Classification

Learning From Data
Chapter 8

Outline

- 8.1 Statistical Learning Theory Formulation
- 8.2 Classical Formulation
- 8.3 Methods for Classification
- 8.4 Summary

Classification

- Input sample $x=(x_1, x_2, \ldots, x_d)$ is classified to one (and only one) of $J$ groups.
- Concerned with relationship between the class-membership label and feature vector.
- Goal is to estimate the mapping $x \rightarrow y$ (decision rule) using labeled training data $(x_i, y_i)$.

Two-class problem

- Output of the system
  - Takes on values $y=\{0, 1\}$.
- Learning machine
  - Implement a set of indicator functions $f(x, \omega)$.
- Loss function
  $$L(y, f(x, \omega)) = \begin{cases} 0 & \text{if } y = f(x, \omega) \\ 1 & \text{if } y \neq f(x, \omega) \end{cases}$$
Risk Functional

\[ R(\omega) = \int L(y, f(x, \omega)) p(x, y) dx dy \]

Learning is the problem of finding the function \( f(x, \omega) \) that minimizes the probability of misclassification using only the training data.

Implementation of methods using SRM requires:

- Specification of a nested structure on a set of indicator approximating functions.
- Minimization of the empirical risk (misclassification error) for a given element of a structure.
- Estimation of prediction risk.

Classical formulation of the classification

- Conditional density
  \[ p(x \mid y = 0) = p(x, \alpha^*) \]
  \[ p(x \mid y = 1) = p(x, \beta^*) \]
- Prior probability
  \[ P(y = 0), \quad P(y = 1) \]
- Posterior probability
  \[ p(y = 0 \mid x) = \frac{p_0(x, \alpha^*) P(y = 0)}{p(x)} \]
  \[ p(y = 1 \mid x) = \frac{p_1(x, \beta^*) P(y = 1)}{p(x)} \]

Decision rule

\[ f(x) = \begin{cases} 0 & \text{if} \quad p_0(x, \alpha^*) P(y = 0) > p_1(x, \beta^*) P(y = 1) \\ 1 & \text{otherwise} \end{cases} \]

- Applies the ERM inductive principle \textit{indirectly} to first estimate the densities.
- Conceptually flawed in estimating decision boundary via density estimation.
8.1 Statistical Learning Theory Formulation

- Goal is to estimate an indicator function or decision boundary \( f(x, \omega_0) \).
- Use SRM inductive principle.
  \[ S_1 \subset S_2 \subset \ldots \subset S_m \subset \ldots \]
  \( h_1 \leq h_2 \leq \ldots \leq h_m \leq \ldots \) \( h_m \) is VC-dimension of \( S_m \)
- Constructive methods should select an element of a structure \( S_m = f(x, \omega_0) \) and an indicator function \( f(x, \omega_0^m) \).
- Bound on prediction risk
  \[ R(\omega_0^m) \leq R_{emp}(\omega_0^m) + \Phi(n/h_m) \]

SLT Formulation (Cont’d)

- Two strategies for minimizing the bound
  - Keep the confidence interval fixed and minimize the empirical risk.
    - Include all statistical and neural network methods using dictionary representation.
    - With high-dimensional data, first project the data onto the low-dimensional subspace (i.e., \( m \) features) and then perform modeling in this subspace.
  - Keep the value of the empirical risk fixed (small) and minimize the confidence interval.
    - Requires a special structure such that the value of the empirical risk is kept small for all approximating functions.

