ISTA 421/521 Introduction to Machine Learning

Lecture 25:
Projection
Principle Components Analysis

Clay Morrison

clayton@sista.arizona.edu

Gould-Simpson 819

Phone 621-6609

24 November 2014

Main parametric modeling frameworks

- Minimizing a Loss function
  - Linear model
  - Linear least mean squares
- Maximum Likelihood
  - Probabilistic model of uncertainty (noise, error)
  - Maximize the likelihood w.r.t. parameters
  - Linear model with additive Gaussian noise
- Bayesian Approach
  - Treat parameters as random variables
  - Use Bayes Theorem to combine likelihood & prior to find posterior distribution
- Estimation Techniques (often used in Bayesian approaches)
  - Gradient methods (Widrow-Hoff (1st), Newton-Rhapson (2nd))
  - Laplace Approximation
  - Monte Carlo estimation of expectation; Metropolis-Hastings

Approaches to Avoiding Over-fitting
i.e., how to achieve ***generalization***
- Regularization
- Cross Validation (estimating the generalization error)

- Classification (& Regression)
- Clustering
- Projection

predicting output

Main algorithmic families of Machine Learning

Dimension Reduction

- Projection methods are part of a broader class of ***Dimension Reduction*** methods.
- Recall that a dimension is a "feature", and the value along the dimension is the feature value.
- What problems are dimension reduction methods used to address?

The CURSE OF DIMENSIONALITY

- A major problem!
- If the data **x** lie in high dimensional space, then an enormous amount of data is required to learn distributions or decision rules.
- Example:
  - 50 dimensions. Each dimension has 20 "levels"
  - Total of 20^{50} cells!
  - But the number of data samples will be far less.
  - There will not be enough data samples to estimate the parameters.

(Adapted from Alan L. Yuille Stat 231, Fall 2004)

The CURSE OF DIMENSIONALITY

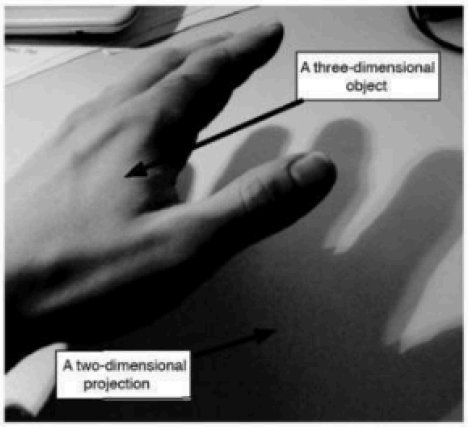
- One way to deal with dimensionality is to assume that we know the form of the probability distribution.
- For example: a Gaussian model with N dimensions has $N+N(N-1)/2$ parameters to estimate. Requires $O(N^2)$ data to learn reliably. *May* be practical.

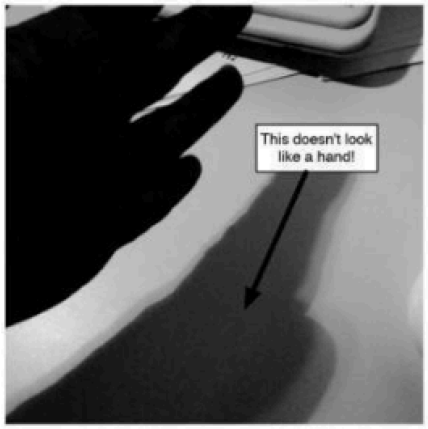
(Adapted from Alan L. Yuille Stat 231, Fall 2004)

Dimension Reduction

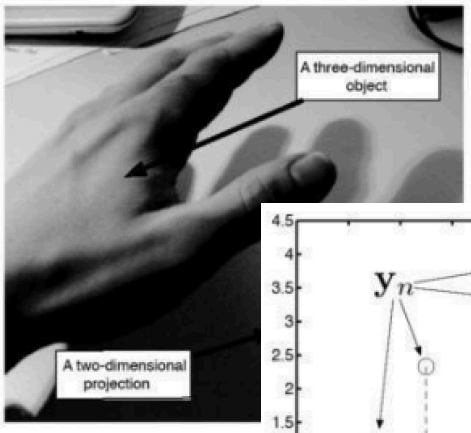
- Another PROBLEM
  - Can't visualize data in more than 3 dimensions!
- **Techniques for dimension reduction**:
  - Principle Components Analysis (PCA)
  - Fisher's Linear Discriminant
  - Multi-dimensional Scaling
  - Independent Component Analysis
- **Approach**: Project data into a lower-dimensional space.
- **Goal**: Preserve as much "interesting" structure in data as possible

