PCA
Reminders

• Homework 8: Reinforcement Learning
  – Out: Fri, Apr 10
  – Due: Wed, Apr 22 at 11:59pm

• Homework 9: Learning Paradigms
  – Out: Wed, Apr. 22
  – Due: Wed, Apr. 29 at 11:59pm
  – Can only be submitted up to 3 days late, so we can return grades before final exam

• Today’s In-Class Poll
  – http://poll.mlcourse.org
ML Big Picture

Learning Paradigms:
What data is available and when? What form of prediction?
- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction

Problem Formulation:
What is the structure of our output prediction?
- boolean Binary Classification
- categorical Multiclass Classification
- ordinal Ordinal Classification
- real Regression
- ordering Ranking
- multiple discrete Structured Prediction
- multiple continuous (e.g. dynamical systems)
- both discrete & (e.g. mixed graphical models) cont.

Facets of Building ML Systems:
How to build systems that are robust, efficient, adaptive, effective?
1. Data prep
2. Model selection
3. Training (optimization / search)
4. Hyperparameter tuning on validation data
5. (Blind) Assessment on test data

Theoretical Foundations:
What principles guide learning?
- probabilistic
- information theoretic
- evolutionary search
- ML as optimization

Big Ideas in ML:
Which are the ideas driving development of the field?
- inductive bias
- generalization / overfitting
- bias-variance decomposition
- generative vs. discriminative
- deep nets, graphical models
- PAC learning
- distant rewards

Application Areas:
- Key challenges?
- NLP, Speech, Computer Vision, Robotics, Medicine, Search
# Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>( \mathcal{D} = { x^{(i)}, y^{(i)} }_{i=1}^N ) \quad ( x \sim p^<em>(\cdot) ) and ( y = c^</em>(\cdot) )</td>
</tr>
<tr>
<td>Regression</td>
<td>( y^{(i)} \in \mathbb{R} )</td>
</tr>
<tr>
<td>Classification</td>
<td>( y^{(i)} \in {1, \ldots, K} )</td>
</tr>
<tr>
<td>Binary classification</td>
<td>( y^{(i)} \in {+1, -1} )</td>
</tr>
<tr>
<td>Structured Prediction</td>
<td>( y^{(i)} ) is a vector</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>( \mathcal{D} = { x^{(i)} }_{i=1}^N ) \quad x \sim p^*(\cdot)</td>
</tr>
<tr>
<td>Clustering</td>
<td>predict ( { z^{(i)} }_{i=1}^N ) where ( z^{(i)} \in {1, \ldots, K} )</td>
</tr>
<tr>
<td>Dimensionality Reduction</td>
<td>convert each ( x^{(i)} \in \mathbb{R}^M ) to ( u^{(i)} \in \mathbb{R}^K ) with ( K &lt;&lt; M )</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>( \mathcal{D} = { x^{(i)}, y^{(i)} }<em>{i=1}^{N_1} \cup { x^{(j)} }</em>{j=1}^{N_2} )</td>
</tr>
<tr>
<td>Online</td>
<td>( \mathcal{D} = {(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots } )</td>
</tr>
<tr>
<td>Active Learning</td>
<td>( \mathcal{D} = { x^{(i)} }_{i=1}^N ) and can query ( y^{(i)} = c^*(\cdot) ) at a cost</td>
</tr>
<tr>
<td>Imitation Learning</td>
<td>( \mathcal{D} = {(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots } )</td>
</tr>
<tr>
<td>Reinforcement Learning</td>
<td>( \mathcal{D} = {(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots } )</td>
</tr>
</tbody>
</table>
DIMENSIONALITY REDUCTION
PCA Outline

• **Dimensionality Reduction**
  – High-dimensional data
  – Learning (low dimensional) representations

• **Principal Component Analysis (PCA)**
  – Examples: 2D and 3D
  – Data for PCA
  – PCA Definition
  – Objective functions for PCA
  – PCA, Eigenvectors, and Eigenvalues
  – Algorithms for finding Eigenvectors / Eigenvalues

• **PCA Examples**
  – Face Recognition
  – Image Compression
High Dimension Data

Examples of high dimensional data:
  – High resolution images (millions of pixels)
High Dimension Data

Examples of high dimensional data:

– Multilingual News Stories
  (vocabulary of hundreds of thousands of words)
High Dimension Data

Examples of high dimensional data:

– Brain Imaging Data (100s of MBs per scan)

Image from (Wehbe et al., 2014)
High Dimension Data

Examples of high dimensional data:
– Customer Purchase Data
Learning Representations

PCA, Kernel PCA, ICA: Powerful unsupervised learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.

