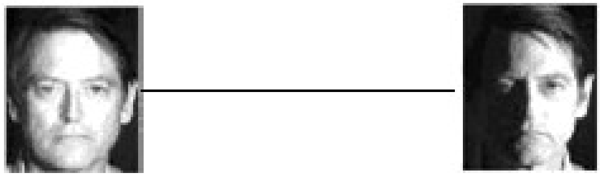
Face Authentication using Eigenfaces, Distance Classifiers and Support Vector Machines

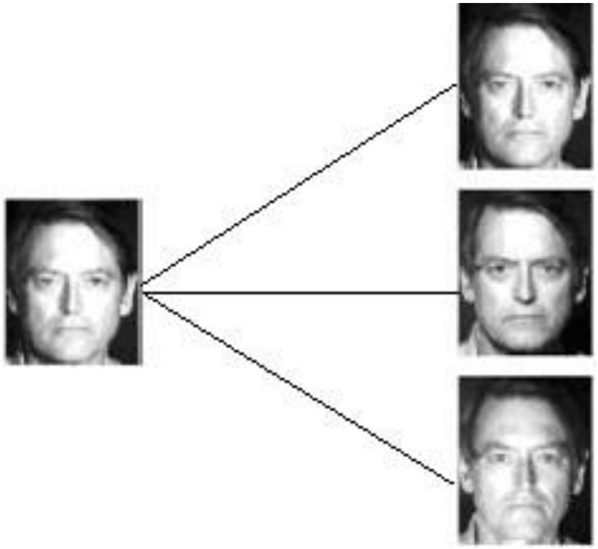
Ashutosh Modi
Arun Muralidharan
Shubhendu Trivedi

Project Guide: Dr (Mrs.) K. R. Joshi
Mentor: Mr Sumedh Kulkarni

Recognition is a Super-Set of Authentication

- Face Verification involves a one to one check that compares a query image with a template that the user claims to be



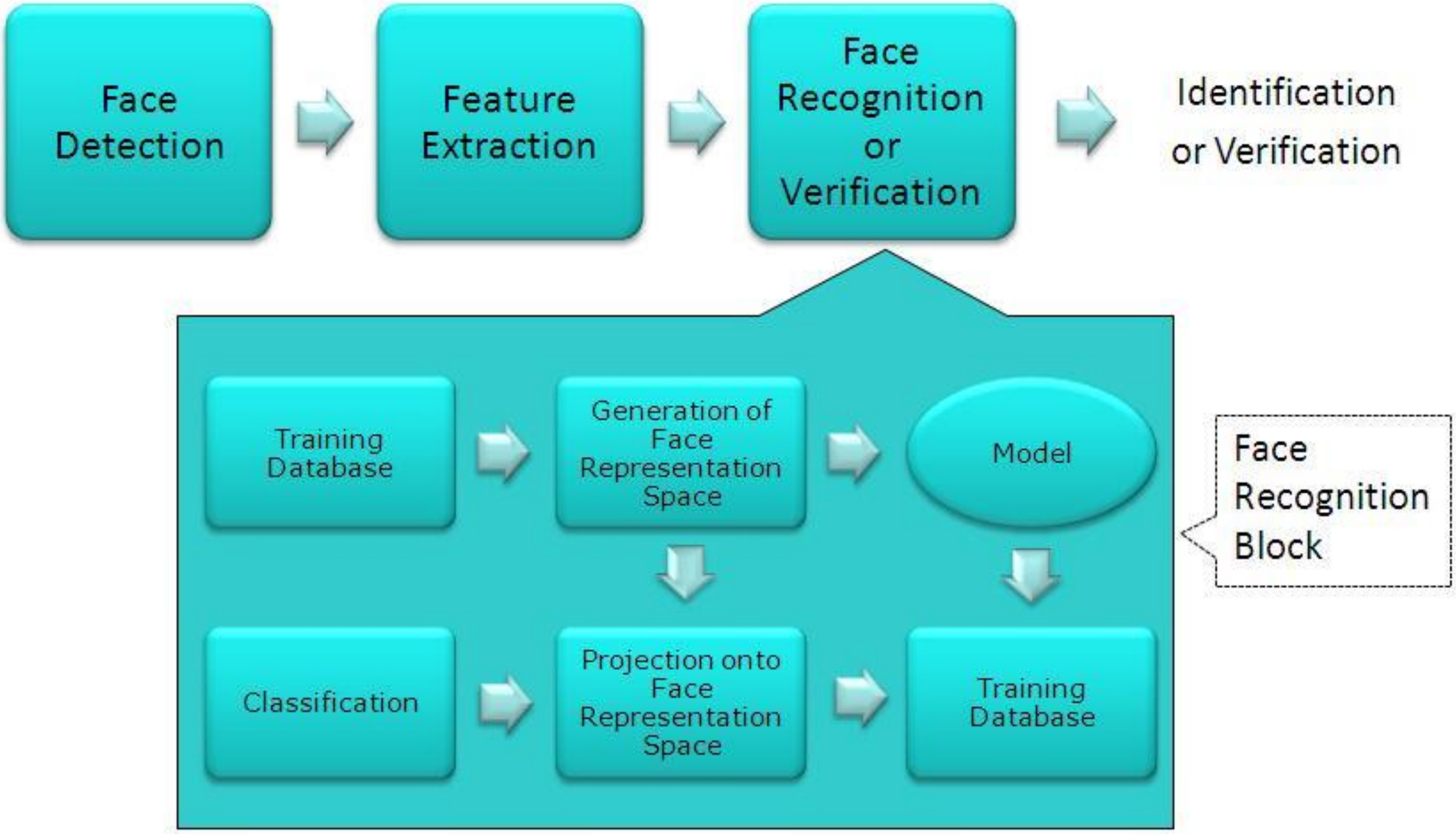


- Face Recognition involves a one to many comparison of a query image with a template library

Major Tasks

- ~~Face Detection/ Segmentation~~
- Feature Extraction
- Classification

Problem Overview


Face Detection
Feature Extraction
Face Recognition or Verification
Identification or Verification
Training Database
Generation of Face Representation Space
Model
Face Recognition Block
Classification
Projection onto Face Representation Space
Training Database