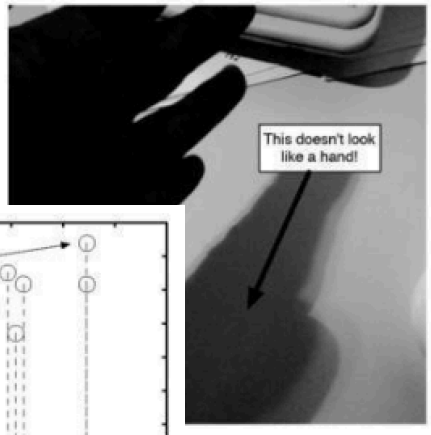
Intuition of Projection

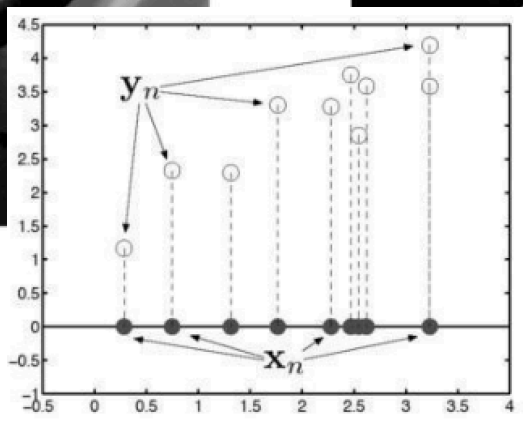




Intuition of Projection





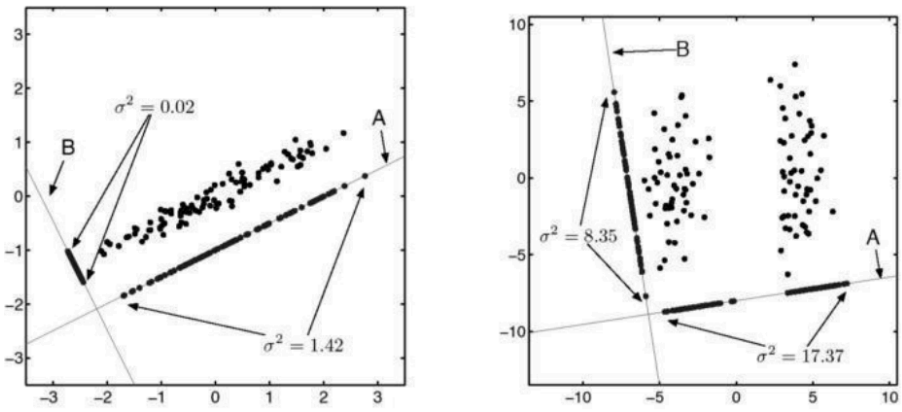


Principle Components Analysis

- PCA is the most commonly used dimension reduction technique.
- (Also called the Karhunen-Loeve transform)
- Principle:
  - **Linear**, **orthogonal** projection method to reduce the number of parameters
  - Criteria of interestingness: **maximize variability** along new axes
- Properties
  - Can be viewed as a rotation (and selection/removal) of the existing axes to new positions in the space defined by the original variables.
  - New axes are orthogonal and represent the directions with maximum variability.
  - Can also be viewed as a way to transfer a set of correlated variables into a new set of uncorrelated variables

Variables ≡ features ≈ dimension ≡ axes

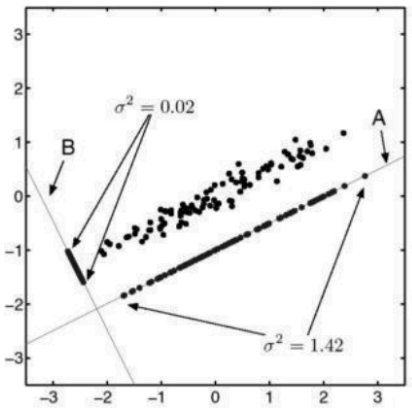
Why care about variance?

