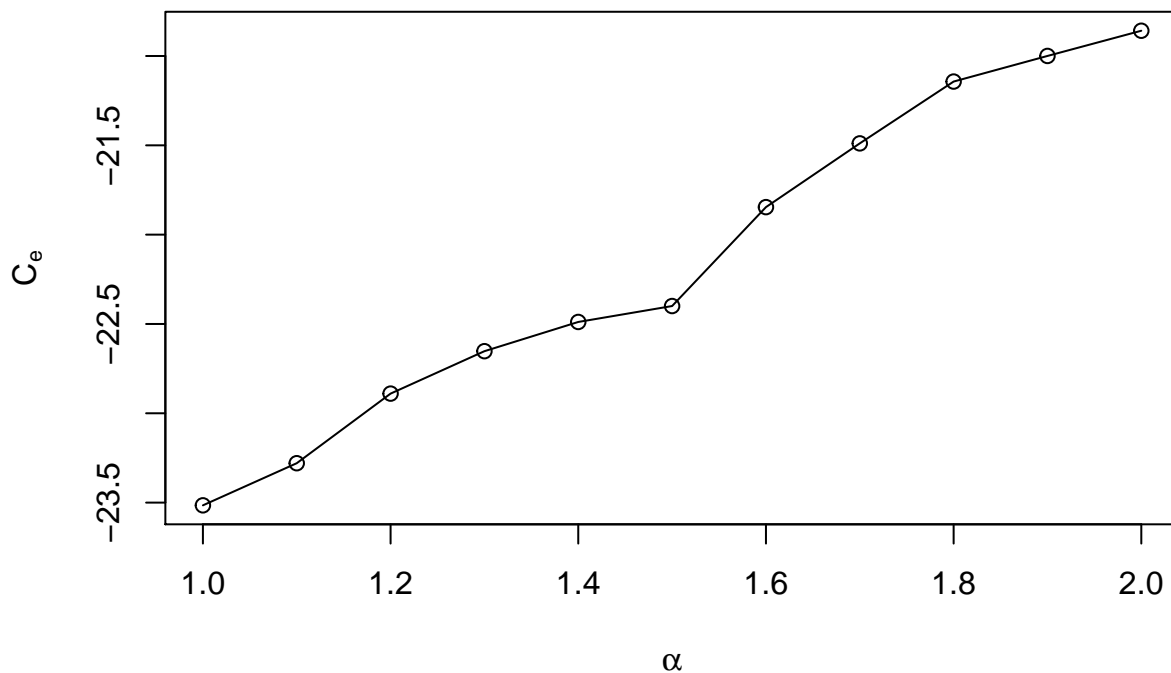
Crabs ARI_d



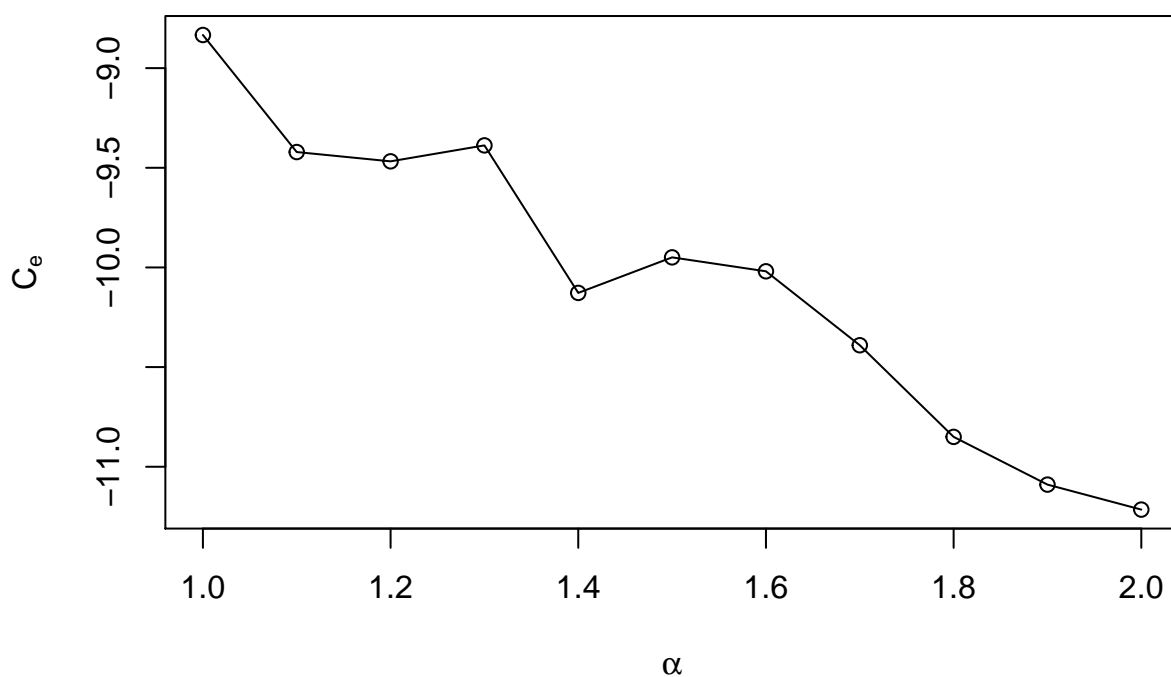
۷-۵ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر α