- **Generic face recognition/authentication system configuration**

Feature Extraction and Representation

Features in Face Recognition

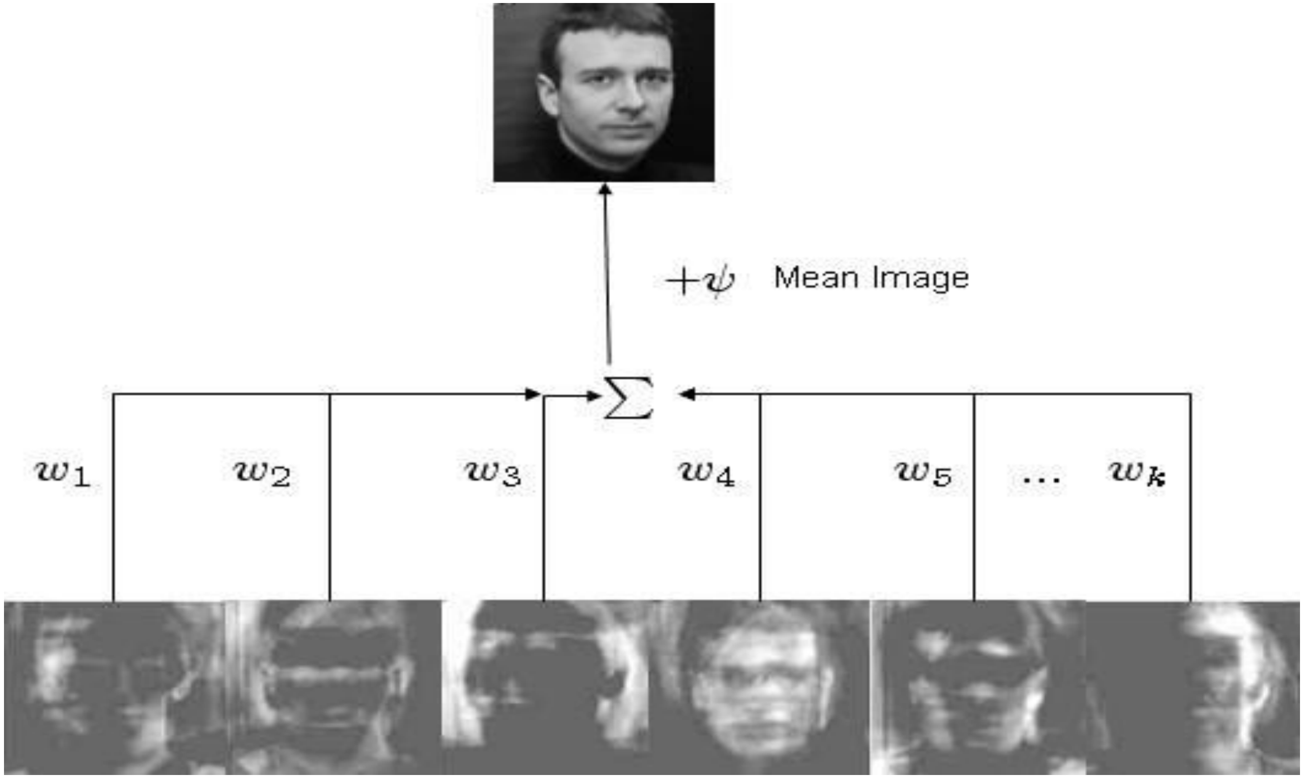
- Global Features (Appearance):
  - PCA
  - ICA
  - LDA
- Local Features:
  - Gabor Wavelets
  - Active Appearance Models (Model)
  - Elastic Bunch Graph (Model)

An Information Theory Approach

- Considers face recognition as a 2-D problem
- Involves encoding face onto some other space
- Use of both intuitive and non-intuitive features

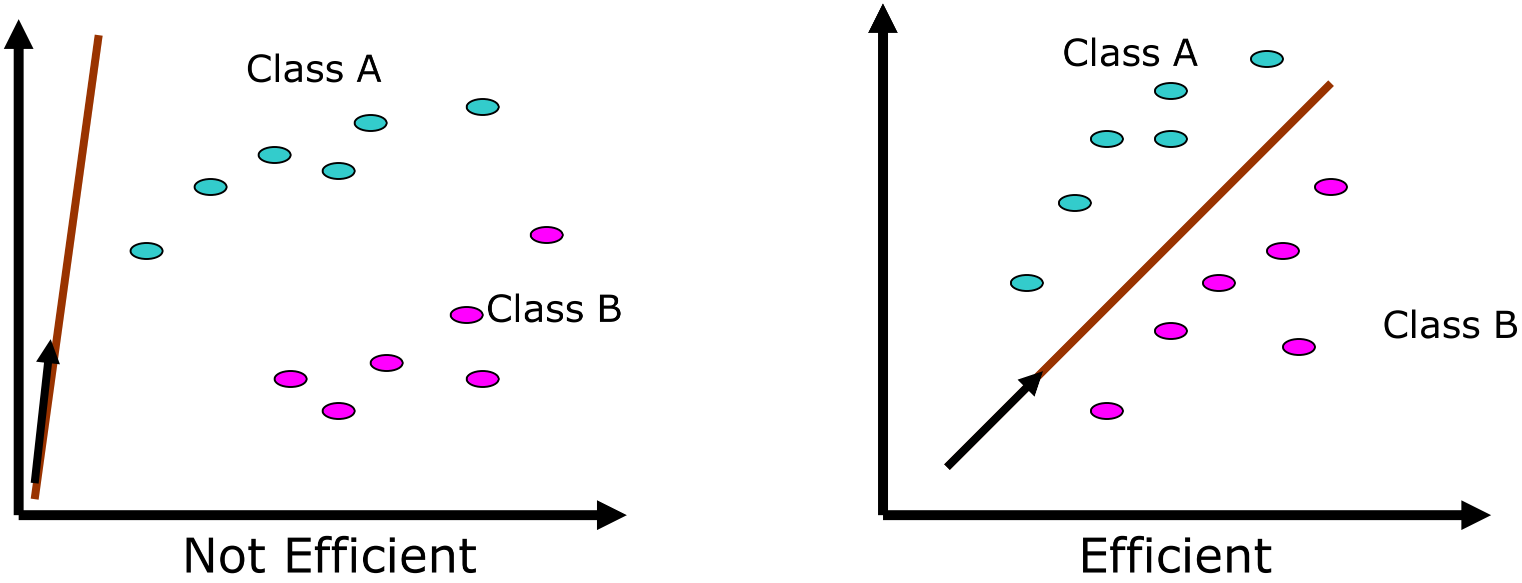
Eigen Faces : The Idea

- Representing a face as a weighted combination of "basis" faces. Idea similar to Fourier Series.



Eigen Faces : Basics

- An image is a point in high dimensional space. An N x N image is a point in R^{NXN}
- PCA seeks directions efficient for representing the data
- PCA reduces the dimensions



Eigen Faces : Finding Eigenvectors

- Obtain set of M training images

$M_1=$

$M_2=$

. . .

$M_N=$

Eigen Faces : Finding Eigenvectors

- Convert each face image into a vector (N x N matrix into a N^2 x 1 vector)

$$= \begin{Bmatrix} A_1 \\ A_2 \\ . \\ . \\ A_n \end{Bmatrix}$$

$$= \begin{Bmatrix} A_1 \\ A_2 \\ . \\ . \\ A_n \end{Bmatrix}$$

$$= \begin{Bmatrix} A_1 \\ A_2 \\ . \\ . \\ A_n \end{Bmatrix}$$

Eigen Faces : Finding Eigenvectors

$$I_i = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}_{N \times N} \xrightarrow{\text{concatenation}} \begin{bmatrix} a_{11} \\ \vdots \\ a_{1N} \\ \vdots \\ a_{2N} \\ \vdots \\ a_{NN} \end{bmatrix}_{N^2 \times 1} = \Gamma_i$$

Eigen Faces – Finding Eigenvectors

- Compute the average face *Ψ*

$$\psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma i$$

- Subtract mean face from each face vector to obtain Φ

$$\Phi i = \Gamma i - \psi$$

Eigen Faces – Finding Eigenvectors

- Compute the Covariance matrix C = AA^T using:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_N \Phi_N^T$$