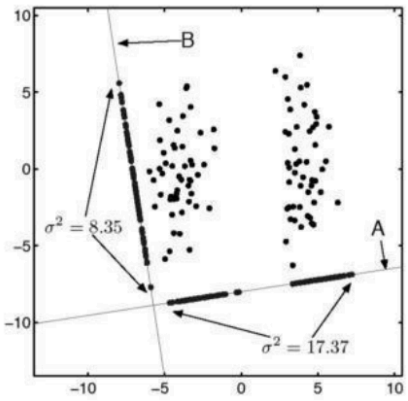


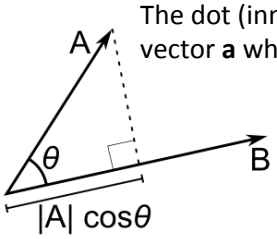
- We don't know the structure of interest ahead of time. But structure tends to be associated with variance.
- PCA uses variance as a proxy for interest.

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_x)^2$$

Projection!







The dot (inner) project is a measure of the length of vector **a** when we *project* it onto **b**

$$\mathbf{y}_n = [y_{n1},\ y_{n2}]^\mathsf{T}$$ Datum in original spatial coordinates

Datum in new spatial coordinates

$$x_n = w_1 y_{n1} + w_2 y_{n2}$$

Map from one space to the other

$$x_n = \mathbf{w}^\mathsf{T}\mathbf{y}_n \qquad \mathbf{w} = [w_1,\ w_2]^\mathsf{T}$$

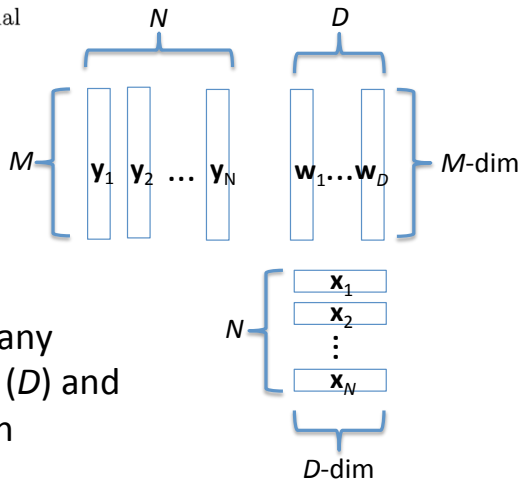
Principle Components Analysis

large *small*

projecting from M to D dimensions,
PCA will define D vectors, $\mathbf{w}_d$ N-dimensional

$$\mathbf{x}_n = [\ x_{n1}, \ldots, x_{nD}]^\mathsf{T}$$

$$x_{nd} = \mathbf{w}_d^\mathsf{T}\mathbf{y}_n$$



The learning task: to choose how many dimensions we want to project into (*D*) and find a projection vector, $\mathbf{w}_d$, for each

2 Constraints

1 PCA uses variance in the projected space as the criterion to choose $\mathbf{w}_d$, but all $\mathbf{w}_d$'s must be mutually orthogonal.

$$\mathbf{w}_i^\mathsf{T}\mathbf{w}_j = 0, \ \forall j \neq i$$

For example:
$\mathbf{w}_1$ is the projection that makes the variance in x_{n1} as high as possible
$\mathbf{w}_2$ will also maximize the variance, but must be orthogonal to $\mathbf{w}_1$ ($\mathbf{w}_1^\mathsf{T}\mathbf{w}_2$=0)
$\mathbf{w}_3$ will also maximize the variance, but must be orthogonal to $\mathbf{w}_1$ and $\mathbf{w}_2$ …

2 PCA also imposes the constraint that each $\mathbf{w}_i$ must have length 1 (not a restriction on the technique, rather, its needed to properly define the optimization)

$$\mathbf{w}_i^\mathsf{T}\mathbf{w}_i = 1$$

Following FCML, we'll derive an expression for the variance of x_n for one dimension (D=1), which we'll then maximize

$$x_n = \mathbf{w}^\mathsf{T}\mathbf{y}_n$$

Defining the Variance of x_n

Assume **y** has 0 mean: $\bar{\mathbf{y}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n = 0$

Enforce this by subtracting $\bar{\mathbf{y}}$ from every $\mathbf{y}_n$.

