Chapter 6. Classification and Prediction

- Overview
- Classification algorithms and methods
  - Decision tree induction
  - Bayesian classification
  - Lazy learning and kNN classification
  - Online learning: Winnow
  - Support Vector Machines (SVM)
- Others
- Ensemble methods
Online learning: Winnow

- PAC learning vs online learning (mistake bound model)

- Winnow: an online learning algorithm for learning linear separator
- Prediction same as perceptron
- Perceptron: additive weight update
- Winnow: multiplicative weight update
Winnow

• Learning disjunction
  – $x_1 \lor x_2 \lor \ldots \lor x_r$ out of $n$ variables
  – Mistake bound $2+3r(\log n)$
  – Most useful when lot of irrelevant variables

• Learning $r$-of-$k$ threshold functions

• Learning a box

• References:
  – Wolfgang Maass and Manfred K. Warmuth, Efficient Learning with Virtual Threshold Gates
Support Vector Machines: Overview

• A relatively new classification method for both separable and non-separable data

• Features
  – Sound mathematical foundation
  – Training time can be slow but efficient methods are being developed
  – Robust and accurate, less prone to overfitting

• Applications: handwritten digit recognition, speaker identification, ...
Support Vector Machines: History

- Vapnik and colleagues (1992)
  - Groundwork from Vapnik-Chervonenkis theory (1960 – 1990)
- Problems driving the initial development of SVM
  - Bias variance tradeoff, capacity control, overfitting
  - Basic idea: accuracy on the training set vs. capacity

- A Tutorial on Support Vector Machines for Pattern Recognition, Burges, Data Mining and Knowledge Discovery, 1998
Linear Support Vector Machines

- Problem: find a linear hyperplane (decision boundary) that best separate the data
Linear Support Vector Machines

- Which line is better? B1 or B2?
- How do we define better?
Support Vector Machines

- Find hyperplane maximizes the margin
Support Vector Machines Illustration

- A separating hyperplane can be written as
  \[ W \bullet X + b = 0 \]
  where \( W = \{w_1, w_2, ..., w_n\} \) is a weight vector and \( b \) a scalar (bias)

- For 2-D it can be written as
  \[ w_0 + w_1 x_1 + w_2 x_2 = 0 \]

- The hyperplane defining the sides of the margin:
  \[ H_1: w_0 + w_1 x_1 + w_2 x_2 = 1 \]
  \[ H_2: w_0 + w_1 x_1 + w_2 x_2 = -1 \]

- Any training tuples that fall on hyperplanes \( H_1 \) or \( H_2 \) (i.e., the sides defining the margin) are \textbf{support vectors}
Support Vector Machines

For all training points:
\[ \mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \text{ for } y_i = +1 \]
\[ \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ for } y_i = -1 \]

\[ y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \]

Margin = \[ \frac{2}{\| \mathbf{w} \|} \]
Support Vector Machines

• We want to maximize: 
  \[ \text{Margin} = \frac{2}{\| \vec{w} \|} \]

  – Equivalent to minimizing: 
  \[ \| \vec{w} \|^2 \]

  – But subjected to the constraints: 
  \[ y_i (\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0 \]

• Constrained optimization problem
  – Lagrange reformulation

\[
L_P \equiv \frac{1}{2} \| w \|^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{l} \alpha_i
\]
Support Vector Machines

• What if the problem is not linearly separable?

• Introduce slack variables to the constraints:

\[ \vec{w} \cdot \vec{x}_i + b \geq +1 - \xi_i \text{ for } y_i = +1 \]
\[ \vec{w} \cdot \vec{x}_i + b \leq -1 + \xi_i \text{ for } y_i = -1 \]

• Upper bound on the training errors:

\[ \sum_i \xi_i \]
Nonlinear Support Vector Machines

• What if decision boundary is not linear?
• Transform the data into higher dimensional space and search for a hyperplane in the new space
SVM—Kernel functions

- Instead of computing the dot product on the transformed data tuples, it is mathematically equivalent to instead applying a kernel function $K(X_i, X_j)$ to the original data, i.e., $K(X_i, X_j) = \Phi(X_i) \Phi(X_j)$

- Typical Kernel Functions

  Polynomial kernel of degree $h$: $K(X_i, X_j) = (X_i \cdot X_j + 1)^h$

  Gaussian radial basis function kernel: $K(X_i, X_j) = e^{-\|X_i - X_j\|^2/2\sigma^2}$

  Sigmoid kernel: $K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$

- SVM can also be used for classifying multiple (> 2) classes and for regression analysis (with additional user parameters)
Support Vector Machines: Comments and Research Issues

- Robust and accurate with nice generalization properties
- Effective (insensitive) to high dimensions
  - Complexity characterized by # of support vectors rather than dimensionality
- Scalability in training
- Extension to regression analysis
- Extension to multiclass SVM
SVM Related Links

- SVM web sites
  - www.kernel-machines.org
  - www.kernel-methods.net
  - www.support-vector.net
  - www.support-vector-machines.org

- Representative implementations
  - LIBSVM: an efficient implementation of SVM, multi-class classifications
  - SVM-light: simpler but performance is not better than LIBSVM, support only binary classification and only C language
  - SVM-torch: another recent implementation also written in C.
SVM—Introduction Literature

- “Statistical Learning Theory” by Vapnik: extremely hard to understand, containing many errors too

  - Better than the Vapnik’s book, but still written too hard for introduction, and the examples are not-intuitive

- The book “An Introduction to Support Vector Machines” by N. Cristianini and J. Shawe-Taylor
  - Also written hard for introduction, but the explanation about the Mercer’s theorem is better than above literatures

- The neural network book by Haykins
  - Contains one nice chapter of SVM introduction