SLT Formulation (Cont’d)

- Binary classification problem
  - Goal is to minimize the misclassification error.
    \[ R(\omega) = \sum_{i=1}^{n} |f(x_i, \omega) - y_i| \text{ or} \]
    \[ \sum_{i=1}^{n} [f(x_i, \omega) - y_i]^2 \]
  - Perceptron algorithm
    - Simple optimization procedure for finding \( f(x, \omega^p) \) providing zero misclassification error if
      - training data is *linearly separable*.
      - indicator function is \( f(x, \omega) = \hat{f}(\omega \cdot x) \)
    - Prerequisite
      \[ x(k) \in \mathbb{R}^d, \quad y(k) \in \{-1, 1\} \]
      initial weights set to small random values
    - Weight update rule
      \[ w(k+1) = \begin{cases} w(k) & \text{if } y(k)(w(k) \cdot x(k)) > 0 \\ w(k) + y(k)x(k) & \text{if } y(k)(w(k) \cdot x(k)) < 0 \end{cases} \]
  - When the data is not separable and/or the optimal decision boundary is nonlinear, does not provide an optimal solution.
MLP classifier

- Can form flexible nonlinear decision boundaries.
- Approximate the indicator function by a well-behaved sigmoid function.
- Empirical risk functional

\[ R = \sum_{i=1}^{n} [s(g(x_i, w, V)) - y_i]^2 \]

\[ g(x, w, V) = \sum_{i=1}^{m} w_i s(x \cdot v_i) + w_0 \]

- Sigmoid activations of hidden units enable the construction of a flexible nonlinear decision boundary.
- Output sigmoid approximates the discontinuous indicator function.

In neural networks, a common procedure for classification decisions is to use sigmoid output.

\[ f(x) = I[s(g(x, w^*, V^*)) - \theta] \]

Output of the trained network is interpreted as an estimate of the posterior probability.

\[ s(g(x, w^*, V^*)) = \hat{P}(y = 1 | x) \]

General prescription for implementing constructive methods

1. Specify a (flexible) class of approximating functions for constructing a (nonlinear) decision boundary.
2. Choose a nonlinear optimization method for selecting the best function from class (1), i.e., the function providing the smallest empirical risk.
3. Select continuous error functional suitable for the optimization methods chosen in (2).
4. Select the best predictive model from a class of functions (1) using the first strategy for minimizing SLT bound.

Classification methods use a continuous error functional that only approximates the true one.

A classification method using such approximation will be successful only if minimizing of the error functional selected in (3) also minimizes true empirical risk.
8.2 Classical Formulation

- Based on statistical decision theory
  - Provides the foundation for constructing optimal decision rules minimizing risk.
  - Strictly applies only when all distributions are known.
  - Concerned with constructing decision rules.
- Estimate the required distributions from the data and use them within the framework of statistical decision theory.

### Classical Formulation (Cont’d)

- When $x$ is not observed
  
  $r(x) = \begin{cases} 
  0 & \text{if } P(y = 0) > P(y = 1) \\
  1 & \text{otherwise}
  \end{cases}
  $

- When $x$ is observed
  
  $r(x) = \begin{cases} 
  0 & \text{if } P(y = 0 | x) > P(y = 1 | x) \\
  1 & \text{otherwise}
  \end{cases}
  
  = \begin{cases} 
  0 & \text{if } p(x | y = 0)P(y = 0) > p(x | y = 1)P(y = 1) \\
  1 & \text{otherwise}
  \end{cases}
  
  = \begin{cases} 
  0 & \text{if } \frac{P(x | y = 0)}{P(y = 0)} > \frac{P(y = 1)}{p(x | y = 1)} \\
  1 & \text{otherwise}
  \end{cases}

**Misclassification with unequal costs**

If $x \in R_i$, the expected cost is

$q_i = C_{i0} \int_{R_0} p(x | y = i) dx + C_{i1} \int_{R_1} p(x | y = i) dx$

overall risk is

$\sum_i q_i P(y = i) = \sum_i \left[ \int_{R_0} C_{i0} P(y = i) p(x | y = i) dx + \int_{R_1} C_{i1} P(y = i) p(x | y = i) dx \right]$

Risk is minimized if region $R_0$ is defined such that $x \in R_0$ whenever

$\left\{ \sum_i C_{i0} P(y = i) p(x | y = i) \right\} < \left\{ \sum_i C_{i1} P(y = i) p(x | y = i) \right\}$

$r(x) = \begin{cases} 
0 & \text{if } \frac{p(x | y = 0)}{p(x | y = 1)} > \frac{C_{10} - C_{11} P(y = 1)}{C_{01} - C_{00} P(y = 0)} \\
1 & \text{otherwise}
\end{cases}$
Posterior distributions

Two strategies
- Estimate the prior probabilities and class conditional densities and plug them into Bayes’ rule.
- Estimate posterior densities directly using training data from all the classes.