برای کاهش بعد به دو بعد

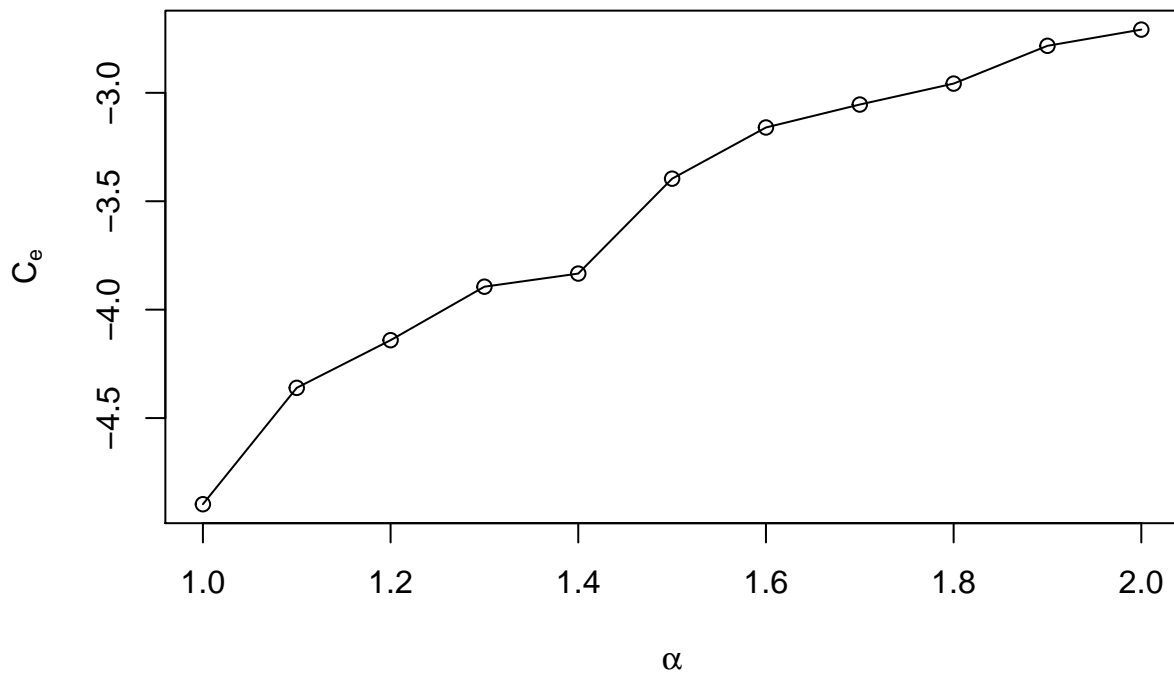
Thyroid C_e vs. α



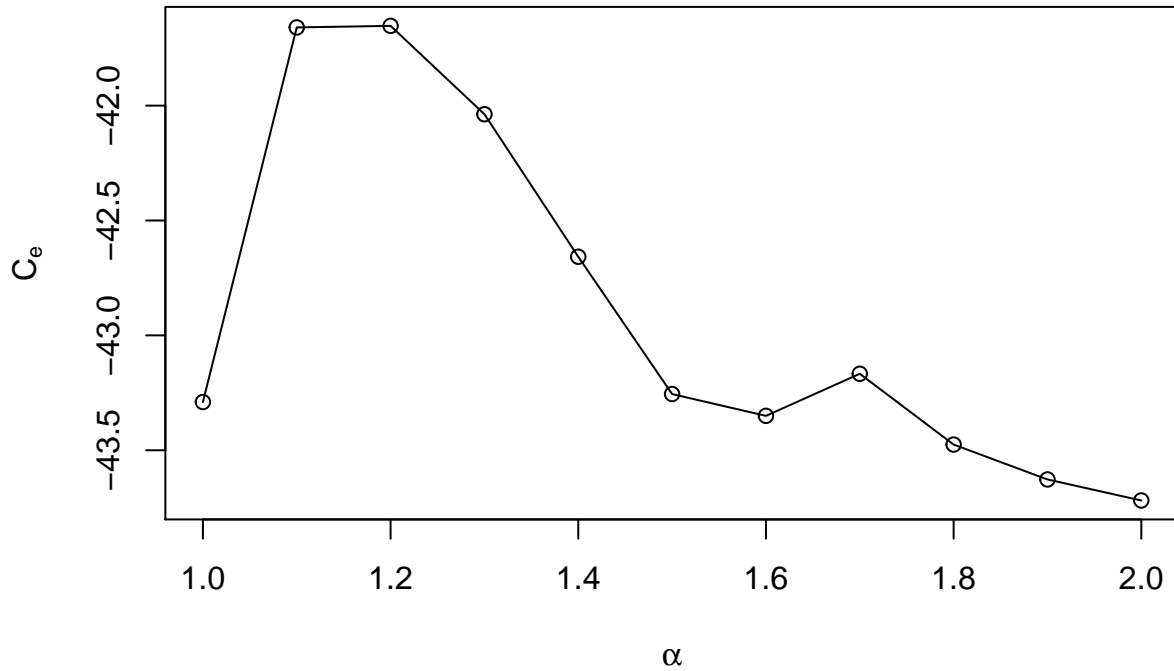
Iris C_e vs. α



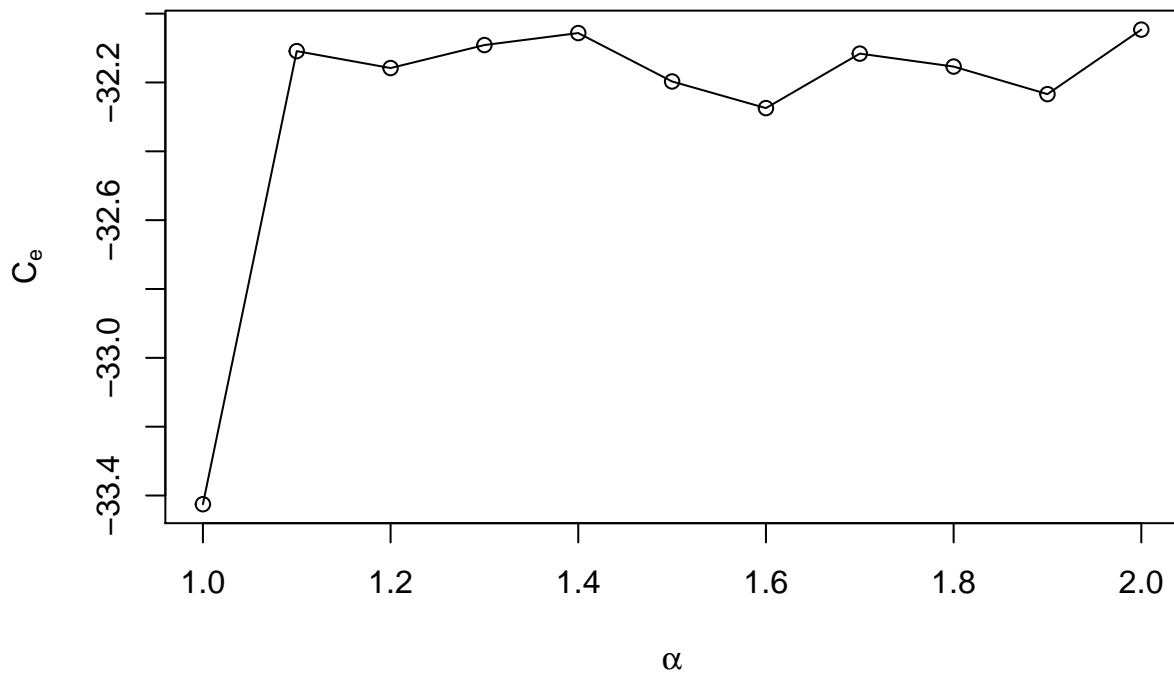
Diabetes C_e vs. α



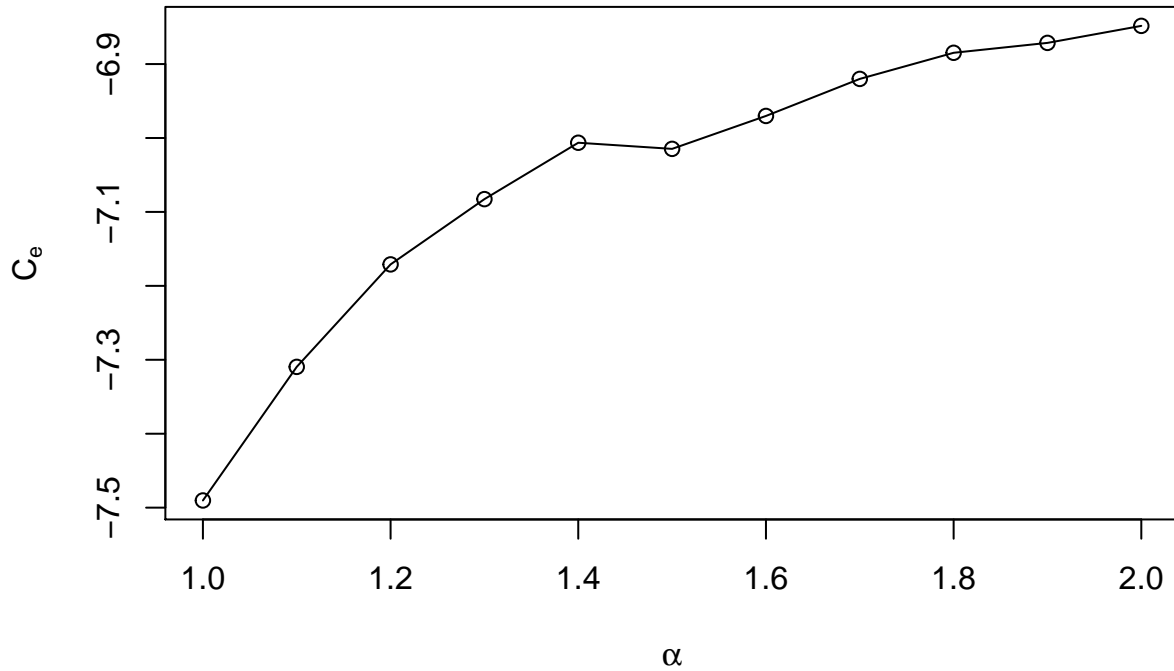
Swiss Banknotes C_e vs. α



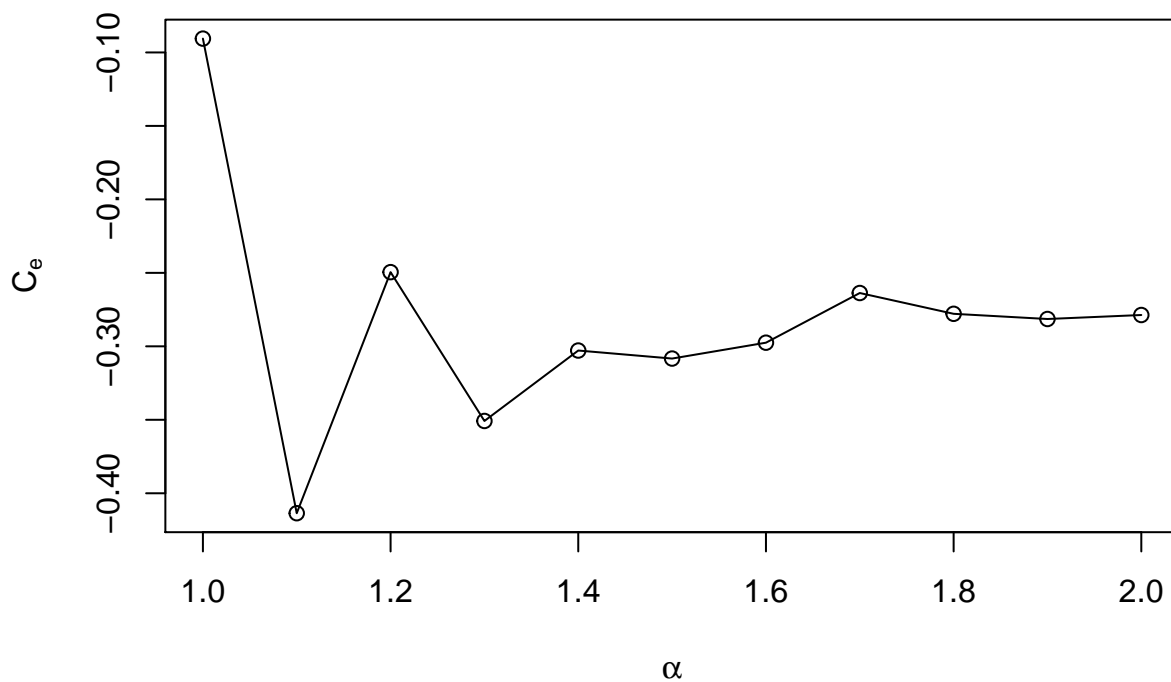
Seeds C_e vs. α



Mice Protein Expression C_e vs. α

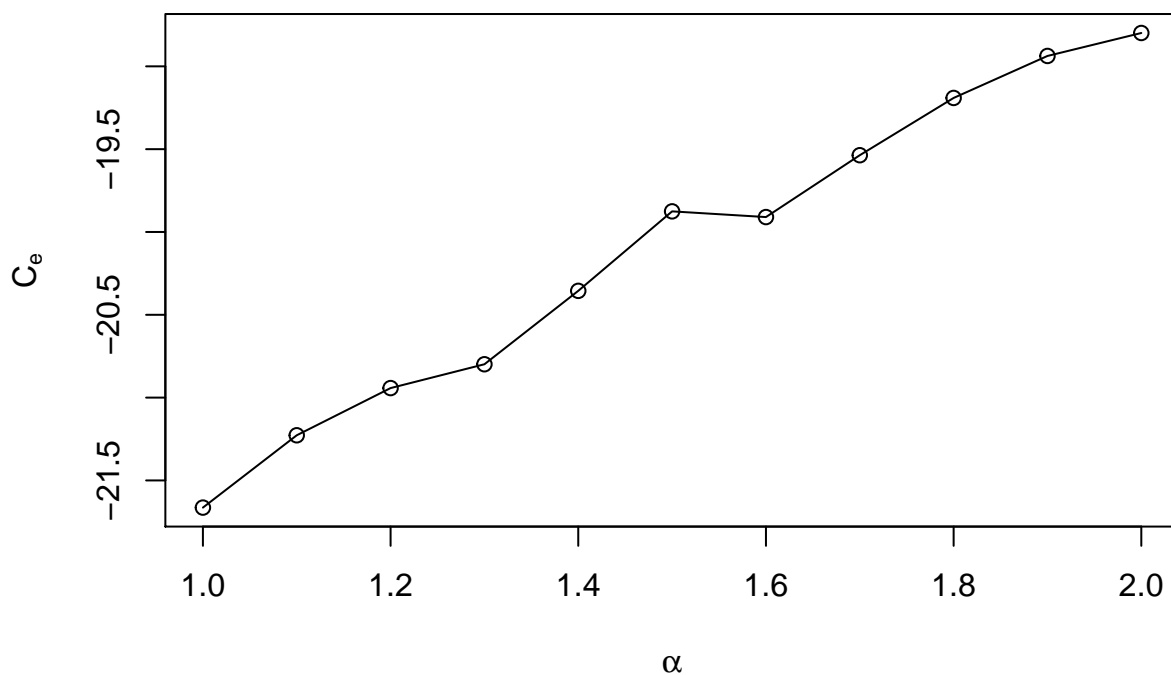


Crabs C_e vs. α

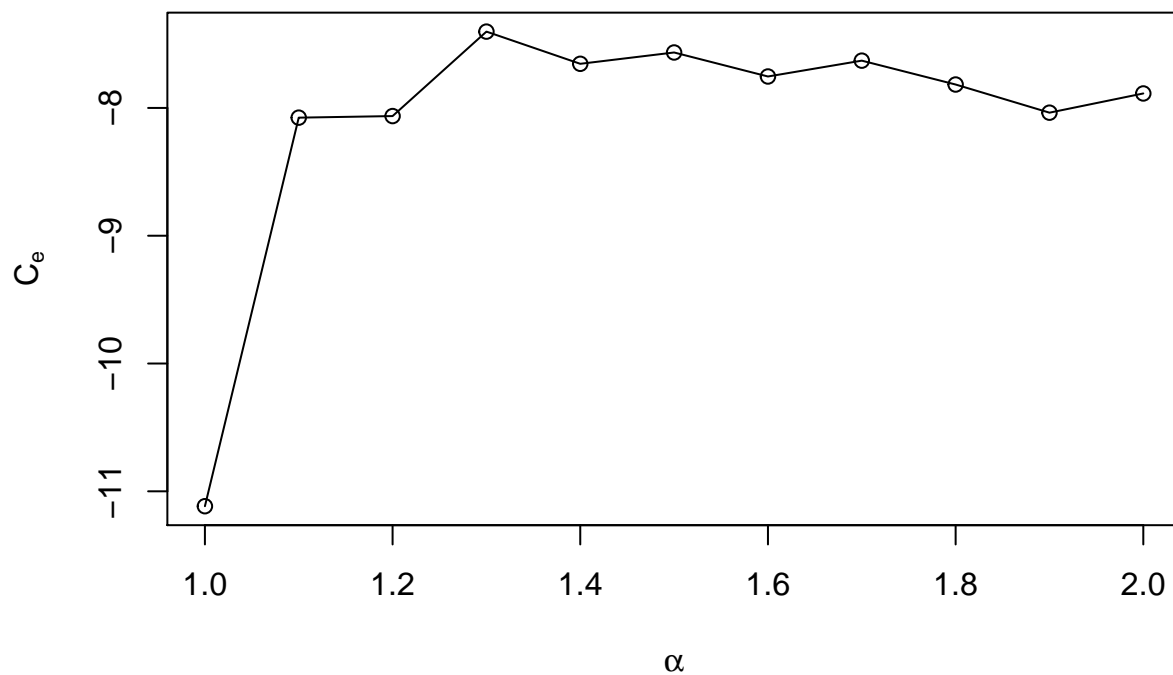


۸-۵ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر α
برای کاهش بعد به سه بعد

