Principal Component Analysis

4-8-2016
PCA: the setting

Unsupervised learning

- Unlabeled data

Dimensionality reduction

- Simplify the data representation
Change of basis examples so far

Support vector machines

- Data that's not linearly separable in the standard basis may be (approximately) linearly separable in a transformed basis.
- The kernel trick sometimes lets us work with high-dimensional bases.

Approximate Q-learning

- When the state space is too large for Q-learning, we may be able to extract features that summarize the state space well.
- We then learn values as a linear function of the transformed representation.
Change of basis in PCA

This looks like the change of basis from linear algebra.

- PCA performs an affine transformation of the original basis.
  - Affine $≡$ linear plus a constant

The goal:

- find a new basis where most of the variance in the data is along the axes.
- Hopefully only a small subset of the new axes will be important.
PCA change of basis illustrated
PCA: step one

First step: center the data.

- From each dimension, subtract the mean value of that dimension.
- This is the "plus a constant" part, afterwards we'll perform a linear transformation.
- The centroid is now a vector of zeros.
PCA: step two

The hard part: find an orthogonal basis that's a linear transformation of the original, where the variance in the data is explained by as few dimensions as possible.

- Orthogonal basis: all axes are perpendicular.
- Linear transformation of a basis: rotate \((m - 1)\) angles
- Explaining the variance: data varies a lot along some axes, but much less along others.
PCA: step three

Last step: reduce the dimension.

- Sort the dimensions of the new basis by how much the data varies.
- Throw away some of the less-important dimensions.
  - Could keep a specific number of dimensions.
  - Could keep all dimensions with variance above some threshold.
- This results in a projection into the subspace of the remaining axes.
Computing PCA: step two

● Construct the covariance matrix.
  ○ $m \times m$ ($m$ is the number of dimensions) matrix.
  ○ Diagonal entries give variance along each dimension.
  ○ Off-diagonal entries give cross-dimension covariance.

● Perform eigenvalue decomposition on the covariance matrix.
  ○ Compute the eigenvectors/eigenvalues of the covariance matrix.
  ○ Use the eigenvectors as the new basis.
Covariance matrix example

$$\mathbf{C} = \frac{1}{n} (\mathbf{X}(\mathbf{X}^T))$$

<table>
<thead>
<tr>
<th>data</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$X$</th>
<th>$X^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
<td>-4</td>
<td>1</td>
<td>2</td>
<td></td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-5</td>
<td></td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>-2</td>
<td>6</td>
<td>-7</td>
<td>6</td>
<td>-3</td>
<td></td>
<td>-2</td>
<td>6</td>
</tr>
</tbody>
</table>

| | 4 | 8 | -2 |
| | 3 | 0 | 6 |
| | 7.8 | 3.2 | 8 |
| | 3.2 | 18.8 | -1.2 |
| | 8 | -1.2 | 26.8 |
Linear algebra review: eigenvectors

Eigenvectors are vectors that the matrix doesn’t rotate.

If $X$ is a matrix, and $v$ is a vector, then $v$ is an eigenvector of $X$ iff there is some constant $\lambda$, such that:

$$Xv = \lambda v$$

$\lambda$, the amount by which $X$ stretches the eigenvector is the eigenvalue.
Linear algebra review: eigenvalue decomposition

If the matrix $(X)(X^T)$ has eigenvectors $\mathbf{v}_i$ with eigenvalues $\lambda_i$ for $i \in \{1, \ldots, m\}$, then the following vectors form an orthonormal basis:

$$\frac{X^T \mathbf{v}_i}{\sqrt{\lambda_i}}$$

The key point: computing the eigenvectors of the covariance matrix gives us the optimal (linear) basis for explaining the variance in our data.

Sorting by eigenvalue $\lambda_i$ tells us the relative importance of each dimension.
PCA change of basis illustrated
When does PCA fail?
Exam questions

Topics coming later today.

Lectures since the last exam:
machine learning intro
decision trees
perceptrons
backpropagation
analyzing backprop
naive Bayes
k nearest neighbors
support vector machines
value iteration

Q-learning
approximate Q-learning
MCTS for MDPs
POMDPs
particle filters
hierarchical clustering
EM, k-means, and GNG
principal component analysis