Machine Learning

کاهش ابعاد

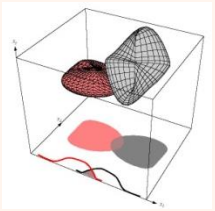
Dimensionality Reduction

یادگیری ماشین

دانشگاه شهید بهشتی
پژوهشکده‌ی فضای مجازی
پاییز ۱۴۰۱
احمد محمودی ازناوه

فهرست مطالب

- مزایای کاهش ابعاد
- انتخاب خصیصه
- استخراج خصیصه
- تحلیل مؤلفه‌ی اصلی
- تحلیل تفکیک خطی
- تجزیه به مقادیر تکین
- روش‌های غیر خطی
- تحلیل عاملی
- تغییر مقیاس داده‌های چند بعدی



نکبت ابعاد!(مشقت ابعاد)

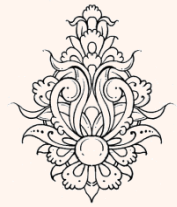
- از لحاظ نظری، افزایش ابعاد منجر به بهبود عملکرد دسته‌بندی می‌شود، اما در عمل همیشه این گونه نیست.

Curse of dimensionality

- انتظار می‌رود در یک فرآیند ایده‌آل دسته‌بندی یا رگرسیون از خصیصه‌های بی‌اهمیت صرفنظر شود و فرآیند «کاهش ابعاد» به صورت جداگانه مورد نیاز نباشد. با این وجود کاهش ابعاد به دلایل زیر مورد توجه قرار می‌گیرد:

مزایای کاهش ابعاد

- «کاهش حجم محاسبات»: حافظه‌ی مصرفی و حجم محاسبات به تعداد (N) و ابعاد (d) داده‌ها بستگی دارد.
  - زمان محاسبات
  - حافظه‌ی مورد نیاز
- «صرفه‌جویی در جمع‌آوری داده»: حذف داده‌های غیرضروری
- «مقاوم‌بودن»(robustness): مدل‌های ساده، هنگامی که داده‌های آموزشی کم باشد، «مقاوم‌تر» می‌باشند؛ قدرت پیش‌بینی برای تعداد مشخصی داده، با افزایش ابعاد، کاهش می‌یابد.

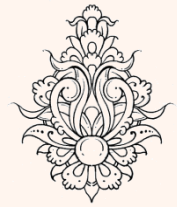


مزایای کاهش ابعاد(ادامه...)

- «استخراج دانش»: با تعداد خصیصه‌های کم‌تر، در مورد داده‌ها و فرآیندهای مربوط به آن درک بهتری وجود خواهد داشت. گاهی این خصیصه‌ها را می‌توان به صورت «عوامل پنهان» در نظر گرفت که متغیرهای قابل مشاهده از آن‌ها نشأت می‌گیرند.

Hidden or latent factor

- هنگامی که تعداد خصیصه‌ها(بدون از دست دادن اطلاعات) کم‌تر باشد، «ساختار داده‌ها» بهتر درک می‌شود. داده‌های پرت و غیرمعمول بهتر تشخیص داده می‌شود؛ قابلیت نمایش بهتری دارند.



(انتخاب –استخراج) خصیصه

Feature Selection vs Extraction

- انتخاب خصیصه:
  - K خصیصه‌ی مهم‌تر (k<d) انتخاب می‌شود.
  - الگوریتم‌های انتخاب زیرمجموعه
- استخراج خصیصه:
  - K خصیصه‌ی جدید، استخراج می‌شود.
  - نگاشت از فضای n-بعدی به فضای k-بعدی
  - روش‌های استخراج خصیصه نیز از دیدگاه‌های مختلفی قابل طبقه‌بندی هستند، روش‌های خطی در برابر روش‌های غیرخطی و یا روش‌های بی‌نظارت در برابر روش‌های بانظارت

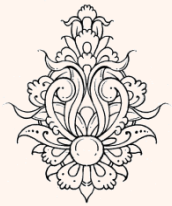
انتخاب زیرمجموعه

- در انتخاب زیرمجموعه، هدف انتخاب **بهترین** زیرمجموعه، زیرمجموعه‌ای با کم‌ترین ابعاد و درست‌ترین نتیجه، می‌باشد.
  - 2^d زیرمجموعه، در یک مجوعه‌ی d-عضوی وجود دارد، بررسی تمام حالات به جز زمانی که d کوچک باشد، امکان‌پذیر نیست.

Forward search

- جستجوی رو به جلو:
  - در گام نخست، مجموعه‌ی خصیصه‌ها، F در حالت اولیه $\emptyset$ در نظر گرفته می‌شود.
  - در هر گام بهترین خصیصه به مجموعه‌ی خصیصه‌ها افزوده می‌شود. (میزان خطای (E(F)) کم‌تر)
  - برای بررسی خطا باید از داده‌های validation استفاده کرد.

$$j = \arg\min_i E(F \cup x_i)$$
$$\text{Add } x_j \text{ to } F \text{ if } E(F \cup x_j) < E(F)$$



انتخاب زیرمجموعه (ادامه...)

- جستجوی رو به عقب: **Backward search**
  - در گام نخست، مجموعه‌ی خصیصه‌ها، F در حالت اولیه تمامی خصیصه‌ها در نظر گرفته می‌شود.
  - در هر گام بدترین خصیصه از مجموعه‌ی خصیصه‌ها حذف می‌شود.
- هنگامی که تعداد خصیصه‌ها زیاد است، روش جستجوی رو به جلو ترجیح داده می‌شود.
- انتخاب زیرمجموعه به صورت بانظارت است.
- در کاربردهایی که یک خصیصه به تنهایی اطلاعات مفیدی ندارد، انتخاب خصیصه مفید نیست.(مانند تشخیص چهره)

Iris data: Single feature

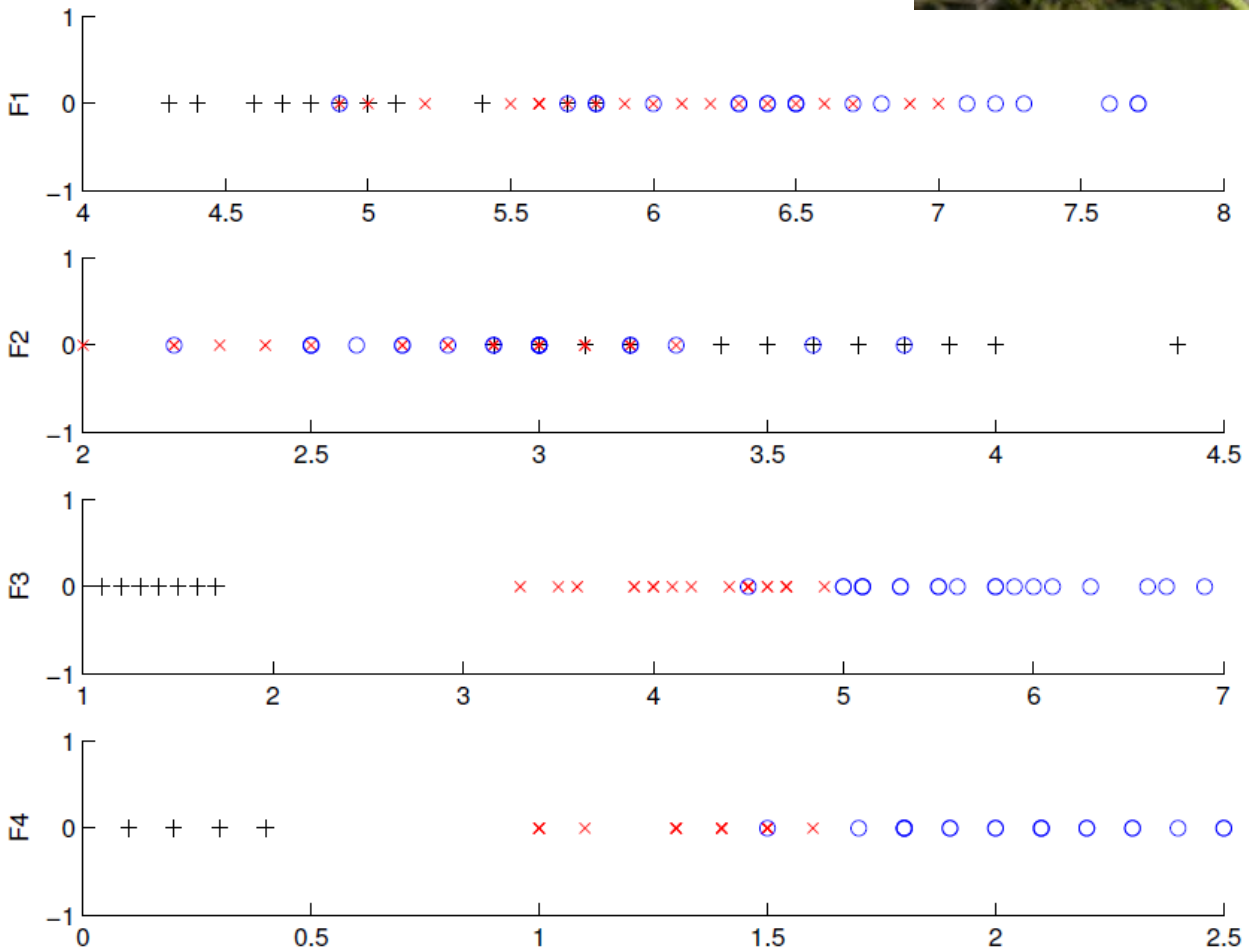
مثال

0.76

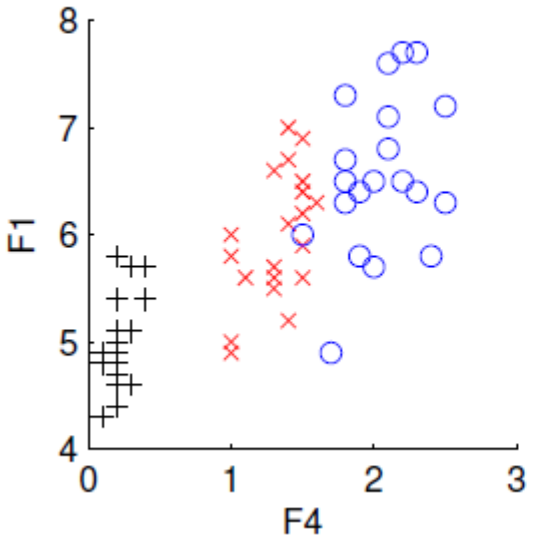
0.57

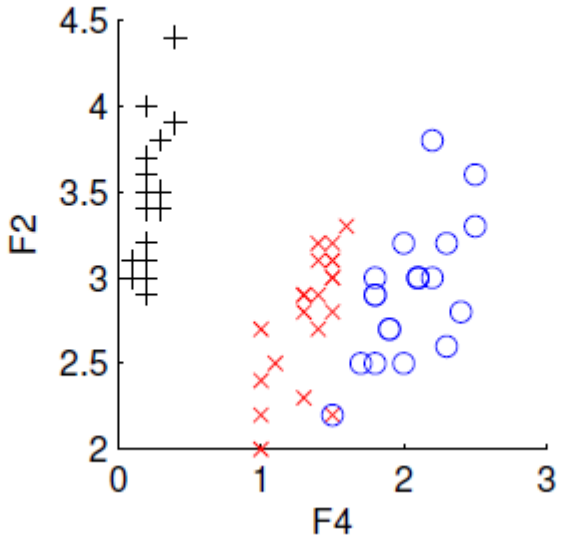
0.92

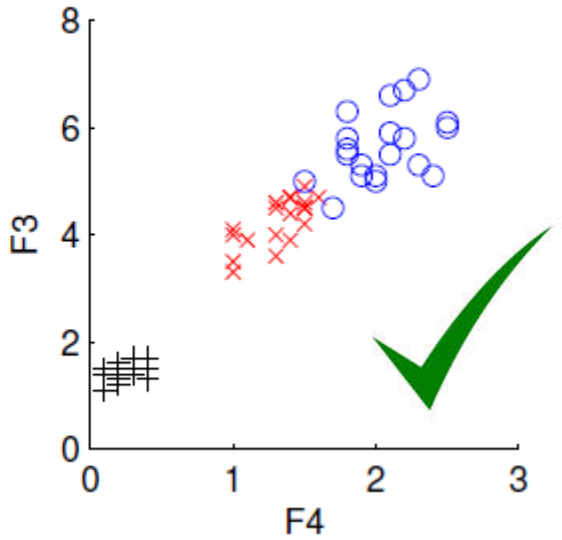
0.94



Iris data: Add one more feature to F4







0.87 ***0.94*** ***0.96***

در صورت اضافه کردن خصیصه‌ی بعدی نتایج افت می‌کند!

