



Lecture 11: Dimensionality Reduction

DD2431

A. Maki

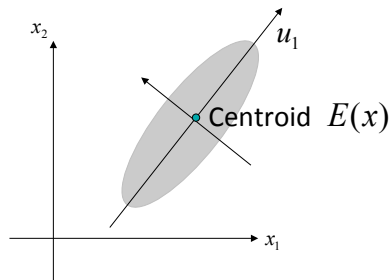
Autumn, 2015

Our keywords today:

- Unsupervised learning
- Dimensionality reduction
 - Principal Component Analysis (PCA)
- Discriminant function
 - Similarity measures: angle, projection length
- Subspace Methods

Principal Component Analysis (PCA)

1. Maximizing variance



Mean vector of x : $E(x) = (1/r) \sum x$ (where r is the number of samples)

Covariance matrix: $\Sigma = E((x - E(x))(x - E(x))^T)$

1. Maximum variance criterion

Reduce the effective number of variables
(only dealing with components with larger variances)

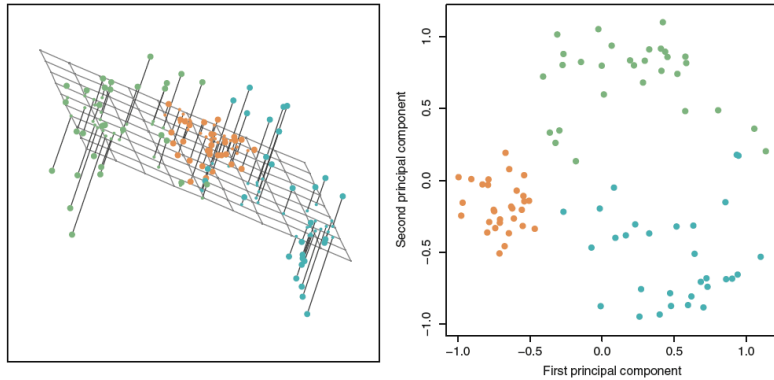
$$\begin{aligned}
 & E((x^T u_i - E(x^T u_i))^2) \rightarrow \text{Maximize } (i = 1, \dots, p) \\
 & = E((u_i^T (x - E(x)))^2) \\
 & = u_i^T \underbrace{E((x - E(x))(x - E(x))^T)}_{\text{Covariance matrix}} u_i = u_i^T \Sigma u_i
 \end{aligned}$$

Condition: $u_i^T u_j = \delta_{ij}$

$\max[\text{tr}(U^T \Sigma U)]$

The transformation matrix U consists of p columns that are **eigenvectors** of the **covariance matrix**, Σ , corresponding to largest eigenvalues.

Example 3-d to 2-d: Ninety observations simulated in 3-d



The first 2 principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane.

Figure from
An Introduction to Statistical Learning (James et al.)

2. Minimum squared distance criterion

Averaged squared error between x and its approximation to be minimized by a set $\{u_1, \dots, u_p\}$

$$E(\|x - x'\|^2) \rightarrow \text{minimize } (i = 1, \dots, p)$$

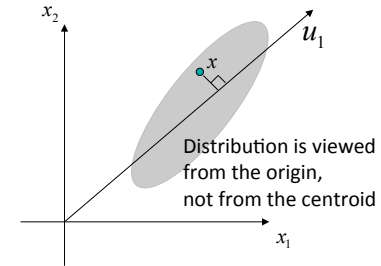
$$\text{Approximated } x' = \sum_{i=1}^p (x^T u_i) u_i$$

$$\|x'\|^2 = \|x\|^2 - \|\tilde{x}\|^2 \rightarrow \text{maximize}$$

The basis consists of p columns that are **eigenvectors** of the **autocorrelation matrix, Q** , corresponding to largest eigenvalues.