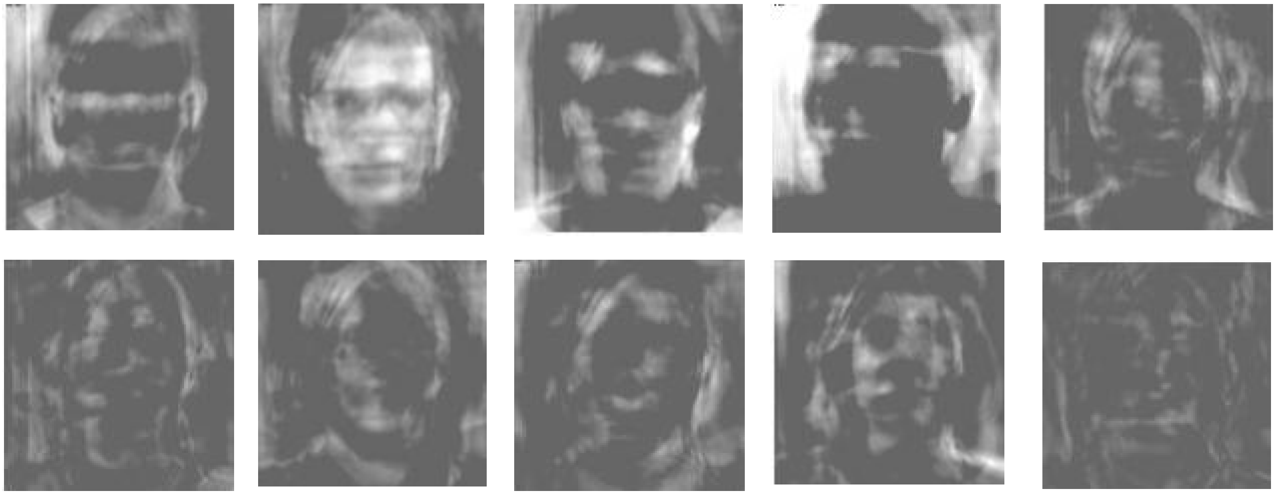
  - A can be given as $[\Phi_1, \Phi_2 \ldots \Phi_M]$
  - Covariance is a matrix of $N^2 x N^2$ while A of N x M

Eigen Faces – Finding Eigenvectors

- Compute Eigenvectors v_i of AA^T ($N^2 x N^2$)
- Computationally too expensive
- Compute Eigenvectors u_i of A^TA instead (MxM)
- $v_i=A\ u_i$
- Keep 'k' most significant eigenvectors

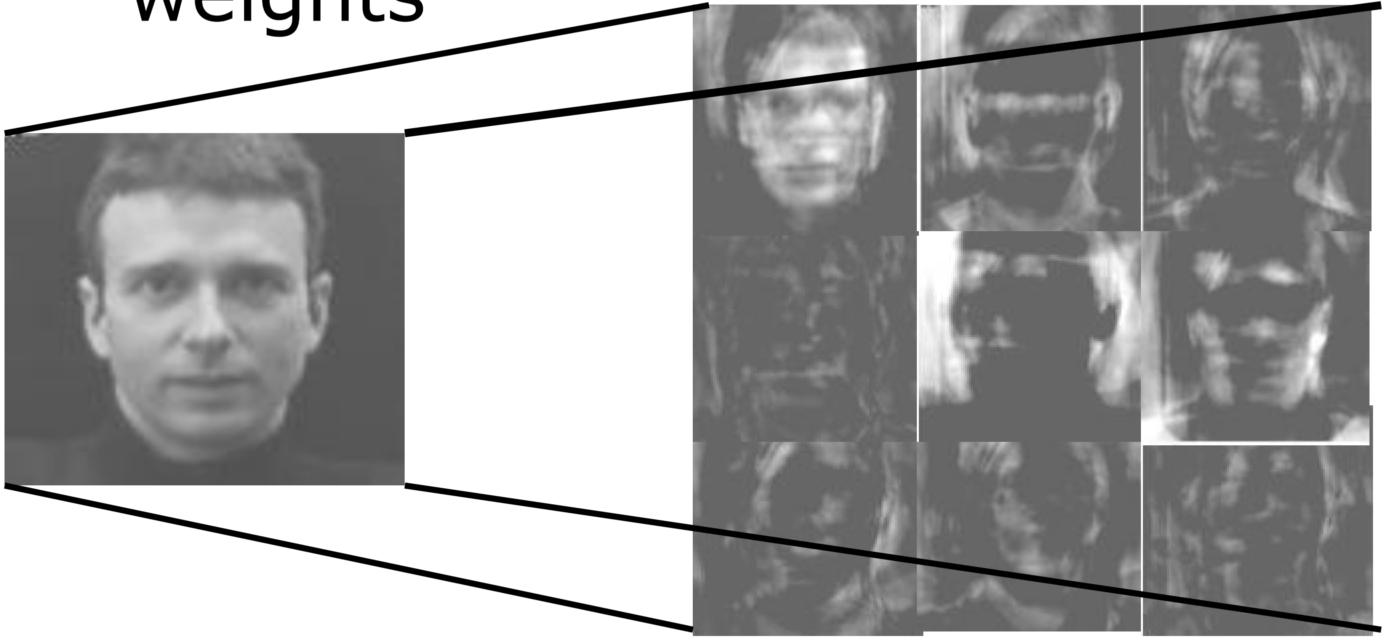
Eigen Faces – Finding Eigenvectors

- Eigenvectors obtained have some component of each face and look face like. Hence these are called Eigenfaces.

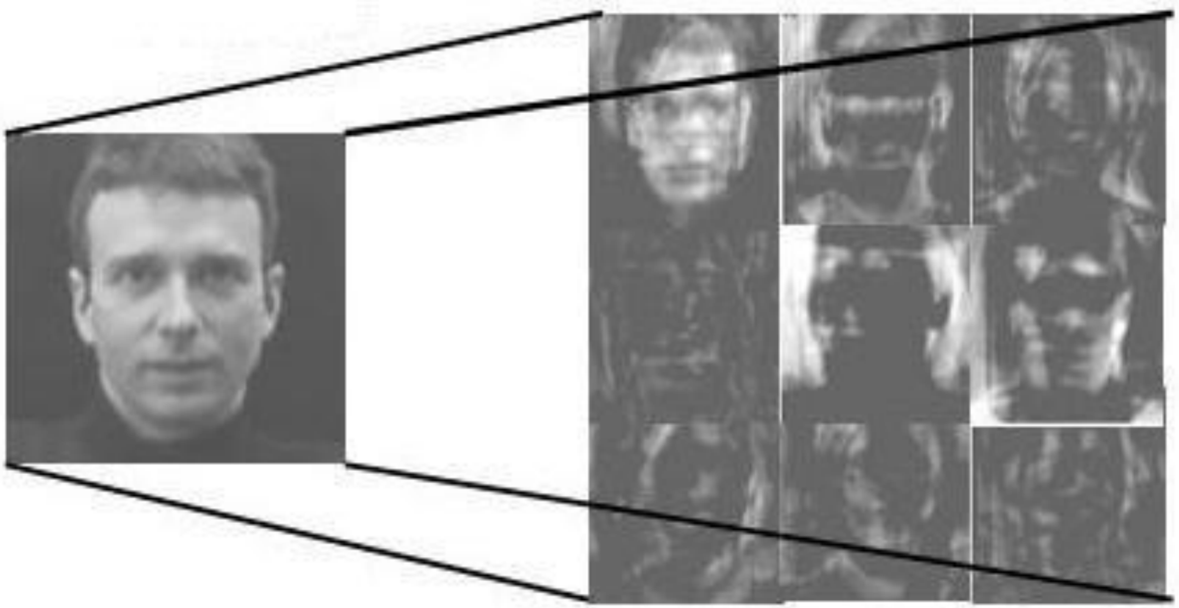


Eigen Faces – Finding Weights

- Each face is projected onto the eigen-space to find out associated weights



Eigen Faces – Finding Weights



- This can be calculated as:

$$\omega_k = u_k^T(\Gamma - \Psi)$$

Final Face Representation

- Each face is represented as a vector of weights

$$\Omega^T = [\omega_1, \omega_1 \dots\dots, \omega_M]$$

- This feature vector is an information theoretic feature and captures intuitive as well as non intuitive face features