در بسیاری موارد انتخاب خصیصه‌ها به نوع دسته‌بند بستگی دارد.

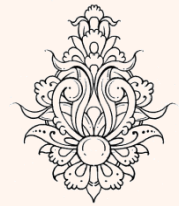
در صورت کوچک بودن دسته‌داده، خصیصه‌ی انتخاب شده، می‌تواند به نحوه‌ی تقسیم پایگاه به دو دسته‌ی training و validation مربوط باشد.

تحلیل مؤلفه‌های اصلی

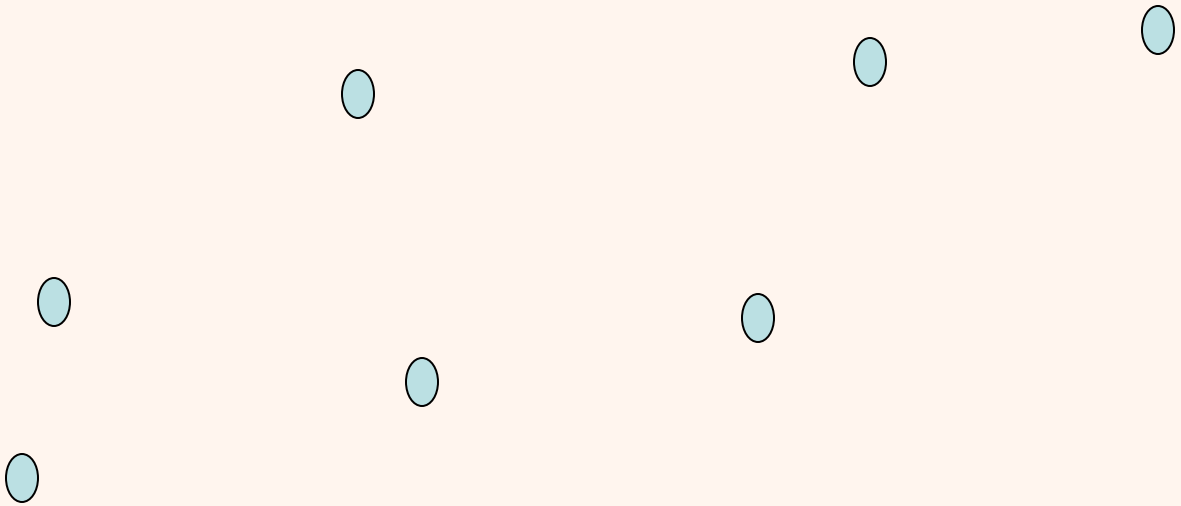
- هدف نگاشت داده‌ی d-بعدی به فضای k-بعدی است (k<d)، به گونه‌ای که کم‌ترین میزان اتلاف رخ دهد.
  - نگاشت x در راستای w:

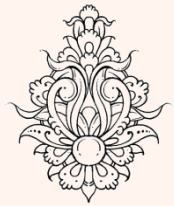
$$z = w^T x$$

- این راستا به گونه‌ای انتخاب می‌شود که Var(z) بیشینه شود، راستایی که داده در امتداد آن بیشترین تغییرات را داشته باشد.
  - این مسأله باعث می‌شود، تفاوت نمونه‌ها آشکارتر شود.
- این شیوه‌ی کاهش بعد به صورت «بی‌نظارت» است.



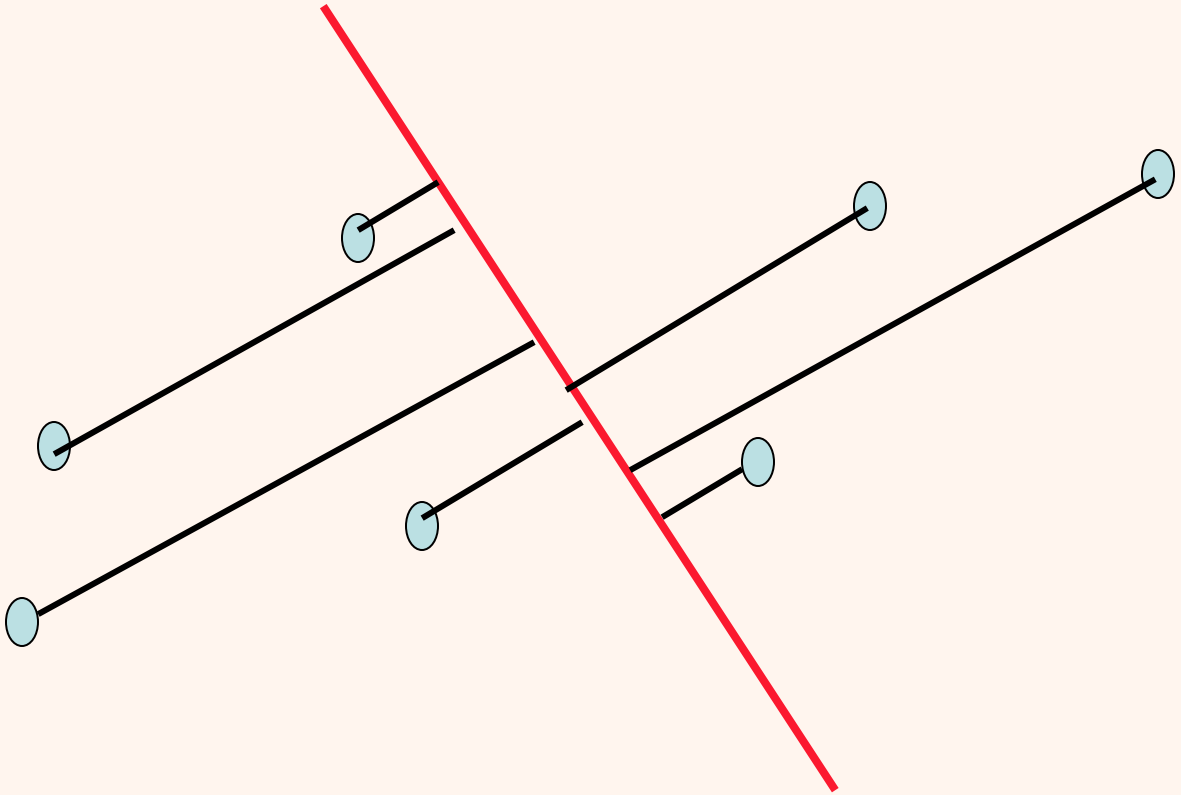
تحلیل مؤلفه‌های اصلی(ادامه...)





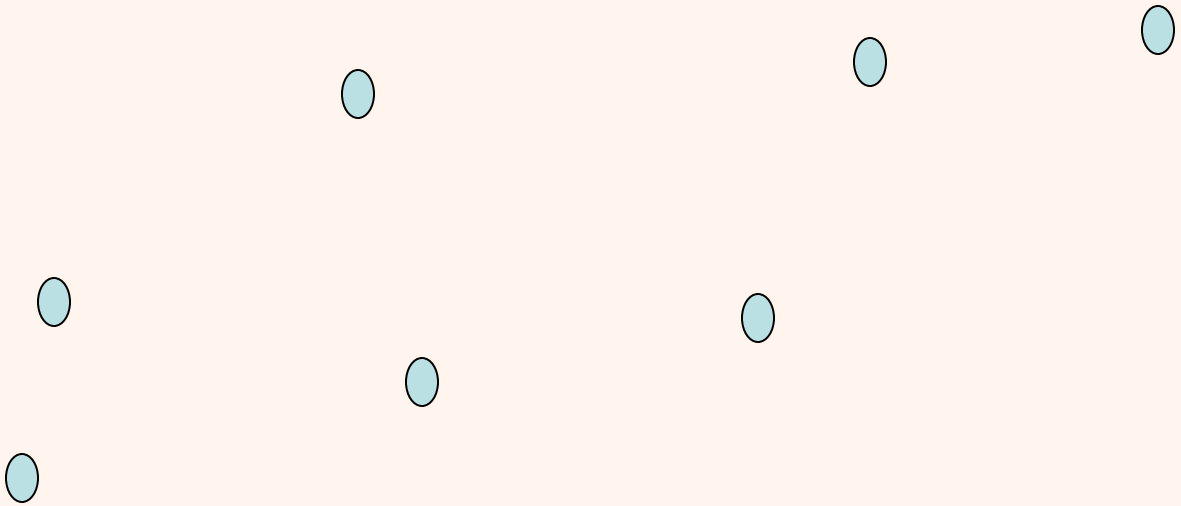
http://www.public.asu.edu/~jye02/

تحلیل مؤلفه‌های اصلی(ادامه...)



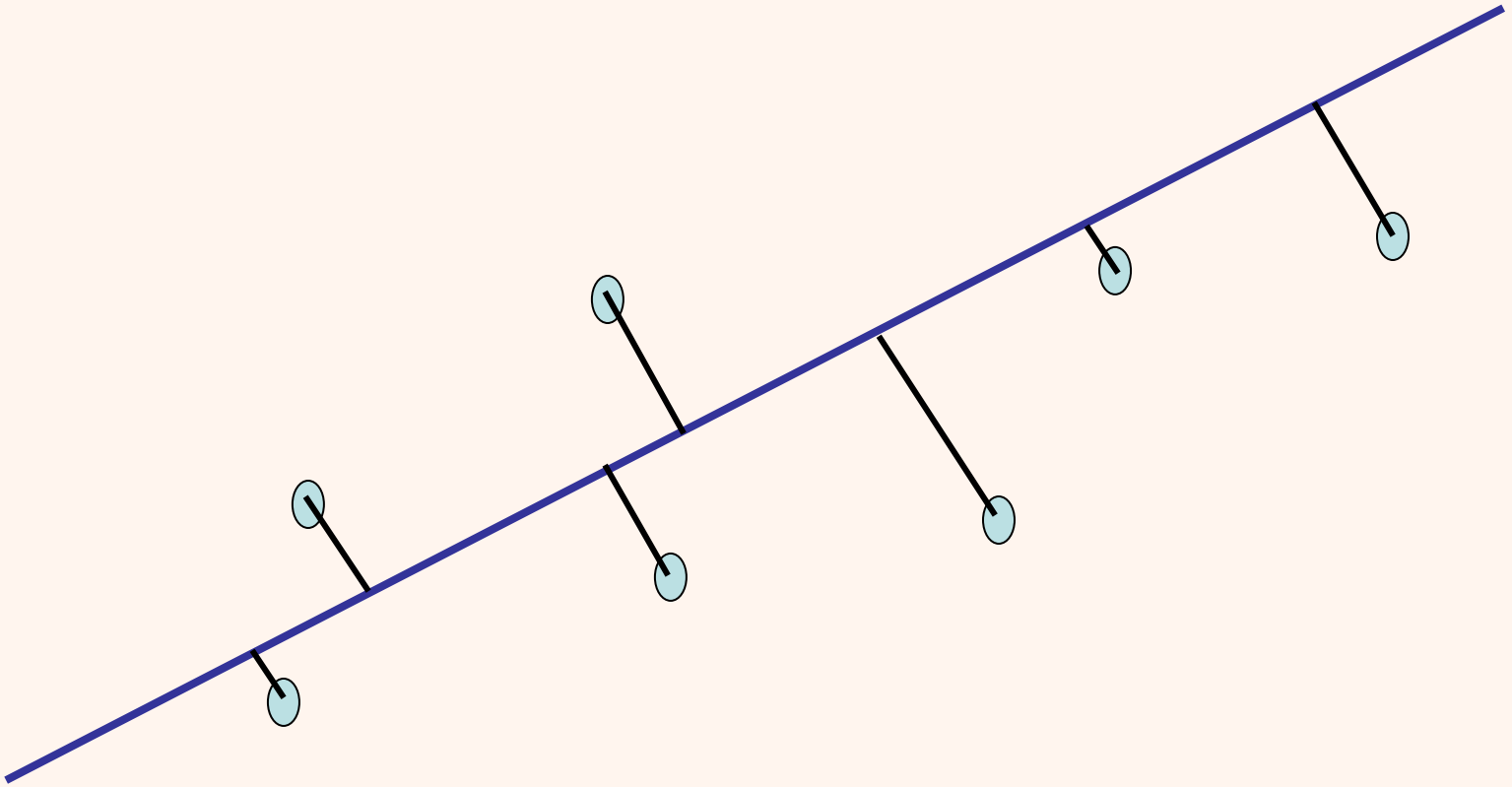
http://www.public.asu.edu/~jye02/

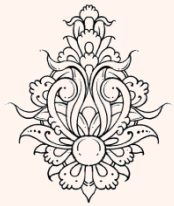
تحلیل مؤلفه‌های اصلی(ادامه...)



http://www.public.asu.edu/~jye02/

تحلیل مؤلفه‌های اصلی(ادامه...)





http://www.public.asu.edu/~jye02/

تحلیل مؤلفه‌های اصلی(ادامه...)