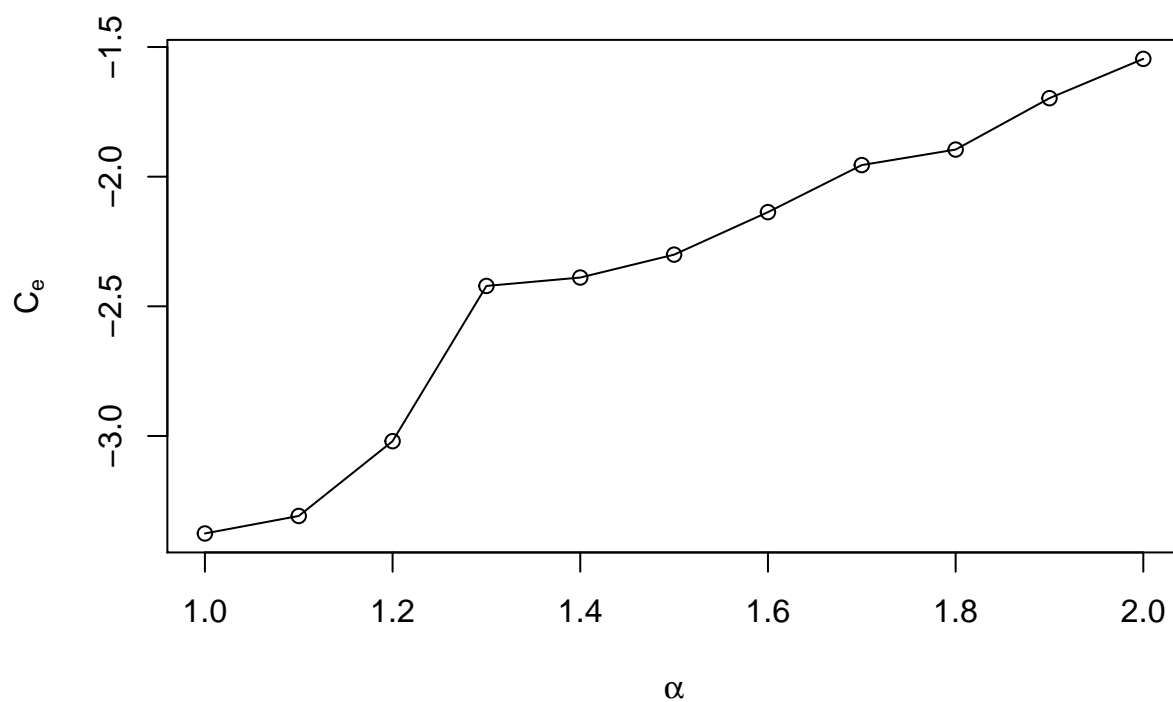
Thyroid C_e vs. α



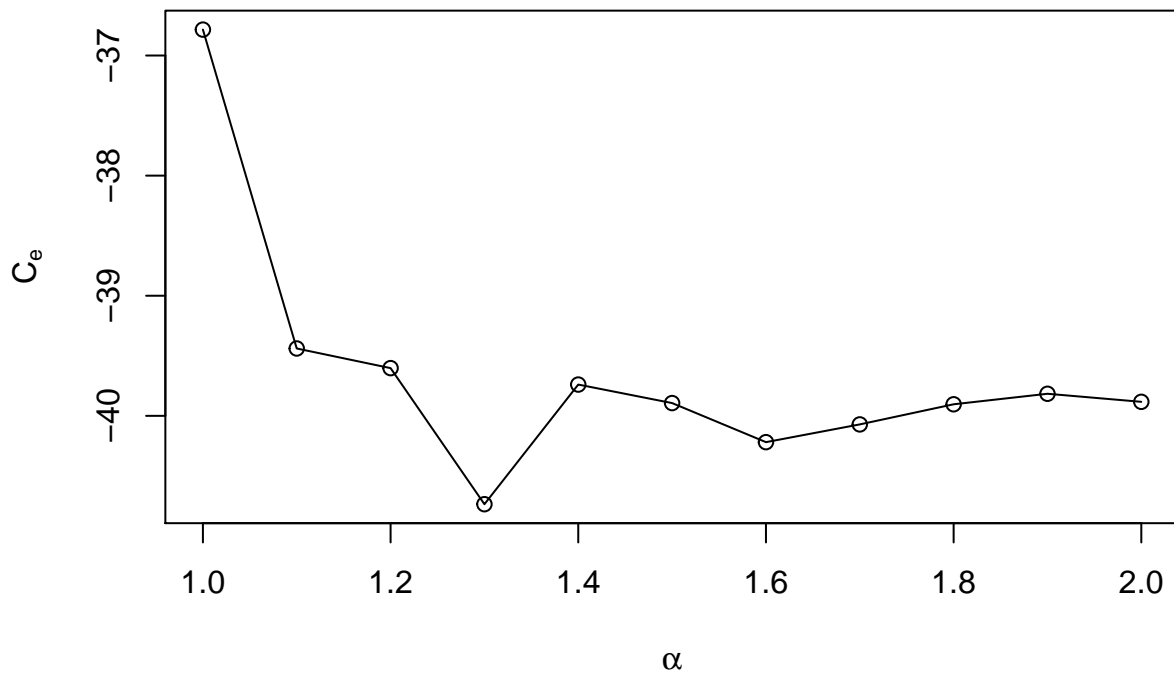
Iris C_e vs. α



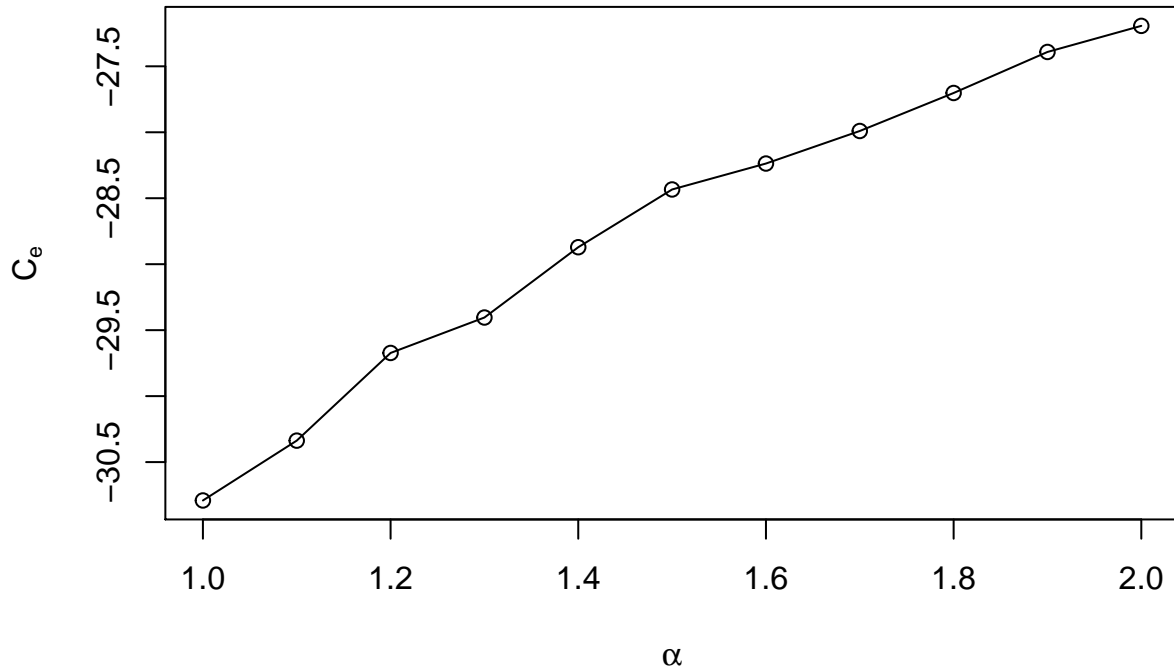
Diabetes C_e vs. α



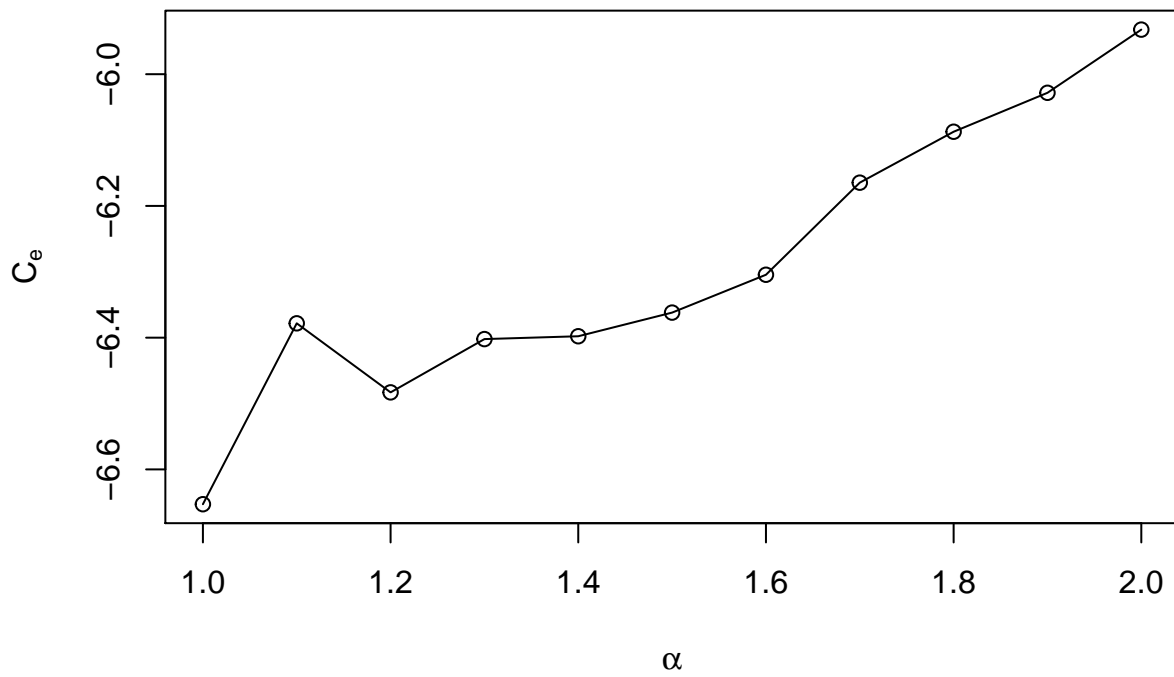
Swiss Banknotes C_e vs. α



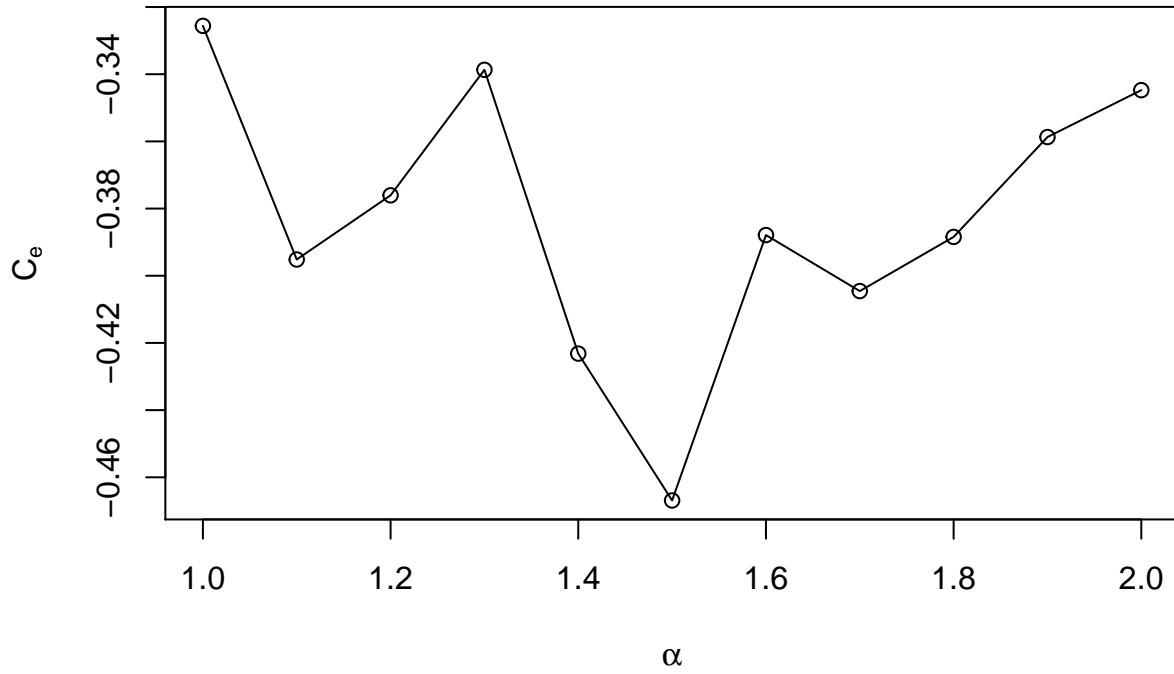
Seeds C_e vs. α



Mice Protein Expression C_e vs. α



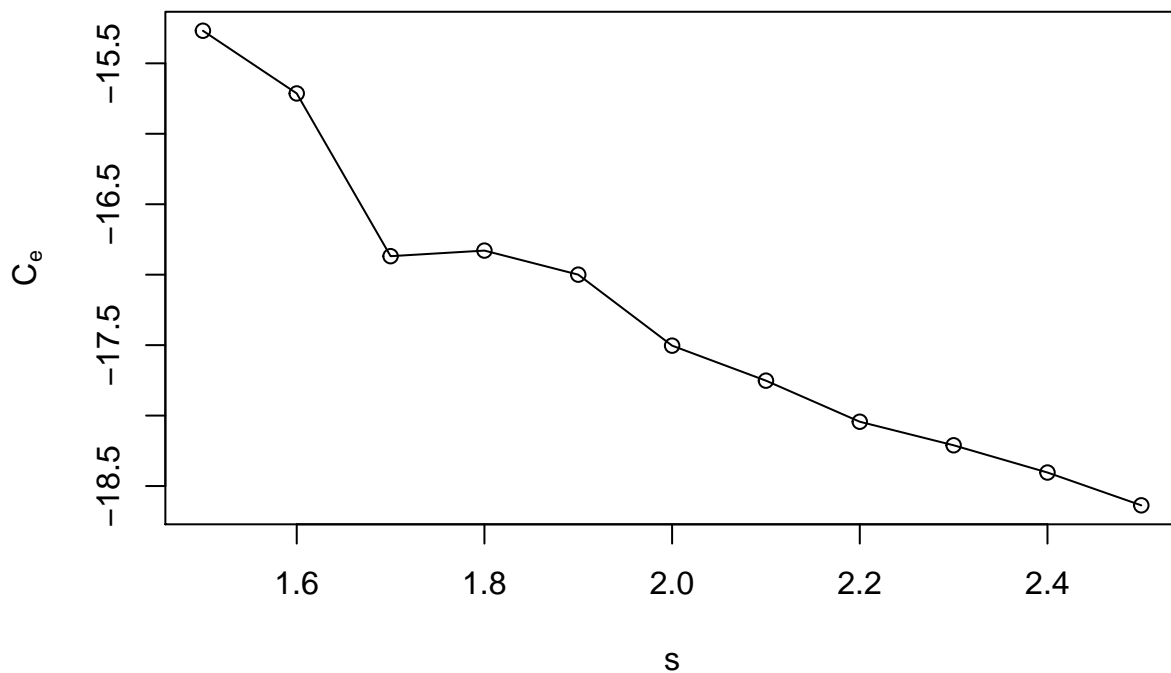
Crabs C_e vs. α



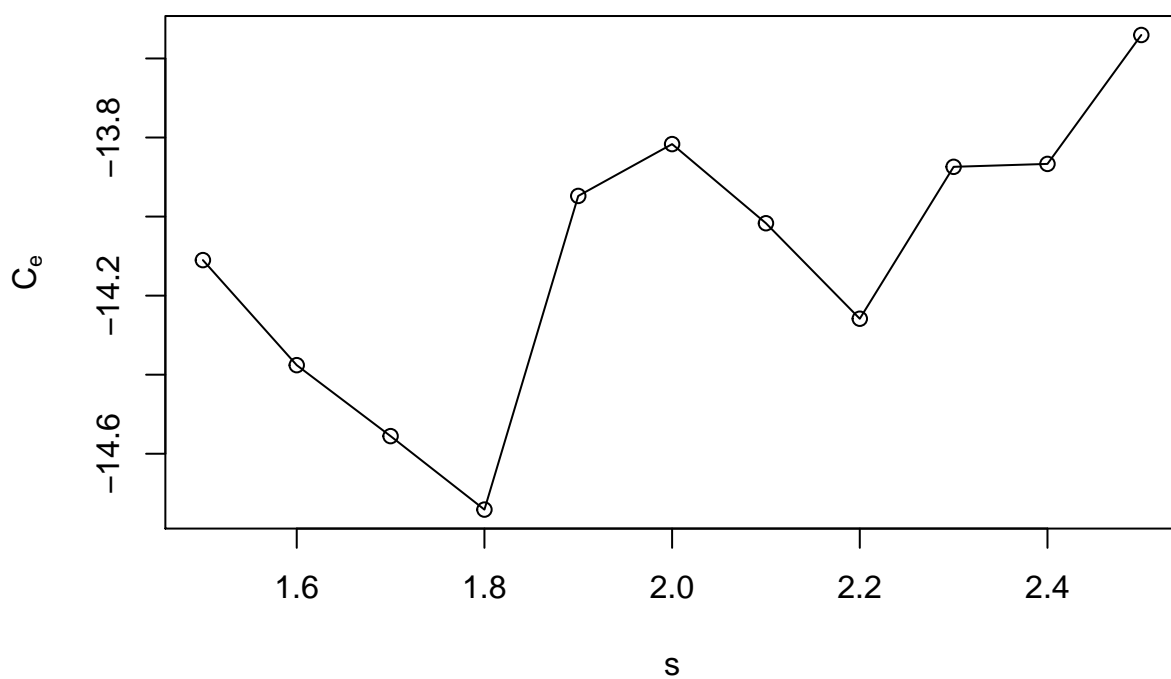
۹-۵ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر s برای