Classification Task

Classification Methods Used

- Distance Classifiers
  - City-Block Distance
  - Euclidean Distance
  - Mahalanobis Distance
- Support Vector Machines

Distance Measures

- City- Block Metric

$$||x - y||_{c-b} = \sum_{i=1}^{D} |x_i - y_i|$$

- Euclidean Distance: Special case of the Minkowski Metric

$$\|x - y\|_e = (\sum_{i=1}^{D} |x_i - y_i|^2)^{1/2}$$

Distance Measures

- Mahalanobis Distance:
  - Takes in to account covariance between variables.
  - Eliminates problems of scale and correlation inherent to Euclidean norm

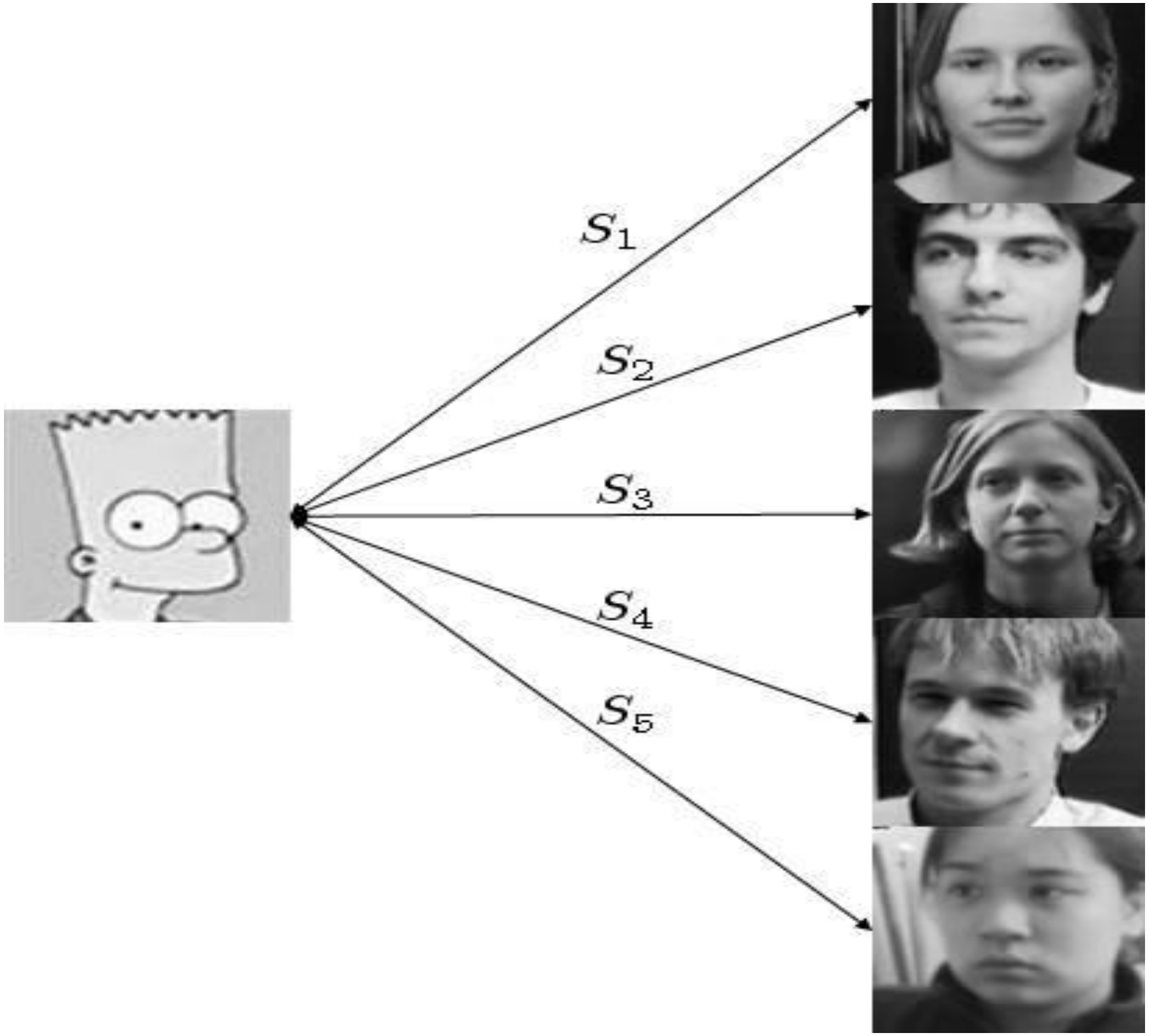
$$d(x,y) = ((x-y)^T C^{-1} (x-y))^{1/2}$$

Distance Measures : Classification

- Find distance measure of incoming probe image feature vector with every image feature vector in the database
- Choose the face for which

  $e_r = \min_l ||\Omega - \Omega^l||$
- Decide threshold Θ empirically.
  - If $e_r < \Theta$ recognise the probe image as best match
  - If $e_r > \Theta$ probe image is not in data-base

Need for Threshold



Support Vector Machines

Feature Representation

- Each Individual is a class and distribution of each face is approximated
- This makes recognition a K class problem
- This would formulate our problem in a difference space which captures dissimilarities between two images