The variance of **x** is then $\sigma_x^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2$ $\quad \sigma_x^2 = \frac{1}{N}\sum_{n=1}^{N}x_n^2$ $\quad x_n = \mathbf{w}^\mathsf{T}\mathbf{y}_n$

We can then simplify this expression using the assumption $\bar{\mathbf{y}} = 0$

$$\begin{aligned}\bar{x} &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{w}^\mathsf{T}\mathbf{y}_n \\ &= \mathbf{w}^\mathsf{T}\left(\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\right) \\ &= \mathbf{w}^\mathsf{T}\bar{\mathbf{y}} = 0.\end{aligned}$$

Plug in definition of x_n as inner product

$$\begin{aligned}\sigma_x^2 &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^\mathsf{T}\mathbf{y}_n)^2 \\ &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{w}^\mathsf{T}\mathbf{y}_n\mathbf{y}_n^\mathsf{T}\mathbf{w} \\ &= \mathbf{w}^\mathsf{T}\left(\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^\mathsf{T}\right)\mathbf{w}\end{aligned}$$

C is the sample covariance matrix (keeping in mind $\bar{\mathbf{y}}$=0 here)

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^\mathsf{T} \qquad \sigma_x^2 = \mathbf{w}^\mathsf{T}\mathbf{C}\mathbf{w},$$

Maximizing the Variance of x_n

Goal is to maximize $\sigma_x^2 = \mathbf{w}^\mathsf{T}\mathbf{C}\mathbf{w}$ $\quad\quad$ $\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^\mathsf{T}$

Remember constraint 2: $\mathbf{w}_i^\mathsf{T}\mathbf{w}_i = 1$

If we did not have this, we could just keep increasing **w** to increase σ^2 !

Incorporate constraint as a Lagrange multiplier!

$$L = \mathbf{w}^\mathsf{T}\mathbf{C}\mathbf{w} - \lambda(\mathbf{w}^\mathsf{T}\mathbf{w} - 1)$$

Now, maximize: take partial derivative, set to 0, rearrange:

$$\frac{\partial L}{\partial \mathbf{w}} = 2\mathbf{C}\mathbf{w} - \lambda\mathbf{w} = \mathbf{0}$$

$$\mathbf{C}\mathbf{w} = \lambda\mathbf{w}$$

But we want to solve for 0!
What to do?

The eigenvector/eigenvalue Equation for square matrix **A**

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

Eigenvectors & Eigenvalues

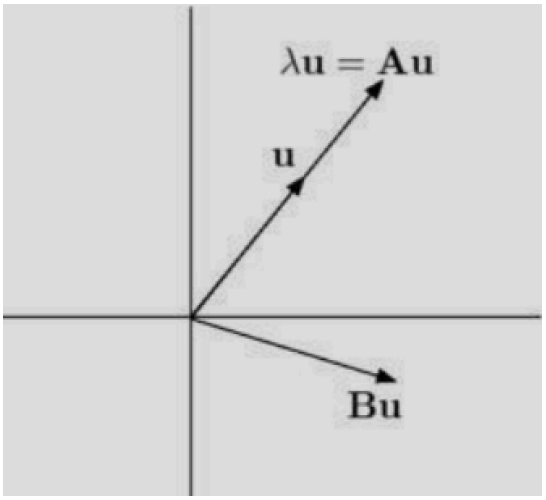
$$\lambda_i\mathbf{u}_i = \mathbf{A}\mathbf{u}_i$$

A is a square, symmetric, positive definite NxN matrix
Solving this equation results in pairs of eigenvalues (λ_i) and eigenvectors ($\mathbf{u}_i$)

- There will be N pairs.
- The N eigenvectors will be mutually orthogonal.

The geometric view:
Multiplying an N-dimensional vector **u** by an NxN square matrix, **B**, generally results in a "rotation" of **u**.

The eigenvector/-value pairs for a square matrix **A** are the vectors **u** for which applying the rotation **A** only results in a change in length of **u**; the magnitude of this change is given by the scalar λ.



