



Reducing Data Dimension

Required reading:

- “A Tutorial on PCA,’ J. Schlens,
<http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf>

Machine Learning 10-601

April 7, 2008

Tom M. Mitchell

Machine Learning Department

Carnegie Mellon University



Outline

- Feature selection
 - Single feature scoring criteria
 - Search strategies
- Unsupervised dimension reduction using all features
 - Principle Components Analysis
 - Singular Value Decomposition
 - Independent components analysis
- Supervised dimension reduction
 - Fisher Linear Discriminant
 - Hidden layers of Neural Networks



Dimensionality Reduction

Why?

- Learning a target function from data where some features are irrelevant - reduce variance, improve accuracy
- Wish to visualize high dimensional data
- Sometimes have data whose “intrinsic” dimensionality is smaller than the number of features used to describe it - recover intrinsic dimension



Supervised Feature Selection

Supervised Feature Selection

Problem: Wish to learn $f: X \rightarrow Y$, where $X = \langle X_1, \dots, X_N \rangle$

But suspect not all X_i are relevant

Approach: Preprocess data to select only a subset of the X_i

- Score each feature, or subsets of features
 - How?
- Search for useful subset of features to represent data
 - How?

Scoring Individual Features X_i

Common scoring methods:

- Training or cross-validated accuracy of single-feature classifiers $f_i: X_i \rightarrow Y$

- Estimated mutual information between X_i and Y :

$$\hat{I}(X_i, Y) = \sum_k \sum_y \hat{P}(X_i = k, Y = y) \log \frac{\hat{P}(X_i = k, Y = y)}{\hat{P}(X_i = k) \hat{P}(Y = y)}$$

- χ^2 statistic to measure independence between X_i and Y
- Domain specific criteria
 - Text: Score “stop” words (“the”, “of”, ...) as zero
 - fMRI: Score voxel by T-test for activation versus rest condition
 - ...

Choosing Set of Features to learn $F: X \rightarrow Y$

Common methods:

Forward1: Choose the n features with the highest scores

Forward2:

- Choose single highest scoring feature X_k
- Rescore all features, conditioned on the set of already-selected features
 - E.g., $\text{Score}(X_i | X_k) = I(X_i, Y | X_k)$
 - E.g., $\text{Score}(X_i | X_k) = \text{Accuracy}(\text{predicting } Y \text{ from } X_i \text{ and } X_k)$
- Repeat, calculating new scores on each iteration, conditioning on set of selected features

Choosing Set of Features

Common methods:

Backward1: Start with all features, delete the n with lowest scores

Backward2: Start with all features, score each feature conditioned on assumption that all others are included.

Then:

- Remove feature with the lowest (conditioned) score
- Rescore all features, conditioned on the new, reduced feature set
- Repeat

Feature Selection: Text Classification

Approximately 10^5 words in English

[Rogati&Yang, 2002]