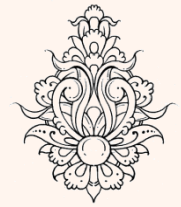
- در راستای w، پراکندگی داده ماکزیمم می‌شود:

$$\begin{aligned}\mathrm{Var}(z) = \mathrm{Var}(w^T x) &= \mathrm{E}[(w^T x - w^T \mu)^2] \\ &= \mathrm{E}[(w^T x - w^T \mu)(w^T x - w^T \mu)] \\ &= \mathrm{E}[w^T (x - \mu)(x - \mu)^T w] \\ &= w^T \mathrm{E}[(x - \mu)(x - \mu)^T] w \\ &= w^T \Sigma w\end{aligned}$$

where $\mathrm{Cov}(x) = \Sigma$

- در این حالت تنها راستا است که اهمیت دارد، در نتیجه برای یافتن پاسخ یکتا، باید شرط زیر نیز برقرار باشد:

$$\|w\| = 1$$



تحلیل مؤلفه‌های اصلی(ادامه...)

- در نتیجه برای اولین مؤلفه‌ی اساسی رابطه‌ی زیر به دست می‌آید: $\max_{\mathbf{w}_1} \mathbf{w}_1^T \Sigma \mathbf{w}_1 - \alpha\left(\mathbf{w}_1^T \mathbf{w}_1 - 1\right)$
- با مشتق گرفتن نسبت به $\boldsymbol{w_1}$ و برابر صفر قرار دادن آن

$$2\Sigma \mathbf{w}_1 - 2\alpha \mathbf{w}_1 = 0$$

- در نتیجه

$$\Sigma \mathbf{w}_1 = \alpha \mathbf{w}_1$$

در نتیجه W_1 یکی از بردارهای ویژه‌ی ماتریس Σ می‌باشد

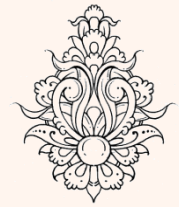
- از طرفی $\mathbf{w}_1^T \Sigma \mathbf{w}_1 = \alpha$ ، در واقع واریانس در راستای $\mathbf{w}_1$ برابر مقدار ویژه‌ی متناظر با آن است.

اولین مؤلفه‌ی اصلی، برابر بردار ویژه‌ی ماتریس کواریانس با بیشترین مقدار ویژه است.

تحلیل مؤلفه‌های اصلی(ادامه...)

- برای یافتن دومین مؤلفه‌ی اصلی، علاوه بر شرایط پیش باید بر راستای اولین مؤلفه‌ی اساسی هم عمود باشد، در این حالت داده‌های نگاشت شده «ناهمبسته» (uncorrelated) خواهند بود.
- برای یافتن دومین مؤلفه‌ی اصلی(w_2)، باید $Var(z_2)$ ماکزیمم شود، مشروط به متعامد بودن بر اولین مؤلفه‌ی اصلی و $\|w_2\|=1$

$$\max_{\mathbf{w}_2} \mathbf{w}_2^T \Sigma \mathbf{w}_2 - \alpha\left(\mathbf{w}_2^T \mathbf{w}_2 - 1\right) - \beta\left(\mathbf{w}_2^T \mathbf{w}_1 - 0\right)$$



تحلیل مؤلفه‌های اصلی(ادامه...)

$$\max_{\mathbf{w}_2} \mathbf{w}_2^T \Sigma \mathbf{w}_2 - \alpha\left(\mathbf{w}_2^T \mathbf{w}_2 - 1\right) - \beta\left(\mathbf{w}_2^T \mathbf{w}_1 - 0\right)$$

- پس از مشتق گرفتن خواهیم داشت:

$$2\Sigma \mathbf{w}_2 - 2\alpha \mathbf{w}_2 - \beta \mathbf{w}_1 = 0$$

- با ضرب در $\mathbf{w}_1^T$

$$\mathbf{w}_1^T \Sigma \mathbf{w}_2 - 2\alpha \mathbf{w}_1^T \mathbf{w}_2 - \beta \mathbf{w}_1^T \mathbf{w}_1 = 0$$

$$\mathbf{w}_1^T \mathbf{w}_2 = 0$$

$$\mathbf{w}_1^T \Sigma \mathbf{w}_2 - \beta \mathbf{w}_1^T \mathbf{w}_1 = 0$$

$$\mathbf{w}_1^T \Sigma \mathbf{w}_2 = \mathbf{w}_2^T \Sigma \mathbf{w}_1 = \lambda_1 \mathbf{w}_2^T \mathbf{w}_1 = 0$$

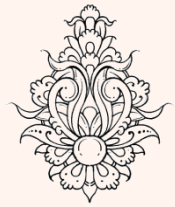
$$\beta = 0$$

$$\Sigma \mathbf{w}_2 = \alpha \mathbf{w}_2$$

دومین مؤلفه‌ی اصلی، برابر بردار ویژه‌ی ماتریس کواریانس با بیشترین مقدار ویژه در رده‌ی دوم است، به همین ترتیب سایر مقادیر ویژه به دست می‌آیند.

تحلیل مؤلفه‌های اصلی(ادامه...)

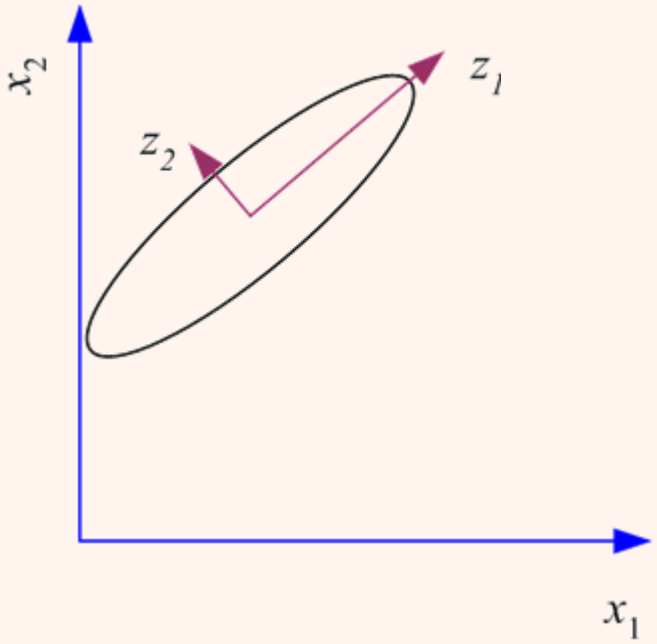
- در صورتی که ماتریس متقارن باشد، بردارهای ویژه‌ی آن متعامد هستند.
- در صورتی که ماتریس positive definite باشند، مقادیر ویژه همگی مثبت خواهند بود.
- در صورتی که ماتریس singular باشد، به اندازه‌ی rank ماتریس مقادیر ویژه‌ی غیرصفر خواهیم داشت.



تحلیل مؤلفه‌های اصلی(ادامه...)

$$z = W^T(x - m)$$

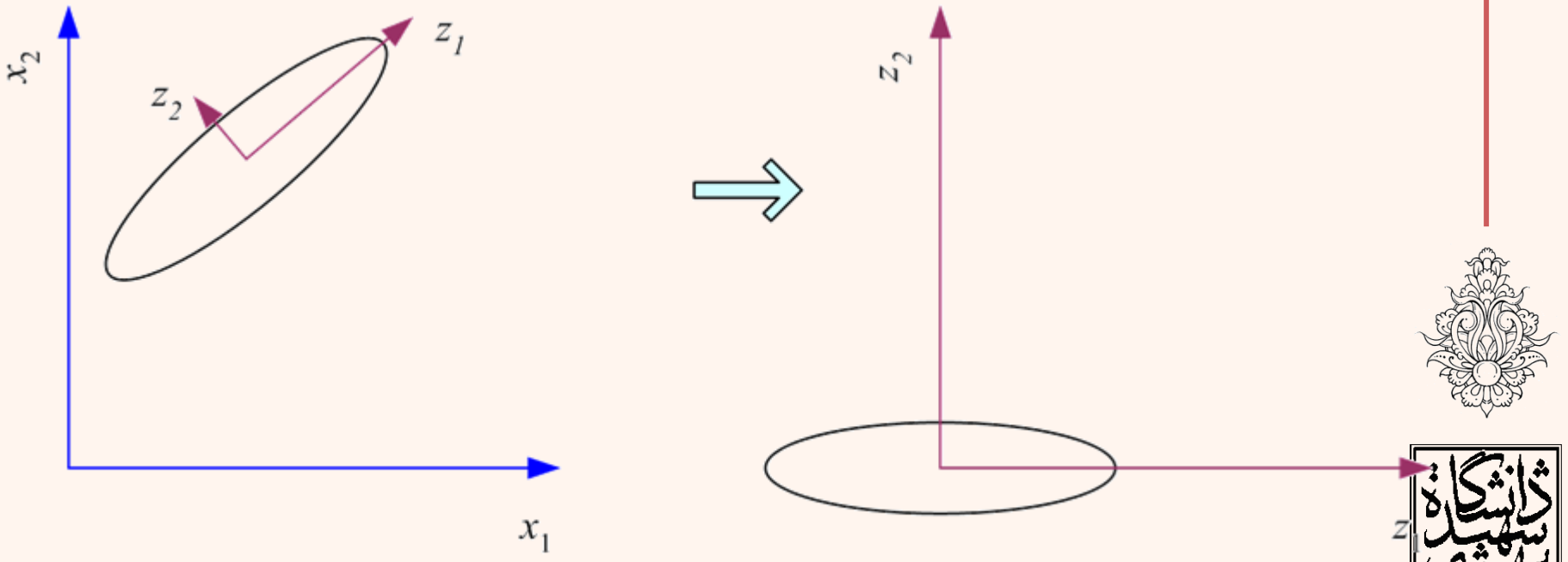
- ستون‌های W، بردارهای ویژه‌ی ماتریس کواریانس هستند.



تحلیل مؤلفه‌های اصلی(ادامه...)

$$z = W^T(x - m)$$

- ستون‌های W، بردارهای ویژه‌ی ماتریس کواریانس هستند.



کاهش بعد

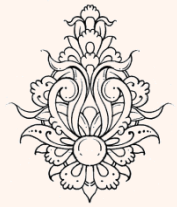
- در صورتی که |S| کوچک باشد، می‌توان نتیجه گرفت برخی مقادیر ویژه، کوچک هستند. در نتیجه داده‌ها در راستای بردار ویژه‌ی متناظر با آن واریانس کمی دارد و قابل صرفنظر کردن است.
  - در این حالت K مؤلفه‌ی پرارزش انتخاب می‌شوند، با فرض آن که مقادیر ویژه به صورت صعودی مرتب شده باشند.