Maximizing the Variance of x_n

Goal is to maximize $\sigma_x^2 = \mathbf{w}^\mathsf{T}\mathbf{C}\mathbf{w}$

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^\mathsf{T}$$

$$\frac{\partial L}{\partial \mathbf{w}} = 2\mathbf{C}\mathbf{w} - \lambda\mathbf{w} = \mathbf{0}$$
$$\mathbf{C}\mathbf{w} = \lambda\mathbf{w}$$

The eigenvector/eigenvalue equation for square matrix **A**

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

But what is the interpretation of λ ?

$$\sigma_x^2 = \mathbf{w}^\mathsf{T}\mathbf{C}\mathbf{w}$$

$$\mathbf{w}^\mathsf{T}\mathbf{w} = 1$$
Constraint 2 (again)

$$\sigma^2\mathbf{w}^\mathsf{T}\mathbf{w} = \mathbf{w}^\mathsf{T}\mathbf{C}\mathbf{w}$$

$$\sigma^2\mathbf{w} = \mathbf{C}\mathbf{w}$$

So λ corresponds to the variance of the data in the projected space defined by **w**

If we find the *M* eigenvector/eigenvalue pairs of the covariance matrix **C**, the pair with the highest eigenvalue corresponds to the projection with the maximal variance $\mathbf{w}_1$. The 2nd highest eigenvalue corresponds to $\mathbf{w}_2$, then $\mathbf{w}_3$, etc.

The PCA Algorithm

PCA on a set of data objects, $\mathbf{y}_1, \ldots, \mathbf{y}_N$

$$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\mathsf{T}$$

1. Transform the M-dimensional data to have zero mean by subtracting $\bar{\mathbf{y}}$ from each object wher e $\bar{\mathbf{y}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n$.

2. Compute the sample covariance matrix $\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^\mathsf{T}$ (or $\mathbf{C} = \frac{1}{N}\mathbf{Y}^\mathsf{T}\mathbf{Y}$)

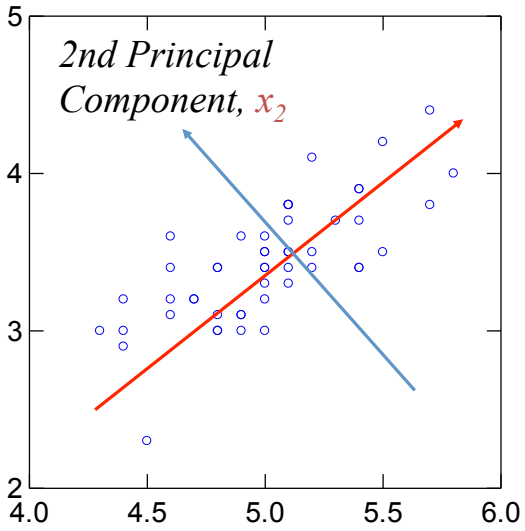
3. Find the M eigenvector/eigenvalue pairs of the covariance matrix.

`numpy.linalg.eig` in python, `eigs` in Matlab

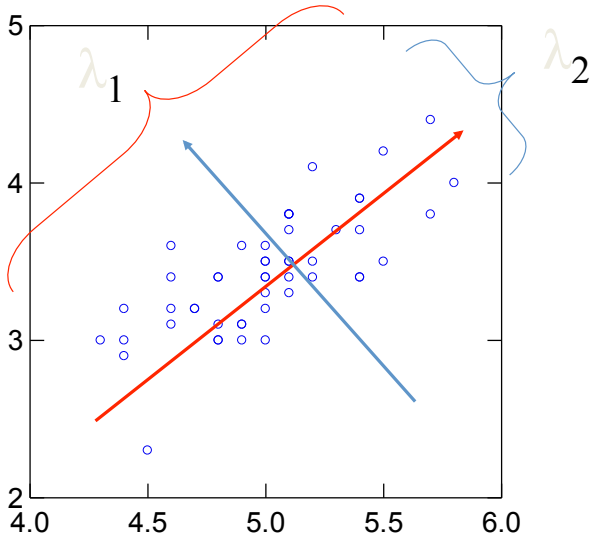
4. Find the eigenvectors corresponding to the D highest eigenvalues, $\mathbf{w}_1, \ldots, \mathbf{w}_D$.