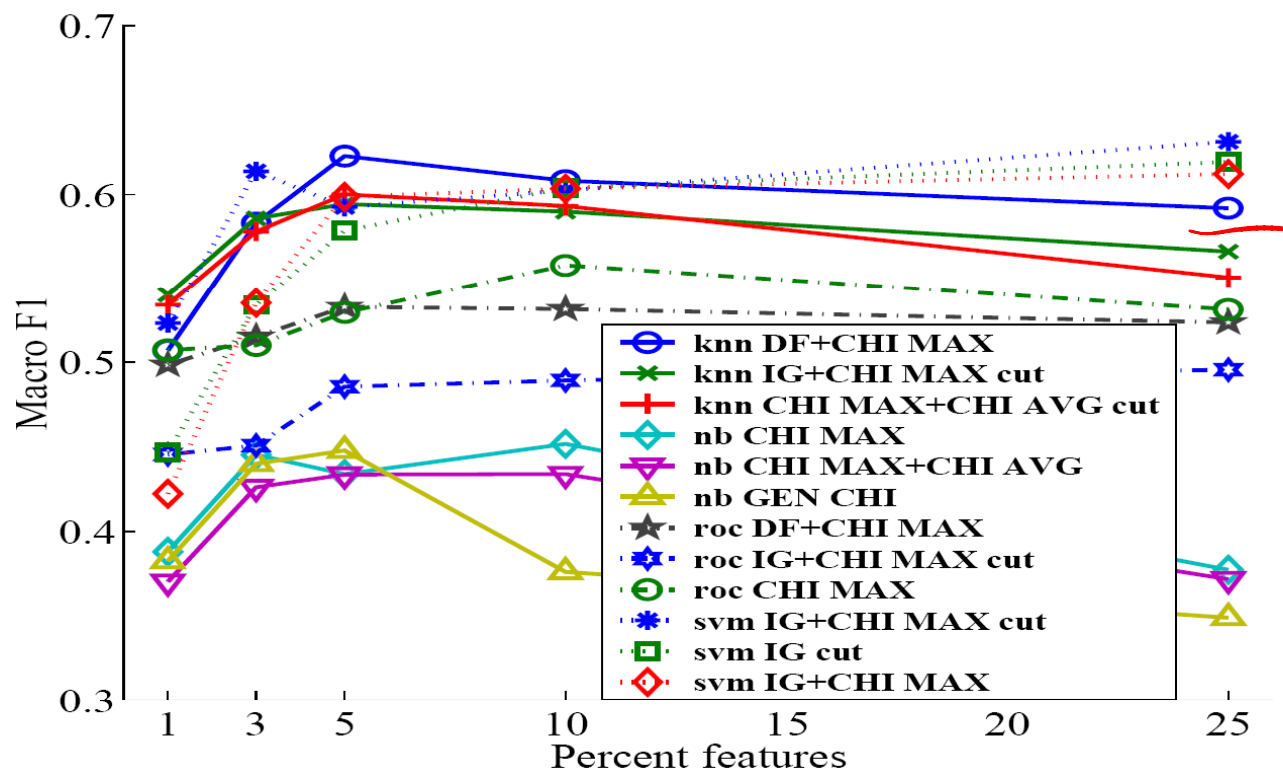


Figure 2: Top 3 feature selection methods for Reuters-21578 (Macro F1)

IG=information gain, $\text{chi} = \chi^2$, DF=doc frequency,

Impact of Feature Selection on Classification of fMRI Data

[Pereira et al., 2005]

Accuracy classifying
category of word read
by subject

#voxels	mean	subjects							
		233B	329B	332B	424B	474B	496B	77B	86B
50	0.735	0.783	0.817	0.55	0.783	0.75	0.8	0.65	0.75
100	0.742	0.767	0.8	0.533	0.817	0.85	0.783	0.6	0.783
200	0.737	0.783	0.783	0.517	0.817	0.883	0.75	0.583	0.783
300	0.75	0.8	0.817	0.567	0.833	0.883	0.75	0.583	0.767
400	0.742	0.8	0.783	0.583	0.85	0.833	0.75	0.583	0.75
800	0.735	0.833	0.817	0.567	0.833	0.833	0.7	0.55	0.75
1600	0.698	0.8	0.817	0.45	0.783	0.833	0.633	0.5	0.75
all (~2500)	0.638	0.767	0.767	0.25	0.75	0.833	0.567	0.433	0.733

Table 1: Average accuracy across all pairs of categories, restricting the procedure to use a certain number of voxels for each subject. The highlighted line corresponds to the best mean accuracy, obtained using 300 voxels.

Voxels scored by p-value of regression to predict voxel value from the task

Approach 2: Regularization

Key idea: add L1 penalty to learning objective, to penalize large weights

- L1 penalty = sum of magnitudes of weights
- think about L1 vs L2 for logistic regression...

$$\frac{1}{1 + \exp(\sum_i w_i x_i)}$$

$\frac{L1}{4}$	$\frac{L2}{8}$	$\rightarrow .2x_3 + .2x_4$
	\rightarrow	$x_3 = x_4$
4	16	$\rightarrow .4x_3 + 0x_4$

$$\arg \max_w \left[\sum_{k \text{ example}} \log P(y^k | x^k) \right] + \underbrace{c|w|^2}_{L2}$$

$c|w| \rightarrow L1$







Summary: Supervised Feature Selection

Approach 1: Preprocess data to select only a subset of the X_i

- Score each feature
 - Mutual information, prediction accuracy, ...
- Find useful subset of features based on their scores
 - Greedy addition/deletion of features to pool
 - Considered independently, or in context of other selected features

Approach 2: use L1 regularization of parameters

Always do feature selection using training set only (why?)

- Often use nested cross-validation loop:
 - Outer loop to get unbiased estimate of final classifier accuracy
 - Inner loop to get unbiased feature scores for feature selection



Unsupervised Dimensionality Reduction



Unsupervised mapping to lower dimension

Differs from feature selection in two ways:

- Instead of choosing subset of features, create new features (dimensions) defined as functions over all features
- Don't consider class labels, just the data points

Principle Components Analysis

- Idea:
 - Given data points in d -dimensional space, project into lower dimensional space while preserving as much information as possible
 - E.g., find best planar approximation to 3D data
 - E.g., find best planar approximation to 10^4 D data
 - In particular, choose projection that minimizes the squared error in reconstructing original data

PCA: Find Projections to Minimize Reconstruction Error

→ Assume data is set of d -dimensional vectors, where n th vector is

$$\mathbf{x}^n = \langle x_1^n \dots x_d^n \rangle$$

We can represent these in terms of any d orthogonal basis vectors

$$\mathbf{x}^n = \sum_{i=1}^d z_i^n \mathbf{u}_i; \quad \mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$$