Feature Representation

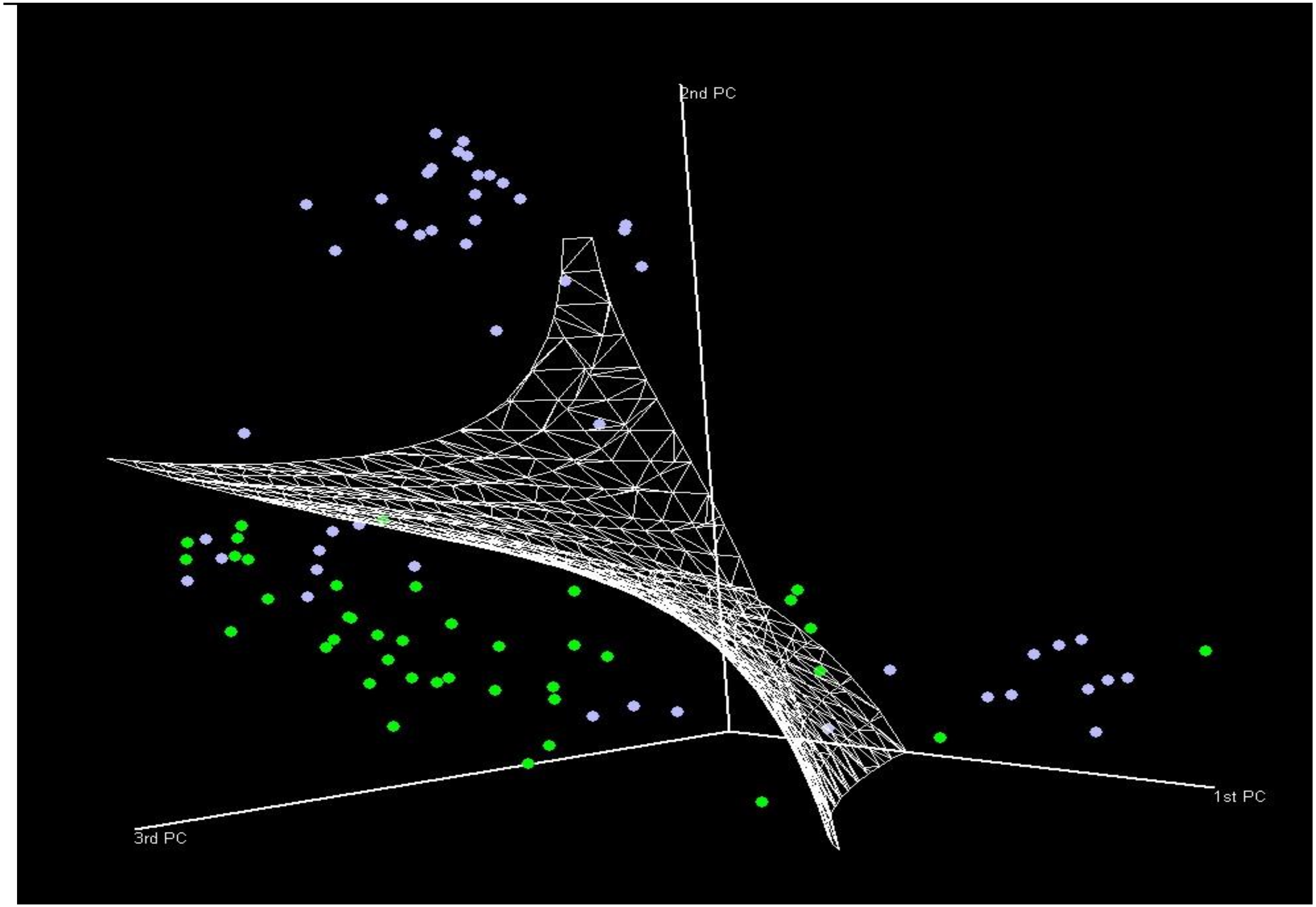
- Define two classes:

$$C_1 = t_i \; i \; t_j \, j t_i \; v \; t_j$$

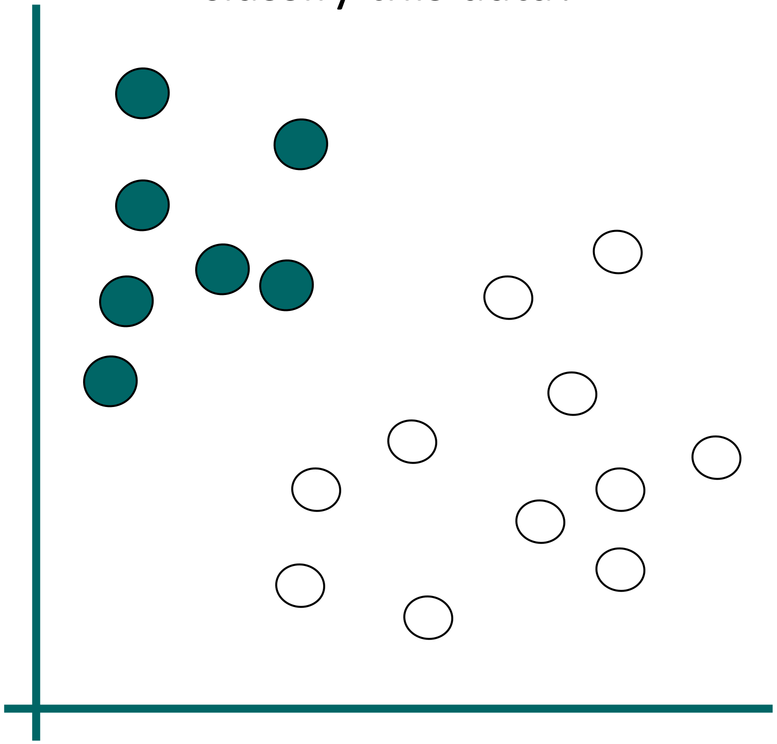
$$C_2 = t_i \; i \; t_j \, j t_i \; ¿ \; t_j$$

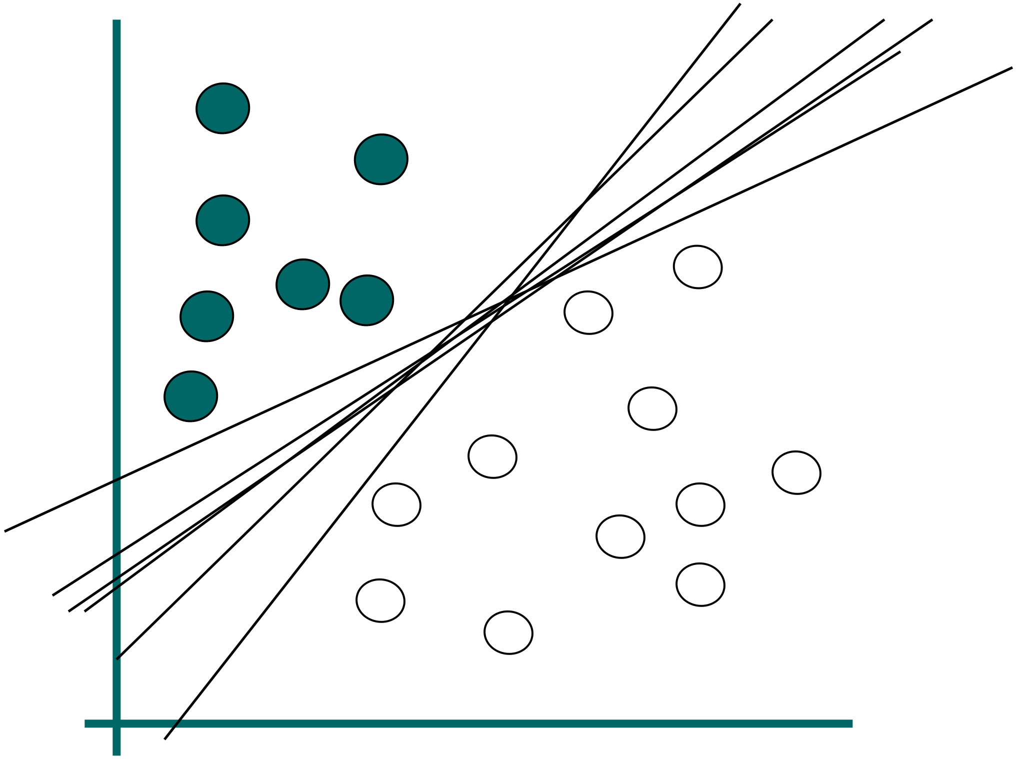
- Classes C1 and C2 are inputs to the SVM algorithm which will generate a decision surface
- Thus basically given two images p1 and p2 the classifer estimates if they are of the same person

Support Vector Machines



How would you classify this data?