Proportion of Variance (PoV)

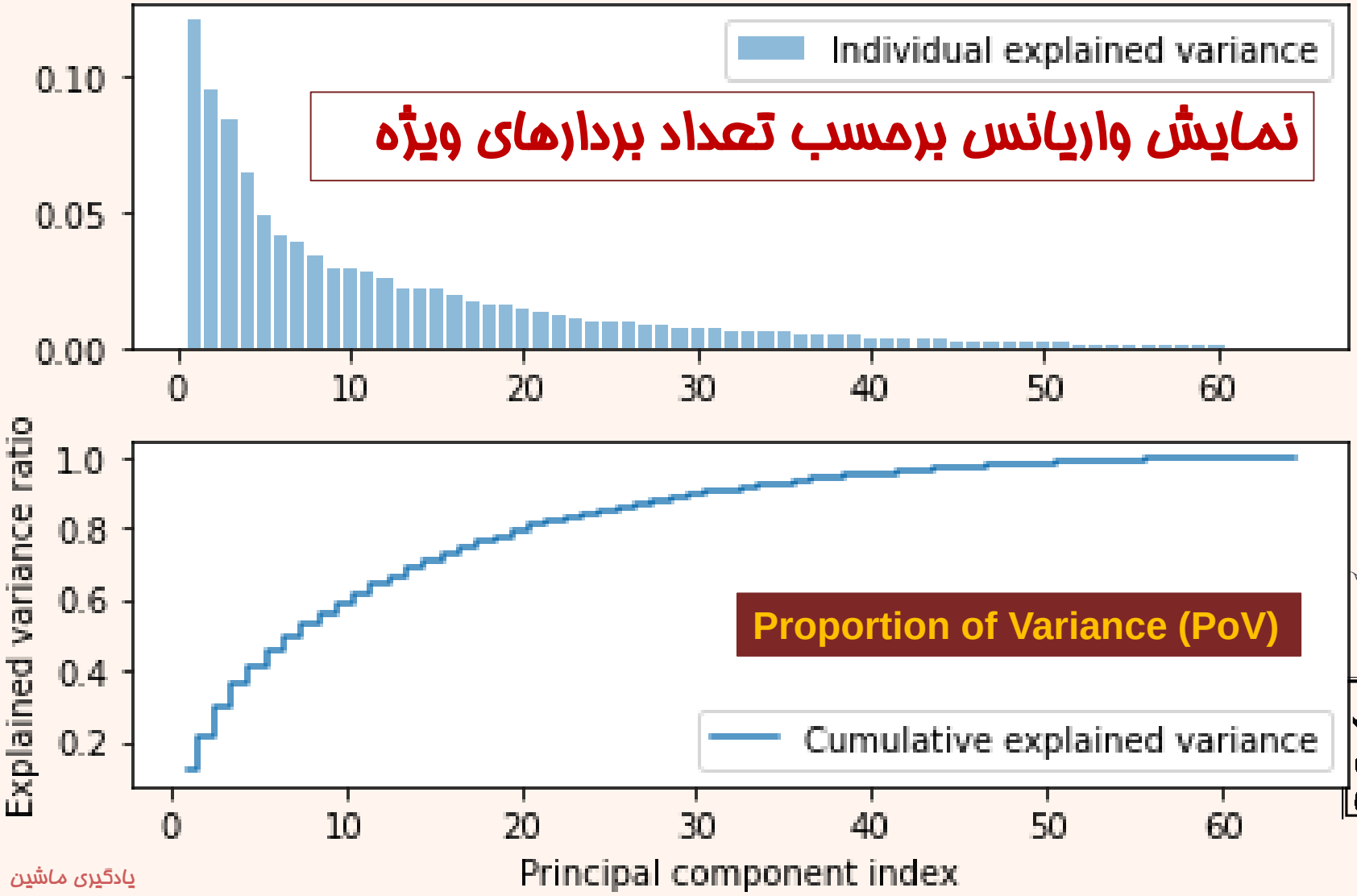
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

PoV>0.9

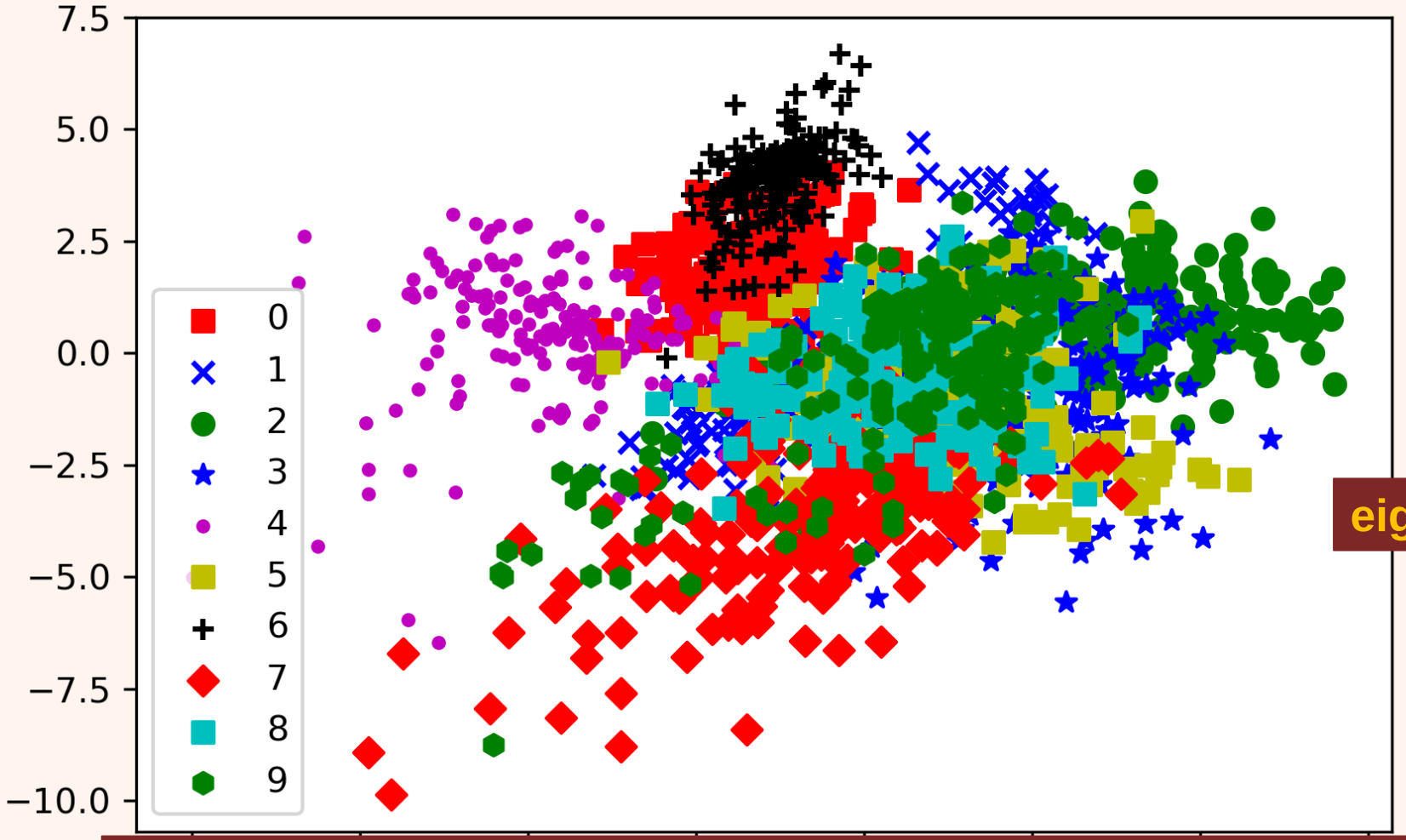
  - در کاربردهای نظیر پردازش تصویر یا صوت، معمولاً کاهش ابعاد قابل توجه است.



Scree graph


Individual explained variance
نمایش واریانس برحسب تعداد بردارهای ویژه
0.10
0.05
0.00
0
10
20
30
40
50
60
Explained variance ratio
1.0
0.8
0.6
0.4
0.2
Proportion of Variance (PoV)
Cumulative explained variance
Principal component index

Visual analysis



eigendigits

در صورتی که سه بعد نخست، حاوی بخش عمده‌ای از واریانس باشند،
می‌توان داده‌ها از آن‌ها برای «بررسی دیداری» بهره برد.

چند نکته

- علاوه بر در نظر گرفتن PoV، می‌توان بردارهای ویژه‌ای که مقدار ویژه‌ی متناظر آن از یک حدآستانه (به عنوان مثال میانگین واریانس) کم‌تر است را حذف نمود.
- در صورتی که واریانس در ابعاد مختلف تغییرات زیادی داشته باشند، بیش از مقدار همبستگی بر روی مؤلفه‌ی اصلی اثرگذار خواهد بود.
  - در این شرایط می‌توان از بردارها و مقادیر ویژه‌ی «ماتریس همبستگی» (R) استفاده کرد یا این که داده‌ها را به گونه‌ای نرمال کرد که همگی واریانس یکسان داشته باشند.

تحلیل مؤلفه‌های اصلی(ادامه...)

- از زاویه‌ی دیگری نیز می‌توان به این مسأله نگاه کرد: هدف یافتن ماتریس تبدیلی است که داده‌های را به گونه‌ای نگاشت کند که در فضای جدید «ناهمبسته» باشند.

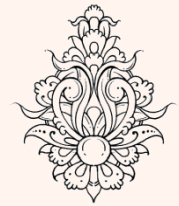
$$\mathbf{z} = \mathbf{W}^T \mathbf{x} \qquad \mathrm{Cov}(\mathbf{z}) = \mathbf{D}'$$

ماتریس قطری

- $\mathbf{C}_{d\times d}$ ماتریسی است که ستون‌هایش بردارهای ویژه‌ی ماتریس کواریانس است:

$$\mathbf{C}^T\mathbf{C} = \mathbf{I}$$

$$\mathbf{S} = \mathbf{S}\mathbf{C}\mathbf{C}^T$$



ادامه

تحلیل مؤلفه‌های اصلی(ادامه...)

$$\mathbf{S} = \mathbf{S}\mathbf{C}\mathbf{C}^T$$
$$= S[\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_d]C^T$$
$$= [\mathbf{S}\mathbf{c}_1, \mathbf{S}\mathbf{c}_2, \ldots, \mathbf{S}\mathbf{c}_d]\mathbf{C}^T$$
$$= [\lambda_1\mathbf{c}_1, \lambda_2\mathbf{c}_2, \ldots, \lambda_d\mathbf{c}_d]\mathbf{C}^T$$
$$= \lambda_1\mathbf{c}_1\mathbf{c}_1^T + \lambda_2\mathbf{c}_2\mathbf{c}_2^T + \ldots + \lambda_d\mathbf{c}_d\mathbf{c}_d^T$$
$$= \mathbf{C}\mathbf{D}\mathbf{C}^T$$

ماتریس قطری که عناصر روی قطر اصلی مقادیر ویژه‌ی ماتریس کواریانس هستند.

Spectral decomposition

$$\mathbf{C}^T\mathbf{S}\mathbf{C} = \mathbf{D}$$

$$\mathbf{z} = \mathbf{W}^T\mathbf{x}, \quad \mathrm{Cov}(\mathbf{z}) = \mathbf{W}^T\mathbf{S}\mathbf{W}$$

$$\mathbf{W} = \mathbf{C}$$

$$\mathrm{Cov}(\mathbf{z}) = \mathbf{D}$$

فاصله‌ی Mahalanobis

$$(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\sum^{-1}(\boldsymbol{x}-\boldsymbol{\mu})=c^2$$

$$\sum=\boldsymbol{QDQ^T}$$

ماتریس شامل بردارهای ویژه $\sum$ (در ستون‌ها)

$$\sum^{-1}=\boldsymbol{QD}^{-1}\boldsymbol{Q}^{\mathrm{T}}$$

ماتریس شامل مقادیر ویژه $\sum$ (قطراصلی)

$$\sum^{-1}=\mathbf{QD}^{-½}\mathbf{D}^{-½}\mathbf{Q}^{\mathrm{T}}$$

$$C^{2=}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\ \mathbf{QD}^{-½}\mathbf{D}^{-½}\mathbf{Q}^{\mathrm{T}}\ (\boldsymbol{x}-\boldsymbol{\mu})$$

$$C^{2=}[(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\ \mathbf{QD}^{-½}][\mathbf{D}^{-½}\mathbf{Q}^{\mathrm{T}}\ (\boldsymbol{x}-\boldsymbol{\mu})]$$

$$C^{2=}[(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\ \mathbf{QD}^{-½}]\ [(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\ \mathbf{QD}^{-½}]^{\mathrm{T}}$$

$$[(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\ \mathbf{QD}^{-½}]$$

تبدیل:

۱- چرخش در راستای بردارهای ویژه

۲- تغییر مقیاس بر اساس مقادیر ویژه(واریانس در راستای بردار ویژه)

چند نکته

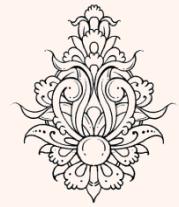
- PCA، نسبت به نویز به شدت حساس است.
  - یک روش ساده حذف داده‌های پرت با استفاده از فاصله‌ی Mahalanobis پیش از محاسبه‌ی ماتریس کواریانس است.
- در میان تمام بردارهای متعامد، PCA کم‌ترین میزان خطا را دارد.