PCA: given $M < d$. Find $\langle \mathbf{u}_1 \dots \mathbf{u}_M \rangle$

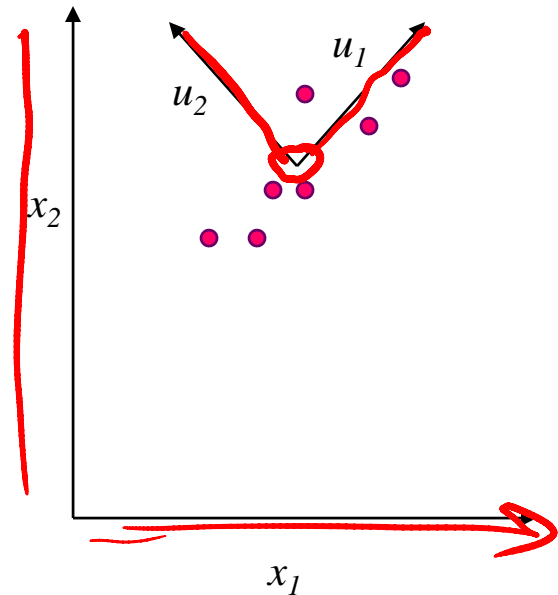
that minimizes $E_M \equiv \sum_{n=1}^N \|\mathbf{x}^n - \hat{\mathbf{x}}^n\|^2$

where $\hat{\mathbf{x}}^n = \bar{\mathbf{x}} +$

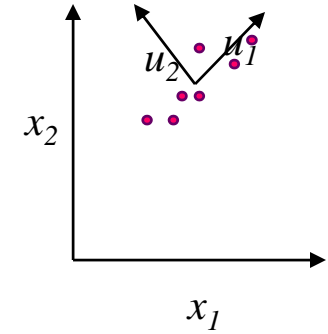
$$\sum_{i=1}^M z_i^n \mathbf{u}_i$$

Mean

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}^n$$



PCA



PCA: given $M < d$. Find $\langle \mathbf{u}_1 \dots \mathbf{u}_M \rangle$
 that minimizes $E_M \equiv \sum_{n=1}^N \|\mathbf{x}^n - \hat{\mathbf{x}}^n\|^2$
 where $\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^M z_i^n \mathbf{u}_i$

Note we get zero error if $M=d$, so all error is due to missing components.

Therefore, $E_M = \sum_{i=M+1}^d \sum_{n=1}^N [\mathbf{u}_i^T (\mathbf{x}^n - \bar{\mathbf{x}})]^2$

M < d

Example

u of these

$= \sum_{i=M+1}^d \mathbf{u}_i^T \Sigma \mathbf{u}_i$

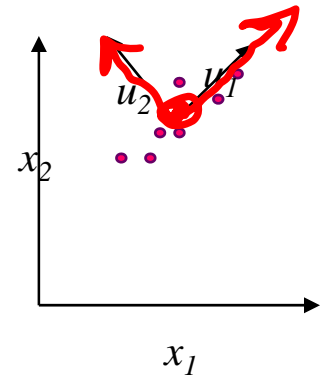
This minimized when \mathbf{u}_i is eigenvector of Σ , the covariance matrix of X .
 i.e., minimized when:

$\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$

↑ scalar

Covariance matrix: $\Sigma = \sum_n (\mathbf{x}^n - \bar{\mathbf{x}})(\mathbf{x}^n - \bar{\mathbf{x}})^T$

PCA



$$\text{Minimize } E_M = \sum_{i=M+1}^d \mathbf{u}_i^T \Sigma \mathbf{u}_i$$

$$\rightarrow \Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$$

↙ Eigenvector of Σ
↘ Eigenvalue (scalar)

$$\rightarrow E_M = \sum_{i=M+1}^d \lambda_i$$

PCA algorithm 1:

1. $X \leftarrow$ Create $N \times d$ data matrix, with one row vector x^n per data point
2. $X \leftarrow$ subtract mean \bar{x} from each row vector x^n in X
3. $\Sigma \leftarrow$ covariance matrix of X
4. Find eigenvectors and eigenvalues of Σ
5. PC's \leftarrow the M eigenvectors with largest eigenvalues