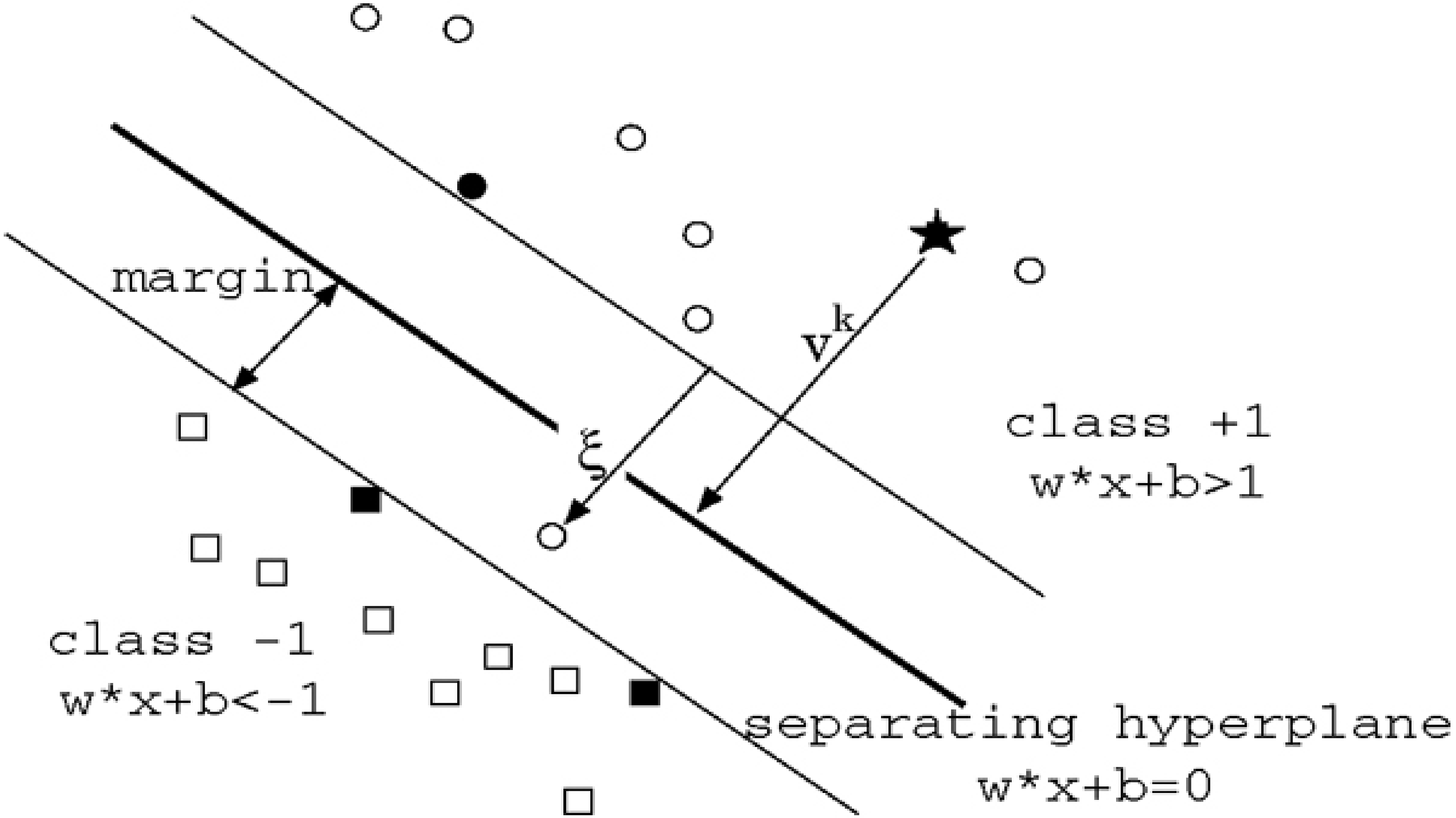




Intuitions about Classification

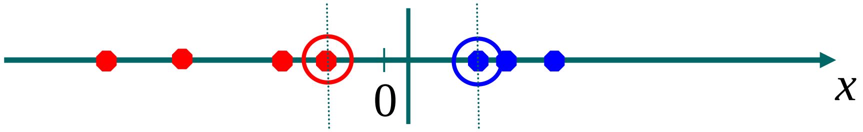
- Confidence in correct Prediction
- Highly separated data set

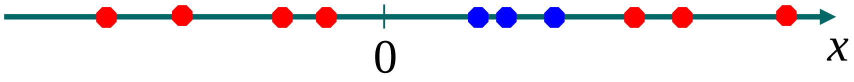
SVM: Intuitions about Classification



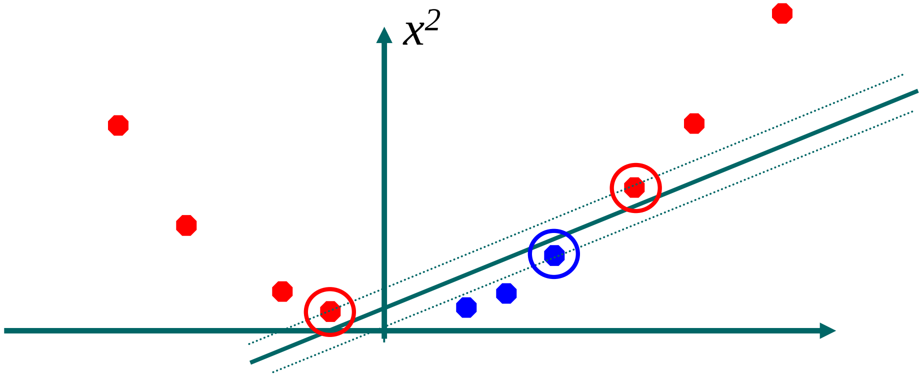
Non-Linear SVMs

- Datasets that are linearly separable with some noise work out great:

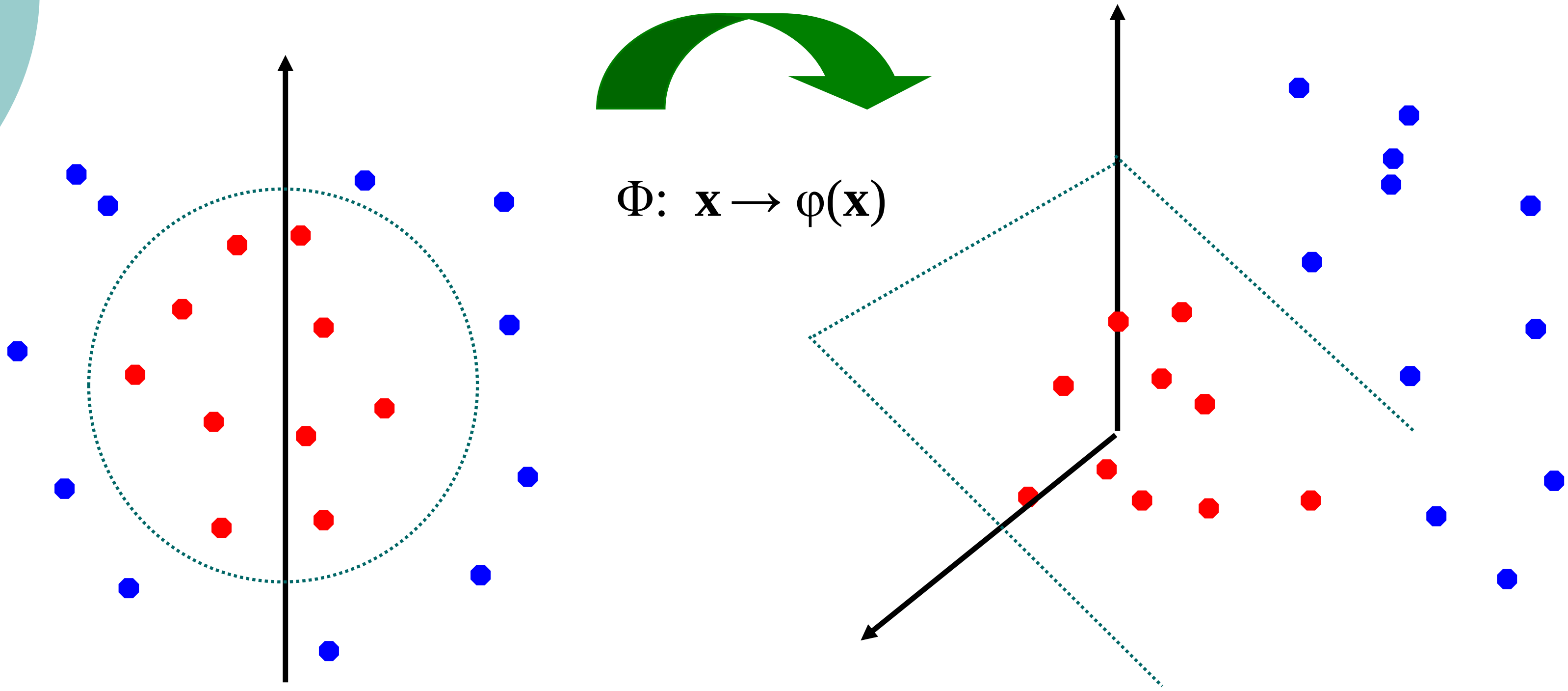




- Map the data to a higher dimensional space



- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



$\Phi\text{:}\ \mathbf{x} \rightarrow \varphi(\mathbf{x})$

Support Vector Machines

- Use Structural Risk Minimization giving better generalization
- Optimization is guaranteed
- Low VC Dimension
- Computational Cycles are lesser as compared to A.N.N
- Best off the shelf learning algorithm for classification and regression problems

Popular Mercer Kernels

- Radial Basis Functions
- Linear Kernels
- Polynomial Kernels
- Multiple Kernels

Final Optimization Problem (General Case)

$$\max_{\alpha} \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0,$$

Steps to apply SVM

- Conduct scaling on data if needed
- Use RBF Kernel first
- Use Cross-Validation to fit parameters C and γ
- Use these values of best parameters to train the whole training set
- Test for images in the designated test set.

Cross Validation

- Follows from Learning Theory
- Choosing hypothesis with low training error might be risky
- Allows to choose hypothesis from hypotheses class H with best generalization error and avoids over-fitting

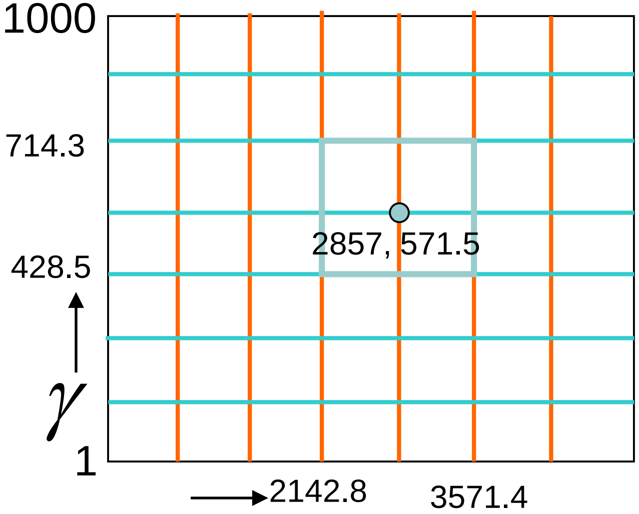
Cross Validation Methods

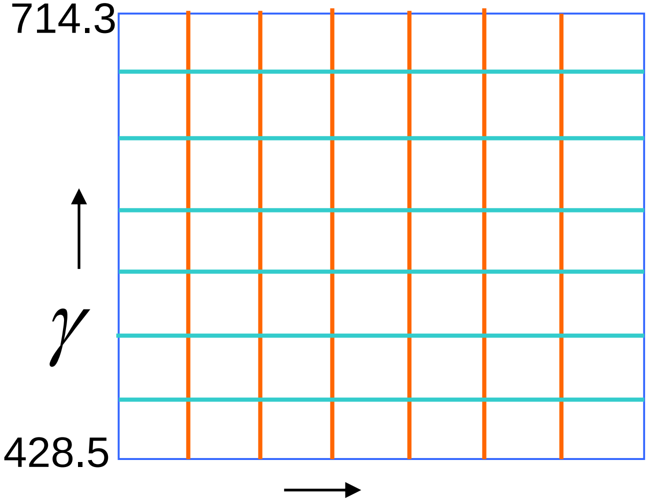
- Hold-Out Cross Validation
  - Wastes training data
  - Suited when training set is large
- K-Fold Cross Validation
  - Utilizes Data better
  - Suited for medium sized training sets
- One Hold Out Cross Validation
  - Uses data best
  - Suited when training data is scarce
  - Computationally very expensive

Hold - Out Cross Validation

- Randomly split training set into disjoint test and training sets (Ratio 25%,33% to 75,67% recommended)
- Train the hypotheses class H on this training set
- Test H on this test set and select h with least generalization error
- Go back and train h on the entire training set