Two approaches for two strategies
- Parametric methods.
- Adaptive flexible methods.

Direct Estimation
- Estimation of posterior densities can be done using regression methods.
- For two-class case
  \[ g(x) = E(Y | X = x) = 0 \cdot P(Y = 0 | x) + 1 \cdot P(Y = 1 | x) = P(Y = 1 | x) \]

Parametric Regression
- Linear regression can be used for classification for non-gaussian distributions.
  - Fisher linear discriminant provides an approximation for the posterior probability, but biased, providing an accurate classification rule.
  - May provide a poor decision boundary.

Adaptive Regression
- Accurate regression does not guarantee a low classification risk.
- Use the data to estimate the conditional expectation.
- For two-class problems
  \[ R_1(\omega) = \frac{1}{n} \sum_{i=1}^{n} (\hat{g}_i(x_i, \omega) - y_i)^2, \quad \hat{g}_1(x, \omega^*) = P(y = 1 | x) \]
  \[ R_0(\omega) = \frac{1}{n} \sum_{i=1}^{n} (\hat{g}_0(x_i, \omega) - (1 - y_i))^2, \quad \hat{g}_0(x, \omega^*) = P(y = 0 | x) \]
8.3 Methods for Classification

8.3.1 Regression-Based Methods
- Methods based on continuous numerical optimization can be cast in the form of multiple-response regression.
- Most popular approach to classification.

8.3.2 Tree-Based Methods
- Based on a greedy optimization strategy.
- CART is an example.

8.3.3 Nearest-Neighbor and Prototype Methods
- Local methods for classification.
- Estimate the decision boundary locally.
- Nearest-neighbors classification, Kohonen’s learning vector quantization

MLP Classifiers

- Use $1$-of-$J$ output encoding and sigmoid output units.

- Practical hints and implementation issues
  - Pre-scaling of input variables
    - Scale input data to the range $[-0.5, 0.5]$.
    - Typically pre-scaled to zero mean, unit variance.
    - Helps to avoid premature saturation and speeds up training.
  - Alternative target output values
    - Training outputs are set to values 0.1 and 0.9.
    - Avoid long training time and extremely large weights during training.

- Initialization
  - Network parameters are initialized to small random values.
  - Choice of initialization range has subtle regularization effect.

- Stopping rules
  - During training
    - Training should proceed as long as the decreasing continuous loss function reduces the empirical misclassification error.
  - Early stopping
    - used as a form of complexity control (model selection).
- Multiple local minima
  - Main factor complicating empirical risk minimization as well as model selection.
  - Use the misclassification error rather than squared error loss during model selection.
- Learning rate and momentum term
  - Affects local minima found by backpropagation training.
  - Optimal choice of these is problem-dependent.

**RBF Classifiers**

- Use multiple output regression to build a decision boundary.
- Form of discriminant functions

\[ g_k(x, w_k) = \sum_{j=1}^{m} w_{jk} K\left(\frac{x - v_j}{\alpha_j}\right) + w_{0k} \]

\[ K(x) = \exp\left(-\frac{x^2}{2}\right) \]

**RBF... (Cont’d)**

- Characteristics
  - Implements local decision boundaries.
  - For fixed values of basis function parameters, \( w_{jk} \) are estimated via linear least squares.
  - Complexity can be determined by the number of basis functions \( m \).
  - Possible to use resampling techniques to estimate prediction risk to perform model selection.
  - Typically use normalized basis functions
    - Allows to be interpreted as a type of density mixture model.