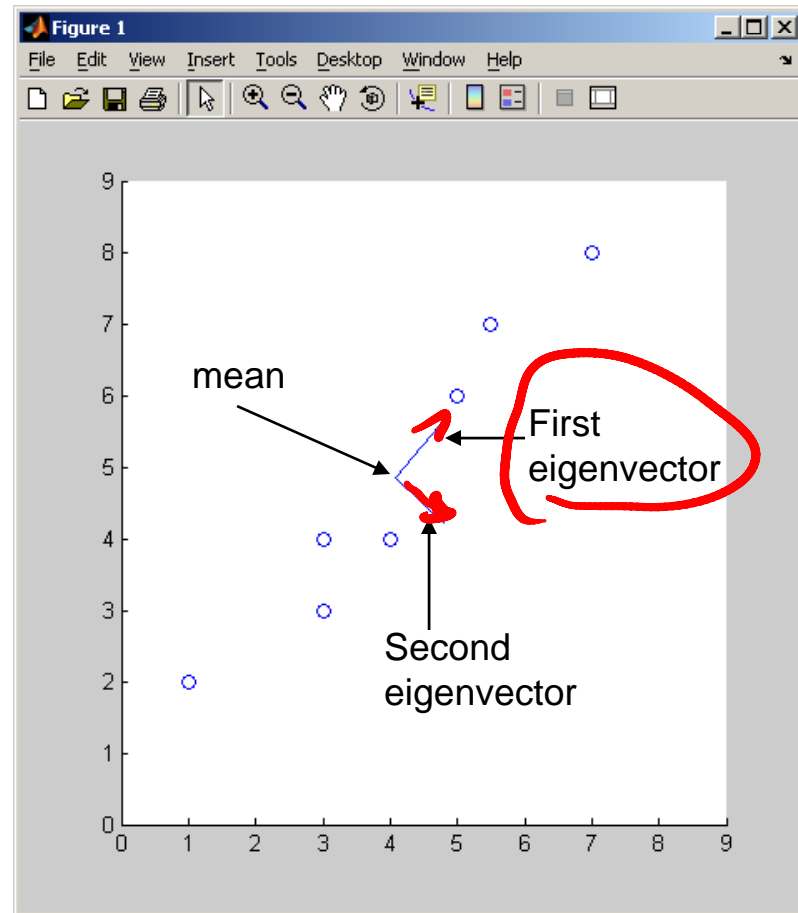
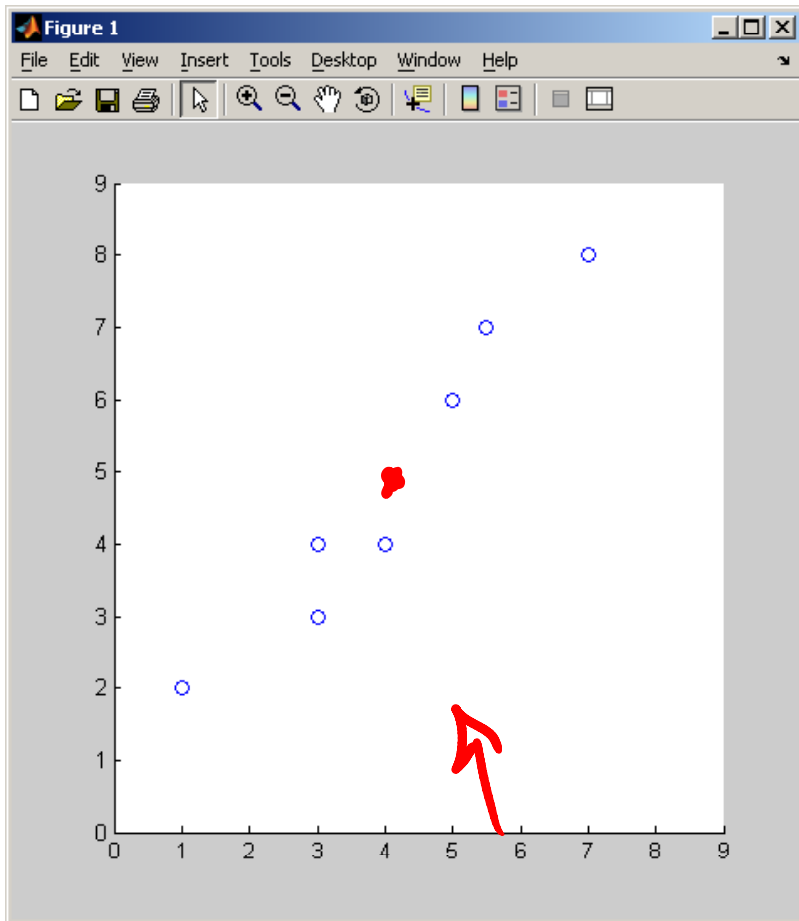
examp
dims
of feats