Principal Component Analysis (PCA)

2. Min. approximation error



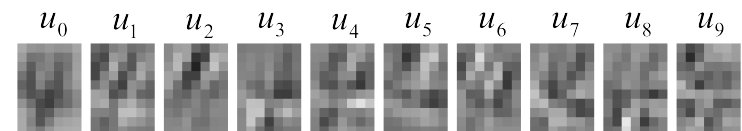
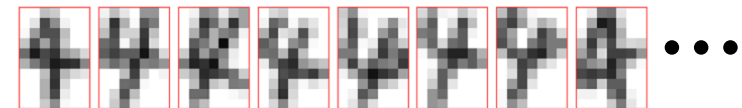
Autocorrelation matrix: $Q = E(xx^T)$

PCA example 1: Hand-written digits



Feature extraction

Pattern vectors: normalized & blurred patterns



(figure by Y. Kurosawa)

Example 2: Human face classification

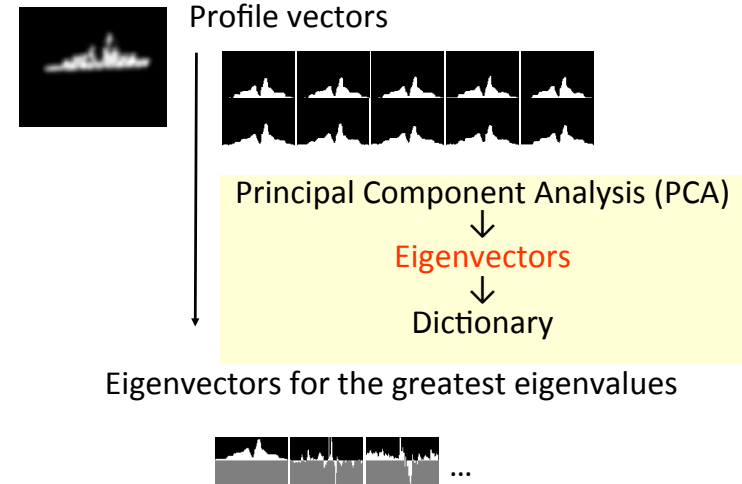
Basis vectors of a person: someone's *dictionary*



(Eigenvectors from a large collection of his face)

(figure by K. Fukui)

Example 3: Ship classification (profiles)



Concept of subspace

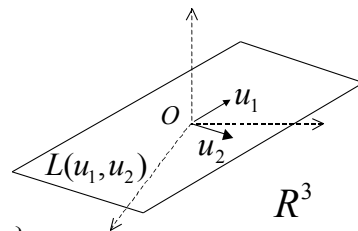
Subspace L is a collection of n -d vectors:
spanned by a basis, a set of linearly independent vectors

$$L(b_1, \dots, b_p) = \{z \mid z = \sum_{i=1}^p \xi_i b_i\} \quad (\xi_i \in \mathbb{R}, b_i \in \mathbb{R}^n)$$

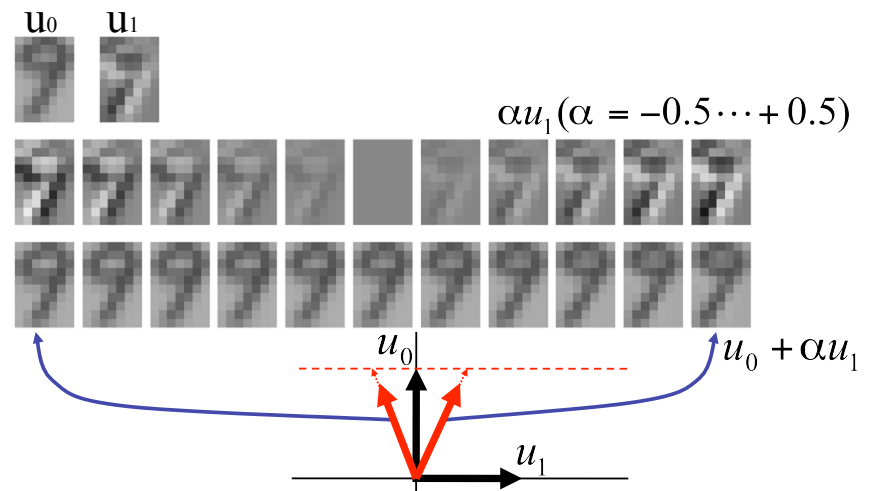
Dimension of a subspace:
the number of base vectors

