EE445 Mod3-Lec2: Principal Component Analysis & Regression

References:

- [CE-OptMod]: Chapter: 5.3.2
- Additional reference: Chapter 15 of "A Course in ML" by Hal Daumé (http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf)

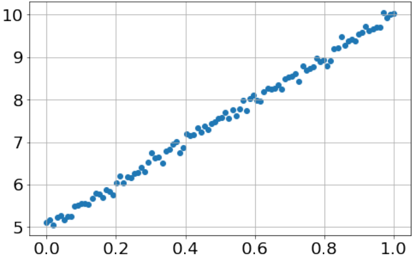
Outline

1. Principal Component Analysis
2. Principal Component Regression

What is PCA?

- Principal component analysis (PCA) is a technique of **unsupervised learning** widely used to "discover" the most important, or informative, directions in a data set.
- Aside **unsupervised learning**: i.e., learning from data without labels or observations—essentially with only features x and no observations y
- There are many reasons you may want to perform `PCA` on a data set
  - to visualize the data in a lower-dimensional space,
  - to understand the sources of variability in the data,
  - to understand correlations between different coordinates of the data points, etc.

What is PCA?



- the majority of the variation of the data is contained in the direction at about 45 degrees from the x-axis
- In contrast, the direction at about 135 degrees contains very little variation.

go to: `https://setosa.io/ev/principal-component-analysis/`

What is PCA? Example

- Suppose we are given dataset $\{x^{(1)}, \ldots, x^{(m)}\}$ of attributes of m different types of vehicles, such as their maximum speed, turn radius, and so on.
- Let $x^{(i)} \in \mathbb{R}^n$ with $n \ll m$
- Unknown to us, two different attributes—some x_i and x_j—respectively give a car's
  1. maximum speed measured in miles per hour,
  2. and the maximum speed measured in kilometers per hour.
- These two attributes are therefore almost linearly dependent, up to only small differences introduced by rounding off to the nearest mph or kph
- Thus the data really lies approximately on an $n-1$ dimensional subspace.
- **How can we automatically detect, and perhaps remove, this redundancy?**

Data Preprocessing: Why?

- It is important to preprocess the data to normalize its mean and variance
- Standardizing the features to have mean zero with a standard deviation of one is important when we compare measurements that have different units.
- Variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias.

Data Preprocessing: How

Let $(z^{(1)}, \ldots, z^{(m)})$ be the original raw data, then preprocessing goes as follows:

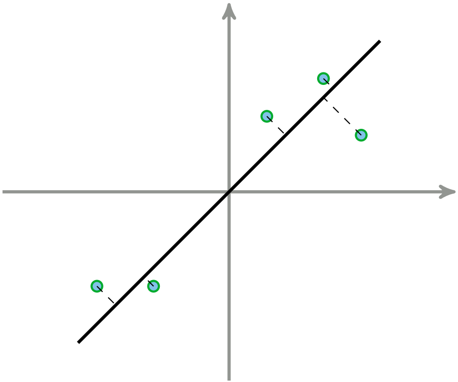
1. Let $\mu = \frac{1}{m}\sum_{i=1}^{m} z^{(i)}$
2. Define $\tilde{x}^{(i)} = z^{(i)} - \mu$
3. Let $\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(\tilde{x}_j^{(i)})^2$
4. Define $x^{(i)} = (\tilde{x}_1^{(i)}/\sigma_1, \ldots, \tilde{x}_n^{(i)}/\sigma_n)$

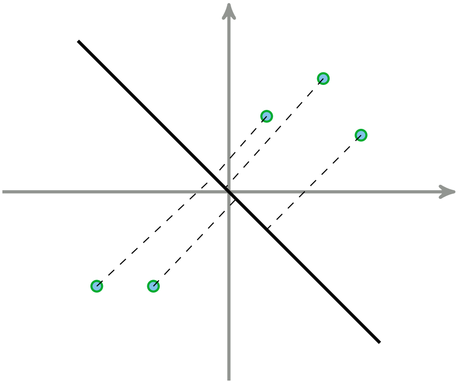
- Steps 1-2 zero out the mean of the data
- Steps 3-4 rescale each coordinate to have unit variance, which ensures that different attributes are all treated on the same "scale."

How do we compute the "major axis of variation"?