کاهش بعد گسسته به دو بعد

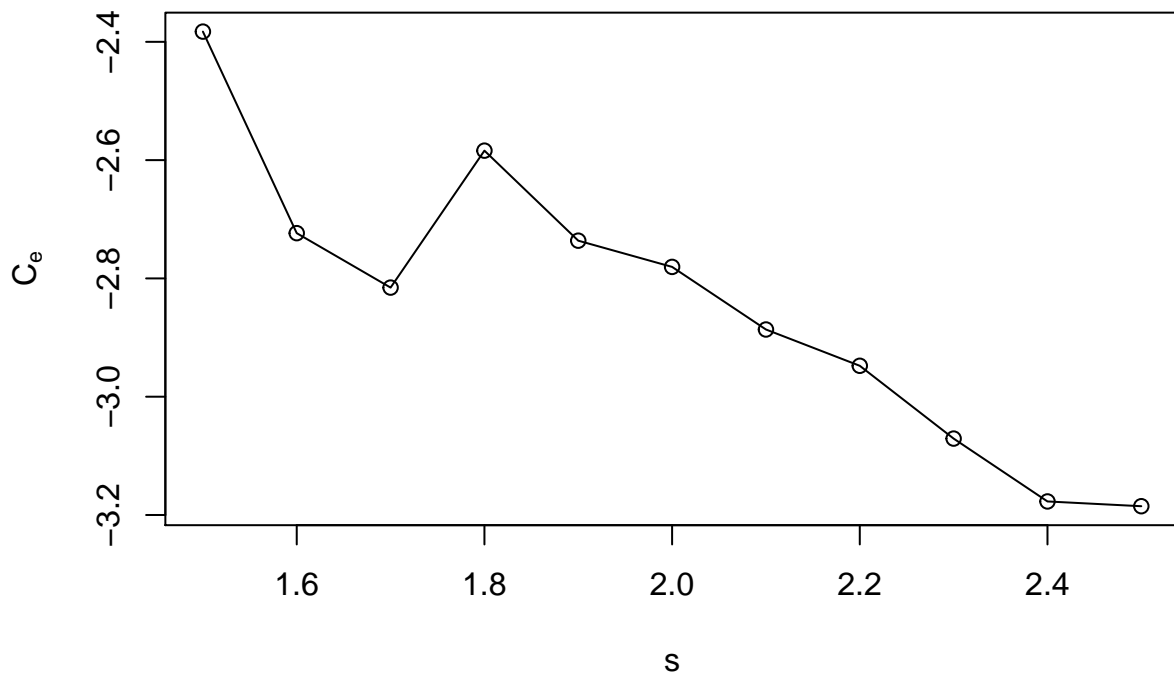
Thyroid C_e vs. s



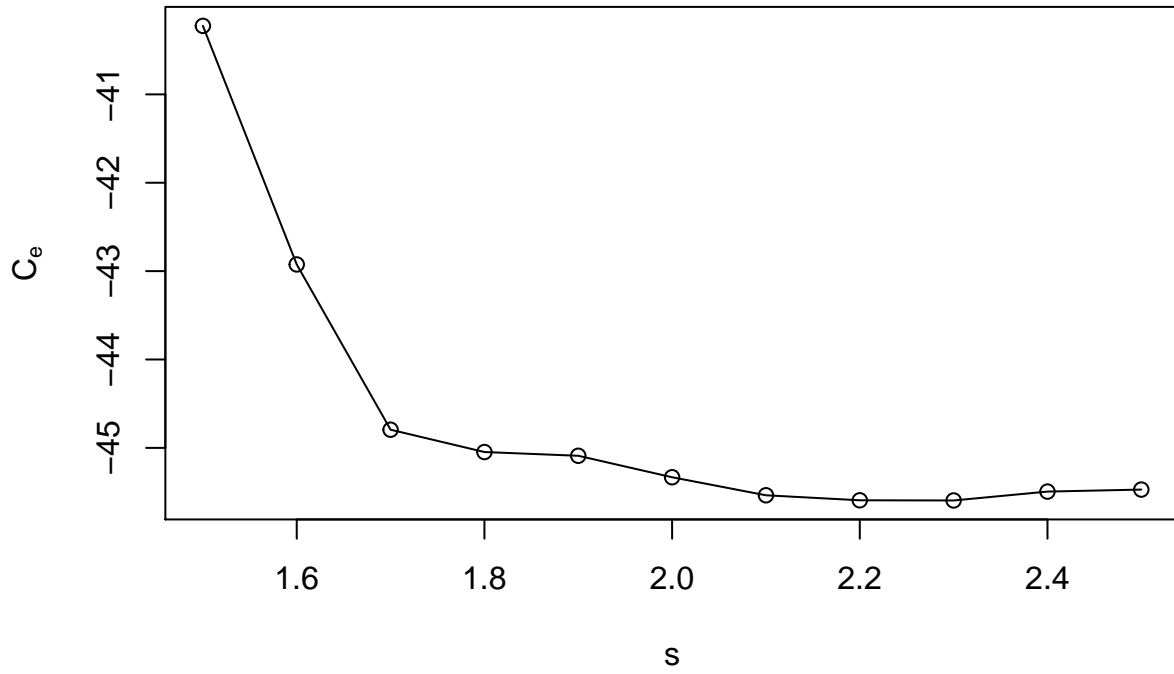
Iris C_e vs. s



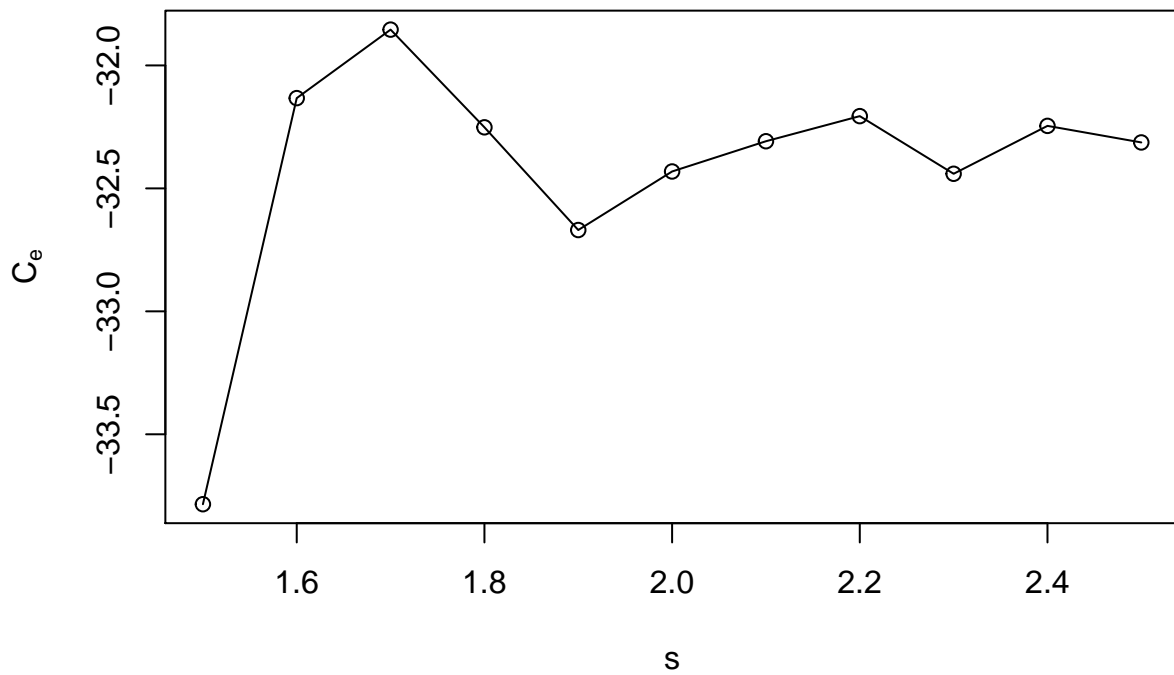
Diabetes C_e vs. s



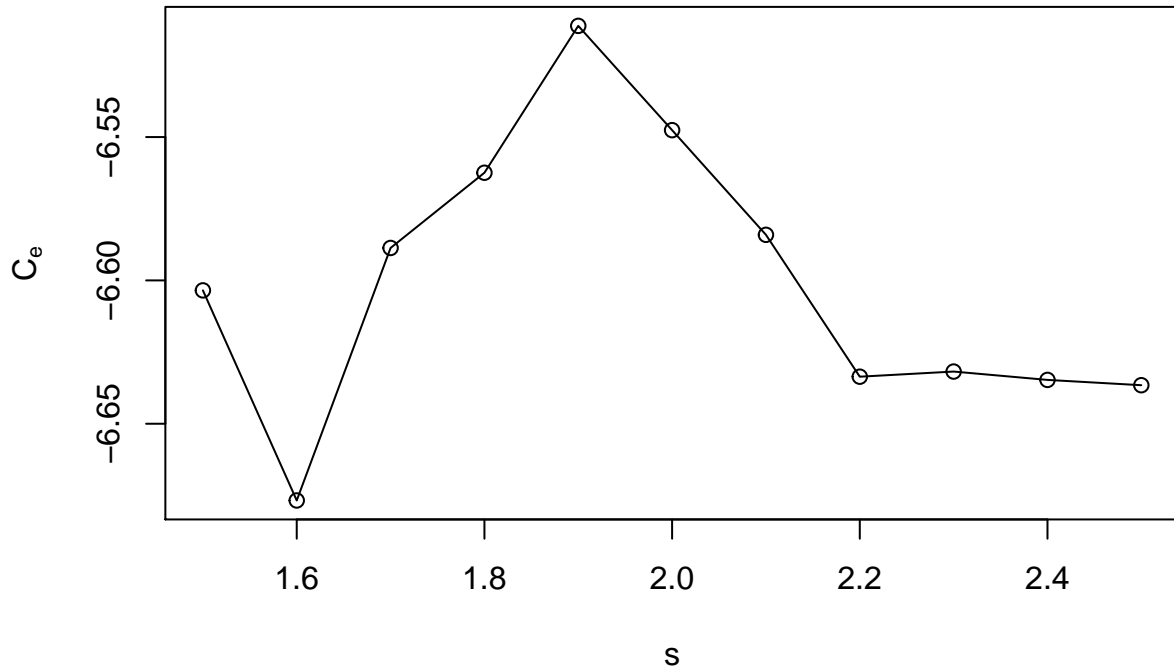
Swiss Banknotes C_e vs. s

