CPSC 340: Machine Learning and Data Mining

More PCA
Fall 2018
1. Decision trees
2. Naïve Bayes classification
3. Ordinary least squares regression
4. Logistic regression
5. Support vector machines
6. Ensemble methods
7. Clustering algorithms
8. Principal component analysis
9. Singular value decomposition
10. Independent component analysis (bonus)
Last Time: Latent-Factor Models

- Latent-factor models take input data ‘X’ and output a basis ‘Z’:

  ![Diagram of Latent-Factor Model](image)

  - Usually, ‘Z’ has fewer features than ‘X’.

- Uses: dimensionality reduction, visualization, factor discovery.

<table>
<thead>
<tr>
<th>Trait</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Openness</td>
<td>Being curious, original, intellectual, creative, and open to new ideas.</td>
</tr>
<tr>
<td>Conscientiousness</td>
<td>Being organized, systematic, punctual, achievement-oriented, and dependable.</td>
</tr>
<tr>
<td>Extraversion</td>
<td>Being outgoing, talkative, sociable, and enjoying social situations.</td>
</tr>
<tr>
<td>Agreeableness</td>
<td>Being affable, tolerant, sensitive, trusting, kind, and warm.</td>
</tr>
<tr>
<td>Neuroticism</td>
<td>Being anxious, irritable, temperamental, and moody.</td>
</tr>
</tbody>
</table>

https://new.edu/resources/big-5-personality-traits
Principal component analysis (PCA) is a linear latent-factor model:

- These models “factorize” matrix $X$ into matrices $Z$ and $W$:
  \[ X \sim Z W \]
  \[ x_i \sim W^\top z_i \]
  \[ x_{ij} \approx <w_j^\top z_i> \]

- We can think of rows $w_c$ of $W$ as ‘$k$’ fixed “part” (used in all examples).
- $z_i$ is the “part weights” for example $x_i$: “how much of each part $w_c$ to use”.

\[ \hat{x}_i = z_{i1} w_1 + z_{i2} w_2 + z_{i3} w_3 + z_{i4} w_4 + z_{i5} w_5 + z_{i6} w_6 + z_{i7} w_7 \]

"parts" are called "principal components" or "factors"
Last Time: PCA Geometry

• When \( k=1 \), the \( W \) matrix defines a line:
  – We choose ‘\( W \)’ as the line minimizing squared distance to the data.
  – Given ‘\( W \)’, the \( z_i \) are the coordinates of the \( x_i \) “projected” onto the line.
Last Time: PCA Geometry

- When $k=2$, the $W$ matrix defines a plane:
  - We choose ‘$W$’ as the plane minimizing squared distance to the data.
  - Given ‘$W$’, the $z_i$ are the coordinates of the $x_i$ “projected” onto the plane.

http://www.nlpca.org/fig_pca_principal_component_analysis.png
Last Time: PCA Geometry

- When $k=2$, the $W$ matrix defines a plane:
  - Even if the original data is high-dimensional, we can visualize data “projected” onto this plane.
PCA Objective Function

• In PCA we minimize the squared error of the approximation:

\[ \hat{f}(W, z) = \sum_{i=1}^{\hat{\delta}} \| W^\top z_i - x_i \|^2 \]

• This is equivalent to the k-means objective:
  – In k-means, \( z_i \) only has a single ‘1’ value and other entries are zero.

• But in PCA, \( z_i \) can be any real number.
  – We approximate \( x_i \) as a linear combination of all means/factors.
PCA Objective Function

• In PCA we minimize the squared error of the approximation:

\[ f(W, Z) = \sum_{i=1}^{n} \| W^T z_i - x_i \|^2 = \sum_{i=1}^{n} \sum_{j=1}^{d} (\langle w_j^i, z_i \rangle - x_{ij})^2 \]

• We can also view this as solving ‘d’ regression problems:
  – Each \( w_j^i \) is trying to predict column ‘j’ of ‘X’ from the basis \( z_i \).
    • The output “\( y_i \)” we try to predict here is actually the features “\( x_i \)”.
    • So we have ‘d’ sums inside the sum over ‘n’.