$$p = \dim(L) \ll n$$

Conveniently represented
by orthonormal basis $\{u_1, \dots, u_p\}$

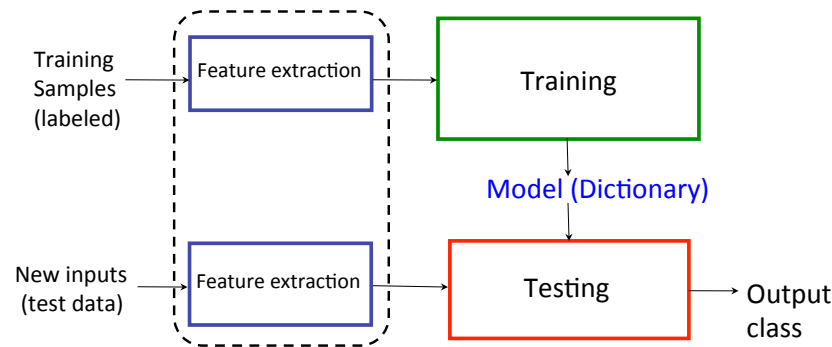


- Variations of "9" covered by a 2-d subspace



(figure by Y. Kurosawa)

Background: Schematic of classification



Testing phase

- Various ways to measure the distance
 - Euclidean / Mahalanobis distance
 - Angle between vectors
 - Projection length on **subspaces**
 - ...
- Classification methods
 - **Discriminant function**
 - **Subspace method**
 - ...

Training phase

- Given: Limited number of **labeled data**
(samples whose classes are known)
- The dimensionality often too high for limited number of samples

One approach to this is to find redundant variables and discard them, i.e. dimensionality reduction (without losing essential information)

Information compression: to extract the class characteristics and throw away the rest!

Nearest Neighbor methods (revisiting)

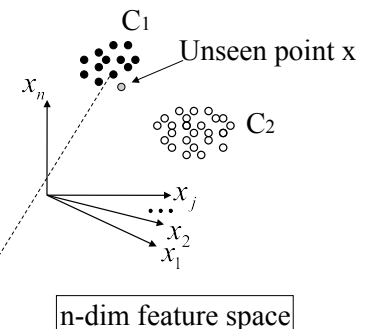
- Binary classification

- N_1 samples of class C_1
- N_2 samples of class C_2

- Unseen data x

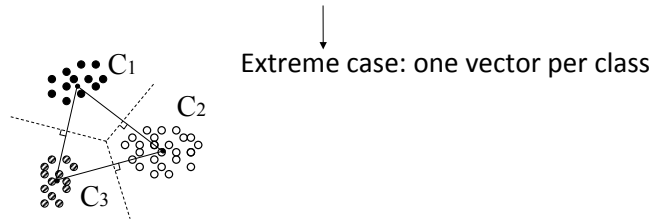
- Compute distances to $N_1 + N_2$ samples

- Find the **nearest neighbour**
→ classify x to the same class



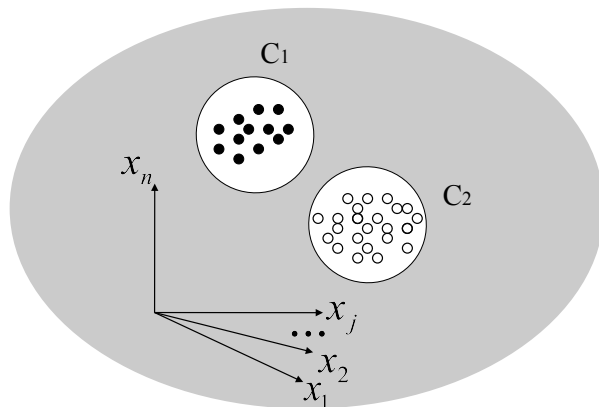
Discriminant function

- Need to remember all the samples?
 - In k -NN we simply used all the training data
 - Still cover only a small portion of possible patterns
- Define a class by a few representative patterns
 - e.g. the centroid of class distribution



Setting the “don’t know” category

- Reject if the distance is above the threshold



Formulation: one prototype per class

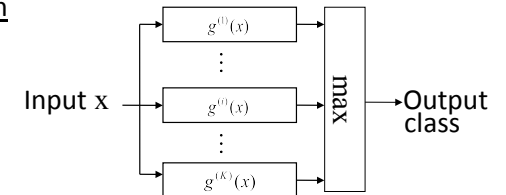
- K classes: $C^{(1)}, \dots, C^{(K)}$
- K prototypes: $a^{(1)}, \dots, a^{(K)}$

Consider **Euclidean distances** between the new input x and the prototypes: $\|x - a^{(i)}\|^2 = \|x\|^2 - 2a^{(i)T}x + \|a^{(i)}\|^2$

→ Choose the class that minimises **the distance**.