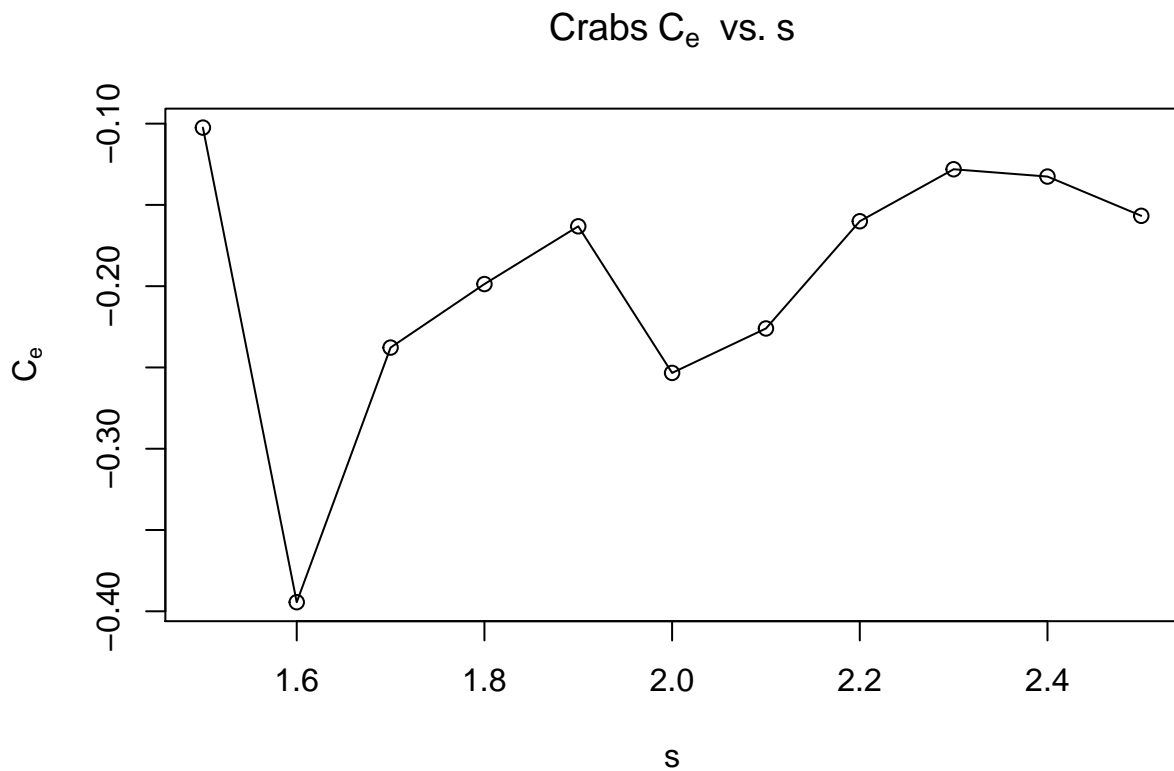


Seeds C_e vs. s

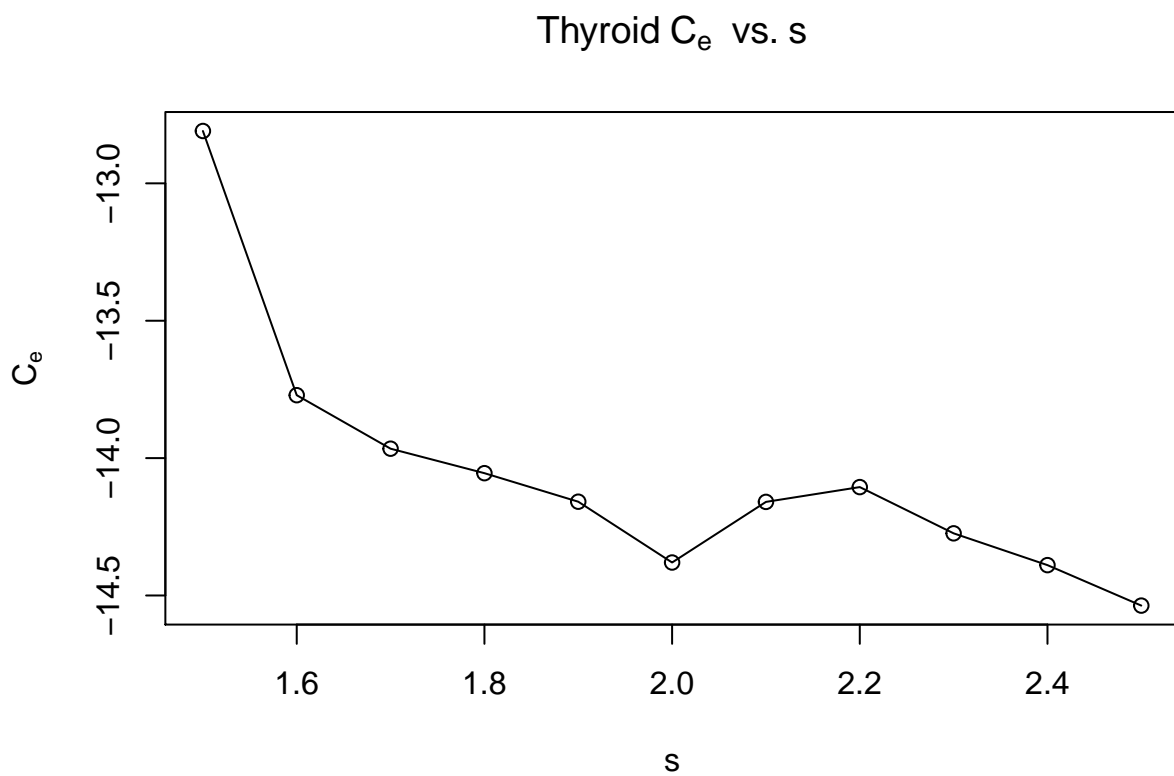


Mice Protein Expression C_e vs. s

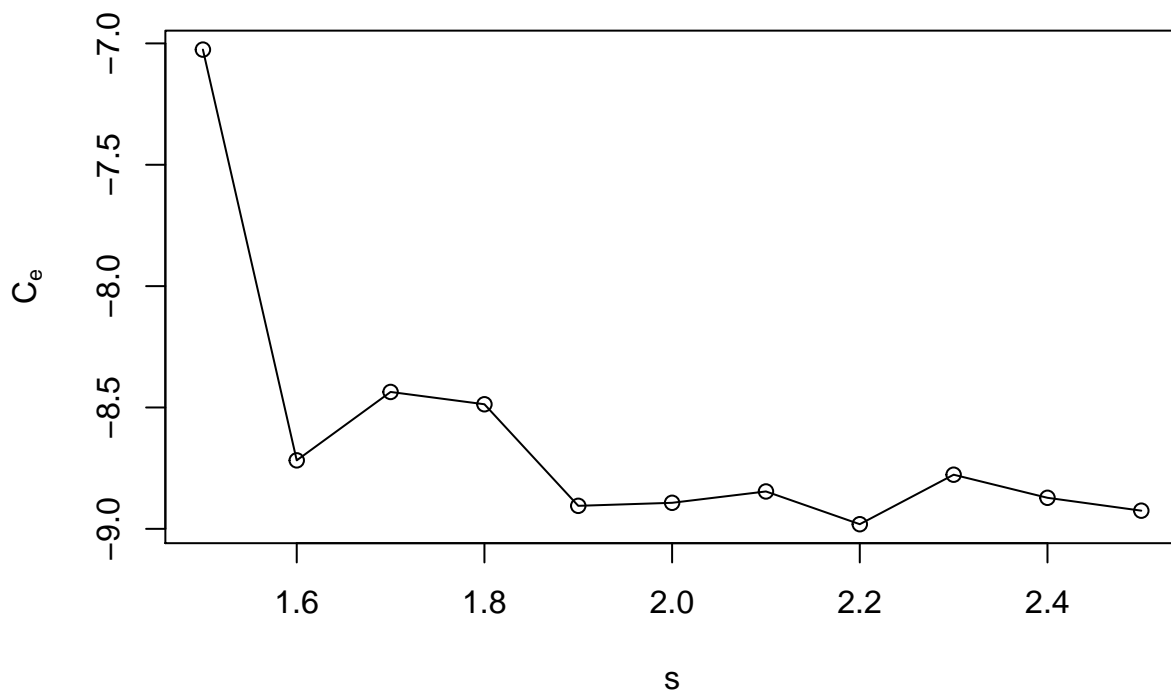




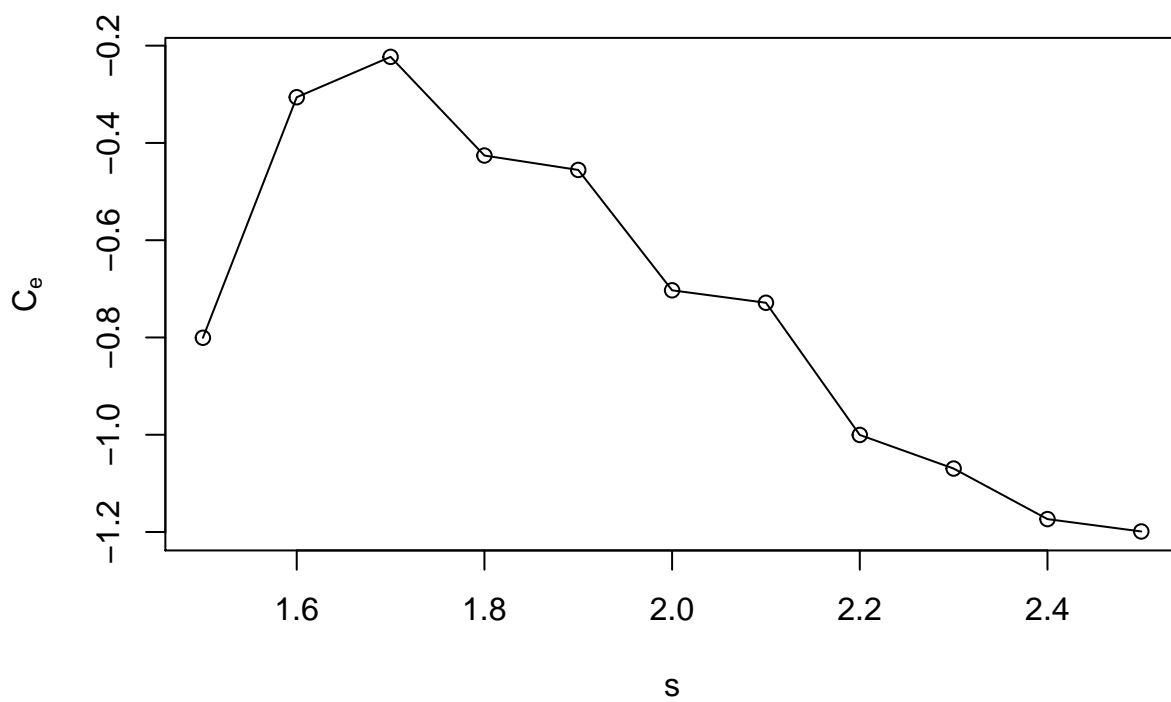
۵-۱۰ بررسی تابعیت عملکرد کاهش بعد C_e به ازای تغییر s
برای کاهش بعد گسسته به سه بعد