Useful for:

- Visualization
- More efficient use of resources (e.g., time, memory, communication)
- Statistical: fewer dimensions $\rightarrow$ better generalization
- Noise removal (improving data quality)
- Further processing by machine learning algorithms

Slide from Nina Balcan
Shortcut Example

https://www.youtube.com/watch?v=MIJN9pEfPfE

Photo from https://www.springcarnival.org/booth.shtml
PRINCIPAL COMPONENT ANALYSIS (PCA)
PCA Outline

• **Dimensionality Reduction**
  – High-dimensional data
  – Learning (low dimensional) representations

• **Principal Component Analysis (PCA)**
  – Examples: 2D and 3D
  – Data for PCA
  – PCA Definition
  – Objective functions for PCA
  – PCA, Eigenvectors, and Eigenvalues
  – Algorithms for finding Eigenvectors / Eigenvalues

• **PCA Examples**
  – Face Recognition
  – Image Compression
Principal Component Analysis (PCA)

In case where data lies on or near a low d-dimensional linear subspace, axes of this subspace are an effective representation of the data.

Identifying the axes is known as **Principal Components Analysis**, and can be obtained by using classic matrix computation tools (Eigen or Singular Value Decomposition).

Slide from Nina Balcan
2D Gaussian dataset

Slide from Barnabas Poczos
$1^{\text{st}}$ PCA axis

Slide from Barnabas Poczos
2\textsuperscript{nd} PCA axis

Slide from Barnabas Poczos
### Data for PCA

\[ D = \{ x^{(i)} \}_{i=1}^{N} \quad x^{(i)} \in \mathbb{R}^M \]

\[ X = \begin{bmatrix}
(x^{(1)})^T \\
(x^{(2)})^T \\
\vdots \\
(x^{(N)})^T
\end{bmatrix} \]

We assume the data is **centered**

\[ \mu = \frac{1}{N} \sum_{i=1}^{N} x^{(i)} = 0 \]

**Q:** What if your data is **not** centered?

**A:** Subtract off the sample mean
The sample covariance matrix is given by:

\[
\Sigma_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_{j}^{(i)} - \mu_{j}) (x_{k}^{(i)} - \mu_{k})
\]

\[\Sigma \in \mathbb{R}^{M \times M}\]

Since the data matrix is centered, we rewrite as:

\[
\Sigma = \frac{1}{N} \mathbf{X}^T \mathbf{X}
\]
Principal Component Analysis (PCA)

Whiteboard

- Strawman: random linear projection
- PCA Definition
- Objective functions for PCA
Maximizing the Variance

Quiz: Consider the two projections below

1. Which maximizes the variance?  
2. Which minimizes the reconstruction error?

Option A

Option B
Background: Eigenvectors & Eigenvalues

For a square matrix $A$ (n x n matrix), the vector $v$ (n x 1 matrix) is an eigenvector iff there exists eigenvalue $\lambda$ (scalar) such that:

$$Av = \lambda v$$

The linear transformation $A$ is only stretching vector $v$.

That is, $\lambda v$ is a scalar multiple of $v$. 
Principal Component Analysis (PCA)

Whiteboard

– PCA, Eigenvectors, and Eigenvalues
PCA

Equivalence of Maximizing Variance and Minimizing Reconstruction Error

Claim: Minimizing the reconstruction error is equivalent to maximizing the variance.

Proof: First, note that:

$$||x^{(i)} - (v^T x^{(i)})v||^2 = ||x^{(i)}||^2 - (v^T x^{(i)})^2$$ (1)

since $v^T v = ||v||^2 = 1$.

Substituting into the minimization problem, and removing the extraneous terms, we obtain the maximization problem.

$$v^* = \operatorname{argmin}_{v:||v||^2=1} \frac{1}{N} \sum_{i=1}^{N} ||x^{(i)} - (v^T x^{(i)})v||^2$$ (2)

$$= \operatorname{argmin}_{v:||v||^2=1} \frac{1}{N} \sum_{i=1}^{N} ||x^{(i)}||^2 - (v^T x^{(i)})^2$$ (3)

$$= \operatorname{argmax}_{v:||v||^2=1} \frac{1}{N} \sum_{i=1}^{N} (v^T x^{(i)})^2$$ (4)
PCA: the First Principal Component

To find the first principal component, we wish to solve the following constrained optimization problem (variance minimization).

$$v_1 = \arg\max_{v: \|v\|^2 = 1} v^T \Sigma v$$  \hspace{1cm} (1)

So we turn to the method of Lagrange multipliers. The Lagrangian is:

$$\mathcal{L}(v, \lambda) = v^T \Sigma v - \lambda (v^T v - 1)$$  \hspace{1cm} (2)

Taking the derivative of the Lagrangian and setting to zero gives:

$$\frac{d}{dv} (v^T \Sigma v - \lambda (v^T v - 1)) = 0$$  \hspace{1cm} (3)

$$\Sigma v - \lambda v = 0$$  \hspace{1cm} (4)

$$\Sigma v = \lambda v$$  \hspace{1cm} (5)

Recall: For a square matrix $A$, the vector $v$ is an eigenvector iff there exists eigenvalue $\lambda$ such that:

$$Av = \lambda v$$  \hspace{1cm} (6)
Algorithms for PCA

How do we find principal components (i.e. eigenvectors)?

