Final Project

- Implement the Mahalanobis Subspace Classifier
- Use Three Different Real Data Sets
- Values of Features are Real Numbers
- Dimension of the tuples should be between 10 and 20
- Number of Classes should be between 4 and 10
- Number of Tuples Per Class in Training Set should be greater than 10 times the number of parameters per class needed by the classifier
- Test Set and Training Set should be about the same size

Use Only Essential Features For Each Class

- Let there be N features and C be the set of K classes
- Let L_n be the range set for the n feature
- Let $\mathcal{M} = \times_{n=1}^{N} L_n$ be the measurement space.
- Shuffle the class tagged data sequence
- Split the shuffled class tagged data sequence into a Training Set and Test Set
- Using the Training set, take the features one by one and build a Mahalanobis Subspace classifier using the Training Set
- Use the Test Set to determine how well the classifier does with each measurement space component

Indexed Relation List Notation

- $I = \{1, \ldots, N\}$ is the index set for the features
- $R = \langle x_1, \ldots, x_Z \rangle$ is the training measurement sequence
- (I, R) is the indexed relation list for the training measurement space sequence
- Let $J \subset I$, $|J| = T$
- Then $\pi_I(I, R) = (J, S)$ where
  - $S = \langle y_1, \ldots, y_Z \mid y_z = (y_{z1}, \ldots, y_{zT}) = (x_{zt} : t \in J) \rangle$

Projecting

- Suppose there are 6 features
- We wish to project to features 1,4, and 5; $Q = \{1, 4, 5\}$
- The relative coordinate projection of tuple $x = (x_1, x_2, x_3, x_4, x_5, x_6)$ is (x_1, x_4, x_5)
- We can write the projection in a general way
    - Let Q be the index list of essential features
    - In the example above $Q = \{1, 4, 5\}$
    - The indices are ordered in ascending order
    - The relative coordinate projection of tuple $(x_1, \ldots, x_N)$ onto the essential features specified in Q is given by
    - $\pi_Q(I, (x_1, \ldots, x_N)) = (Q, (x_i : i \in Q))$

Use Only Essential Features For Each Class

- Let $I = \{1, \ldots, N\}$ be the index set for the features
- Let $d : I \times (I \times \mathcal{M}) \rightarrow C$ be the class that the classifier assigns to $x \in \mathcal{M}$ when only using feature i
- Let $d(i, \pi_{\{i\}}(I, x))$ be the class that the classifier assigns to x when only using feature $i \in I$
- Let $\{\langle x_1, \ldots x_Z \rangle, \langle c_1, \ldots, c_Z \rangle\}$ be the Test Set

Using the test set calculate an N row by K column table $A : I \times C \rightarrow \mathbb{R}$ defined by

$$A(i \mid c) = \frac{|\{z \in [1, Z] \mid d(i, \pi_{\{i\}}(x_z)) = c = c_z \text{ when feature } i \text{ is used}\}|}{|\{z \in [1, Z] \mid c_z = c\}|}$$

Given class c, $A(i \mid c)$ is the estimated probability of correct identification when using feature i

Project To Essential Features

- Suppose there are 6 features
- Features 1,4, and 5 are essential; $Q = \{1, 4, 5\}$
- Relative Coordinate Project to components 1,4 and 5
- The relative coordinate projection of tuple $x = (x_1, x_2, x_3, x_4, x_5, x_6)$ is (x_1, x_4, x_5)
- We can write the projection in a general way
  - Let Q be the index list of essential features
  - In the example above $Q = \{1, 4, 5\}$
  - The indices are ordered in ascending order
  - The relative coordinate projection of tuple $(x_1, \ldots, x_N)$ onto the essential features specified in Q is given by
  - $\pi_Q(I, (x_1, \ldots, x_N)) = (Q, x_i : i \in Q)$

Project To Subspace

- Let $0 < \theta << 1$ be the threshold that determines when classification accuracy is large enough
- If for any class c and feature index i, $A(i \mid c) > \theta$, then feature i can be used in playing a role in the classification for class c
- $J_c = \{i \in I \mid A(i \mid c) > \theta\}$
- The subspace for class c is then indexed by J_c
- The new training set is then $\{\langle \pi_{J_c}(I, x_1), \pi_{J_c(I,x_2)}, \ldots \pi_{J_c(I,x_z)} \rangle, \langle c_1, \ldots, c_Z \rangle\}$

Class Mean and Covariance

- Organize the Training set by class
- After omitting the non-essential features, the number of dimensions for y from class k is N_k
- $R_k = \langle y_{k1}, \ldots y_{kZ_k} \rangle$ training set for class k; $y_{kz}^{N_k \times 1}$

$$\mu_k = \frac{1}{Z_k} \sum_{z=1}^{Z_k} y_{kz}$$

$$\Sigma_k = \frac{1}{Z_k - 1} \sum_{z=1}^{Z_k} (y_{kz} - \mu)^{N_k \times 1} (y_{kz} - \mu)'^{1 \times N_k}$$

Finding Subspaces

- For each class c
- Use the relative coordinate projection of the training set to determine the class covariance matrix Σ_c
- Use Σ_c to do a Principle Components
- E_c is selected so that the eigenvalue fraction

$$\frac{\sum_{n=1}^{E_c} \sigma_n}{\sum_{n=1}^{N} \sigma_n}$$

  is just greater than the user specified fraction f
- Define the class subspace to be the span of the first E_c eigenvectors of Σ_c