Reconstruction error

$$\sum_t \left\| \hat{\mathbf{x}}^t - \mathbf{x}^t \right\|$$

- Karhunen-Loève expansion و Hotelling transform نام‌های دیگری هستند برای مفاهیم مشابه به کار می‌روند.
- در common principal components برای همه‌ی کلاس‌ها مؤلفه‌های اساسی یکسانی در نظر گرفته می‌شود، با این تفاوت که واریانس هر کلاس متفاوت در نظر گرفته می‌شود.

$$S_i = \mathbf{C}\mathbf{D}_i\mathbf{C}^T$$



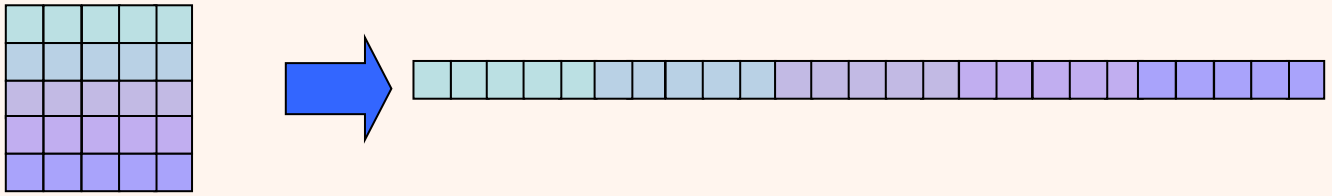
کاربرد PCA در شناسایی چهره

پایگاه داده‌ی ORL

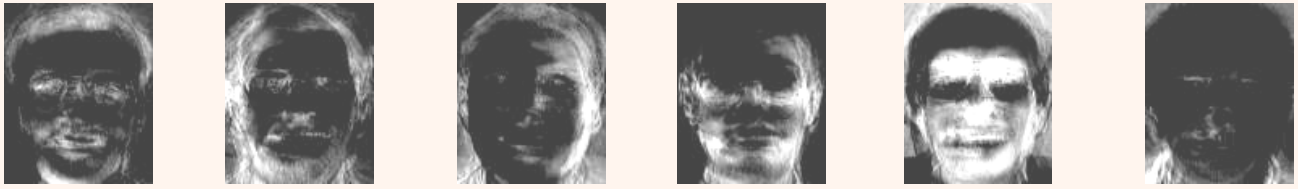
میانگین چهره‌ها

M. Turk, A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

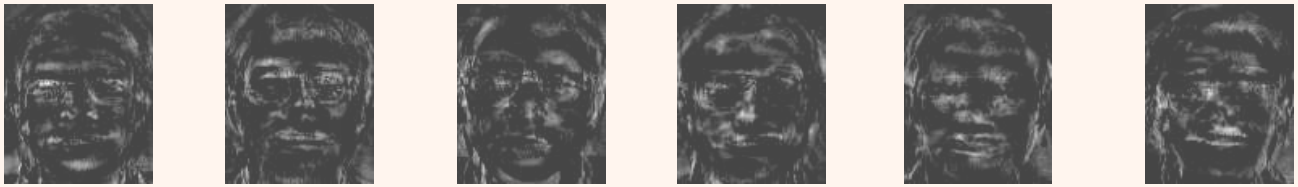
کاربرد PCA در شناسایی چهره(ادامه...)



Eigenfaces



۶ ۵ ۴ ۳ ۲ ۱



۴۰ ۳۹ ۳۸ ۳۷ ۳۶ ۳۵

کاربرد PCA در شناسایی چهره(ادامه...)

تصویراصلی

=

تصویرمیانگین

+(2.4

اولین مؤلفه‌ی اساسی

−0.05

دومین مؤلفه‌ی اساسی

+1.9

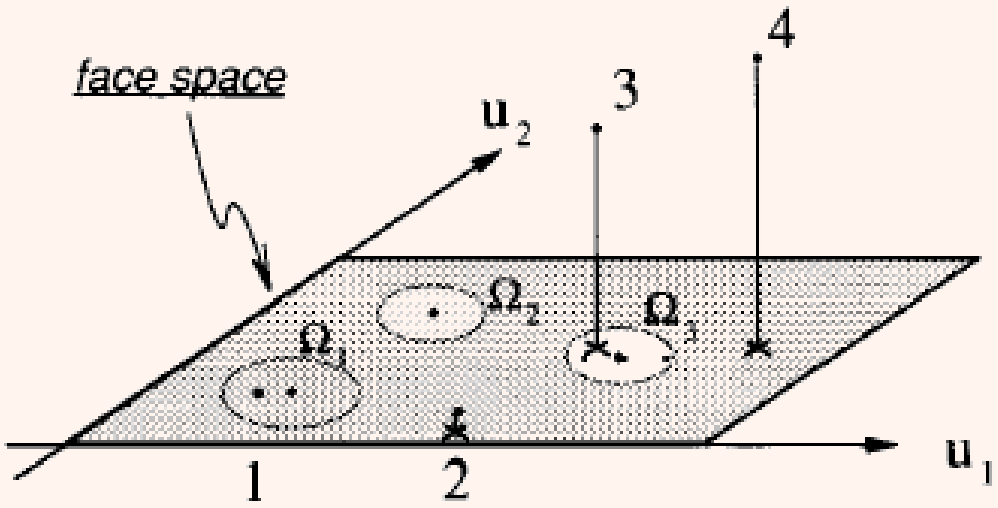
سومین مؤلفه‌ی اساسی

+ .... −0.012

)10^3

آخرین مؤلفه‌ی اساسی

تشخیص چهره



$$\|\hat{\mathbf{x}} - \mathbf{x}\|$$

Feature Embedding

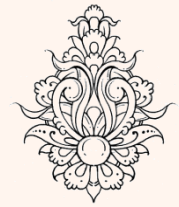
- ماتریس داده‌ها به صورت $\mathbf{X}_{N\times d}$ است.
- در صورت این که میانگین داده‌ها صفر باشد، ماتریس کواریانس خصیصه‌ها $\mathbf{X}^T\mathbf{X}_{d\times d}$ می‌باشد، در نتیجه
- با ضرب طرفین در $\mathbf{X}$

$$\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{w}_i = \lambda_i \mathbf{w}_i$$

$$\left(\mathbf{X}\mathbf{X}^T\right)\mathbf{X}\mathbf{w}_i = \lambda_i \mathbf{X}\mathbf{w}_i$$

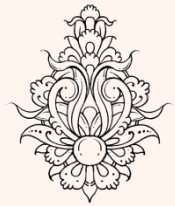
- در نتیجه $\mathbf{X}\mathbf{w}_i$ بردار ویژه‌ی $\mathbf{X}\mathbf{X}^T_{N\times N}$ با مقدار ویژه‌ی λ_i است.
- در این حالت بردار ویژه، مختصات نمونه‌ها در راستای $\mathbf{w}_i$ را نشان می‌دهد.

$$\left(\mathbf{X}\mathbf{X}^T\right)\mathbf{X}\mathbf{w}_i = \lambda_i \mathbf{X}\mathbf{w}_i$$



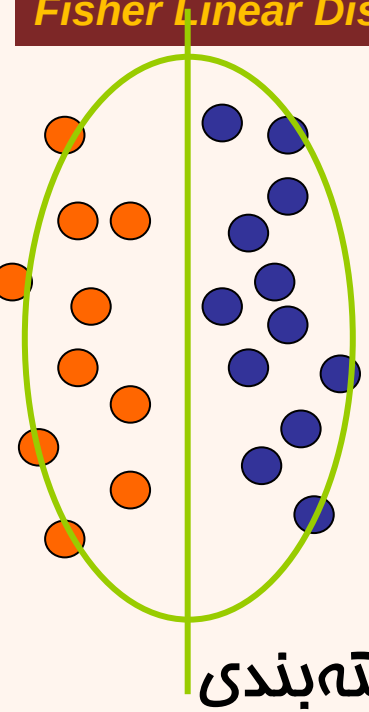
Feature Embedding

- ثابت می‌شود، رتبه‌ی ماتریس حداکثر $\min(d, N)$ می‌باشد.
- برای یک پایگاه داده حاوی چهل تصویر ۲۵۶×۲۵۶
  - ماتریس کواریانس خصیصه‌ها ۶۵۵۳۶×۶۵۵۳۶ خواهد بود.
  - در حالی که ماتریس شباهت نمونه‌ها ۴۰×۴۰ می‌باشد.
  - این ماتریس، شباهت دوبه‌دو نمونه‌ها را نشان می‌دهد، از این نظر می‌توان گفت این شیوه داده‌های dبعدی را در فضای kبعدی به گونه‌ای قرار می‌دهد که فاصله‌ی بین آن‌ها حفظ می‌شود.
  - این ماتریس شامل ضرب داخلی دو به دوی همه نمونه‌های آموزش است.



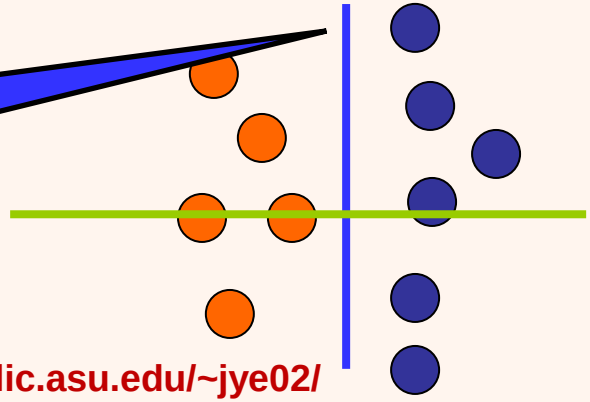
تحلیل تفکیک خطی

دسته‌بندی دو کلاسه



Ronald Aylmer Fisher

- آیا PCA برای دسته‌بندی مناسب است؟
  - راستای نگاشت بر اساس واریانس، انتخاب می‌شود.
  - در این میان ممکن است اطلاعات دسته‌ها از بین بروند.
- تحلیل تفکیک خطی، «بانظارت» است و برای دسته‌بندی به کار می‌رود.
  - هدف آن کاهش بعد همراه با حفظ اطلاعاتی است که بین دسته‌ها تمایز قائل می‌شود.



در این راستا دو کلاس همپوشانی دارند

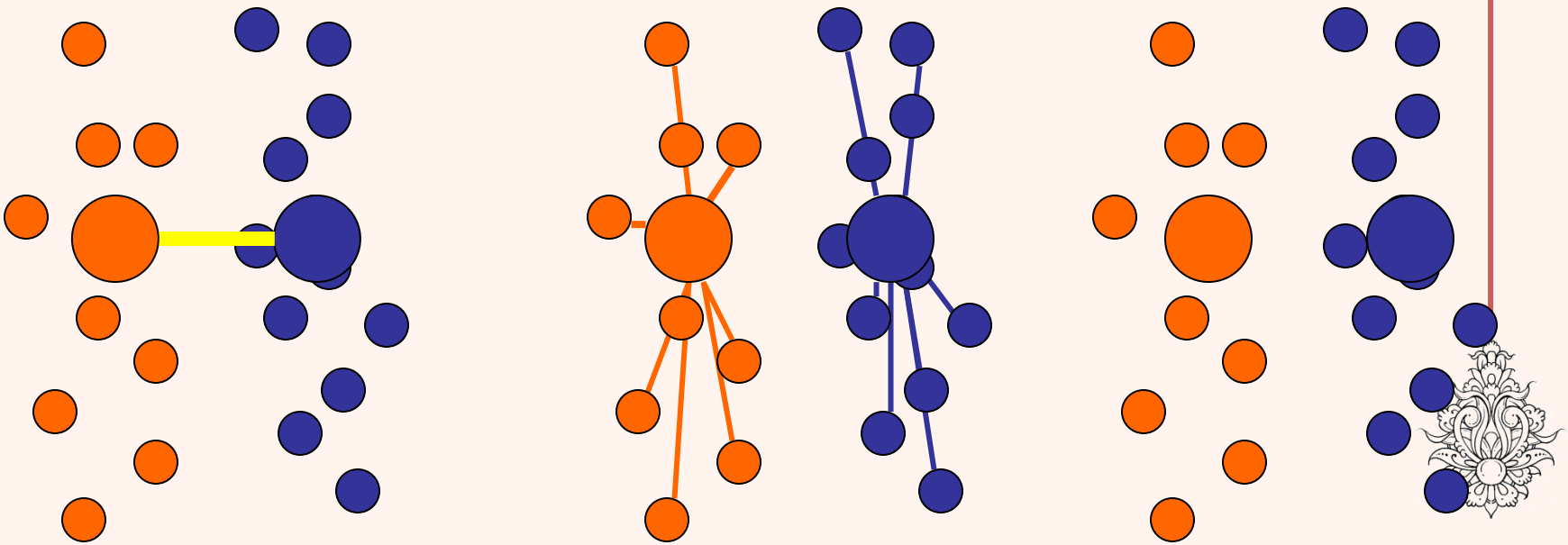
در این راستا دو کلاس بدون خطا دسته‌بندی می‌شوند

http://www.public.asu.edu/~jye02/

کاهش ابعاد برای دسته‌بندی

دسته‌بندی دو کلاسه

برای انتخاب راستای مناسب برای نگاشت، باید اطلاعات دسته‌ها نیز در نظر گرفته شود.



http://www.public.asu.edu/~jye02/

تحلیل تفکیک خطی (ادامه...)

