

Lecture 16.

Classification (II): Practical Considerations

Outline

- Classifier design issues
 - Which classifier to use?
 - What order of a particular classifier to use?
 - Which set of parameters to use?
- Features
 - Feature transformation
 - Feature dimension reduction
 - Removing irrelevant features
 - Removing redundant features
- Output labels
 - Output encoding
 - Two-class vs. multi-class pattern classification

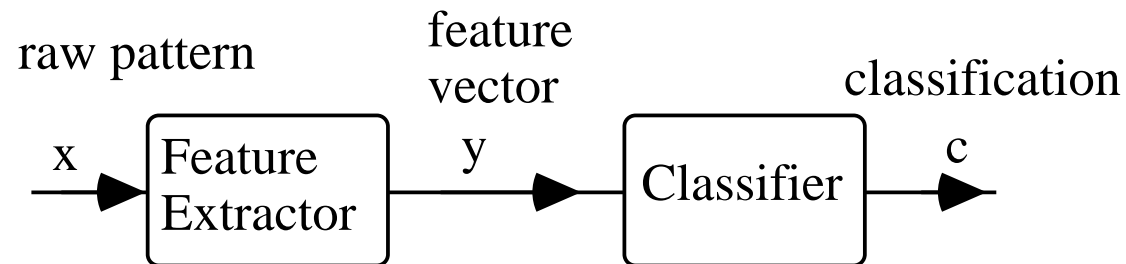
Classifier Design Issues

- Which classifier to use?
 - Performance:
 - cross-validation may be used to facilitate performance comparison.
 - Each classifier should have been fully developed
 - Cost:
 - CPU time for developing and executing the algorithm
 - Memory and storage requirements may prevent some classifier to be used

Classifier Design Issues

- Order Selection
 - Appears in almost all classifiers
 - # of neighbors in kNN
 - # of Gaussian mixtures in each class in ML classifier
 - # of hidden layers, hidden neurons
 - Cross-validation can be used to offer hints for selecting the order of classifiers.
- Which set of parameters to use
 - When classifiers are adaptively trained (such as a MLP), different set of parameters may results due to multiple training runs.
 - Again, CV may be used to aid the selection of which set of parameters to use.

Feature Representation



A typical pattern classification system

- Proper feature representation is essential.
- Feature Transformation: Exposing important features from among unimportant features.
- Feature Selection: Select among a set of given features. Bad feature will confuse classifier
- A feature = a particular dimension of the feature vector. E.g. feature vector $x = [x_1, x_2, x_3]$. Then x_i , $i = 1, 2, 3$ will be the features.

Symbolic Feature Encoding

Many classifiers allow only numerical input values. Features that are represented with symbols must be *encoded* in numerical form. eg. {red, green, blue}, {G, A, C, T}

1. Real number encoding: map each feature symbol into a quantized number in real line. E.g. red $\rightarrow -1$, green $\rightarrow 0$, and blue $\rightarrow +1$.
2. 1-in-N encoding: e.g. red $\rightarrow [1\ 0\ 0]$, green $\rightarrow [0\ 1\ 0]$, and blue $\rightarrow [0\ 0\ 1]$
3. Fuzzy encoding: e.g. red $\rightarrow [1\ 0.5\ 0\ 0\ 0]$, green $\rightarrow [0\ 0.5\ 1\ 0.5\ 0]$, and blue $\rightarrow [0\ 0\ 0\ 0.5\ 1]$

Feature Transformation: Why?

- To expose structures inherent in the data,
 - make difficult classification problem easier to solve.
 - Requires in-depth understanding of the data and extensive trial-and-error
- To equalize the influence of different features.
 - Feature value ranges should often be normalized
 - To have zero mean in each dimension
 - To have same (or similar) ranges or standard deviation in each dimension
- Linear transformation = projection onto basis
- Nonlinear transformation
 - Potentially more powerful. But no easy rule to follow.

Feature Transformation: How?