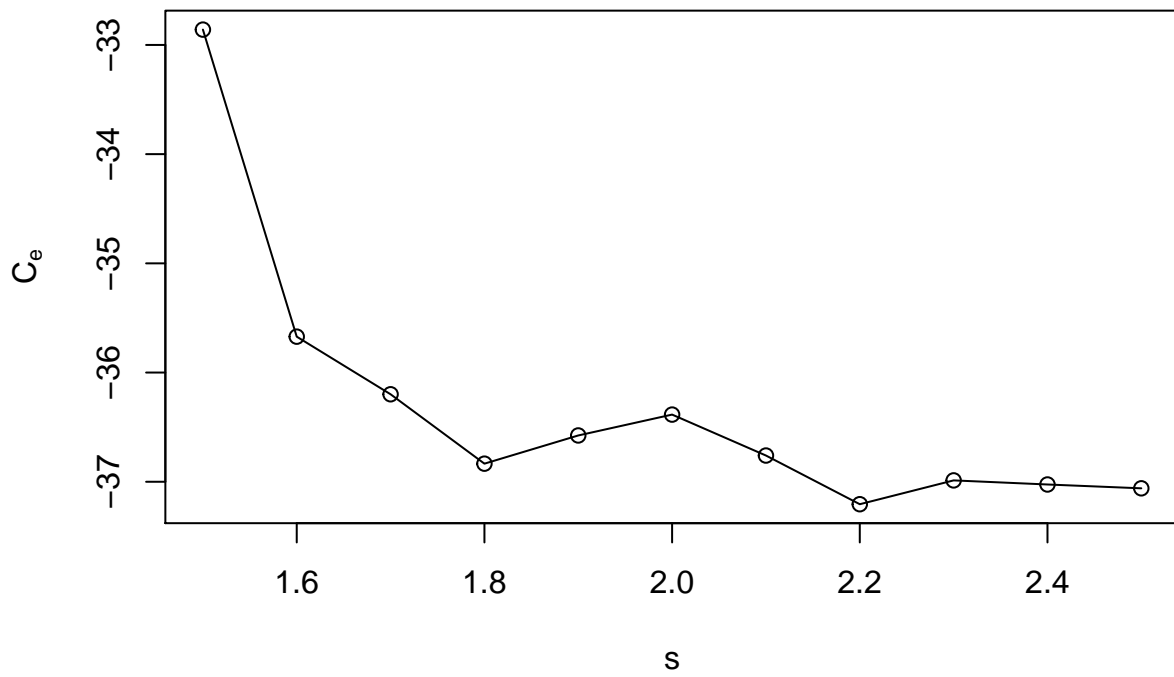
Iris C_e vs. s



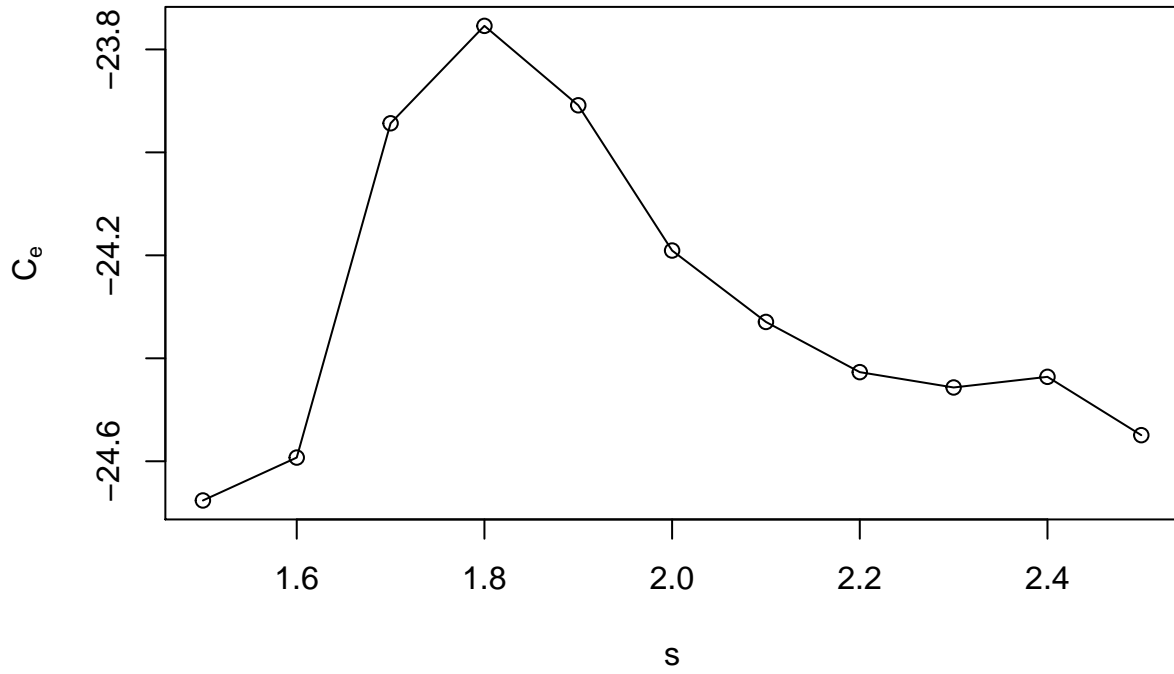
Diabetes C_e vs. s



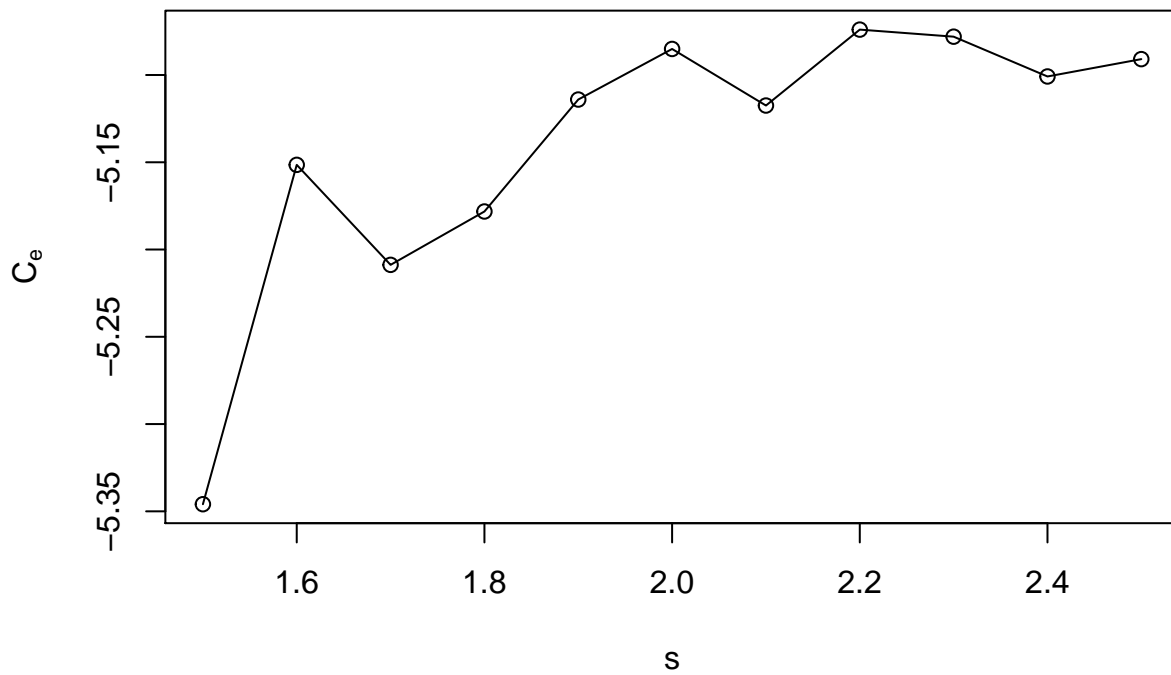
Swiss Banknotes C_e vs. s



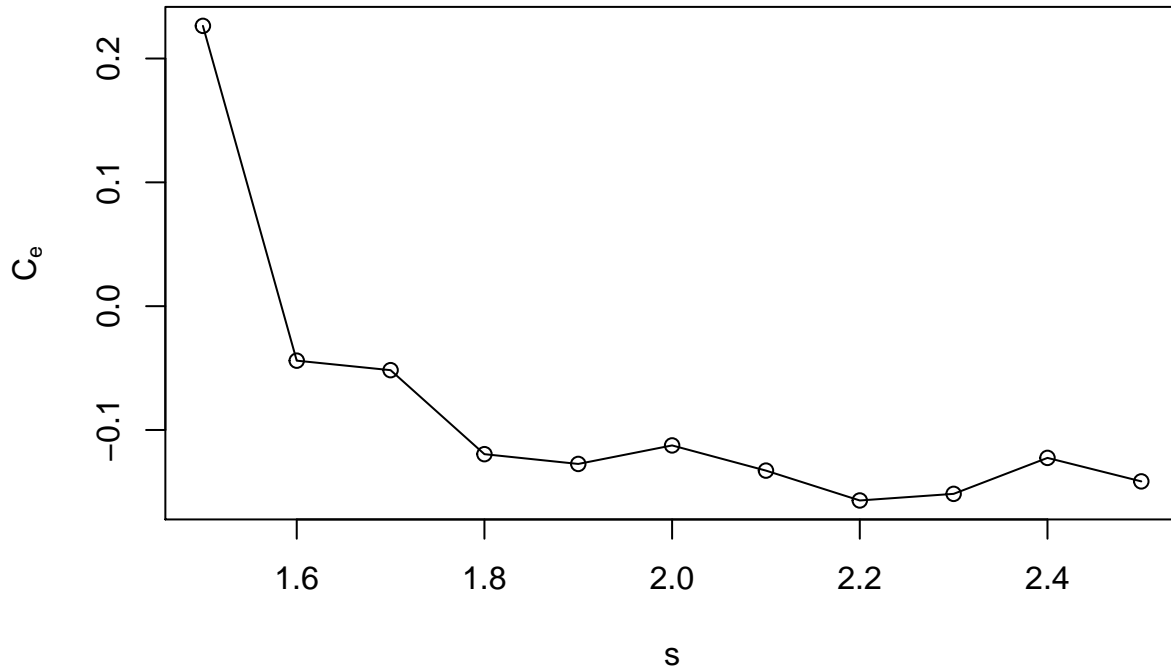
Seeds C_e vs. s



Mice Protein Expression C_e vs. s



Crabs C_e vs. s



منابع و مراجع

- [1] <ftp://statgen.ncsu.edu/pub/thorne/molevoclass/atchleyoct19.pdf>.
- [2] <http://www.informationweek.com/news/showarticle.jhtml?articleid=175801775>.
- [3] UCI Machine Learning Repository: iris Data Set.
- [4] UCI Machine Learning Repository: Mice Protein Expression Data Set.
- [5] UCI Machine Learning Repository: seeds Data Set.
- [6] Achlioptas, Dimitris. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281. ACM, 2001.
- [7] Achlioptas, Dimitris. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.
- [8] Aggarwal, Charu C. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.
- [9] Aggarwal, Charu C, Wolf, Joel L, and Yu, Philip S. *A new method for similarity indexing of market basket data*. ACM, 1999.
- [10] 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.
- [11] 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.

- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] Brinkman, Bo and Charikar, Moses. On the impossibility of dimension reduction in ℓ_1 . *Journal of the ACM (JACM)*, 52(5):766–788, 2005.
- [20] Brinkman, Bo and Charikar, Moses. On the impossibility of dimension reduction in ℓ_1 . *Journal of the ACM (JACM)*, 52(5):766–788, 2005.
- [21] Broder, Andrei Z. On the resemblance and containment of documents. In *Compression and complexity of sequences 1997. proceedings*, pages 21–29. IEEE, 1997.

- [22] Buhler, Jeremy and Tompa, Martin. Finding motifs using random projections. *Journal of computational biology*, 9(2):225–242, 2002.
- [23] Buldygin, Valeri Vladimirovich and Kozachenko, IU V. *Metric characterization of random variables and random processes*, volume 188. American Mathematical Soc., 2000.
- [24] Chaudhuri, Surajit, Motwani, Rajeev, and Narasayya, Vivek. On random sampling over joins. In *ACM SIGMOD Record*, volume 28, pages 263–274. ACM, 1999.
- [25] Chernoff, Herman et al. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.
- [26] Church, Kenneth Ward and Hanks, Patrick. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] Dhillon, Inderjit S and Modha, Dharmendra S. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1-2):143–175, 2001.

- [32] 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.
- [33] Fama, Eugene F and Roll, Richard. Some properties of symmetric stable distributions. *Journal of the American Statistical Association*, 63(323):817–836, 1968.
- [34] Fama, Eugene F and Roll, Richard. Parameter estimates for symmetric stable distributions. *Journal of the American Statistical Association*, 66(334):331–338, 1971.
- [35] Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. *The elements of statistical learning*, volume 10. Springer series in statistics New York, NY, USA:, 2001.
- [36] Friedman, Jerome H, Baskett, Forest, and Shustek, Leonard J. An algorithm for finding nearest neighbors. *IEEE Transactions on computers*, 100(10):1000–1006, 1975.
- [37] 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.
- [38] Garcia-Molina, Hector. Database systems: the complete book/hector garcia, molina jeffrey d. ullman, jennifer widom, 2002.
- [39] Gracia, Antonio, González, Santiago, Robles, Victor, and Menasalvas, Ernestina. A methodology to compare dimensionality reduction algorithms in terms of loss of quality. *Information Sciences*, 270:1–27, 2014.
- [40] Henzinger, Monika Rauch, Raghavan, Prabhakar, and Rajagopalan, Sridhar. Computing on data streams. *External memory algorithms*, 50:107–118, 1998.
- [41] Hornby, Albert Sydney, editor. *Oxford Advanced Learner's Dictionary of Current English*. Oxford University Press, Oxford, UK, fourth edition, 1989.
- [42] Hubert, Lawrence and Arabie, Phipps. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
- [43] Hubert, Lawrence and Arabie, Phipps. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

- [44] Indyk, Piotr. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *focs*, page 189. IEEE, 2000.
- [45] Indyk, Piotr. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM)*, 53(3):307–323, 2006.
- [46] 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.
- [47] Johnson, William B and Lindenstrauss, Joram. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
- [48] Johnson, William B and Schechtman, Gideon. Embedding l_p into l_1 . *Acta Mathematica*, 149(1):71–85, 1982.
- [49] Kannan, J Feigenbaum S, Strauss, M, and Viswanathan, M. An approximate l_1 -difference algorithm for massive data streams. *Unknown*, Unknown.
- [50] 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.
- [51] 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.
- [52] Li, Ping. *Stable random projections and conditional random sampling, two sampling techniques for modern massive datasets*. Stanford, 2007.
- [53] Li, Ping. Estimators and tail bounds for dimension reduction in l_α ($0 < \alpha \leq 2$) 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.
- [54] 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.

- [55] McCulloch, J Huston. Simple consistent estimators of stable distribution parameters. *Communications in Statistics-Simulation and Computation*, 15(4):1109–1136, 1986.
- [56] McKee, Sally A. Reflections on the memory wall. In *CF'04: Proceedings of the 1st conference on Computing frontiers*, page 162, 2004.
- [57] Milligan, Glenn W and Cooper, Martha C. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21(4):441–458, 1986.
- [58] Muthukrishnan, S. Data streams: Algorithms and applications (foundations and trends in theoretical computer science). *Hanover, MA: Now Publishers Inc*, 2005.
- [59] Newman, Mark EJ. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.
- [60] Rand, William M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.
- [61] Scrucca, Luca, Fop, Michael, Murphy, Thomas Brendan, and Raftery, Adrian E. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):205–233, 2016.
- [62] 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.
- [63] Vempala, Santosh S. *The random projection method*, volume 65. American Mathematical Soc., 2005.
- [64] Venables, W. N. and Ripley, B. D. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.
- [65] 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.
- [66] Yeung, Ka Yee and Ruzzo, Walter L. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17:763–774, 2001.

- [67] 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

کد دریافت داده‌ها

```
rm(list = ls(all.names = TRUE))
library(beepr)

library(rgl)
library(stabledist)
library(MASS)
library(mclust)
library(xlsx)

if(file.exists('data.RData')) {
  load(file = 'data.RData')
} else {

  crabs <- crabs #MASS
  data("diabetes") #mclust
  data("thyroid") #mclust
  data("banknote") #mclust

  irisUci <- read.csv('https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data', header =
    ↪ FALSE)
  names(irisUci) <- names(iris)
  seeds <- read.table('https://archive.ics.uci.edu/ml/machine-learning-databases/00236/seeds_dataset.txt', header =
    ↪ FALSE)

  tmpFile <- tempfile(fileext = '.xls')
  download.file('https://archive.ics.uci.edu/ml/machine-learning-databases/00342/Data_Cortex_Nuclear.xls', destfile =
    ↪ tmpFile)
  MPE <- read.xlsx(tmpFile, sheetIndex = 1)

  rm(tmpFile)
```

```
save(list = ls(), file = 'data.RData')
```

```
}
```

```
beep(4)
```

کد مرتب سازی داده‌ها

```
library(tictoc)
source('Util.R')
d <- list()
apply(thyroid, 2, function(x) {sum(is.na(x) | is.null(x))})
d$thyroid$data <- thyroid[,2:6]
d$thyroid$class <- as.numeric(thyroid$Diagnosis)
d$thyroid$data <- normalizeData(d$thyroid$data)
d$thyroid$name <- 'Thyroid'

apply(irisUci, 2, function(x) {sum(is.na(x) | is.null(x))})
d$iris$data <- irisUci[,1:4]
d$iris$class <- as.numeric(irisUci$Species)
d$iris$data <- normalizeData(d$iris$data)
d$iris$name <- 'Iris'

apply(diabetes, 2, function(x) {sum(is.na(x) | is.null(x))})
d$diabetes$data <- diabetes[,2:4]
d$diabetes$class <- as.numeric(diabetes$class)
d$diabetes$data <- normalizeData(d$diabetes$data)
d$diabetes$name <- 'Diabetes'

apply(banknote, 2, function(x) {sum(is.na(x) | is.null(x))})
d$banknote$data <- banknote[,2:7]
d$banknote$class <- as.numeric(banknote$Status)
d$banknote$data <- normalizeData(d$banknote$data)
d$banknote$name <- 'Swiss Banknotes'

apply(seeds, 2, function(x) {sum(is.na(x) | is.null(x))})
d$seeds$data <- seeds[,1:7]
d$seeds$class <- as.numeric(seeds[,8])
d$seeds$data <- normalizeData(d$seeds$data)
d$seeds$name <- 'Seeds'
```



```

apply(MPE, 2, function(x) {sum(is.na(x) | is.null(x))})
MPEwoNA <- MPE
MPEwoNA[,2:78] <- apply(MPE[,2:78], 2, function(x) { x[is.na(x)] <-
  ↪ mean(x, na.rm = TRUE); x } )
apply(MPEwoNA, 2, function(x) {sum(is.na(x) | is.null(x))})
d$MPEwoNA$data <- MPEwoNA[,2:78]
d$MPEwoNA$class <- as.numeric(MPEwoNA$class)
d$MPEwoNA$data <- normalizeData(d$MPEwoNA$data)
d$MPEwoNA$name <- 'Mice Protein Expression'

apply(crabs, 2, function(x) {sum(is.na(x) | is.null(x))})
d$crabs$data <- crabs[,4:8]
d$crabs$data <- cbind(d$crabs$data, as.numeric(crabs[,2]))
d$crabs$class <- as.numeric(crabs[,1])
d$crabs$data <- normalizeData(d$crabs$data)
d$crabs$name <- 'Crabs'

```

کد توابع مورد استفاده داده‌ها

```

# Sample Data Generation Functions
↪ -----

# _Univariate Normal
↪ -----

sampleDataUniNormal <- function(cf) {

  out <- dataConfiguration(cf)

  cf <- out$cf
  dClass <- out$dClass
  dClassMean <- out$dClassMean
  dClassVar <- out$dClassVar

  d <- matrix(rnorm(cf$nSample * cf$nDimention), nrow = cf$nSample)

  for(i in 1:cf$nClass) {
    for(j in 1:cf$nDimention) {
      d[(cf$classIndex[i]+1):cf$classIndex[i+1], j] <-
        sqrt(dClassVar[i, j, j]) *
        ↪ d[(cf$classIndex[i]+1):cf$classIndex[i+1], j]
    }
    d[(cf$classIndex[i]+1):cf$classIndex[i+1], ] <-

```

```

        d[(cf$classIndex[i]+1):cf$classIndex[i+1], ] +
        matrix(rep(dClassMean[i,], cf$classIndex[i+1] -
        ↪ cf$classIndex[i]),
                ncol = cf$nDimention,
                byrow = TRUE)
    }

    permuteIndex <- sample.int(cf$nSample, size = cf$nSample)
    d <- d[permuteIndex, ]
    dClass <- dClass[permuteIndex]

    return(list(data = d, class = dClass))
}

if(FALSE) {
    d <- sampleDataUniNormal(cf)
    print(head(data.frame(d = d$data, class = d$class)))
    plot(d$data, type = 'p', col = d$class, asp = 1)
    plot3d(d$data, col = d$class, size = 2, type = 's')
    aspect3d(x = 'iso')
}

# _Multivariate Normal
↪ -----

sampleDataMultiNormal <- function(cf) {
    out <- dataConfiguration(cf, multiVariate = TRUE)

    cf <- out$cf
    dClass <- out$dClass
    dClassMean <- out$dClassMean
    dClassVar <- out$dClassVar

    d <- matrix(0, nrow = 0, ncol = cf$nDimention)

    for(i in 1:cf$nClass) {
        d <- rbind(d, mvrnorm(cf$classIndex[i+1] - cf$classIndex[i],
                               mu = out$dClassMean[i,], Sigma =
                               ↪ out$dClassVar[i, , ])
        )
    }

    permuteIndex <- sample.int(cf$nSample, size = cf$nSample)
    d <- d[permuteIndex, ]
    dClass <- dClass[permuteIndex]

    return(list(data = d, class = dClass))
}

```

```

}

if(FALSE) {
  d <- sampleDataMultiNormal(cf)
  print(head(data.frame(d = d$data, class = d$class)))
  plot(d$data, type = 'p', col = d$class, asp = 1)
  plot3d(d$data, col = d$class, size = 2, type = 's')
  aspect3d(x = 'iso')
}

# _Alpha Univariate
↪ -----

sampleDataUniStable <- function(cf, alpha) {

  out <- dataConfiguration(cf)

  cf <- out$cf
  dClass <- out$dClass
  dClassMean <- out$dClassMean
  dClassVar <- out$dClassVar

  d <- matrix(rstable(cf$nSample * cf$nDimention, alpha = alpha, beta
↪ = 0),
             nrow = cf$nSample)

  for(i in 1:cf$nClass) {
    for(j in 1:cf$nDimention) {
      d[(cf$classIndex[i]+1):cf$classIndex[i+1], j] <-
        sqrt(dClassVar[i, j, j]) *
↪ d[(cf$classIndex[i]+1):cf$classIndex[i+1], j]
    }
    d[(cf$classIndex[i]+1):cf$classIndex[i+1], ] <-
      d[(cf$classIndex[i]+1):cf$classIndex[i+1], ] +
      matrix(rep(dClassMean[i,], cf$classIndex[i+1] -
↪ cf$classIndex[i]),
             ncol = cf$nDimention,
             byrow = TRUE)
  }

  permuteIndex <- sample.int(cf$nSample, size = cf$nSample)
  d <- d[permuteIndex, ]
  dClass <- dClass[permuteIndex]

  return(list(data = d, class = dClass))
}