• Power iteration (aka. Von Mises iteration)
  – finds each principal component one at a time in order

• Singular Value Decomposition (SVD)
  – finds all the principal components at once
  – two options:
    • Option A: run SVD on $X^TX = \Sigma$
    • Option B: run SVD on $X$
      (not obvious why Option B should work…)

• Stochastic Methods (approximate)
  – very efficient for high dimensional datasets with lots of points
Background: SVD

Singular Value Decomposition (SVD)

For any arbitrary matrix $\mathbf{A}$, SVD gives a decomposition:

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T$$  \hspace{1cm} (1)

where $\boldsymbol{\Lambda}$ is a diagonal matrix, and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices.
SVD for PCA

For any arbitrary matrix $A$, SVD gives a decomposition:

$$A = U\Lambda V^T$$

(1)

where $\Lambda$ is a diagonal matrix, and $U$ and $V$ are orthogonal matrices.

Suppose we obtain an SVD of our data matrix $X$, so that:

$$X = U\Lambda V^T$$

(1)

Now consider what happens when we rewrite $\Sigma = \frac{1}{N}X^TX$ terms of this SVD.

$$\Sigma = \frac{1}{N}X^TX$$

(2)

$$= \frac{1}{N}(U\Lambda V^T)^T(U\Lambda V^T)$$

(3)

$$= \frac{1}{N}(V\Lambda U^T)(U\Lambda V^T)$$

(4)

$$= \frac{1}{N}V\Lambda^T\Lambda V^T$$

(5)

$$= \frac{1}{N}V(\Lambda)^2V^T$$

(6)

Above we used the fact that $U^TU = I$ since $U$ is orthogonal by definition.

We find that $(\Lambda)^2$ is a diagonal matrix whose entries are $\Lambda_{ii} = \lambda_i^2$ the squares of the eigenvalues of the SVD of $X$. Further, both $X$ and $X^TX$ share the same eigenvectors in their SVD.

Thus, we can run SVD on $X$ without ever instantiating the large $X^TX$ to obtain the necessary principal components more efficiently.
Principal Component Analysis (PCA)

\[(X X^T)v = \lambda v\], so \(v\) (the first PC) is the eigenvector of sample correlation/covariance matrix \(X X^T\)

Sample variance of projection \(v^T X X^T v = \lambda v^T v = \lambda\)

Thus, the eigenvalue \(\lambda\) denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

Eigenvalues \(\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots\)

- The 1st PC \(v_1\) is the eigenvector of the sample covariance matrix \(X X^T\) associated with the largest eigenvalue.
- The 2nd PC \(v_2\) is the eigenvector of the sample covariance matrix \(X X^T\) associated with the second largest eigenvalue.
- And so on ...
For $M$ original dimensions, sample covariance matrix is $M \times M$, and has up to $M$ eigenvectors. So $M$ PCs.

Where does dimensionality reduction come from? Can ignore the components of lesser significance.

You do lose some information, but if the eigenvalues are small, you don’t lose much
- $M$ dimensions in original data
- calculate $M$ eigenvectors and eigenvalues
- choose only the first $D$ eigenvectors, based on their eigenvalues
- final data set has only $D$ dimensions

Variance ($\%$) = ratio of variance along given principal component to total variance of all principal components
PCA EXAMPLES
Projecting MNIST digits

Task Setting:

1. Take 25x25 images of digits and project them down to K components
2. Report percent of variance explained for K components
3. Then project back up to 25x25 image to visualize how much information was preserved
Projecting MNIST digits

Task Setting:
1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points
3. Here we look at all ten digits 0 - 9
Projecting MNIST digits

Task Setting:
1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points
3. Here we look at just four digits 0, 1, 2, 3
Learning Objectives

Dimensionality Reduction / PCA

You should be able to...
1. Define the sample mean, sample variance, and sample covariance of a vector-valued dataset
2. Identify examples of high dimensional data and common use cases for dimensionality reduction
3. Draw the principal components of a given toy dataset
4. Establish the equivalence of minimization of reconstruction error with maximization of variance
5. Given a set of principal components, project from high to low dimensional space and do the reverse to produce a reconstruction
6. Explain the connection between PCA, eigenvectors, eigenvalues, and covariance matrix
7. Use common methods in linear algebra to obtain the principal components