5. Create the dth dimension for object n in the projection, $x_{nd} = \mathbf{w}_d^\mathsf{T}\mathbf{y}_n$ (or $\mathbf{X} = \mathbf{Y}\mathbf{W}$, where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_D]$, i.e. the $M \times D$ matrix created by placing the D eigenvectors alongside one another and $\mathbf{X}$ is the $N \times D$ matrix defined as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\mathsf{T}$)

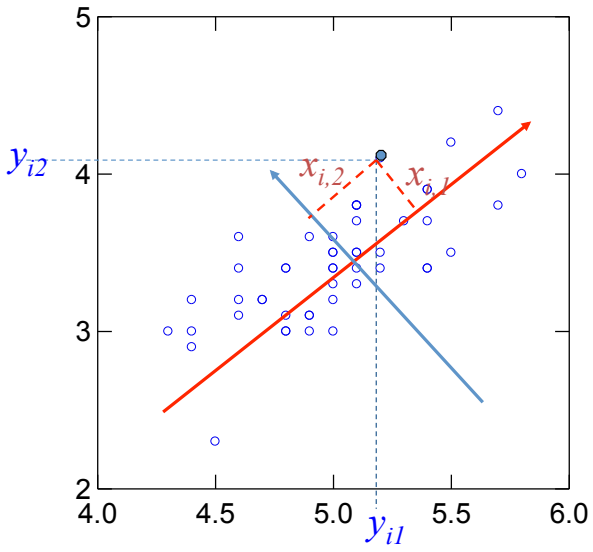
PCA Eigenvectors/Principle Components



PCA Eigenvalues



PCA Projections of data

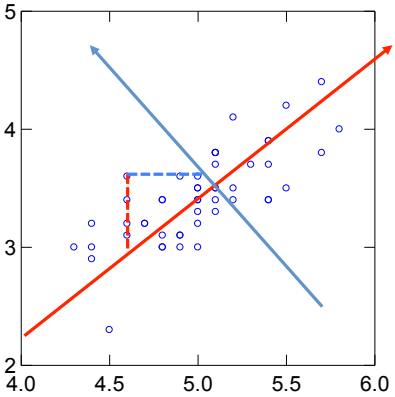


The Least-Squares View

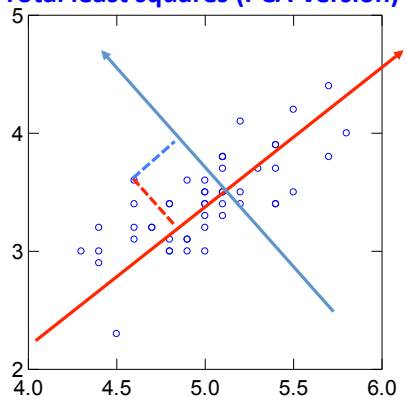
Principle Components are a series of linear least-squares fits to a sample, each orthogonal to all previous ones.

HOWEVER: Unlike linear regression, where the data are projected down to the line perpendicular to the original axes, instead they're projected perpendicularly to the principle component axis.

Least Squares

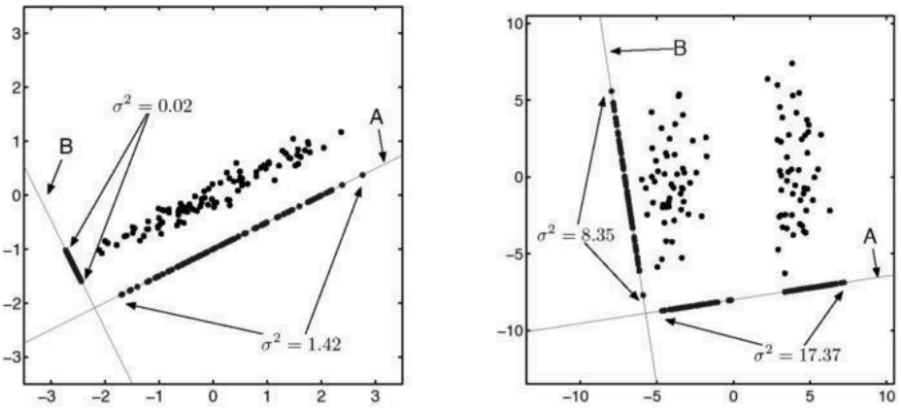


Total least squares (PCA-version)



The "PCA version" is called "Total least squares" or "rigorous least squares" or "orthogonal regression"; in this model, observational errors (deviations from the line) are w.r.t. both dependent and independent variables, rather than the dependent variable(s) alone.