**Tree-Based Methods**

- Adaptively split input space into disjoint regions in order to construct a decision boundary.
- Based on a greedy optimization procedure.
- Splitting process can be represented as a binary tree.
- Following the growth of the tree, pruning occurs as a form of model selection.
- Growing and pruning strategy provides better classification accuracy than just growing alone.
Tree-Based Methods

- Pruning criteria provides an estimate of the prediction risk while the growing criteria roughly reflects empirical risk.
- Resulting classifier has a binary tree representation.

CART

- Popular approach to construct a binary-tree-based classifier.
- Greedy search employs a recursive partitioning strategy.
- Cost function
  - Measure node impurity.
    - Give a measurement of how homogeneous a node $t$ is with respect to the class labels of the training data in the region of node $t$.

CART (Cont’d)

- Examples
  - $Q(t) = 1 - \max_j p(j \mid t)$ ("misclassification cost")
  - $Q(t) = \sum_{j \neq i} p(i \mid t) p(j \mid t) - \sum_j [p(j \mid t)]^2$ ("Gini function")
  - $Q(t) = \sum_j p(j \mid t) \ln p(j \mid t)$ ("Entropy function")

- Gini and entropy functions are used for practical implementation.
- Gini and entropy functions do not measure the classification risk directly.
Two difficulties when using the empirical misclassification cost

- There are cases where misclassification cost does not decrease for any candidate split, leading to early halting in a poor local minimum.
- The misclassification cost does not favor splits that tend to provide a lower misclassification cost in future splits.

Decrease in impurity

\[ \Delta Q(v, k, t) = Q(t) - Q(t_L) p_L(t) - Q(t_R) p_R(t) \]

\[ p_L(t) = \frac{p(t_L)}{p(t)}, \quad p_R(t) = \frac{p(t_R)}{p(t)} \]

Variable \( x_k \) and the split point \( v \) are selected to maximize the decrease in node impurity.

Pruning

- Performed after growing is completed.
- Implements model selection.
- Based on minimizing penalized empirical risk.
- Performed in a greedy search strategy.

- Every pair of sibling leaf nodes is recombined in order to find a pair that reduces the following.

\[ R_{pen} = R_{emp} + \lambda |T| \]

\( R_{emp} \): misclassification rate for the training data

\( |T| \): number of terminal nodes
CART (Cont’d)

**Procedure**
- **Initialization**
  - Root node consists of whole input space.
  - Estimate proportion of the classes via $p(y(t=0) = y)/n$.
- **Tree growing**
  - Repeat until stopping criterion is satisfied.
    - Perform exhaustive search over all valid nodes in the tree, all split variables, and all valid knot points.
    - Incorporate the daughters into the tree that results in the largest decrease in the impurity using gini or entropy cost function.

CART (Cont’d)

**Tree Pruning**
- Repeat until no more pruning occurs.
  - Perform exhaustive search over all sibling leaf nodes in the tree, measuring the change in model selection criterion.
  - Delete the pair that leads to the largest decrease of model selection criterion.

**Disadvantages**
- Sensitive to coordinate rotations
  - Because CART partitions the space into axis-oriented subregions.
  - Can be alleviated by splitting on linear combinations of input variables.

Nearest-Neighbor and Prototype Methods

**Goal of local methods for classification**
- Construction of local decision boundaries.

**In SLT view, local methods for classification follow the framework of local risk minimization.**

**In classical decision theory, local methods are interpreted as local posterior density estimation.**

Nearest-Neighbor Classification

**Classifies an object based on the class of the $k$ data points nearest to the estimation point $x_0$.**

**Nearness is most commonly measured using euclidean distance metric in $\mathbf{x}$-space.**

**Local decision rule is constructed using the procedure of local risk minimization.**
Nearest... (Cont’d)

Example: two-class problem

- Empirical risk

\[ R_{emp\_local}(w) = \frac{1}{k} \sum_{i=1}^{n} (y_i - w) \cdot K(x_0, x_i) \]

\[ K(x_0, x_i) = \begin{cases} 1 & \text{if } x_i \text{ is one of the } k \text{ nearest data points} \\ 0 & \text{otherwise} \end{cases} \]

\[ w^* = \begin{cases} 1 & \text{if } \frac{1}{k} \sum_{i=1}^{n} y_i K(x_0, x_i) > 0.5 \\ 0 & \text{otherwise} \end{cases} \]