examp

u_1, u_2, \dots, u_n

PCA Example

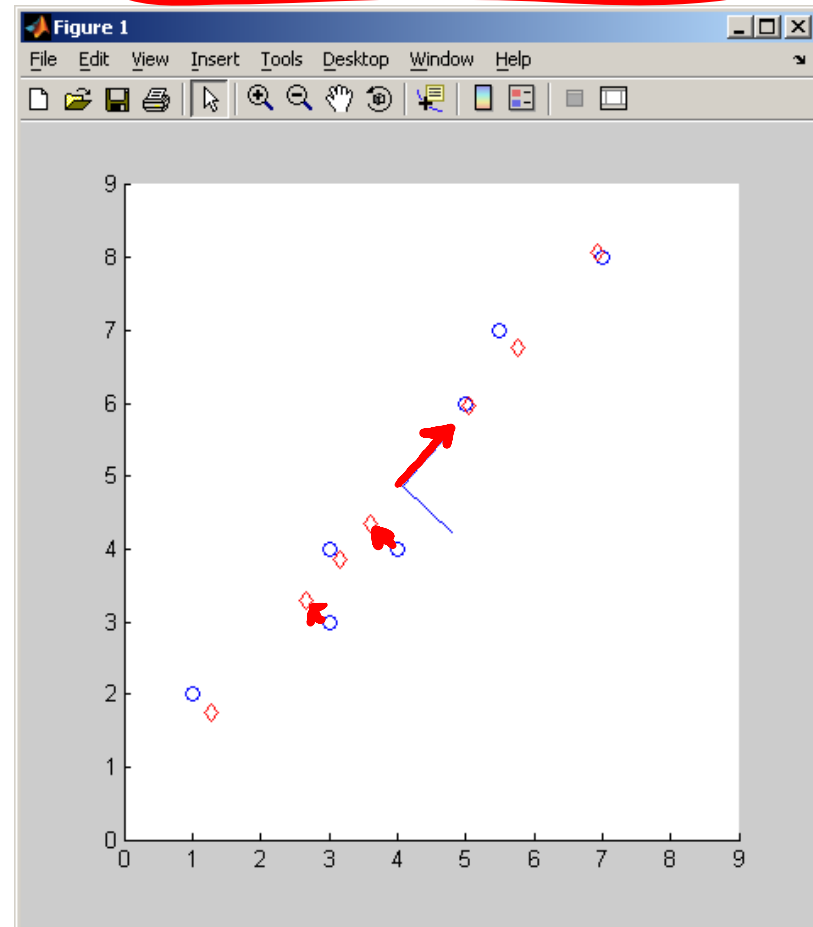
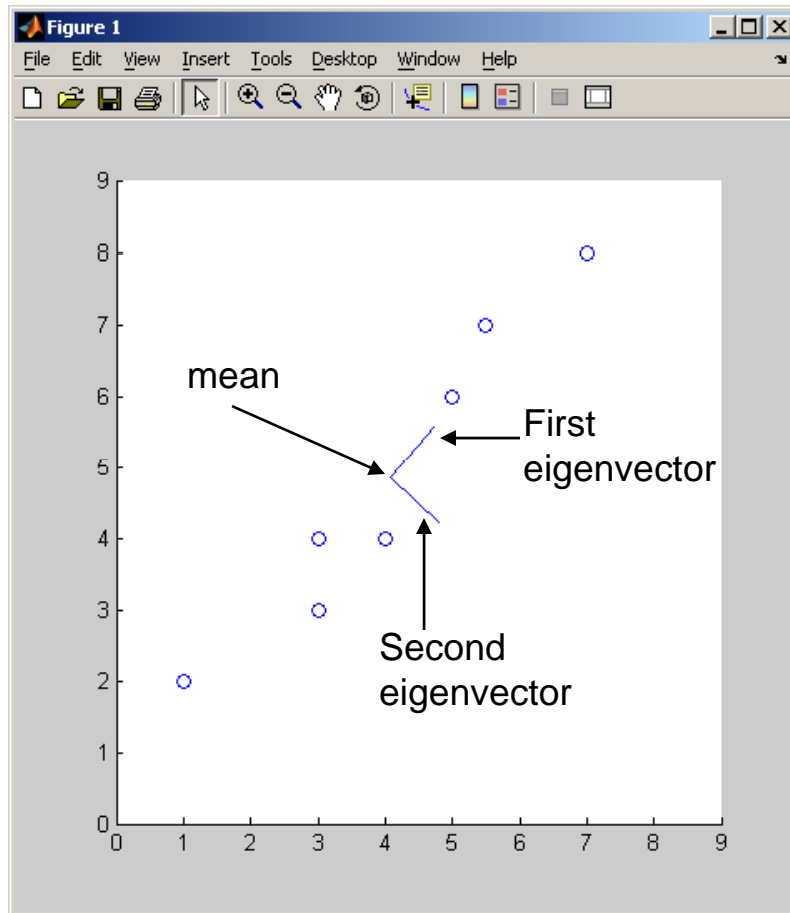
$$\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^M z_i^n \mathbf{u}_i$$



PCA Example

$$\hat{\mathbf{x}}^n = \bar{\mathbf{x}} + \sum_{i=1}^M z_i^n \mathbf{u}_i$$

Reconstructed data using only first eigenvector (M=1)



Very Nice When Initial Dimension Not Too Big

What if very large dimensional data?