- We want to compute the direction on which the data approximately lies.
- One way to pose this problem is as finding the unit vector u so that when the data is projected onto the direction corresponding to u, the variance of the projected data is maximized
- In other words, we would like to choose a direction u so that if we were to approximate the data as lying in the direction/subspace corresponding to u, as much as possible of this variance is still retained.

Toy Example





- projected data still has a fairly large variance, and points are far from origin
- projections have a significantly smaller variance, and are closer to the origin
- Want to o automatically select the direction u corresponding to the left graphic

PCA Warm up: Projecting onto first principle component

- Recall: the length of the projection of x onto u is given by $x^\top u$
- To maximize the variance of the projections, we choose a unit-length u to maximize

$$\frac{1}{m}\sum_{i=1}^{m}((x^{(i)})^\top u)^2 = \frac{1}{m}\sum_{i=1}^{m} u^\top x^{(i)}(x^{(i)})^\top u = u^\top \underbrace{\left(\sum_{i=1}^{m} x^{(i)}(x^{(i)})^\top\right)}_{=:\Sigma = X^\top X} u$$

- Note that $\Sigma = X^\top X$ where

$$X = \begin{bmatrix} — & (x^{(1)})^\top & — \\ & \cdots & \\ — & (x^{(m)})^\top & — \end{bmatrix}$$

- Caution!: the $x^{(i)}$ here are the pre-processed features—i.e., they are the centered and scaled (normalized) features

PCA Warm up: Projecting onto first principle component

- **How**? This is actually an optimization problem given by

$$\max_u \|Xu\|^2 \text{ subject to } \|u\|^2 - 1 = 0 \quad (\text{note that } \|Xu\|^2 = u^\top \Sigma u)$$

- To solve, we write out the "Lagrangian" (more to come in **Module 4**)

$$\mathcal{L}(u, \lambda) = \|Xu\|^2 - \lambda(\|u\|^2 - 1) = u^\top \Sigma u - \lambda(u^\top u - 1)$$

$$\nabla_u \mathcal{L} = 2\Sigma u - 2\lambda u = 0 \implies \Sigma u = \lambda u$$

- Hence, we choose an eigenvector u of Σ that chooses the largest eigenvalue
- u is called the **principal eigenvector**
- **Summary**: we have found that if we wish to find a 1-dimensional subspace with which to approximate the data, we should choose u to be the principal eigenvector of Σ

What about projecting on to $k = 2$ components?

- To get a second dimension, we want to find a new vector v on which the data has maximal variance, but to avoid redundancy, we want $v^\top u = 0$
- Optimization problem:

$$\max_v \|Xv\|^2 \text{ subject to } \|v\|^2 - 1 = 0, \text{ and } u^\top v = 0$$

- Optimality for the Lagrangian $\mathcal{L}(v, \lambda_1, \lambda_2) = \|Xv\|^2 - \lambda_1(\|v\|^2 - 1) - \lambda_2 u^\top v$:

$$\nabla_v \mathcal{L} = 2\underbrace{X^\top X}_{=\Sigma} v - 2\lambda_1 v - \lambda_2 u = 0 \implies 2\underbrace{u^\top \Sigma v}_{=\lambda u^\top v} - 2\lambda_1 \underbrace{u^\top v}_{=0} - \lambda_2 \underbrace{u^\top u}_{=1} = -\lambda_2 \cdot 1 = 0$$

$$\implies \lambda_2 = 0 \implies \Sigma v = \lambda_1 v \implies (\lambda_1, v) \text{ second largest eigenpair}$$

PCA More Generally

- Suppose we wish to project our data on to a k-dimensional subspace ($k < n$)
- We should choose $u_1, \ldots, u_k$ to be the top k eigenvectors of Σ.
- The u_i's form a new, orthogonal basis for the data
- Indeed, recall that Σ is symmetric so we can always choose the u_i's to be orthogonal to one another
- Next, we represent each $x^{(i)}$ in the new basis

$$y^{(i)} = (u_1^\top x^{(i)}, u_2^\top x^{(i)}, \ldots, u_k^\top x^{(i)}) \in \mathbb{R}^k$$

- $x^{(i)}$ are n–dimensional and $y^{(i)}$ are k–dimensional
- PCA is therefore also referred to as a **dimensionality reduction** algorithm.
- Vectors $u_1, \ldots, u_k$ are called the first k **principal components**