Reasons for the success

- Practical problems often have a low intrinsic dimensionality even though they may have many input variables.
- Effect of the curse of dimensionality is not as severe due to the nature of the classification problem.
  - Accurate estimates of conditional probabilities are not necessary for accurate classification.

Learning Vector Quantization (LVQ)

Outline

- Use vector quantization methods to determine initial locations of \( m \) prototype vectors.
- Assign class labels to these prototype.
- Adjust the locations using a heuristic strategy.
  - Reduce the empirical misclassification risk.
  - LVQ1, LVQ2, LVQ3 by Kohonen.

LVQ (Cont’d)

LVQ1

- A stochastic approximation method.
- Given \((x(k), y(k))\): a data point, \( c_j(k) \): prototype center, and prototype labels \( w_j \)
  - Determine the nearest prototype center to the data point.

\[ j = \arg \min_j \| x(k) - c_j(k) \| \]

- Update the location of the nearest prototype.
  - If correctly classified

\[ c_j(k+1) = c_j(k) + \gamma(k)[x(k) - c_j(k)] \]

- Else

\[ c_j(k+1) = c_j(k) - \gamma(k)[x(k) - c_j(k)] \]

- \( k=k+1 \)
Training data sets are generated according to the distribution $x \sim N(0, I)$, $x \in \mathbb{R}^{10}$. Ten training sets are generated, and each data set contains 200 samples. Prediction risk is estimated for each individual classifier using a large test set (2000 samples).
Empirical Comparisons (3)

Waveform Data

- 21 input variables that correspond to 21 discrete time samples taken from a randomly generated waveform.
- Waveform is generated using a random linear combination of two out of three possible component waveforms.

\[ x_{ij} = u_i h_1(j) + (1-u_i) h_2(j) + \varepsilon_j \]

Class 1: \( x_{ij} = u_i h_1(j) + (1-u_i) h_2(j) + \varepsilon_j \)
Class 2: \( x_{ij} = u_i h_1(j) + (1-u_i) h_3(j) + \varepsilon_j \)
Class 3: \( x_{ij} = u_i h_2(j) + (1-u_i) h_3(j) + \varepsilon_j \)

- Ten training sets are generated, and each data set contains 300 samples.
- Prediction risk is estimated using a large test set (2000 samples).

Summary

- Understanding classification methods requires clear separation between conceptual procedure based on the SRM inductive principle and its technical implementation.

Conceptual procedure

- Two necessary things
  - Minimize the empirical classification error (via nonlinear optimization).
  - Estimate accurately future classification error (model selection).
• Technical implementation
  ▲ Complicated by the discontinuous misclassification error functional.
  ◁ Prevents direct minimization of the empirical risk
  ▲ All practical methods use a suitable continuous loss function providing approximation for misclassification error.
  ▲ In model selection, should use classification error loss.
• Accurate estimation of a posterior probabilities is not necessary for accurate classification.
  Good probability estimates are not necessary for good classification; similarly, low classification error does not imply that the corresponding class probabilities are being estimate (even remotely) accurately - Friedman (1997)
• SLT’s explanation for empirical evidence on well-behaving of simple methods
  ● Simple classification methods may not require nonlinear optimization, so the empirical classification error is minimized directly.
  ● Often simple methods provide the same level of empirical misclassification error in the minimization step as more complex methods.
  ● No need to use more complex methods.
  ● Classification problems are inherently less sensitive than regression to optimal model selection.