- Bias, shift of origin: $x' = x - b$
- Linear Transformation (Data Independent) : $y = T x$

$$\text{Rotation: } \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\text{Scaling: } y = a x$$

$$\text{DFT: } y(k) = \sum_{m=0}^{N-1} x(m) \exp\left(-j \frac{2\pi km}{N}\right) \quad 0 \leq k \leq N-1$$

Linear Digital Filtering: FIR, IIR

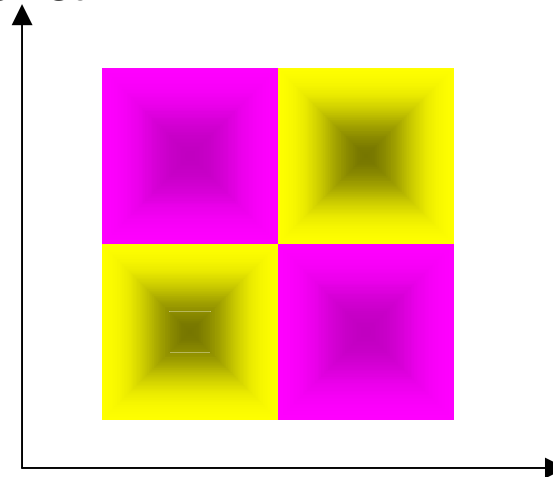
Others: Discrete Cosine Transform, singular value decomposition, etc.

Feature Dimension Reduction

- Irrelevant feature reduction
 - An irrelevant feature (dimension) is one that is uncorrelated with the class label.
 - E.g. the feature value in a dimension is a constant
 - E.g. the feature value in a dimension is random
- Redundant feature reduction
 - If the values of a feature is linearly dependent on remaining features, then this feature can be removed.
 - Method 1. Using principal component analysis (eigenvalue/singular value decomposition)
 - Method 2. Subset selection

Irrelevant Feature Reduction

- If a feature value remains constant, or nearly constant for all the training samples, then this feature dimension can be removed.
- Method:
 - Calculate mean and variance of each feature dimension. If the variance is less than a preset threshold, the feature dimension is marked for removal.
- If the distribution (histogram) of values of a feature corresponding to different classes overlap each other, this feature “may be” subject to removal. However, higher dimensional correlation may exist!



Redundant Feature Reduction

Given a feature matrix

$$\mathbf{X} = [x_1, x_2, \dots, x_M]$$

- Each row = feature (sample) vector.
- Each column = feature

If $x_3 = ax_1 + bx_2$, then

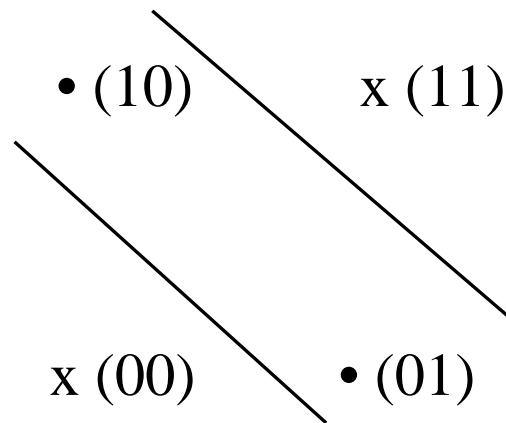
$$[x_1 \ x_2 \ x_3] = [x_1 \ x_2] \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \end{bmatrix}$$

In other words, x_3 is redundant and hence can be removed without affecting the result of classification.

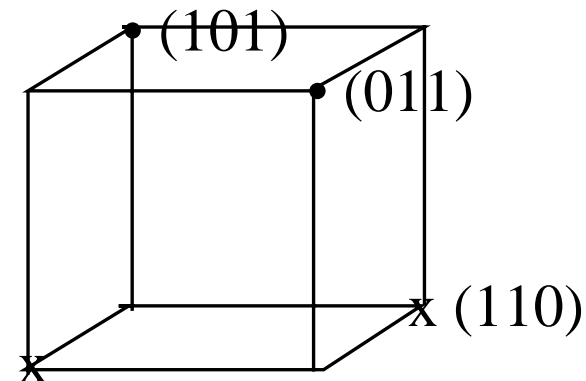
Method:

1. Perform SVD on \mathbf{X} to identify its rank r ($\leq M$).
2. Repeat $M-r$ times:
find i^* , s. t. $\mathbf{X} = [x_{i^*} \ \mathbf{X}_r]$
and the projection error
 $\| \mathbf{X}_r (\mathbf{X}_r^T \mathbf{X}_r)^{-1} \mathbf{X}_r^T \mathbf{X} - \mathbf{X} \|^2$
is minimized.
set $\mathbf{X} = \mathbf{X}_r$.

Higher Dimension Features



Not linearly separable



Linearly separable

Data Sampling

- Samples are assumed to be drawn *independently* from the underlying population.
- Use resampling, i.e. repeated train-and-test partitions to estimate the error rate.
- M-fold cross-validation: partition all available samples into M mutually exclusive sets. Each time, use one set as the testing set, and the remaining as training set. Repeat M times. Take the average testing error as an estimate of the true error rate.
- If sample size < 100 , use *leave-one-out cross validation* where only one sample is used as the test set each time