در LDA، نگاشت به گونه‌ای انجام می‌شود که فاصله‌ی بین دو کلاس حداکثر شده و فاصله‌ی نمونه‌های متعلق به یک کلاس مینیمم گردد.

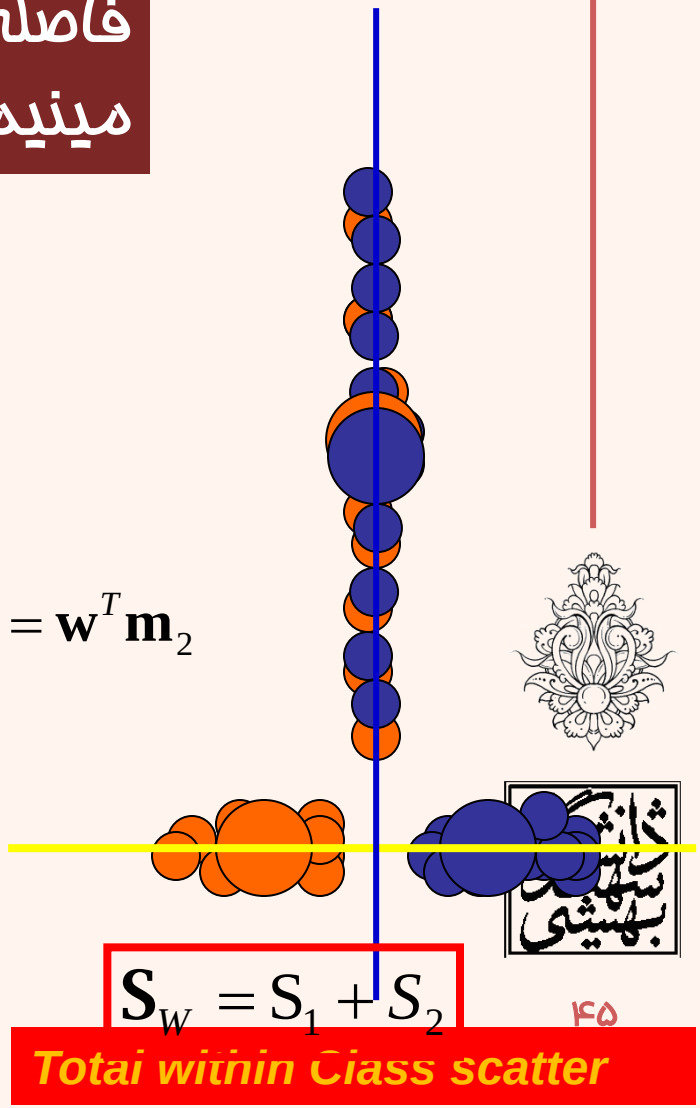
دسته‌بندی دو کلاسه

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

$$m_1 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t r^t}{\sum_t r^t} = \mathbf{w}^T \mathbf{m}_1, \quad m_2 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t (1 - r^t)}{\sum_t (1 - r^t)} = \mathbf{w}^T \mathbf{m}_2$$

$$s_1^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_1)^2 r^t$$

$$s_2^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_2)^2 (1 - r^t)$$



$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

تحلیل تفکیک خطی (ادامه...)

دسته‌بندی دو کلاسه

$$(m_1 - m_2)^2 = (\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2)^2$$

$$= \mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w}$$

$$= \mathbf{w}^T\mathbf{S}_B\mathbf{w}$$

Between class scatter matrix

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$$

$$s_1^2 = \sum_t (\mathbf{w}^T\mathbf{x}^t - m_1)^2 r^t$$

$$= \sum_t \mathbf{w}^T(\mathbf{x}^t - \mathbf{m}_1)(\mathbf{x}^t - \mathbf{m}_1)^T\mathbf{w} r^t$$

$$= \sum_t \mathbf{w}^T\mathbf{S}_1\mathbf{w} r^t$$

Class scatter matrix for C1

$$\mathbf{S}_1 = \sum_t r^t(\mathbf{x}^t - \mathbf{m}_1)(\mathbf{x}^t - \mathbf{m}_1)^T$$

$$s_1^2 + s_2^2 = \mathbf{w}^T\mathbf{S}_W\mathbf{w}$$

$$\mathbf{S}_W = S_1 + S_2$$

Total within Class scatter

تحلیل تفکیک خطی (ادامه...)

دسته‌بندی دو کلاسه

- هدف ماکزیمم کردن رابطه‌ی زیر است:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\left|\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)\right|^2}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

- با مشتق گرفتن:

$$\frac{\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \left(2(\mathbf{m}_1 - \mathbf{m}_2) - \frac{\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \mathbf{S}_W \mathbf{w} \right) = 0$$

Scalar

$$\mathbf{w} = c\mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

یادآوری: جداساز خطی

$$\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$$

when $p(\mathbf{x} \mid C_i) \sim \mathcal{N}(\mu_i, \Sigma)$

بدین‌ترتیب، برای دسته‌های نرمال، LDA جداساز بهینه است.

دسته‌بندی برای بیش از دو کلاس

- زمانی که تعداد کلاس‌ها بیشتر از دو باشد: برای کاهش ابعاد، ماتریس $\mathbf{W}_{d\times k-1}$ برای نگاشت مورد استفاده قرار می‌گیرد:

$$\mathbf{z} = \mathbf{W}^T\mathbf{x}$$

$$\mathbf{S}_W = \sum_{i=1}^{K}\mathbf{S}_i \qquad \mathbf{S}_i = \sum_t r_i^t\left(\mathbf{x}^t - \mathbf{m}_i\right)\left(\mathbf{x}^t - \mathbf{m}_i\right)^T$$

Within-class scatter

Between-class scatter:

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i\left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^T \qquad \mathbf{m} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{m}_i \qquad N_i = \sum_t r_i^t$$

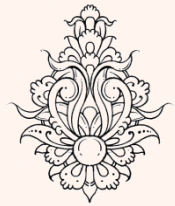
- پس از نگاشت، $\mathbf{W}^T\mathbf{S}_B\mathbf{W}$ و $\mathbf{W}^T\mathbf{S}_W\mathbf{W}$ ماتریس‌های پراکندگی داده «بین‌دسته‌ها» و «درون‌دسته‌ها» خواهند بود.

دسته‌بندی برای بیش از دو کلاس

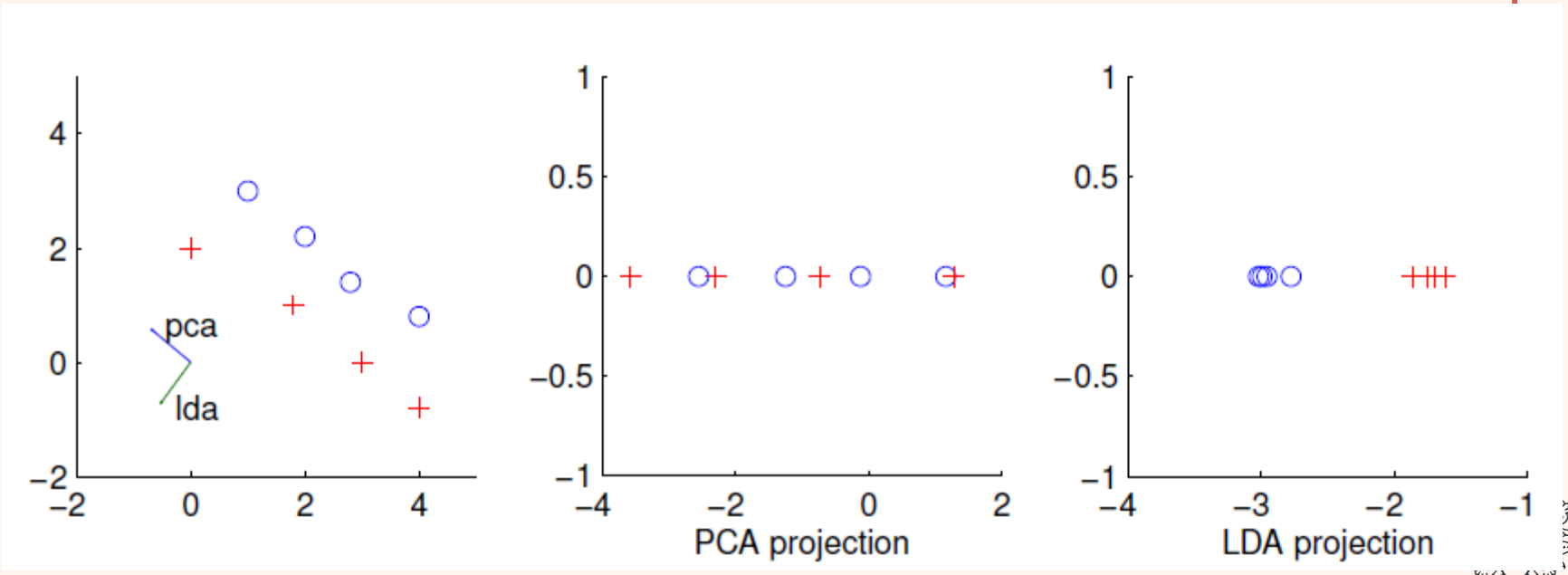
- در نتیجه در صورت بیشینه شدن عبارت زیر، دسته‌بندی به بهترین شکل انجام می‌شود.
  - برای ماتریس کواریانس، دترمینان معیاری است که پراکندگی داده را نشان می‌دهد.

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$$

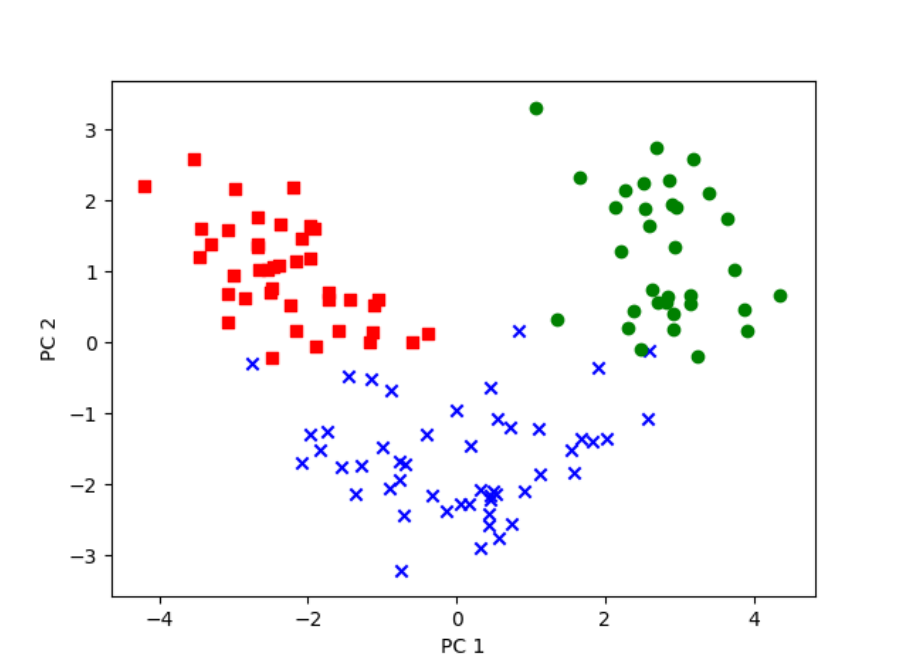
  - در این حالت پاسخ، بردارهای ویژه متناظر با بزرگ‌ترین مقادیر ویژه‌ی ماتریس $\mathbf{S}_W^{-1}\mathbf{S}_B$ خواهد بود.