- e.g., Images ($d \sim 10^4$)

Problem:

- Covariance matrix Σ is size ($d \times d$)
- $d=10^4 \rightarrow |\Sigma| = 10^8$

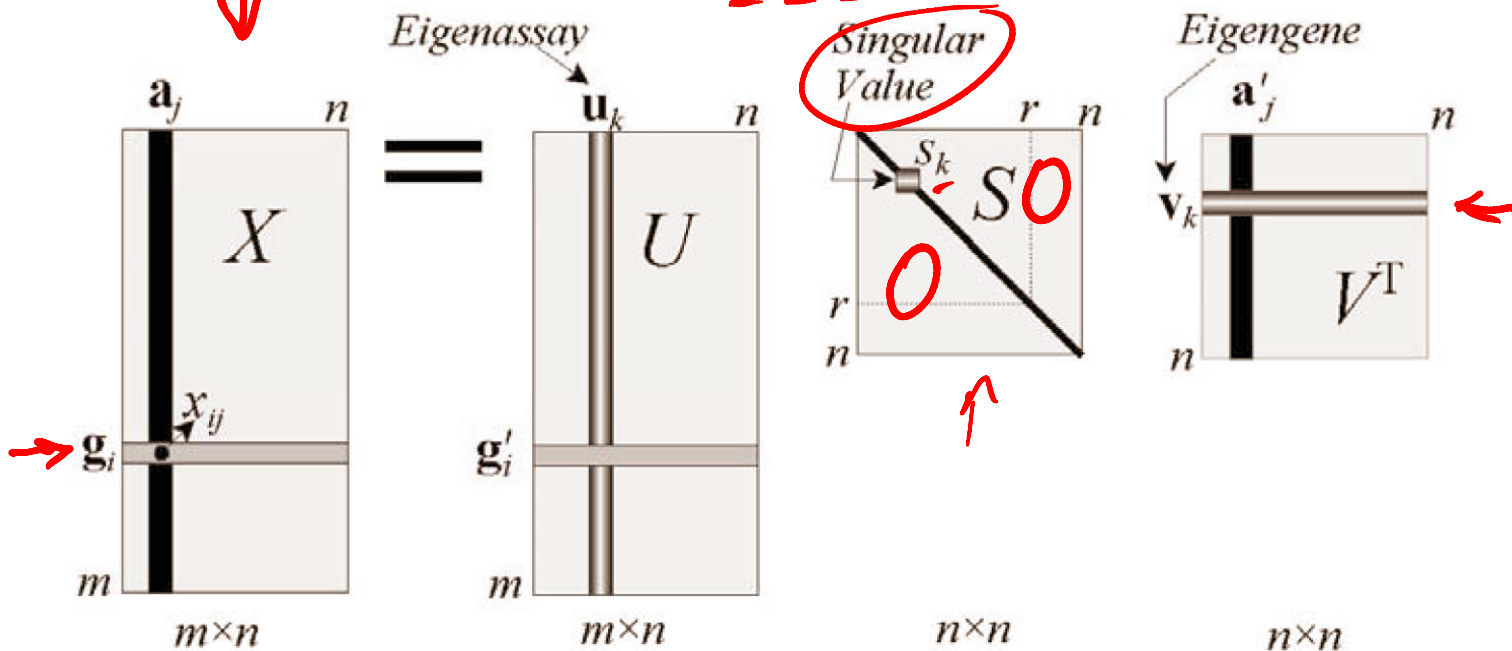
Singular Value Decomposition (SVD) to the rescue!

- pretty efficient algs available, including Matlab SVD
- some implementations find just top N eigenvectors

SVD

data

$$X = USV^T$$



Data X , one row per data point

US gives coordinates of rows of X in the space of principle components

S is diagonal
 $S_k > S_{k+1}$
 S_k^2 is k th largest eigenvalue

Rows of V^T are unit length eigenvectors of $X^T X$

If cols of X have zero mean, then $X^T X - c \Sigma$ and eigenvects are the Principle Components

[from Wall et al., 2003]

Singular Value Decomposition

To generate principle components:

- Subtract mean $\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}^n$ from each data point, to create zero-centered data
- Create matrix X with one row vector per (zero centered) data point
- Solve SVD: $X = USV^T$
- Output Principle components: columns of V (= rows of V^T)
 - Eigenvectors in V are sorted from largest to smallest eigenvalues
 - S is diagonal, with s_k^2 giving eigenvalue for k th eigenvector

Singular Value Decomposition

To project a point (column vector x) into PC coordinates:

