Pattern Classification

- **Statistical Approach**
  - Supervised
    - Basic concepts: Bayesian decision rule (MAP, ML, LDA, QDA), distance, similarity, discriminant, Bayes classifier
  - Unsupervised
    - Basic concepts: distance, similarity, discriminant, Bayes classifier

- **Non-Statistical Approach**
  - Supervised
    - Basic concepts: agglomerative method, winner-takes-all, Kohonen maps, winner-takes-all, Kohonen maps, winner-takes-all, Kohonen maps
  - Unsupervised
    - Basic concepts: distance, similarity, discriminant, Bayes classifier

**Dimensionality Reduction**
- LDF (Perceptron)
- k-means
- Winner-takes-all
- Kohonen maps
- Winner-takes-all
- Kohonen maps
- Winner-takes-all
- Kohonen maps

**Performance Evaluation**
- ROC curve (TP, TN, FN, FP)
- Cross validation

**Classifier Fusion**
- Majority voting
- NB, BKS

**Stochastic Methods**
- Local opt (GD)
- Global opt (SA, GA)

**Decision Tree**
- Syntactic approach
- NN (BP)
- Support Vector Machine
- Deep Learning (DL)

**Mean Shift**

**Review - Bayes Decision Rule**

- Maximum Posterior Probability
- Likelihood Ratio
- Discriminant Function
- Non-parametric

- kNN

For a given $x$, if $P(x|\omega_1) > P(x|\omega_2)$, then $x$ belongs to class 1, otherwise 2.

The classifier will assign a feature vector $x$ to class $\omega_i$ if

- Case 1: Minimum Euclidean Distance (Linear Machine), $\Sigma = \epsilon$
- Case 2: Minimum Mahalanobis Distance (Linear Machine), $\Sigma = I$
- Case 3: Quadratic classifier, $\Sigma = \text{arbitrary}$

For a given $x$, if $k_1|x| > k_2|x|$, then $x$ belongs to class 1, otherwise 2.

Estimate Gaussian (Maximum Likelihood Estimation, MLE), Two-modal Gaussian

Dimensionality reduction
- Performance evaluation
- ROC curve

For a given $x$, if $P(x|\omega_1) > P(x|\omega_2)$, then $x$ belongs to class 1, otherwise 2.

The classifier will assign a feature vector $x$ to class $\omega_i$ if

- Case 1: Minimum Euclidean Distance (Linear Machine), $\Sigma = \epsilon$
- Case 2: Minimum Mahalanobis Distance (Linear Machine), $\Sigma = I$
- Case 3: Quadratic classifier, $\Sigma = \text{arbitrary}$

For a given $x$, if $k_1|x| > k_2|x|$, then $x$ belongs to class 1, otherwise 2.
Nominal Data

- Descriptions that are discrete and without any natural notion of similarity or even ordering

---

CART

- Classification and regression trees
- A generic tree growing methodology
- Issues studied
  - How many splits from a node?
  - Which property to test at each node?
  - When to declare a leaf?
  - How to prune a large, redundant tree?
  - If the leaf is impure, how to classify?
  - How to handle missing data?
Number of Splits

- Binary tree
- Expressive power and comparative simplicity in training

Node Impurity – Occam’s Razor

- The fundamental principle underlying tree creation is that of simplicity: we prefer simple, compact trees with few nodes
- Occam’s (or Ockham’s) razor is a principle attributed to the 14th century logician and Franciscan friar, William of Occam. Ockham was the village in the English county of Surrey where he was born.
- The principle states that “Entities should not be multiplied unnecessarily.”
- "when you have two competing theories which make exactly the same predictions, the one that is simpler is the better.”
- Stephen Hawking explains in A Brief History of Time: “We could still imagine that there is a set of laws that determines events completely for some supernatural being, who could observe the present state of the universe without disturbing it. However, such models of the universe are not of much interest to us mortals. It seems better to employ the principle known as Occam’s razor and cut out all the features of the theory which cannot be observed.”
- Everything should be made as simple as possible, but not simpler

Property Query and Impurity Measurement

- We seek a property query T at each node N that makes the data reach the immediate descendent nodes as pure as possible
- We want \( i(N) \) to be 0 if all the patterns reach the node bear the same category label
- Entropy impurity (information impurity)

\[
i(N) = - \sum \omega_j \log_2 P(\omega_j)
\]

\( P(\omega_j) \) is the fraction of patterns at node N that are in category \( \omega_j \)
Other Impurity Measurements

- Variance impurity (2-category case)
  \[ i(N) = P(\omega_1) \bar{y}_1(N) - P(\omega_2) \bar{y}_2(N) \]

- Gini impurity
  \[ i(N) = \sum_{\omega} P(\omega) \bar{y}_\omega(N) - \left( \sum_{\omega} P^2(\omega) \right) \]

- Misclassification impurity
  \[ i(N) = 1 - \max_\omega P(\omega \mid N) \]

Choose the Property Test?

- Choose the query that decreases the impurity as much as possible
  \[ \Delta(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R) \]

- \( N_L, N_R \): left and right descendent nodes
- \( i(N_L), i(N_R) \): impurities
- \( P_L \): fraction of patterns at node \( N \) that will go to \( N_L \)
- Solve for extrema (local extrema)

Example

- Node \( N \):
  - 90 patterns in \( \omega_1 \)
  - 10 patterns in \( \omega_2 \)
- Split candidate:
  - 70 \( \omega_1 \) patterns & 0 \( \omega_2 \) patterns to the right
  - 20 \( \omega_1 \) patterns & 10 \( \omega_2 \) patterns to the left
When to Stop Splitting?

- Two extreme scenarios
  - Overfitting (each leaf is one sample)
  - High error rate
- Approaches
  - Validation and cross-validation
  - 90% of the data set as training data
  - 10% of the data set as validation data
  - Use threshold
    - Unbalanced tree
    - Hard to choose threshold
  - Minimum description length (MDL)
    - \( i(N) \) measures the uncertainty of the training data
    - Size of the tree measures the complexity of the classifier itself

\[
\text{MDL} = \alpha \cdot \text{sce} + \sum_{i=1}^{\text{leaf nodes}} i(N)
\]

When to Stop Splitting? (cont)

- Use stopping criterion based on the statistical significance of the reduction of impurity
  - Use chi-square statistic
  - Whether the candidate split differs significantly from a random split

\[
\chi^2 = \sum_{i=1}^{\text{leaf nodes}} \frac{(n_i - n_e)}{n_i}
\]

Pruning

- Another way to stop splitting
- Horizon effect
  - Lack of sufficient look ahead
- Let the tree fully grow, i.e. beyond any putative horizon, then all pairs of neighboring leaf nodes are considered for elimination
Instability

Other Methods

• Quinlan’s ID3
• C4.5 (successor and refinement of ID3)
http://www.rulequest.com/Personal/

MATLAB Routine

• classregtree
• Classification tree vs. Regression tree
  – If the target variable is categorical or numeric

Random Forest

- Potential issue with decision trees
- Prof. Leo Breiman
- Ensemble learning methods
  - Bagging (Bootstrap aggregating): Proposed by Breiman in 1994 to improve the classification by combining classifications of randomly generated training sets
  - Random forest: bagging + random selection of features at each node to determine a split

MATLAB Implementation

- B = TreeBagger(nTrees, train_x, train_y);
- pred = B.predict(test_x);
Reference