Covariance Matrix of $y^{E_c \times 1}$

- $y^{E_c \times 1} = S_c' x$
- We need its covariance matrix so that we can use its inverse in the Mahalanobis distance calculation

$$\begin{aligned}
\Sigma_y &= S_c' \Sigma S_c \\
&= S_c' (T_c \Lambda T_c') S_c \\
&= (S_c' T_c) \Lambda (T_c' S_c) \\
&= \begin{pmatrix} I^{E_c \times E_c} & 0^{E_c \times N - E_c} \end{pmatrix} \Lambda^{N \times N} \begin{pmatrix} I^{E_c \times E_c} \\ 0^{N - E_c \times E_c} \end{pmatrix} \\
&= Diagonal(\lambda_1, \lambda_2, \ldots, \lambda_{E_c})
\end{aligned}$$

The inverse covariance matrix Σ_y^{-1}

$$\Sigma_y^{-1} = Diagonal(\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_{E_c}^{-1})$$

Class c Mahalanobis Distance and P-value for y

$$d_c^2(y) = y' Diagonal(\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_{E_c}^{-1}) y$$

- $\langle y_{c1}, \ldots, y_{cZ_c} \rangle$ Projected Training Measurement Sequence
- $\langle d_c^2(y_{c1}), \ldots d_c^2(y_{cZ_c}) \rangle$ The Mahalanobis distances
- Sort in Ascending order $\langle d_{(1)}^2, \ldots d_{(Z_c)}^2 \rangle$
- Test Set y
- Mahalanobis distance to mean of class c $d_c^2(y)$
- If $d_c^2(y) < d_{(1)}^2$ p-value $p_c(y) = \frac{1}{Z_c}$
- If $d_c^2(y) > d_{(Z_c)}^2$ p-value $p_c(y) = 1$
- If $d_{(z-1)}^2 \leqslant d_c^2(y) < d_{(z)}^2$, p-value $p_c(y) = \frac{z - .5}{Z_c}$

Class Assignment

- Select a p-value $.5 < p_0$
- If $p_c(y) > p_0$, y cannot be assigned to class c
- Assign y to allowable class c for which $p_c(y)$ is minimal

Review 1

- For class c
- Leave out components of x whose associated classification accuracy is too low
- The covariance matrix of the reduced x is Σ_c
- Eigenvector Eigenvalue decomposition $\Sigma_c = T_c \Lambda_c T_c'$
- Choose a fraction f of the variance to be preserved
- E_c is the smallest number satisfying $\frac{\sum_{n=1}^{E_c} \lambda_{cn}}{\sum_{n=1}^{N} \lambda_{cn}} \geqslant f$
- Define S_c to be the first E_c columns of T_c
- $y = S_c' x$
- y has covariance matrix $S_c' \Sigma_c S_c = Diagonal(\lambda_{c1}, \ldots \lambda_{cE_c})$
- The inverse covariance matrix is $Diagonal(\lambda_{c1}^{-1}, \ldots, \lambda_{cE_c}^{-1})$

Review 2

- Squared Mahalanobis distance
  - $d_c^2(y) = y' \mathit{Diagonal}(\lambda_{c1}^{-1}, \ldots, \lambda_{cE_c}^{-1}) y$
- Training sequence for class c, $\langle x_{c1}, \ldots x_{cZ_c} \rangle$
- $y = S_c' x$ Take the projection of x to the relative coordinates of S_c
- Compute Squared Mahalanobis $d_c^2(y_{c1}), \ldots d_c^2(y_{cZ_c})$
  - Ascending order $\langle d_{(1)}^2, \ldots d_{(Z_c)}^2 \rangle$
  - Select $p_0 = \frac{z}{Z_c}$, $z > Z_c/2$
  - $s_{critical,c}(p_0) = d_{(Z_c - z)}^2$
- New x, $y = S_c' x$
- If $d_c^2(y) > s_{critical,c}$ class c is not allowable for x
- Assign x to allowable class c for which $p_c(y)$ is minimal
- If there is no allowable class, assign x to reserve class

Experimental Protocol

- There are parameters
  - θ threshold for classification accuracy of single feature
  - Fraction f for eigenvalue ratio
  - p_0 threshold probability for disallowing a class
- For each class of each data set plot the identification accuracy and the fraction of reserve decisions as a function of the parameters
  - Fix a value for two of them and then plot accuracy versus the third parameter value
  - Change the value for two of them and then plot accuracy versus the third parameter value
  - Write out the confusion matrix (with a reserved decision column)

Confusion Matrix With Reserved Decision

		ASSIGNED						
		1	2	$\cdots$	k	$\cdots$	K	r
	1			$\cdots$		$\cdots$		
	$\vdots$							
TRUE	j			$\cdots$	$P_{TA}(j,k)$	$\cdots$		$P_{TR}(j,r)$
	$\vdots$			$\vdots$		$\vdots$		
	K			$\cdots$		$\cdots$		

Reject Rate: $P(Reject) = \sum_{c \in C} P_{TR}(c,r)$ $\quad$ $P(Correct|Read) = \dfrac{\sum_{c \in C} P_{TA}(c,c)}{1 - P_R(r)}$

Read Rate: $P(Read) = 1 - P_R(r)$ $\quad$ $P_{R|T}(r \mid c) = \dfrac{P_{TR}(c,r)}{P(c)}$

Experimental Protocol

- For each data set decide what the best parameters are
  - Report the confusion matrix (with a reserved decision column)
  - For each class c, report the class conditional identification accuracy $P_{A|T}(c \mid c) = \frac{P_{AT}(c,c)}{P(c) - P(r \mid c)}$
  - Report the overall identification accuracy: $P(Correct)$
  - The class conditional reserve decision rate $P_{RT}(r \mid c)$
  - The Reject Rate: $\sum_{c \in C} P_{RT}(r, c)$
- For each data set, plot the accuracy as a function of Read Rate (parameter p_0 controls this)
- In the narrative of the report tell the story of your findings