Simple Examples

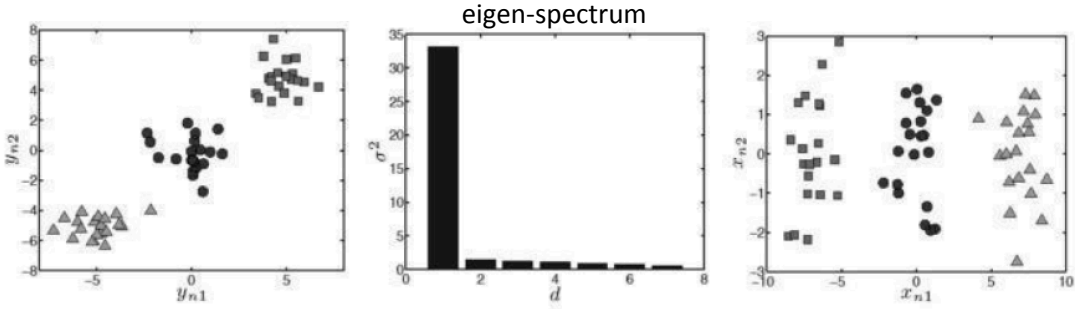


This is 2d data, so there are 2 eigenvectors (2 possible principle components)

Note: the principle component eigenvectors only give directions -- they're plotted as offset axes for convenience.

Another Example

eigen-spectrum



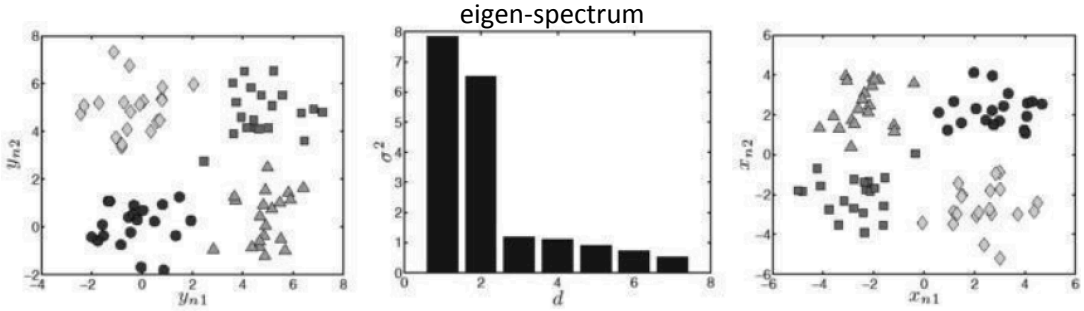
(a) First two dimensions of the data objects $\mathbf{y}_n$

(b) The seven eigenvalues (variances of the projected dimensions)

(c) The data projected onto the first two principal components

These are actually 7-dimensional data;
the other 5 dimensions are sampled from $\mathcal{N}(0, 1)$.
So structure in first two dimensions and symmetric noise in rest.

Another Example



(a) First two dimensions of the data objects $\mathbf{y}_n$

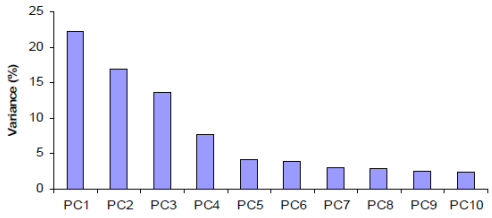
(b) The seven eigenvalues (variances of the projected dimensions)

(c) The data projected onto the first two principal components

Note:
(1) The data labels are for visualization, PCA does NOT discover them
(2) Although we generate the data (same as before) with dimensions 3-7 random, the order of the dimensions makes no difference (WHY?)

Choosing D

- If for visualization, choose the top 1-3 eigen-spectrum, but can't really visualize in higher than 3 dimensions.
- More generally, the eigen-spectrum provides information about potential interestingness/ informativeness, but is subjective.



- Try to find objective measure: if PCA is first step of classification or regression (supervised method), then utility of different D values (cross-validation!)

Eigenfaces!

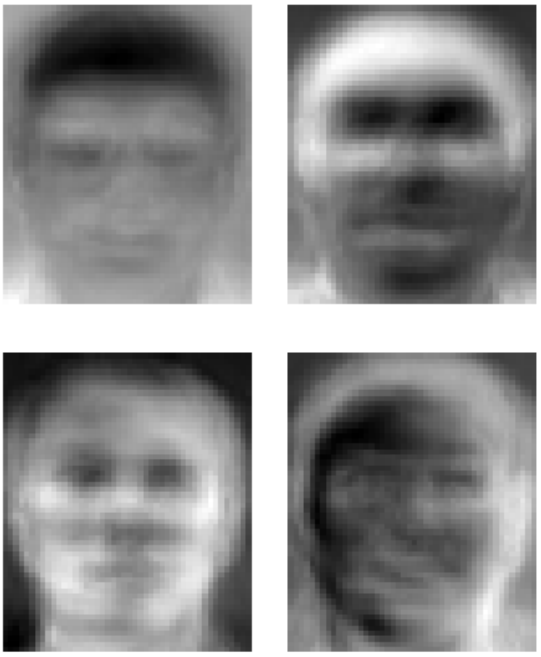
A dataset of M face images, where each image is a set of N pixel intensities.