Summary: PCA Algorithm

- Pre-process the raw data $(z^{(1)}, \ldots, z^{(m)})$
  1. Recenter the data: define $\tilde{x}^{(i)} = z^{(i)} - \mu$ where $\mu = \frac{1}{m}\sum_{i=1}^{m} z^{(i)}$
  2. Rescale/normalize: define $x^{(i)}$ with entries $x_j^{(i)} = \tilde{x}_j^{(i)}/\sigma_j$ where $\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(\tilde{x}_j^{(i)})^2$
- Run PCA
  1. Compute the covariance matrix $\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}(x^{(i)})^\top = \frac{1}{m}X^\top X$
  2. Compute the eigenvalues and (orthonormal) eigenvectors of Σ
  3. Retain k eigenvectors with largest eigenvalues V_k
  4. Project X onto the principal component space

Alternative Derivation via Reconstruction Error

- Rather than maximizing variance, we may want to minimize reconstruction error
- 1–dimensional case: we are looking for a single projection direction u
- projected data: $y = Xu$ where each y_i is the position of the i-th feature vector along u
- To project back into the original space we do $yu^\top = Xuu^\top$—i.e., $yu^\top$ is the *reconstructed* value
- **Reconstruction Error**:

$$\|X - yu^\top\|^2 = \|X - Xuu^\top\|^2 = \|X\|^2 + \underbrace{\|Xuu^\top\|^2}_{=\|X\|^2} - 2\mathrm{Tr}(X^\top Xuu^\top)$$

$$\implies \|X - yu^\top\|^2 = 2\underbrace{\|X\|^2}_{\text{constant}} - 2u^\top X^\top Xu$$

- This is equivalent to minimizing $\|Xu\|^2$

Connections with SVD

- Facts: for a symmetric matrix $\Sigma = \Sigma^\top$,
  - the singular values are the absolute values of the eigenvalues and $\Sigma = U\Lambda V^\top$ where $U = V$
  - if $\Sigma \geq 0$, then $\lambda_i \geq 0$
  - if $\Sigma \succ 0$, then $\lambda_i > 0$ and U, V, Λ are all square non-singular matrices matrices
- Indeed, $\Sigma^\top\Sigma = \Sigma^2$ so that $\sigma_i(\Sigma) = \sqrt{\lambda_i(\Sigma^2)} = \lambda_i(\Sigma)$

Use the SVD to scale up!

- Often we have very large data sets—i.e., Σ might be very big in terms of dimension
- Problems: Computing eigenvectors is slow, and computing Σ could have numerical precision issues
- As an alternative we can use SVD since PCA reduces to SVD

Reducing PCA to SVD

- $\Sigma = X^\top X \in \mathbb{R}^{n\times n}$ is symmetric PSD $\implies$ $\Sigma = Q\Lambda Q^\top$ where $QQ^\top = I$
- Consider the SVD of $X = USV^\top$.

$$\Sigma = X^\top X = (USV^\top)^\top (USV^\top) = VS^\top \underbrace{U^\top U}_{=I} SV^\top = V\Lambda V^\top, \; V \equiv Q$$

- Hence, the rows of $V^\top = Q^\top$ are the eigenvectors of $\Sigma = X^\top X$
  - The right singular vectors of X are the same as the eigenvectors of $X^\top X$
  - The eigenvalues of $X^\top X$ are the squares of the singular values of X
- Thus PCA reduces to computing the SVD of X (without having to form $X^\top X$!).
- Output of PCA is the top k eigenvectors of $X^\top X \iff$ SVD of $X = USV^\top$ gives top k eigenvectors of $X^\top X$ via first k rows of $V^\top$

PCA based Low-Rank Approximations

- The techniques developed for PCA can also be used to produce low-rank matrix approximations.
- We seek matrices $Y, Z^\top$ such that $X = YZ^\top$

1. Preprocess the data $(z^{(1)}, \dots, z^{(m)})$ as before: so that the rows sum to the all-zero vector and, normalize each column
2. Form the covariance matrix $X^\top X$
3. Take the k rows of $Z^\top$ to be the top k principal components of X—the k eigenvectors $u_1, \dots, u_k$ of $X^\top X$ with largest eigenvalues
4. For $i = 1, \dots, m$, the i-th row of Y is defined as the projections $(\langle x^{(i)}, u_1 \rangle, \dots, \langle x^{(i)}, u_k \rangle)$.

Example: Eigenfaces

`Mod3-N3.ipynb`