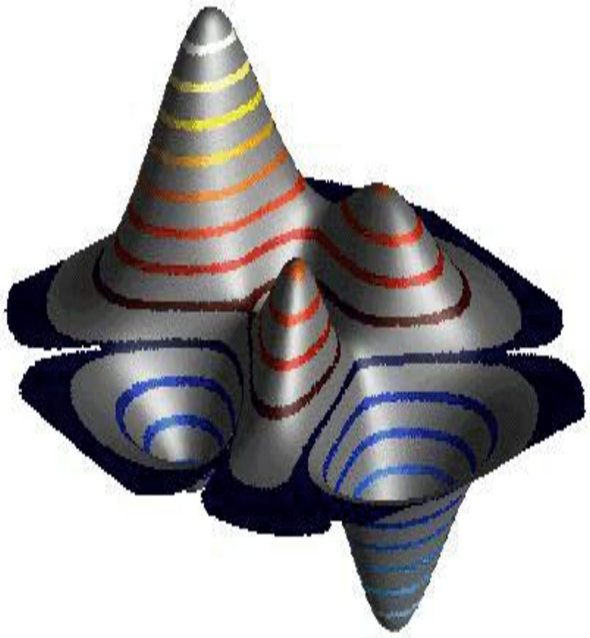
Grid Search

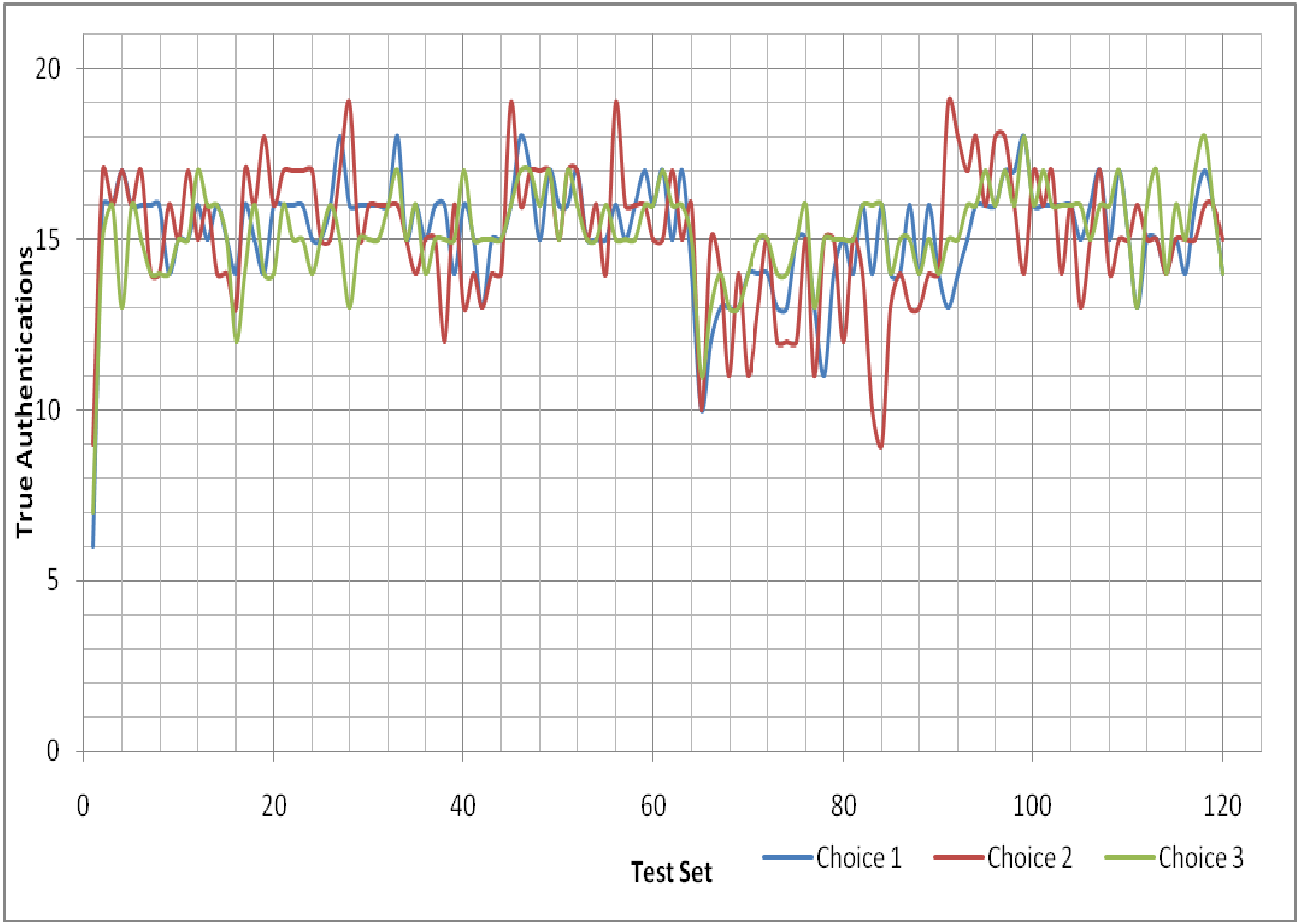




Random Search With Multiple Restarts

- Choose Random point on C/Gamma Axes.
- Find Optima in neighborhood
- Choose another random point
- Repeat process and update optima




20
15
10
5
0
True Authentications
0
20
40
60
80
100
120
Test Set
Choice 1
Choice 2
Choice 3

Training Algorithms

- Projected Conjugated Gradient Chunking Algorithm
- Oslun’s Algorithm
- Sequential Minimal Optimization Algorithm

Recognition Task

- Let the incoming probe be P
- Compute similarity score of P with each of the gallery images

$$\pm_j = \overset{N_s}{\underset{i=1}{X}} \text{®}_i y_i k(s_i g_j \text{¡} p) + b$$

- Recognize probe as person j that has minimum similarity score
- Decide threshold heuristically

Experimental Results

- Databases Tested on:
  - MIT-CBCL (For Distance Classifiers)
  - Non-Standard (Online + Offline)

Experimental Results

Image (Test Set) (In bracket – Actual Image)	Identified as (City-Block)	Identified as (Euclidean)	**Identified as (Mahalanobis)**	**Identified as (SVM)**
1. (9)	9	9	**9**	**9**
2. (2)	2	2	**2**	**2**
3. (-)	6	8	**10**	**10**
4. (-)	9	9	**10**	**9**
5. (3)	3	3	**3**	**3**
6. (1)	1	1	**1**	**1**
7. (10)	10	10	**10**	**10**
8. (10)	10	10	**10**	**10**
9. (5)	5	5	**5**	**5**
10 (6)	6	6	**6**	**6**

Further Work

- Testing on a challenging database
- Using a parallel and/or cascade combination of Gabor Features and Eigenfaces
- Statistically constructing a random image set
- Designing a custom Kernel
- Investigating a multiple Kernel

References

- Vladimir Vapnik, Feature Selection for SVM, Neural and information processing systems proceedings.
- Marti Hearst, Support Vector Machines, IEEE Intelligent Systems
- C.J.C Burges, A Tutorial on Support Vector Machines, Lucent Bell Labs.
- Maria trias, A survery of face verification methods. Project Thesis.
- K. Jonsson, J. Matas, SVM, CVSSP.
- Andrew Ng, CS-229, Stanford Center for Professional Development.
- Lindsay Smith, A tutorial on PCA.
- Jonathon Shlens, Systems Neurobiology Laboratory, Salk Insitute for Biological Studies La Jolla, CA 92037.
- Vladimir Vapnik, Statistical Learning Theory, Springer Varlag, 1995

References

- Vladimir Vapnik, Support Vector Machines. Springer 1998
- Bernhard Scholkopf, Alexander J. Smola, Learning With Kernels, The MIT press.
- Lipo Wang, Advances in Pattern Recognition, Springer 2003.
- Graf, Borer, Normalization in Support Vector Machines, Eurpoean Conference on Computer Vision.
- Wu, Liu, Shum, Auto eye glass Removal in face recognition, IEEE Pattern Recognition and Machine Intelligence
- He, Yan, Hu, Niyogi, and Zhang, Face recognition using laplacianfaces, IEEE transactios on Pattern analysis and machine intelligence, March 2005.
- L.Wiskott, J. andN, Kr¨uger, and C. D. Malsburg, “Face recognition by elastic bunch graph matching,” IEEE Transactions on Pattern analysis and machine intelligence, July 1997.

References

- R. Duda, P. Hart, and D. Stork, Pattern Classification (2nd Edition). Wiley-Interscience, 2000.
- P. Belhumeur, J. P. Hespanha, and DJ.Kriegman, “Eigenfaces vs fisherfaces: Recognition using class specific linear projection,” IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 1997.