  – And we are also learning the features \( z_i \).
    • Each \( z_i \) say how to mix the mean/factor \( w_c \) to approximation example ‘i’.
Principal Component Analysis (PCA)

• Different ways to write the PCA objective function:

\[ f(W,Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (\langle w_j z_i \rangle - x_{ij})^2 \]

\[ = \sum_{i=1}^{n} \| W^T z_i - x_i \|_2^2 \]

\[ = \| Z W - X \|_F^2 \]

(approximating \( x_{ij} \) by \( \langle w_j z_i \rangle \))

(approximating \( x_i \) by \( W^T z_i \))

(approximating \( X \) by \( Z W \))

• We’re picking \( Z \) and \( W \) to approximate the original data \( X \).
  – It won’t be perfect since usually \( k \) is much smaller than \( d \).
Digression: Data Centering (Important)

• In PCA, we assume that the data X is “centered”.
  – Each column of X has a mean of zero.

• It’s easy to center the data:

  \[ \text{Set } \mu_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \quad \text{(mean of column 'j')} \]

  Replace each \( x_{ij} \) with \( (x_{ij} - \mu_j) \)

• There are PCA variations that estimate “bias in each coordinate”.
  – In basic model this is equivalent to centering the data.
PCA Computation: Prediction

• At the end of training, the “model” is the $\mu_j$ and the $W$ matrix.
  – PCA is parametric.

• PCA prediction phase:
  – Given new data $\tilde{X}$, we can use $\mu_j$ and $W$ this to form $\tilde{Z}$:
    
    1. Center: replace each $\tilde{x}_{ij}$ with $(\tilde{x}_{ij} - \mu_j)$
    2. Find $\tilde{Z}$ minimizing squared error:
       $$\tilde{Z} = \tilde{X} W^{\top} (W W^{\top})^{-1}$$

   (means of training data)
   (could just store this $d \times k$ matrix)
PCA Computation: Prediction

• At the end of training, the “model” is the $\mu_j$ and the $W$ matrix.
  – PCA is parametric.

• PCA prediction phase:
  – Given new data $\tilde{X}$, we can use $\mu_j$ and $W$ this to form $\tilde{Z}$:
  – The “reconstruction error” is how close approximation is to $\tilde{X}$:
    $$\| \tilde{Z} W - \tilde{X} \|_F^2$$
  – Our “error” from replacing the $x_i$ with the $z_i$ and $W$.
  – Notice that this means that PCA is parametric (don’t need ‘Z’ at test time).
(pause)
Non-Uniqueness of PCA

• Unlike k-means, we can efficiently find global optima of $f(W,Z)$.
  – Algorithms coming later.

• Unfortunately, there never a unique global optimum.
  – There are actually several different sources of non-uniqueness.

• To understand these, we’ll need idea of “span” from linear algebra.
  – This also helps explain the geometry of PCA.
  – We’ll also see that some global optima may be better than others.
Span of 1 Vector

• Consider a single vector $w_1$ (k=1).
Span of 1 Vector

- Consider a single vector $w_1$ (k=1).
- The $\text{span}(w_1)$ is all vectors of the form $z_iw_1$ for a scalar $z_i$. 
Span of 1 Vector

• Consider a single vector \( w_1 \) (k=1).
• The \( \text{span}(w_1) \) is all vectors of the form \( z_i w_1 \) for a scalar \( z_i \).
• If \( w_1 \neq 0 \), this forms a line.
Span of 1 Vector

• But note that the “span” of many different vectors gives same line.
  – Mathematically: $\alpha w_1$ defines the same line as $w_1$ for any scalar $\alpha \neq 0$.
  