Discriminant function

$$g^{(i)}(x) \equiv a^{(i)T}x - \frac{1}{2} \|a^{(i)}\|^2$$



Direction cosine as similarity

Think of the new input and the prototype as vectors. Compute **cosine** between the input vector x and vector $a^{(i)}$

$$g^{(i)}(x) = \frac{(x^T a^{(i)})}{\|x\| \|a^{(i)}\|} = \cos A$$

“Simple similarity”

$0 \leq \cos^2 A \leq 1$ (The closer it is to 1, the more likely to be in $C^{(i)}$)

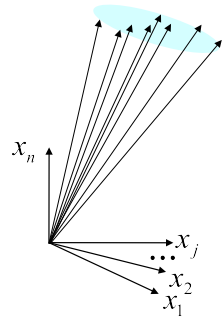
Now let’s extend the class representative to **a set of basis vectors** → spans a subspace

Subspace Methods

- Exploit localization of pattern distributions

Samples in the same class such as a digit (or face images of a person) are similar to each other.

They are localized in a subspace spanned by a set of basis u_i .



u_i : reference vectors
(orthonormal basis)

a.k.a CLAFIC
CLAss-Featuring Information Compression

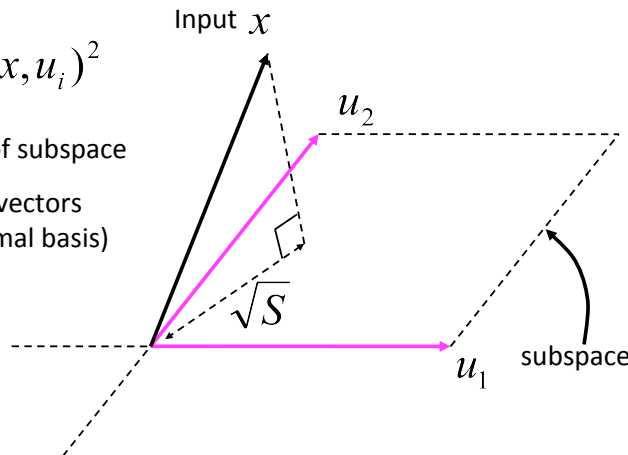
Similarity in Subspace Method

Projection length to the subspace

$$S = \sum_{i=0}^{p-1} (x, u_i)^2$$

p : dimension of subspace

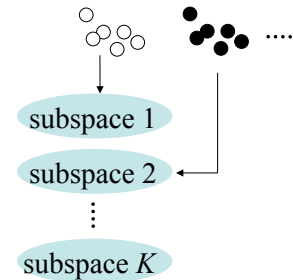
u_i : reference vectors
(orthonormal basis)



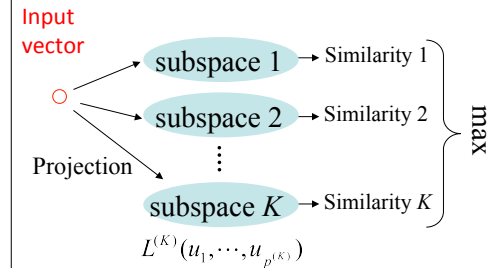
Framework of Subspace Method

- Training: for each class, compute a low-dimensional subspace that represents the distribution in the class.
 $\omega^{(1)}, \dots, \omega^{(K)}$
- Testing: determine the class of new unknown input by comparing which subspace best approximates the input.

Training



Testing

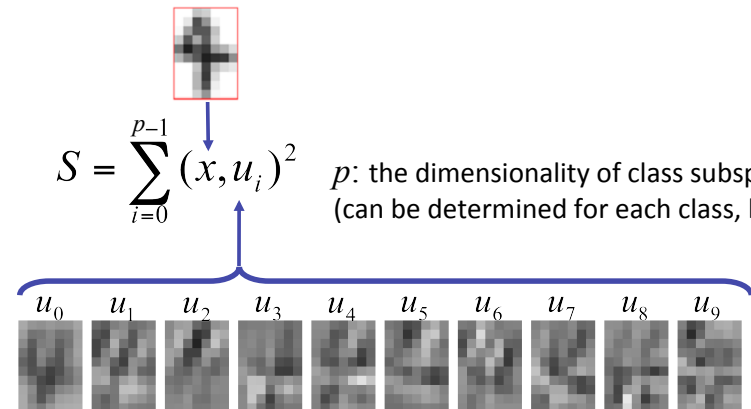


Similarity in Subspace Method (example)

Projection length to the subspace

$$S = \sum_{i=0}^{p-1} (x, u_i)^2$$

p : the dimensionality of class subspace
(can be determined for each class, how?)