Derive the top principle components:
Find the eigenvectors for the components, project the M face images to D-dimensional eigenfaces.

To then use for classification, the original database of images is saved as collections of weights describing the contribution each eigenface makes to that image. New face is presented for classification, its own weights are found by projecting the image into the collection of eigenfaces. A method like Nearest Neighbors can then be used to find the distance between the new image vector and vectors of existing images

Reported result: M=10,000 images, each 100x100 pixels, obtain 10,000 eigenvectors; but typically most faces can be identified using a projection on between 100 and 150 eigenfaces!

M. Turk and A. Pentland (1991). "Face recognition using eigenfaces". Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–591.



A different eigenfaces example from Alan Yuille, using PCA for analysis

- The images of an object under different lighting lie in a low-dimensional space.
- The original images are 256x 256. But the data lies mostly in 3-5 dimensions.
- First we show the PCA for a face under a range of lighting conditions. The PCA components have simple interpretations.
- Then we plot $\frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$ as a function of M for several objects under a range of lighting.

Example of Alan Yuille's face

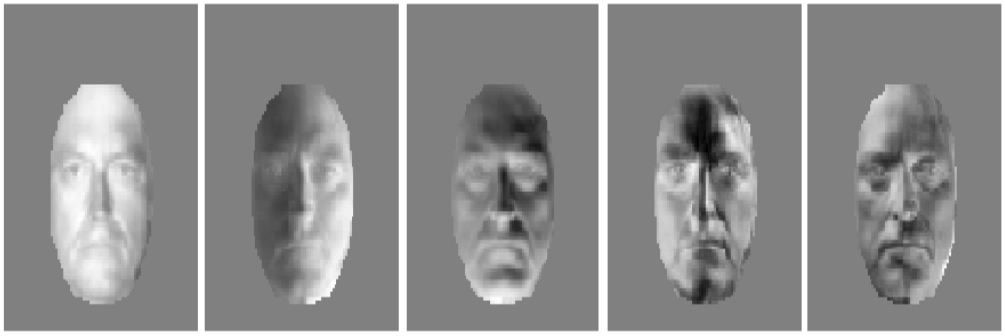


Figure 4: The eigenvectors calculated from the sparse set for the human face. Note that the images were only lit from the right so the eigenvectors are not perfectly symmetric. Observe also that the first three eigenvectors appear to be images of the face illuminated from three orthogonal lighting conditions in agreement with the orthogonal lighting conjecture.

Limitations of PCA

1. Data are real-valued
2. Assumes no missing values in the data

- Scientific data often has missing values
- Movie ratings: 1-5 scale (integer) and most people don't watch and rate EVERY movie!

- We'd like to construct the analysis provided by PCA but that can overcome these limitations

Latent Variable Models

- Mixture models!
- Characteristics of objects not provided in the data: ***latent*** (hidden) variables
- Two types of latent variables:
  1. Real feature of the object, but not measured
  2. "Abstract qualities" that may not correspond directly to any particular real thing (entity, process, relation) but instead arise from our modeling assumptions and might be useful
- Example of type 2: Indicator variables; enabled us to build mixture models but do not necessarily correspond to anything in reality. $q_{nk} = p(z_{nk} = 1|\mathbf{x}_n, \pi, \Delta)$
- PCA is also a type of latent variable model: the dimensions identified are (potentially) useful ways of relating/describing relationships among data; do they correspond to real properties?

Extending PCA

- Possible ways to extend PCA beyond its limitations…
- Extend: create a probabilistic version
- Inference, will require approximation of the posterior; direct solution is intractable
- Possible inference tools: Ch.4 methods: iterative gradient methods (Newton-Rhapson), Laplace approximation, MCMC sampling.
- The book introduces another method: **Variational Bayes**.