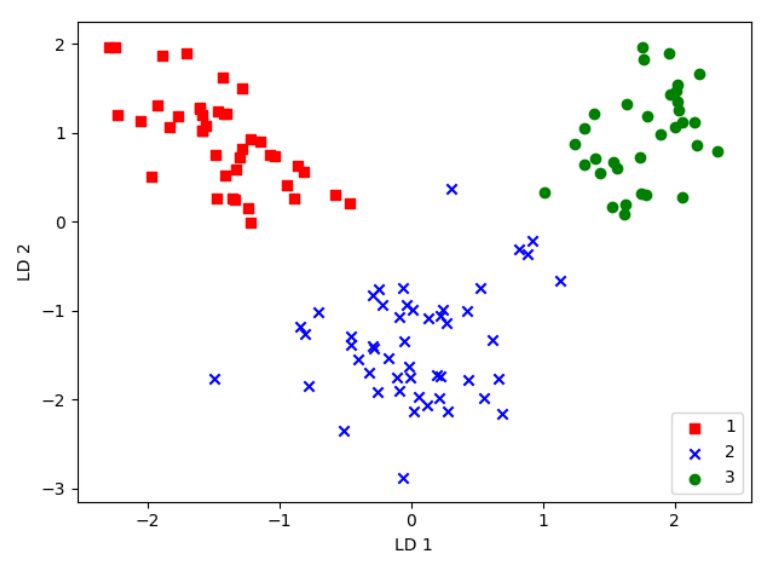


PCA vs LDA



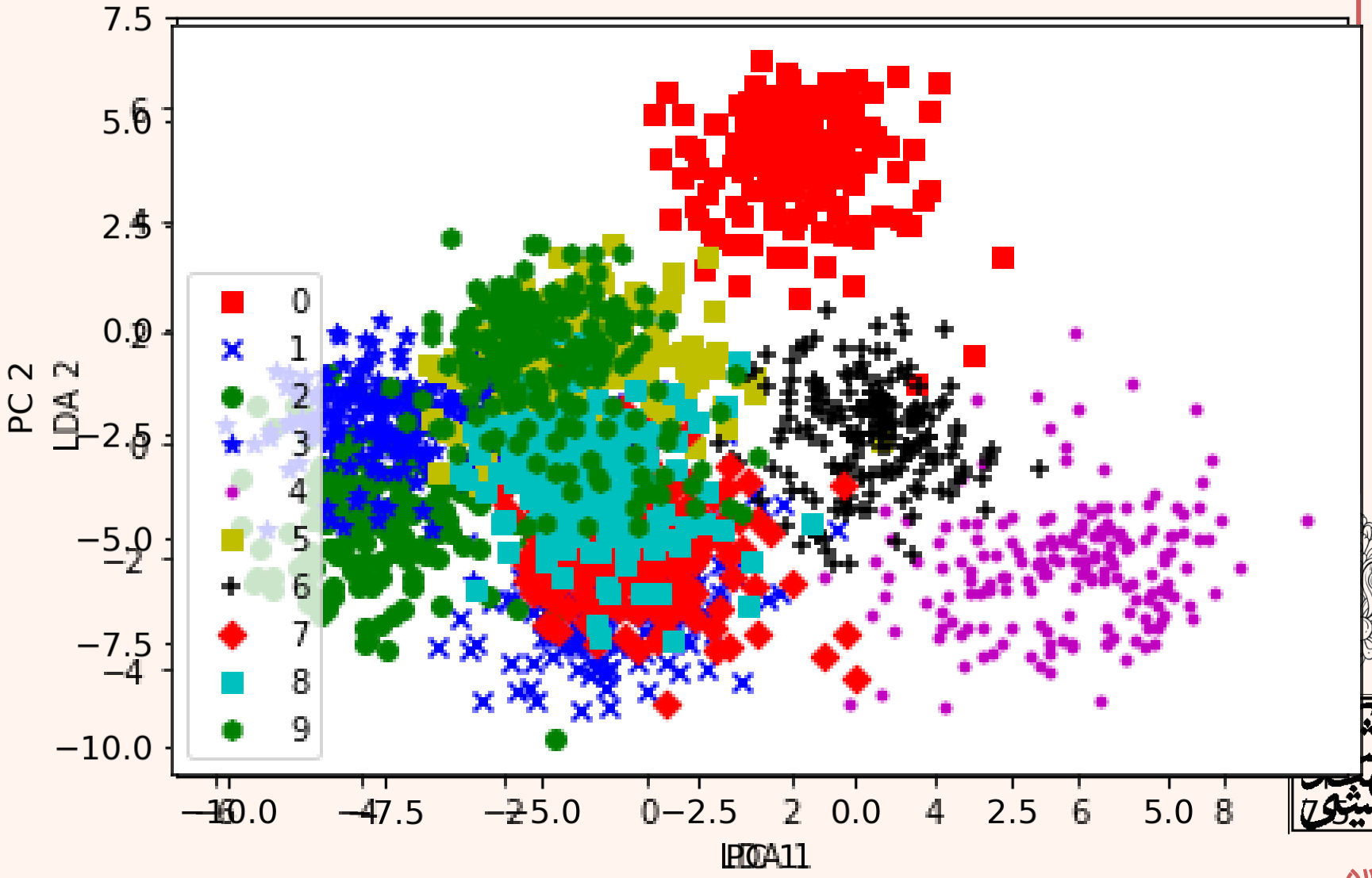
PCA vs LDA





***Raschka, S., and V. Mirjalili. Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and Tensorflow 2, 3rd Edition*. Packt Publishing, 2019.**

مثال



تجزیه‌ی مقدارهای تکین

- با استفاده از SVD، یک ماتریس به سه ماتریس تجزیه می‌شود:

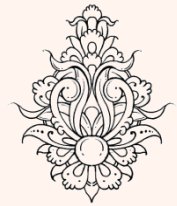
$$\mathbf{X}_{N\times d} = \mathbf{V}_{N\times N}\mathbf{A}_{N\times d}\mathbf{W}^{T}_{d\times d}$$

- $\mathbf{V}$ شامل بردارهای ویژه‌ی XX^T می‌باشد، $\mathbf{W}$ شامل بردارهای ویژه‌ی X^TX است و A مقادیر ویژه را در k عنصر قطری خود دارد.

$$\mathbf{X}\mathbf{X}^T = \left(\mathbf{V}\mathbf{A}\mathbf{W}^T\right)\left(\mathbf{V}\mathbf{A}\mathbf{W}^T\right)^T = \mathbf{V}\mathbf{A}\mathbf{W}^T\mathbf{W}\mathbf{A}^T\mathbf{V}^T = \mathbf{V}\mathbf{E}\mathbf{V}^T$$

$$\mathbf{X}^T\mathbf{X} = \left(\mathbf{V}\mathbf{A}\mathbf{W}^T\right)^T\left(\mathbf{V}\mathbf{A}\mathbf{W}^T\right) = \mathbf{W}\mathbf{A}^T\mathbf{V}^T\mathbf{V}\mathbf{A}\mathbf{W}^T = \mathbf{W}\mathbf{D}\mathbf{W}^T$$

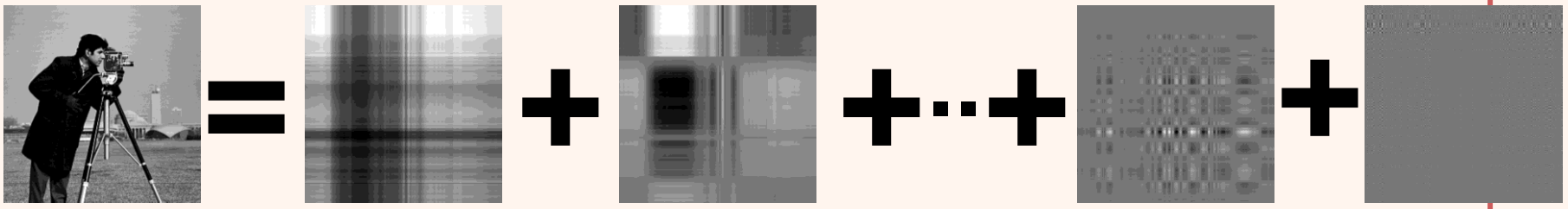
$$\mathbf{E} = \mathbf{A}\mathbf{A}^T \qquad \mathbf{D} = \mathbf{A}^T\mathbf{A}$$



$$\mathbf{X}_{N\times d} = \mathbf{V}_{N\times N}\mathbf{A}_{N\times d}\mathbf{W}^{T}_{d\times d}$$

تجزیه‌ی مقدارهای تکین

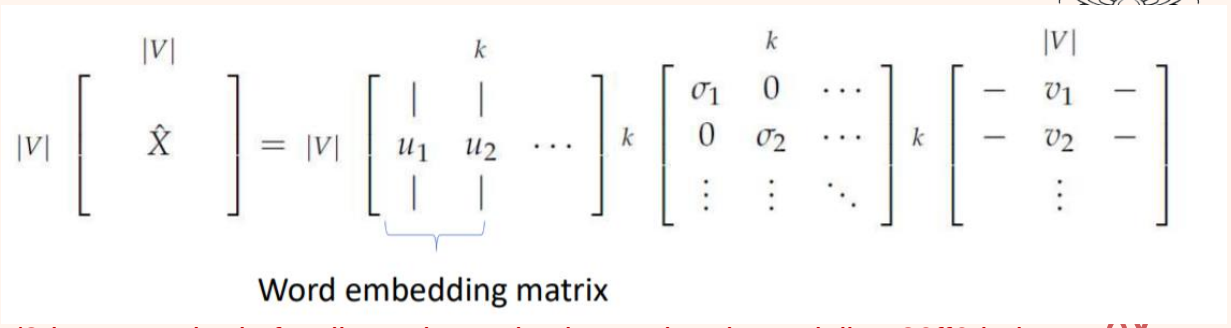
$$\mathbf{X} = a_1\mathbf{v}_1\mathbf{w}_1^T + \ldots + a_k\mathbf{v_k}\mathbf{w}_k^T \qquad \mathbf{X} = \sum_{i=1}^{k} a_k\mathbf{v}_k\mathbf{w}_k^T$$



I enjoy flying,
I like NLP,
I like deep learning

	flying	enjoy	I	NLP	deep	learning	like
flying	0.0	1.0	0.0	0.0	0.0	0.0	0.0
enjoy	1.0	0.0	1.0	0.0	0.0	0.0	0.0
I	0.0	1.0	0.0	0.0	0.0	0.0	2.0
NLP	0.0	0.0	0.0	0.0	0.0	0.0	1.0
deep	0.0	0.0	0.0	0.0	0.0	1.0	1.0
learning	0.0	0.0	0.0	0.0	1.0	0.0	0.0
like	0.0	0.0	2.0	1.0	1.0	0.0	0.0

Co-occurrence matrix A



https://towardsdatascience.com/2-latent-methods-for-dimension-reduction-and-topic-modeling-20ff6d7d547

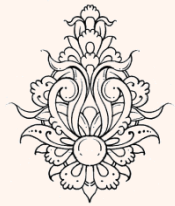
https://github.com/AparGarg99/NLP_BU_6thSem/blob/master/Lab%202/English_Code.ipynb

	0	1	2	3	4	5	6
NLP	-3.471975e-01	0.000000e+00	-1.942887e-01	0.000000e+00	0.000000e+00	-4.183768e-01	8.164966e-01
flying	-1.396622e-01	0.000000e+00	6.739877e-01	-9.992007e-16	-4.440892e-16	5.996402e-01	4.082483e-01
enjoy	1.249001e-16	-3.687707e-01	-2.220446e-16	-8.041284e-01	4.662464e-01	-1.110223e-16	-5.551115e-17
I	-8.340573e-01	-5.277047e-18	2.854104e-01	-7.948171e-16	-1.162003e-15	-2.371134e-01	-4.082483e-01
deep	-4.053355e-01	0.000000e+00	-6.530953e-01	0.000000e+00	0.000000e+00	6.396638e-01	-1.110223e-16
like	1.665335e-16	-9.167567e-01	-2.775558e-17	2.318040e-01	-3.253062e-01	5.551115e-17	1.110223e-16
learning	-1.318390e-15	-1.535102e-01	1.221245e-15	5.473979e-01	8.226726e-01	3.885781e-16	-5.551115e-17