  – PCA solution can only be defined up to scalar multiplication.

  • If $(W,Z)$ is a solution, then $(\alpha W, (1/\alpha)Z)$ is also a solution.

\[ \| (\alpha W)(\frac{1}{\alpha}Z) - X \|_F^2 = \| W - X \|_F^2 \]
Span of 2 Vectors

• Consider two vector $w_1$ and $w_2$ ($k=2$).
Span of 2 Vectors

- Consider **two vector** $w_1$ and $w_2$ ($k=2$).
  - The span($w_1, w_2$) is all vectors of form $z_{i1}w_1 + z_{i2}w_2$ for scalars $z_{i1}$ and $z_{i2}$. 
Span of 2 Vectors

• Consider two vector \( w_1 \) and \( w_2 \) (k=2).
  – The \( \text{span}(w_1, w_2) \) is all vectors of form \( z_{i1}w_1 + z_{i2}w_2 \) for a scalars \( z_{i1} \) and \( z_{i2} \).
Span of 2 Vectors

• Consider two vector \( w_1 \) and \( w_2 \) (k=2).
  – The \( \text{span}(w_1,w_2) \) is all vectors of form \( z_{i1}w_1 + z_{i2}w_2 \) for scalars \( z_{i1} \) and \( z_{i2} \).
  – For most non-zero 2d vectors, \( \text{span}(w_1,w_2) \) is a plane.
    • In the case of two vectors in \( \mathbb{R}^2 \), the plane will be *all* of \( \mathbb{R}^2 \).
Consider two vectors $w_1$ and $w_2$ ($k=2$).

- The span($w_1, w_2$) is all vectors of form $z_{i1}w_1 + z_{i2}w_2$ for scalars $z_{i1}$ and $z_{i2}$.

- For most non-zero 2d vectors, span($w_1, w_2$) is a plane.

  - Exception is if $w_2$ is in span of $w_1$ ("collinear"), then span($w_1, w_2$) is just a line.
Span of 2 Vectors

• Consider two vector $w_1$ and $w_2$ (k=2).
  – The $\text{span}(w_1, w_2)$ is all vectors of form $z_{i1}w_1 + z_{i2}w_2$ for scalars $z_{i1}$ and $z_{i2}$.

– New issues for PCA (k $\geq$ 2):
  • We have label switching: $\text{span}(w_1, w_2) = \text{span}(w_2, w_1)$.
  • We can rotate factors within the plane (if not rotated to be collinear).
Span of 2 Vectors

- 2 tricks to make vectors defining a plane “more unique”:
  - Normalization: enforce that $||w_1|| = 1$ and $||w_2|| = 1$. 
Span of 2 Vectors

• 2 tricks to make vectors defining a plane “more unique”:
  – **Normalization**: enforce that \( ||w_1|| = 1 \) and \( ||w_2|| = 1 \).
Span of 2 Vectors

- 2 tricks to make vectors defining a plane “more unique”:
  - **Normalization**: enforce that $||w_1|| = 1$ and $||w_2|| = 1$.
  - **Orthogonality**: enforce that $w_1^T w_2 = 0$ (“perpendicular”).

- Now I can’t grow/shrink vectors (though I can still reflect).
- Now I can’t rotate one vector (but I can still rotate *both*).
Digression: PCA only makes sense for $k \leq d$

- Remember our clustering dataset with 4 clusters:
  - It doesn’t make sense to use PCA with $k=4$ on this dataset.
    - We only need two vectors $[1 \ 0]$ and $[0 \ 1]$ to exactly represent all 2d points.
      - With $k=2$, I could just set $Z=X$ and $W=I$ to get $ZW=X$ (error of 0).
Span in Higher Dimensions

• In higher-dimensional spaces:
  – Span of 1 non-zero vector $w_1$ is a line.
  – Span of 2 non-zero vectors $w_1$ and $w_2$ is a plane (if not collinear).
    • Can be visualized as a 2D plot.
  – Span of 3 non-zeros vectors \(\{w_1, w_2, w_3\}\) is a 3d space (if not “coplanar”).
  – ...