```

```

if(FALSE) {
  d <- sampleDataUniStable(cf, 1.8)
  print(head(data.frame(d = d$data, class = d$class)))
  plot(d$data, type = 'p', col = d$class, asp = 1)
  plot3d(d$data, col = d$class, size = 2, type = 's')
  aspect3d(x = 'iso')
}

# _Alpha Multivariate
↪ -----

sampleDataMultiStable <- function(cf, alpha) {

  out <- dataConfiguration(cf)

  cf <- out$cf
  dClass <- out$dClass
  dClassMean <- out$dClassMean
  dClassVar <- out$dClassVar

  cf$ndimension <- 2

  theta <- c(0, pi/4, pi/2, 3 * pi / 4)
  weight <- c(1,2,1,1)

  d <-
    matrix(
      rstable(cf$nsample * length(theta), alpha = alpha, beta = 0),
      ncol = length(theta)
    ) %*%
    matrix(
      c(
        (cos(theta) * weight) / sum(abs(cos(theta) * weight)),
        (sin(theta) * weight) / sum(abs(sin(theta) * weight))
      ),
      ncol = cf$ndimension
    )

  for(i in 1:cf$class) {
    d[(cf$classIndex[i]+1):cf$classIndex[i+1], ] <-
      d[(cf$classIndex[i]+1):cf$classIndex[i+1], ] +
      matrix(rep(dClassMean[i,], cf$classIndex[i+1] -
        ↪ cf$classIndex[i]),
        ncol = cf$ndimension,
        byrow = TRUE)
  }
}

```

```

permuteIndex <- sample.int(cf$nSample, size = cf$nSample)
d <- d[permuteIndex, ]
dClass <- dClass[permuteIndex]

return(list(data = d, class = dClass))

}

if(FALSE) {
  d <- sampleDataMultiStable(cf, 1.8)
  print(head(data.frame(d = d$data, class = d$class)))
  plot(d$data, type = 'p', col = d$class, asp = 1)
  plot3d(d$data, col = d$class, size = 2, type = 's')
  aspect3d(x = 'iso')
}

# Dimention Reduction
↪ -----

randomProjection <- function(d, targetNumDimention, alpha = 2) {

  pm <- matrix(rstable(ncol(d) * targetNumDimention, alpha = alpha,
    ↪ beta = 0),
              nrow = ncol(d))

  d %*% pm

}

sRandomProjection <- function(d, targetNumDimention, s = 2) {

  r <- runif(ncol(d) * targetNumDimention, min = 0, max = 2*s)
  r[r <= 1] = -1
  r[r >= (2*s-1)] = 1
  r[r != 1 & r != -1] <- 0
  pm <- matrix(r,
              nrow = ncol(d))

  d %*% pm

}

```

```

# Util Functions
↪ -----

dataConfiguration <- function(cf, multiVariate = FALSE) {

  if(!('classIndex' %in% names(cf))) {
    if(!('classRatio' %in% names(cf))) {
      cf$classRatio <- rep(1/cf$nClass, cf$nClass)
    }
    cf$classIndex <- c(0, round( cumsum(cf$classRatio) * cf$nSample))
  }

  # dClass <- rep(0, cf$nSample)
  dClass <- integer(0)
  for(i in 1:cf$nClass) {
    dClass <- c(dClass, rep(i, cf$classIndex[i+1]- cf$classIndex[i]))
  }

  dClassMean <- matrix(0, nrow = cf$nClass, ncol = cf$nDimention
    ↪ ,byrow = TRUE)
  theta <- 2 * pi / cf$nClass
  radious <- 5
  for(i in 1:cf$nClass) {
    dClassMean[i, 1] <- radious * cos(theta * i)
    dClassMean[i, 2] <- radious * sin(theta * i)
  }
  # dClassMean[,1] <- 10 * seq(0, cf$nClass-1)

  dClassVar <- array(0, dim = c(cf$nClass, cf$nDimention,
    ↪ cf$nDimention))
  if(multiVariate) {
    for(i in 1:cf$nClass) {
      dClassVar[i, , ] <- randCovMat(cf$nDimention)
    }
  } else {
    for(i in 1:cf$nClass) {
      dClassVar[i, , ] <- diag(cf$nDimention)
    }
  }

  return(list(
    cf = cf,
    dClass = dClass,
    dClassMean = dClassMean,
    dClassVar = dClassVar
  ))
}

```

```

randCovMat <- function(nd, sdev = NULL) {

  if (is.null(sdev)) {
    sdev <- rep(1, nd)
  }
  out <- diag(nd)
  index <- data.frame(n = 1:((nd*(nd-1)) / 2))
  index$i <- ceiling((-1+sqrt(1+8*index$n)) / 2)
  index$j <- index$n - ((index$i - 1)^2 + (index$i - 1)) / 2
  index$i <- index$i + 1
  index <- index[sample(index$n, nrow(index)), ]

  for(i in 1:nrow(index)) {
    candidate <- seq(from = -1, to = 1, by = 0.01)
    candidate <- sample(candidate, size = length(candidate))
    for(k in 1:length(candidate)) {
      out[index$i[i],index$j[i]] <- candidate[k]
      out[index$j[i],index$i[i]] <- out[index$i[i],index$j[i]]
      if(det(out) > 0) break
    }
  }
  msdev <- diag(nd) * sdev
  out <- msdev %*% out %*% msdev
}

normalizeData <- function(d) {
  d <- apply(d, 2, function(x) {(x-mean(x))/sd(x)})
  d
}

ARlreport <- function(d, tDimention = 2, alpha = 2) {

  nClass <- length(unique(d$class))

  cl <- kmeans(d$data, nClass, nstart = 20, iter.max = 100)
  ari_d <- (adjustedRandIndex(d$class, cl$cluster))

  d$pdata <- randomProjection(d$data, tDimention, alpha = alpha)
  clp <- kmeans(d$pdata, nClass, nstart = 20, iter.max = 100)
  ari_p <- (adjustedRandIndex(d$class, clp$cluster))

  c_e <- 100 * (ari_d - ari_p)

  data.frame(ari_d = ari_d, ari_p = ari_p, c_e = c_e)
}

ARlreport_s <- function(d, tDimention = 2, s = 2) {

```

```

nClass <- length(unique(d$class))

cl <- kmeans(d$data, nClass, nstart = 20, iter.max = 100)
ari_d <- (adjustedRandIndex(d$class, cl$cluster))

check <- TRUE
while(check){
  check <- tryCatch({
    d$pdata <- sRandomProjection(d$data, tDimention, s = s)
    clp <- kmeans(d$pdata, nClass, nstart = 20, iter.max = 100)
    FALSE
  }, error = function(e) {print('not enough distinct points');
    ↪ TRUE})
}

ari_p <- (adjustedRandIndex(d$class, clp$cluster))

c_e <- 100 * (ari_d - ari_p)

data.frame(ari_d = ari_d, ari_p = ari_p, c_e = c_e)
}

```

کد گزارش گیری

```

iterations <- 1000
tic()
for(dIndex in names(d)) {
  out <- data.frame(ari_d = numeric(0), ari_p = numeric(0), c_e =
    ↪ numeric(0))
  for(i in 1:iterations) {
    out <- rbind(out , ARIreport_s(d[[dIndex]], tDimention = 3, s = 2))
  }
  d[[dIndex]]$ARIreport <- out
  print(dIndex)
}
toc()

save(file = 'SimData_S2D3_1jR.RData', list = c('d'))

# vs. reports
↪ -----

```



```

iterations <- 200
range <- seq(from = 1, to = 2, by = 0.1)
tic()
for(dIndex in names(d)) {
  out <- data.frame(ari_d = numeric(0), ari_p = numeric(0), c_e =
    ↪ numeric(0))
  out1 <- data.frame(alpha = numeric(0),
    ari_d = numeric(0), sd_ari_d = numeric(0),
    ari_p = numeric(0), sd_ari_p = numeric(0),
    c_e = numeric(0), sd_c_e = numeric(0))
  for(j in range){
    for(i in 1:iterations) {
      out <- rbind(out , ARIreport(d[[dIndex]], tDimention = 2, alpha
        ↪ = j))
    }
    out1 <- rbind(out1, data.frame(alpha = j,
      ari_d = mean(out[,1]), sd_ari_d =
        ↪ sd(out[,1]),
      ari_p = mean(out[,2]), sd_ari_p =
        ↪ sd(out[,2]),
      c_e = mean(out[,3]), sd_c_e =
        ↪ sd(out[,3])
    )
  )
  d[[dIndex]]$ARivsAlpha <- out1
  print(dIndex)
}
toc()

save(file = 'SimData_ARivsAlphaD2_1jM.RData', list = c('d'))

```

```
beep(4)
```

```

iterations <- 200
range <- seq(from = 1.5, to = 2.5, by = 0.1)
tic()
for(dIndex in names(d)) {
  out <- data.frame(ari_d = numeric(0), ari_p = numeric(0), c_e =
    ↪ numeric(0))
  out1 <- data.frame(alpha = numeric(0), ari_d = numeric(0), ari_p =
    ↪ numeric(0), c_e = numeric(0))
  for(j in range){
    for(i in 1:iterations) {

```

```

    out <- rbind(out , ARIreport_s(d[[dIndex]], tDimention = 2, s =
    ↪ j))
  }
  out1 <- rbind(out1, data.frame(alpha = j,
                                ari_d = mean(out[,1]), sd_ari_d =
                                ↪ sd(out[,1]),
                                ari_p = mean(out[,2]), sd_ari_p =
                                ↪ sd(out[,2]),
                                c_e = mean(out[,3]), sd_c_e =
                                ↪ sd(out[,3])
                                )
  )
}
d[[dIndex]]$ARivsAlpha <- out1
print(dIndex)
}
toc()

```

```

save(file = 'SimData_ARivsSD2_1jN.RData', list = c('d'))

```

```

beep(4)

```

```

iterations <- 200
range <- seq(from = 1, to = 2, by = 0.1)
tic()
for(dIndex in names(d)) {
  out <- data.frame(ari_d = numeric(0), ari_p = numeric(0), c_e =
  ↪ numeric(0))
  out1 <- data.frame(alpha = numeric(0),
                    ari_d = numeric(0), sd_ari_d = numeric(0),
                    ari_p = numeric(0), sd_ari_p = numeric(0),
                    c_e = numeric(0), sd_c_e = numeric(0))
  for(j in range){
    for(i in 1:iterations) {
      out <- rbind(out , ARIreport(d[[dIndex]], tDimention = 3, alpha
      ↪ = j))
    }
    out1 <- rbind(out1, data.frame(alpha = j,
                                  ari_d = mean(out[,1]), sd_ari_d =
                                  ↪ sd(out[,1]),
                                  ari_p = mean(out[,2]), sd_ari_p =
                                  ↪ sd(out[,2]),
                                  c_e = mean(out[,3]), sd_c_e =
                                  ↪ sd(out[,3])
                                  )
  )
}
}

```

```

    d[[dIndex]]$ARivsAlpha <- out1
    print(dIndex)
  }
  toc()

  save(file = 'SimData_ARivsAlphaD3_1jP.RData', list = c('d'))

  beep(4)

  iterations <- 200
  range <- seq(from = 1.5, to = 2.5, by = 0.1)
  tic()
  for(dIndex in names(d)) {
    out <- data.frame(ari_d = numeric(0), ari_p = numeric(0), c_e =
      ↪ numeric(0))
    out1 <- data.frame(alpha = numeric(0), ari_d = numeric(0), ari_p =
      ↪ numeric(0), c_e = numeric(0))
    for(j in range){
      for(i in 1:iterations) {
        out <- rbind(out , ARIreport_s(d[[dIndex]], tDimention = 3, s =
          ↪ j))
      }
      out1 <- rbind(out1, data.frame(alpha = j,
        ari_d = mean(out[,1]), sd_ari_d =
          ↪ sd(out[,1]),
        ari_p = mean(out[,2]), sd_ari_p =
          ↪ sd(out[,2]),
        c_e = mean(out[,3]), sd_c_e =
          ↪ sd(out[,3])
      )
    )
    d[[dIndex]]$ARivsAlpha <- out1
    print(dIndex)
  }
  toc()

  save(file = 'SimData_ARivsSD3_1jQ.RData', list = c('d'))

  beep(4)

```

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 Mathematics and Computer Science

M. Sc. Thesis

Using Random Projection to Dimension Reduction of Large Scale Data

By

Siamak Dehbod

Supervisor

Dr. A. Mohammadpour

Advisor

Dr. H. Zare

January 2019