$$V^T x$$

If x_i is i^{th} row of data matrix X , then

- (i^{th} row of US) = $V^T x_i^T$
- $(US)^T = V^T X^T$

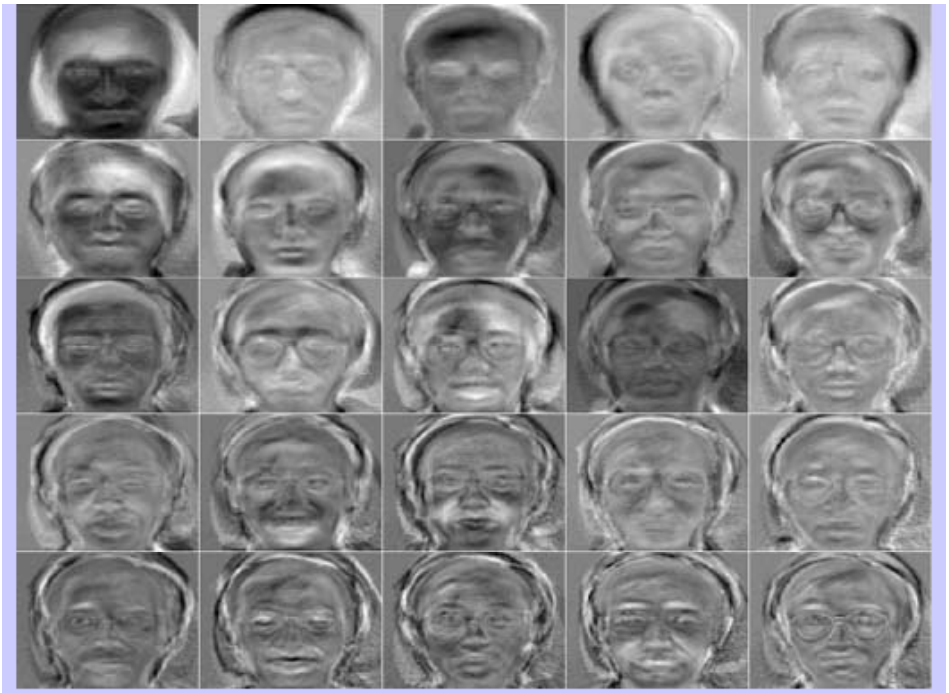
To project a column vector x to M dim Principle Components subspace, take just the first M coordinates of $V^T x$

PCA Example

faces



Eigenfaces



Thanks to Christopher DeCoro
see <http://www.cs.princeton.edu/~cdecoro/eigenfaces/>

Reconstructing a face from
the first N components
(eigenfaces)

Adding 1 additional
PCA component at
each step



In this next image, we show a similar picture, but with each additional face representing an additional 8 principle components. You can see that it takes a rather large number of images before the picture looks totally correct.

Adding 8 additional
PCA components
at each step



Independent Components Analysis

- PCA seeks directions $\langle Y_1 \dots Y_M \rangle$ in feature space X that minimize reconstruction error
- ICA seeks directions $\langle Y_1 \dots Y_M \rangle$ that are most *statistically independent*. I.e., that minimize $I(Y)$, the mutual information between the Y_j :

$$I(Y) = \left[\sum_{j=1}^J H(Y_j) \right] - H(Y)$$

Dimensionality reduction for multiple datasets

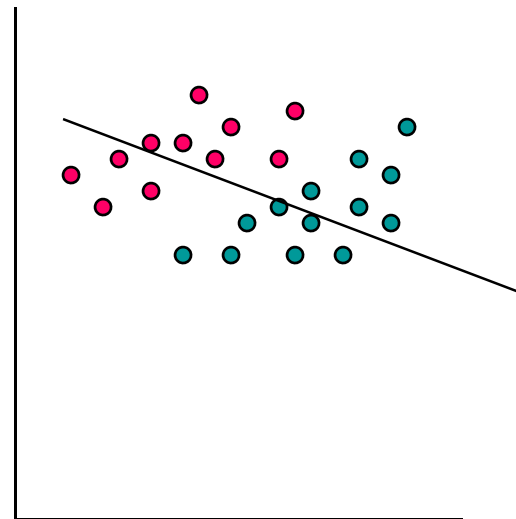
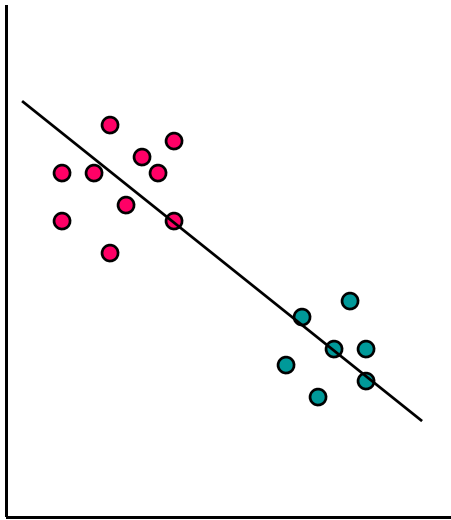
- Generalized SVD
 - Given data sets A and B , find projections of each into common low dimensional basis, to minimize sq reconstruction errors of both
- Canonical correlation analysis



Supervised Dimensionality Reduction

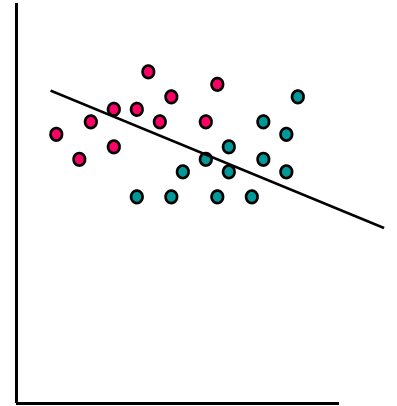
1. Fisher Linear Discriminant

- A method for projecting data into lower dimension to hopefully improve classification
- We'll consider 2-class case



Project data onto vector that connects class means?