• This is how the W matrix in PCA defines lines, planes, spaces, etc.
  – Each time we increase ‘k’, we add an extra “dimension” to the “subspace”.
Making PCA Unique

• We’ve identified several reasons that optimal $W$ is non-unique:
  – I can multiply any $w_c$ by any non-zero $\alpha$.
  – I can rotate any $w_c$ almost arbitrarily within the span.
  – I can switch any $w_c$ with any other $w_{c'}$.

• PCA implementations add constraints to make solution unique:
  – **Normalization**: we enforce that $||w_c|| = 1$.
  – **Orthogonality**: we enforce that $w_c^T w_{c'} = 0$ for all $c \neq c'$.
  – **Sequential fitting**: We first fit $w_1$ (“first principal component”) giving a line.
    • Then fit $w_2$ given $w_1$ (“second principal component”) giving a plane.
    • Then we fit $w_3$ given $w_1$ and $w_2$ (“third principal component”) giving a space.
Basis, Orthogonality, Sequential Fitting

optimal solution with one PC
Any non-parallel line gives optimal solution to second PC (when $d=2$).

I can get 0 error on every data point.

An optimal solution but not orthogonal. (both PCs give similar information)

Optimal solution with one PC.
Any non-parallel line gives optimal solution to second PC (when d=2). I can get 0 error on every data point.

An orthogonal solution (PCs are not redundant) but PCs have nothing to do with data.
Basis, Orthogonality, Sequential Fitting

http://setosa.io/ev/principal-component-analysis
PCA Computation: SVD

• How do we fit with normalization/orthogonality/sequential-fitting?
  – It can be done with the “singular value decomposition” (SVD).
  – Take CPSC 302.

• 4 lines of Python code:
  – mu = np.mean(X, axis=0)
  – X -= mu
  – U, s, Vh = np.linalg.svd(X)
  – W = Vh[:k]

• Computing Z is cheaper now:
  \[ Z = X W^\top (W W^\top)^{-1} = X W^\top \]
  \[ W W^\top = \begin{bmatrix} -w_1 \\ -w_2 \\ \vdots \\ -w_k \end{bmatrix} \begin{bmatrix} w_1^\top \\ w_2^\top \\ \vdots \\ w_k^\top \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = I \]
PCA Computation: other methods

• With **linear regression**, we had the **normal equations**
  – But we also could do it with gradient descent, SGD, etc.

• With **PCA** we have the **SVD**
  – But we can also do it with gradient descent, SGD, etc.

– These other methods typically don’t enforce the uniqueness “constraints”.
  • Sensitive to initialization, don’t enforce normalization, orthogonality, ordered PCs.
    – But you can do this in post-processing if you want.

– Why would we want this? We use our tricks from Part 3 of the course:
  • We can do things like “robust” PCA, “regularized” PCA, “sparse” PCA, “binary” PCA.
  • We can fit huge datasets where SVD is too expensive.
PCA Computation: Alternating Minimization

• With centered data, the **PCA objective** is:

\[ f(W, Z) = \sum_{i=1}^{\hat{n}} \sum_{j=1}^{d} (\langle w_j^i z_i^j \rangle - x_{ij})^2 \]

• In **k-means** we tried to optimize this with **alternating minimization**:
  – Fix “cluster assignments” Z and find the optimal “means” W.
  – Fix “means” W and find the optimal “cluster assignments” Z.

• Converges to a local optimum.
  – But may not find a global optimum (sensitive to initialization).
PCA Computation: Alternating Minimization

• With centered data, the PCA objective is:
  \[ f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} \left( \langle w_j, z_i \rangle - x_{ij} \right)^2 \]

• In PCA we can also use alternating minimization:
  – Fix “part weights” Z and find the optimal “parts” W.
  – Fix “parts” W and find the optimal “part weights” Z.