Dimensionality of a class subspace

Eigenvalues of autocorrelation matrix Q : $\lambda_1 \geq \dots \geq \lambda_p \geq 0$

The number of dimensions to be used:

- Too low \rightarrow low capability to represent the class
- Too high \rightarrow issue of overlapping across classes

Cumulative contributions

$$a(p^{(i)}) = \frac{\sum_{j=1}^{p^{(i)}} \lambda_j}{\sum_{j=1}^p \lambda_j}$$

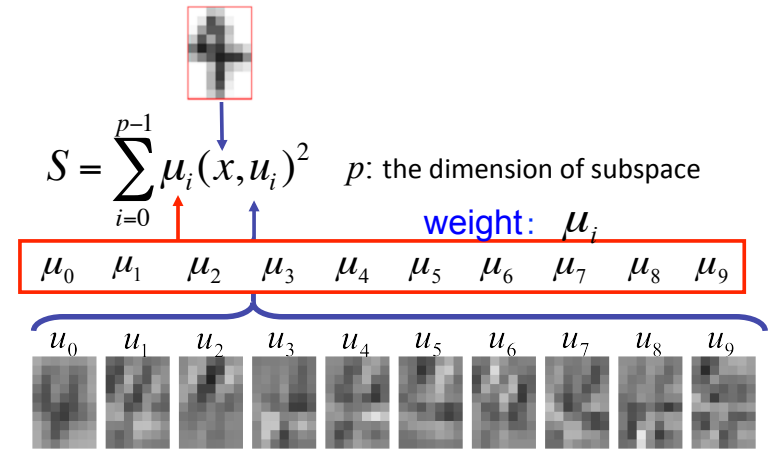
Choose a dimension $p^{(i)}$ for each class $\omega^{(i)}$

$$a(p^{(i)}) \leq \kappa \leq a(p^{(i)} + 1) \quad (\kappa: \text{common value})$$

The projection length to the subspace is made uniform.

Experiments still needed to find a good dimensionality

Similarity in weighted Subspace Method

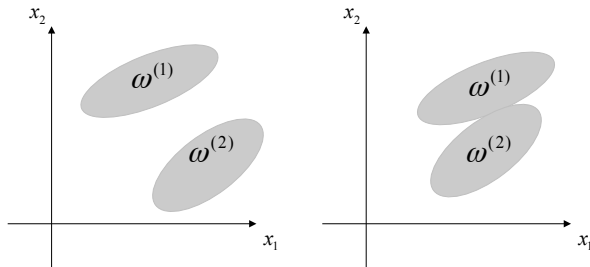


(figure by Y. Kurosawa)

Useful dimension for classification?

Ideal distributions of input pattern vectors:

- Patterns from an identical class be close
- Patterns from different classes be apart



\rightarrow Overlapping distributions harmful for classification

Ratio of between-classes variance to within-class variance

Within-class variance

$$\sigma_W^2 = \frac{1}{r} \sum_{i=1}^K \sum_{x \in \omega^{(i)}} (x - E^{(i)}(x))^T (x - E^{(i)}(x))$$

Total # of samples

Average in class $\omega^{(i)}$

Between-class variance

$$\sigma_B^2 = \frac{1}{r} \sum_{i=1}^K r^{(i)} (E^{(i)}(x) - E(x))^T (E^{(i)}(x) - E(x))$$

Number of samples in class $\omega^{(i)}$

Average overall

Within-class var. between-class var. ratio

$$J_\sigma = \frac{\sigma_B^2}{\sigma_W^2} \quad \begin{array}{l} \text{Between-class variance} \\ \text{Within-class var in ave} \end{array}$$

In short: distance between classes normalized by distance within class

\rightarrow the larger the better!