Fisher Linear Discriminant



Project data onto one dimension, to help classification

$$y = \mathbf{w}^T \mathbf{x}$$

Define class means: $\mathbf{m}_i = \frac{1}{N_i} \sum_{n \in C_i} \mathbf{x}^n$

Could choose \mathbf{w} according to: $\arg \max_{\mathbf{w}} \mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1)$

Instead, Fisher Linear Discriminant chooses: $\arg \max_{\mathbf{w}} \frac{(\mathbf{m}_2 - \mathbf{m}_1)^2}{s_1^2 + s_2^2}$

$$m_i \equiv \mathbf{w}^T \mathbf{m}_i \quad s_i^2 \equiv \sum_{n \in C_i} (x^n - m_i)^2$$

Fisher Linear Discriminant

Project data onto one dimension, to help classification

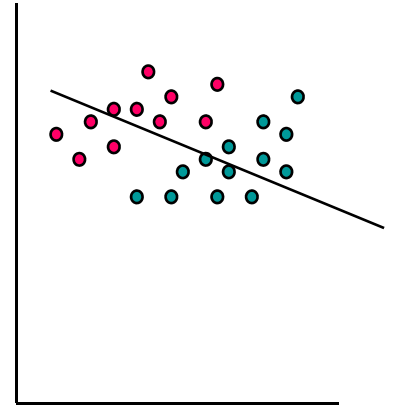
$$y = \mathbf{w}^T \mathbf{x}$$

Fisher Linear Discriminant : $\arg \max_{\mathbf{w}} \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$

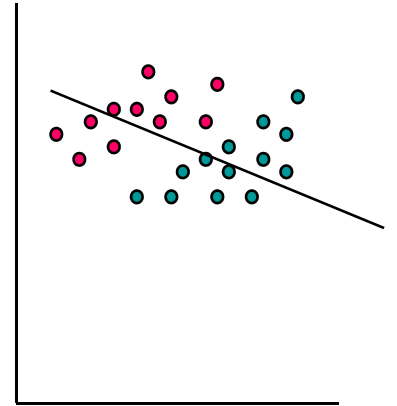
is solved by : $\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$

Where \mathbf{S}_W is sum of within-class *projected covariances*:

$$\mathbf{S}_W \equiv \sum_{n \in C_1} (\mathbf{x}^n - \mathbf{m}_1)(\mathbf{x}^n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}^n - \mathbf{m}_2)(\mathbf{x}^n - \mathbf{m}_2)^T$$



Fisher Linear Discriminant



Fisher Linear Discriminant :
$$\arg \max_{\mathbf{w}} \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

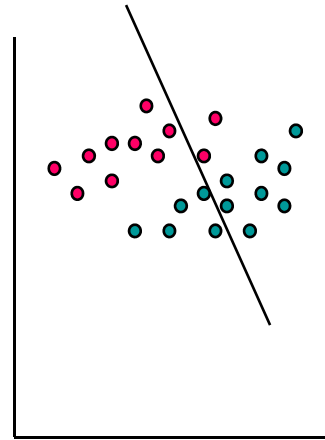
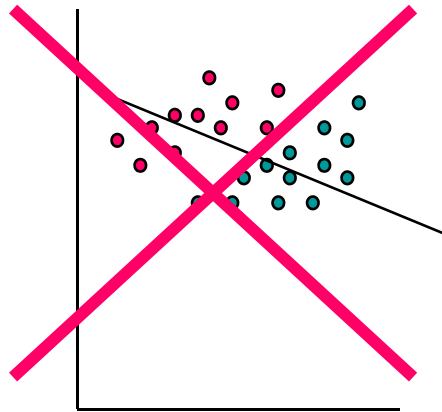
Is equivalent to minimizing sum of squared error if we assume target values are not +1 and -1, but instead N/N_1 and $-N/N_2$

Where N is total number of examples, N_i is number in class i

Also generalized to K classes (and projects data to $K-1$ dimensions)

Summary: Fisher Linear Discriminant

- Choose $n-1$ dimension projection for n -class classification problem
- Use within-class covariances to determine the projection
- Minimizes a different sum of squared error function (the projected within-class variances)



What you should know

- Feature selection
 - Single feature scoring criteria
 - Regularization
 - Search strategies
 - Common approaches: Greedy addition of features, or greedy deletion
- Unsupervised dimension reduction using all features
 - Principle Components Analysis
 - Minimize reconstruction error
 - Singular Value Decomposition
 - Efficient PCA
 - Independent components analysis
- Supervised dimension reduction
 - Fisher Linear Discriminant
 - Project to $n-1$ dimensions to discriminate n classes
 - Hidden layers of Neural Networks
 - Most flexible, local minima issues

Further Readings

- “Singular value decomposition and principal component analysis,” Wall, M.E, Rechtsteiner, A., and L. Rocha, in *A Practical Approach to Microarray Data Analysis* (D.P. Berrar, W. Dubitzky, M. Granzow, eds.) Kluwer, Norwell, MA, 2003. pp. 91-109.
LANL LA-UR-02-4001