• Converges to a local optimum.
  – Which will be a global optimum (if we randomly initialize W and Z).
PCA Computation: Alternating Minimization

• With centered data, the **PCA objective** is:

\[
\mathcal{f}(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (\langle w_j^i, z_i \rangle - x_{ij})^2
\]

• **Alternating minimization** steps:
  - If we fix \( Z \), this is a quadratic function of \( W \) (least squares column-wise):
    \[
    \nabla_W \mathcal{f}(W, Z) = Z^T Z W - Z^T X
    \]
    
    \( \text{(writing gradient as a matrix)} \)
    
    so
    \[
    W = (Z^T Z)^{-1} (Z^T X)
    \]
  - If we fix \( W \), this is a quadratic function of \( Z \) (transpose due to dimensions):
    \[
    \nabla_Z \mathcal{f}(W, Z) = Z W W^T - X W^T
    \]
    
    so
    \[
    Z = X W^T (W W^T)^{-1}
    \]
PCA Computation: Alternating Minimization

• With centered data, the PCA objective is:

\[ f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (\langle w_j, z_i \rangle - x_{ij})^2 \]

• This objective is not jointly convex in \( W \) and \( Z \).
  – You will find different \( W \) and \( Z \) depending on the initialization.
    • For example, if you initialize with \( W_1 = 0 \), then they will stay at zero.

  – But it’s possible to show that all “stable” local optima are global optima.
    • You will converge to a global optimum in practice if you initialize randomly.
      – Randomization means you don’t start on one of the unstable non-global critical points.
    • E.g., sample each initial \( z_{ij} \) from a normal distribution.
PCA Computation: Stochastic Gradient

- For big X matrices, you can also use \textit{stochastic gradient}:

\[
f(W, z) = \sum_{i=1}^{d} \sum_{j=1}^{d} (\langle w_j, z_i \rangle - x_{ij})^2 = \sum_{i,j} (\langle w_j, z_i \rangle - x_{ij})^2 \]

\[
f(\hat{w}_j, \gamma, \hat{z}_i, x_{ij})
\]

On each iteration, pick a random example \(i\) and feature \(j\):

- Set \(w^j \) to \(w^j - \alpha^t \nabla_{w^j} f(w^j, z, x_{ij})\)

- Set \(z_i \) to \(z_i - \alpha^t \nabla_{z_i} f(w^j, z_i, x_{ij})\)

- Other variables stay the same, cost per iteration is only \(O(k)\).
Summary

• **PCA objective:**
  – Minimizes squared error between elements of $X$ and elements of $ZW$.

• **PCA non-uniqueness:**
  – Due to scaling, rotation, and label switching.

• **Orthogonal basis and sequential fitting of PCs (via SVD):**
  – Leads to non-redundant PCs with unique directions.

• **Alternating minimization and stochastic gradient:**
  – Iterative algorithms for minimizing PCA objective.

• Next time: cancer signatures and NBA shot charts.
Making PCA Unique

- PCA implementations add constraints to make solution unique:
  - **Normalization**: we enforce that $||w_c|| = 1$.
  - **Orthogonality**: we enforce that $w_c^T w_{c'} = 0$ for all $c \neq c'$.
  - **Sequential fitting**: We first fit $w_1$ (“first principal component”) giving a line.
    - Then fit $w_2$ given $w_1$ (“second principal component”) giving a plane.
    - Then we fit $w_3$ given $w_1$ and $w_2$ (“third principal component”) giving a space.
    - ...

- Even with all this, the solution is only unique up to sign changes:
  - I can still replace any $w_c$ by $-w_c$:
    - $-w_c$ is normalized, is orthogonal to the other $w_{c'}$, and spans the same space.
  - Possible fix: require that first non-zero element of each $w_c$ is positive.