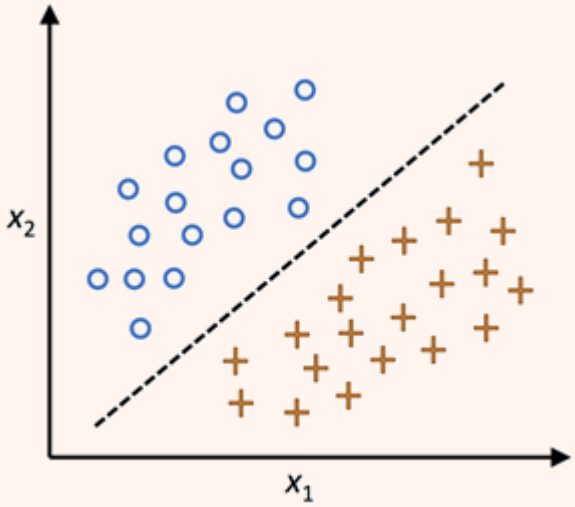
Matrix U (before selecting k singular features)

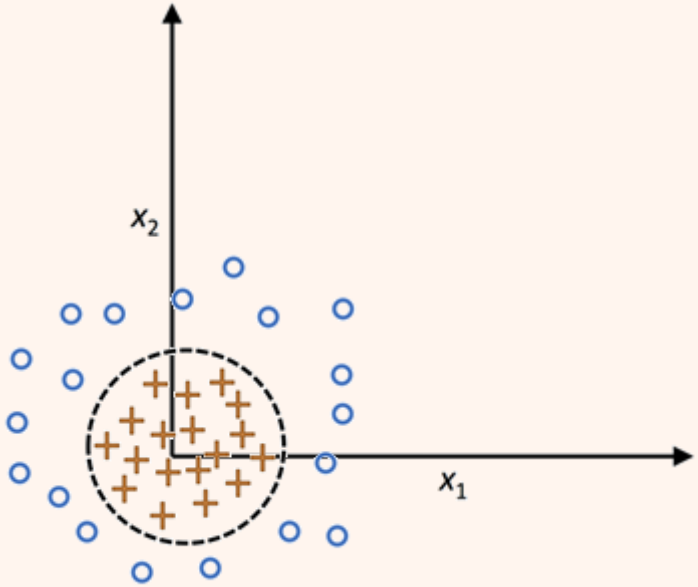
	0	1	2
NLP	-3.471975e-01	0.000000e+00	-1.942887e-01
flying	-1.396622e-01	0.000000e+00	6.739877e-01
enjoy	1.249001e-16	-3.687707e-01	-2.220446e-16
I	-8.340573e-01	-5.277047e-18	2.854104e-01
deep	-4.053355e-01	0.000000e+00	-6.530953e-01
like	1.665335e-16	-9.167567e-01	-2.775558e-17
learning	-1.318390e-15	-1.535102e-01	1.221245e-15

Matrix U (after selecting k singular features)



- زمانی که داده‌ها جدایی‌پذیر خطی نیستند، استفاده از روش‌ها خطی برای کاهش بعد موجب از دست رفتن اطلاعات کلاس‌ها می‌شود.





در این حالت نیز می‌توان داده‌ها را به فضایی با ابعاد بالاتر نگاشت کرد به گونه‌ای که در آن فضا به صورت خطی قابل جداسازی باشند و پس از آن از روش‌های سنتی کاهش بعد بهره برد.

KPCA

Kernel principal component analysis

- در این حالت نیز می‌توان از kernel trick استفاده کرد.

– با توجه به این که در Feature embedding از ماتریس تشابه(ضرب داخلی) نمونه‌های آموزشی استفاده می‌شود با استفاده از توابع هسته‌ی مناسب به راحتی و بدون مواجه شدن با نگاشت به فضایی دیگر، کاهش ابعاد به شیوه‌ی غیرخطی قابل انجام خواهد بود.

$$\phi : \mathbb{R}^d \to \mathbb{R}^k \quad (k \gg d)$$

توابع کرنل ضرب داخلی دو بردار ویژگی در فضایی با ابعاد بالا را بدون نگاشت با آن فضا به دست می‌آورد:

$$k\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \phi\left(\mathbf{x}^{(i)}\right)^T \phi\left(\mathbf{x}^{(j)}\right)$$

توابع کرنل متعارف:

Radial Basis function(RBF)

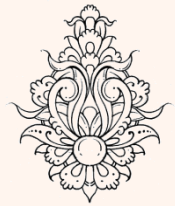
$$\kappa\left(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}\right)=\exp\left(-\gamma\left\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}\right\|^2\right)$$

$$\kappa\left(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}\right)=\left(\boldsymbol{x}^{(i)T}\boldsymbol{x}^{(j)}+\theta\right)^p$$

polynomial kernel

Hyperbolic Tangent Kernel

$$\kappa\left(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}\right)=\tanh\left(\eta\boldsymbol{x}^{(i)T}\boldsymbol{x}^{(j)}+\theta\right)$$



حذف میانگین

$$k_{ij} = \phi_i^T \phi_j$$

- باید توجه داشت که لازم است نمونه‌ها در فضای جدید به گونه‌ای نرمال شود که میانگین آن‌ها صفر باشد:

$$k'_{ij} = \left(\phi_i - \frac{1}{N}\sum_m \phi_m\right)^T \left(\phi_j - \frac{1}{N}\sum_n \phi_n\right)$$

$$k'_{ij} = \phi_i^T \phi_j - \phi_i^T \left[\frac{1}{N}\sum_n \phi_n\right] - \left[\frac{1}{N}\sum_m \phi_m^T\right]\phi_j + \left[\frac{1}{N}\sum_n \phi_n\right]\left[\frac{1}{N}\sum_m \phi_m^T\right]$$

$$k'_{ij} = k_{ij} - \left[\frac{1}{N}\sum_n k_{in}\right] - \left[\frac{1}{N}\sum_m k_{mj}\right] + \frac{1}{N^2}\sum_m\sum_n k_{mn}$$

$$\mathbf{1}_{N[i,j]} = [1/N]_{i,j}$$

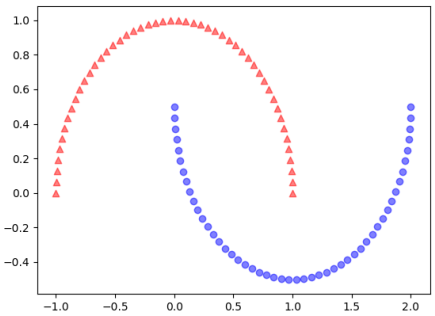
$$\mathbf{K}' = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N$$

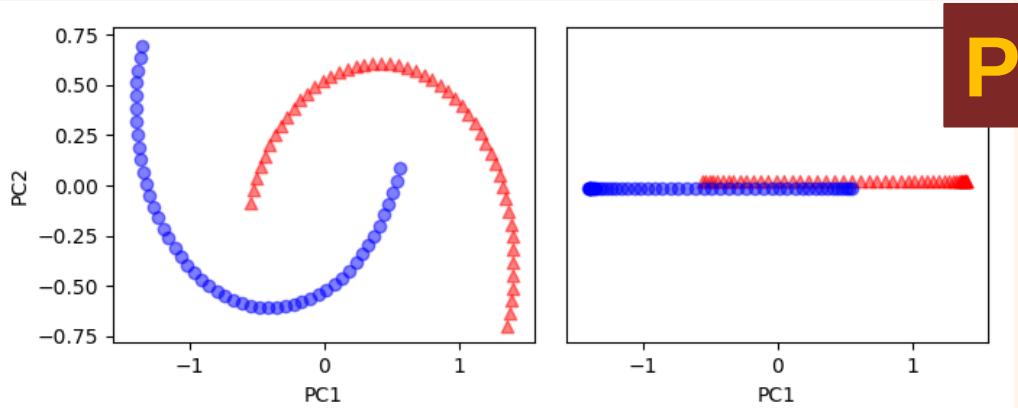
مثال گام به گام

```
def rbf_kernel_pca(X, gamma, n_components):

    sq_dists = pdist(X, 'sqeuclidean')
    mat_sq_dists = squareform(sq_dists)
    K = exp(-gamma * mat_sq_dists)

    N = K.shape[0]
    one_n = np.ones((N, N)) / N
    K = K - one_n.dot(K) - K.dot(one_n) + one_n.dot(K).dot(one_n)


 eigvals, eigvecs = eigh(K)
    eigvals, eigvecs = eigvals[::-1], eigvecs[:, ::-1]


    X_pc = np.column_stack((eigvecs[:, i]
                            for i in range(n_components)))



    return X_pc
```

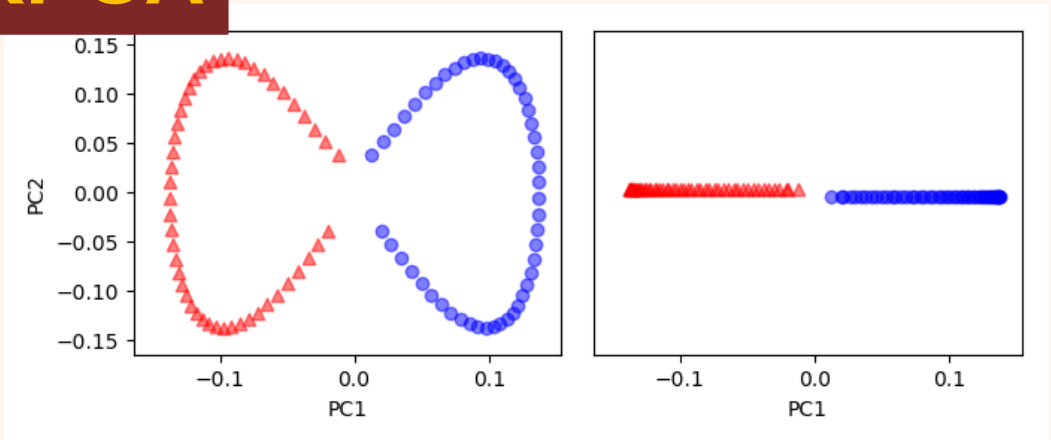
$$\kappa\left(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}\right)=\exp\left(-\gamma\left\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(j)}\right\|^2\right)$$

$$\mathbf{K}'=\mathbf{K}-\mathbf{1}_n\mathbf{K}-\mathbf{K}\mathbf{1}_n+\mathbf{1}_n\mathbf{K}\mathbf{1}_n$$

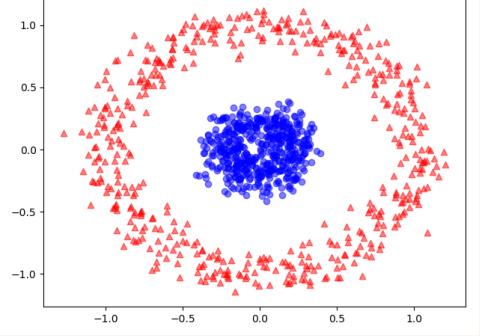




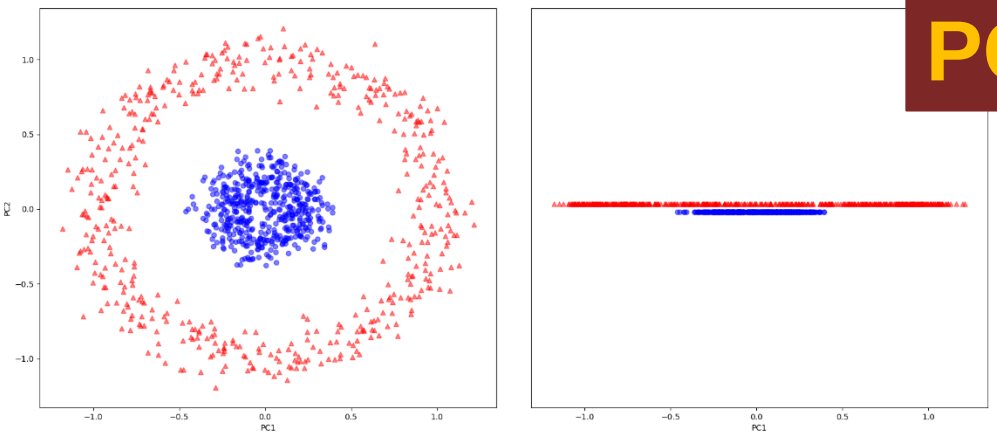
KPCA




```
from sklearn.decomposition import KernelPCA

X, y = make_moons(n_samples=100, random_state=123)
scikit_kpca = KernelPCA(n_components=2, kernel='rbf', gamma=15)
X_skernpca = scikit_kpca.fit_transform(X)
```



PCA



KPCA