Support-Vector Machines

- Haykin chapter 6.
- See Alpaydin chapter 13 for similar content.
- Note: Part of this lecture drew material from Ricardo Gutierrez-Osuna’s Pattern Analysis lectures.

Introduction

- Support vector machine is a linear machine with some very nice properties.
- The basic idea of SVM is to construct a separating hyperplane where the margin of separation between positive and negative examples are maximized.
- Principled derivation: structural risk minimization
  - error rate is bounded by: (1) training error-rate and (2) VC-dimension of the model.
  - SVM makes (1) become zero and minimizes (2).

Optimal Hyperplane

For linearly separable patterns \(\{(x_i, d_i)\}_{i=1}^N\) (with \(d_i \in \{+1, -1\}\)):

- The separating hyperplane is \(w^T x + b = 0\):
  \[
  w^T x + b \geq 0 \quad \text{for } d_i = +1 \\
  w^T x + b < 0 \quad \text{for } d_i = -1
  \]
- Let \(w_o\) be the optimal hyperplane and \(b_o\) the optimal bias.

Distance to the Optimal Hyperplane

From \(w_o^T x_i = -b_o\), the distance from the origin to the hyperplane is calculated as:

\[
d = \|x_i\| \cos(x_i, w_o) = \frac{-b_o}{\|w_o\|}
\]

since

\[
w_o^T x_i = \|w_o\| \|x_i\| \cos(w_o, x_i) = -b_o
\]
The distance from an arbitrary point to the hyperplane can be calculated as:

- When the point is in the positive area:
  \[ r = \|x\| \cos(\mathbf{x}, \mathbf{w}_o) - d = \frac{\mathbf{x}^T \mathbf{w}_o}{\|\mathbf{w}_o\|} + \frac{b_o}{\|\mathbf{w}_o\|} = \frac{\mathbf{x}^T \mathbf{w}_o + b_o}{\|\mathbf{w}_o\|}. \]

- When the point is in the negative area:
  \[ r = d - \|x\| \cos(\mathbf{x}, \mathbf{w}_o) = -\frac{\mathbf{x}^T \mathbf{w}_o}{\|\mathbf{w}_o\|} - \frac{b_o}{\|\mathbf{w}_o\|} = -\frac{\mathbf{x}^T \mathbf{w}_o + b_o}{\|\mathbf{w}_o\|}. \]

### Optimal Hyperplane and Support Vectors (cont'd)

- The optimal hyperplane is supposed to maximize the margin of separation \(\rho\).
- With that requirement, we can write the conditions that \(\mathbf{w}_o\) and \(b_o\) must meet:
  \[ \mathbf{w}_o^T \mathbf{x} + b_o \geq +1 \quad \text{for } d_i = +1 \]
  \[ \mathbf{w}_o^T \mathbf{x} + b_o \leq -1 \quad \text{for } d_i = -1 \]

  Note: \(\geq +1\) and \(\leq -1\), and support vectors are those \(\mathbf{x}^{(s)}\) where equality holds (i.e., \(\mathbf{w}_o^T \mathbf{x}^{(s)} + b_o = +1\) or \(-1\)).

- Since \(r = (\mathbf{w}_o^T \mathbf{x} + b_o)/\|\mathbf{w}_o\|\),
  \[ r = \begin{cases} 
  1/\|\mathbf{w}_o\| & \text{if } d = +1 \\
  -1/\|\mathbf{w}_o\| & \text{if } d = -1 
  \end{cases} \]

- Support vectors: input points closest to the separating hyperplane.
- Margin of separation \(\rho\): distance between the separating hyperplane and the closest input point.

- Margin of separation between two classes is
  \[ \rho = 2r = \frac{2}{\|\mathbf{w}_o\|}. \]

- Thus, maximizing the margin of separation between two classes is equivalent to minimizing the Euclidean norm of the weight \(\mathbf{w}_o\)!
**Primal Problem: Constrained Optimization**

For the training set $T = \{(x_i, d_i)\}_{i=1}^{N}$ find $w$ and $b$ such that

- they minimize a certain value $(1/\rho)$ while satisfying a constraint (all examples are correctly classified):
  - Constraint: $d_i (w^T x_i + b) \geq 1$ for $i = 1, 2, ..., N$.
  - Cost function: $\Phi(w) = \frac{1}{2} w^T w$.

This problem can be solved using the *method of Lagrange multipliers* (see next two slides).

**Mathematical Aside: Lagrange Multipliers**

Turn a constrained optimization problem into an unconstrained optimization problem by absorbing the constraints into the cost function, weighted by the Lagrange multipliers.

Example: Find closest point on the circle $x^2 + y^2 = 1$ to the point $(2, 3)$ (adapted from Ballard, *An Introduction to Natural Computation*, 1997, pp. 119–120).

- Minimize $F(x, y) = (x - 2)^2 + (y - 3)^2$ subject to the constraint $x^2 + y^2 - 1 = 0$.
- Absorb the constraint into the cost function, after multiplying the Lagrange multiplier $\alpha$:
  $$F(x, y, \alpha) = (x - 2)^2 + (y - 3)^2 + \alpha (x^2 + y^2 - 1).$$

**Lagrange Multipliers (cont’d)**

Must find $x, y, \alpha$ that minimizes $F(x, y, \alpha) = (x - 2)^2 + (y - 2)^2 + \alpha (x^2 + y^2 - 1)$. Set the partial derivatives to 0, and solve the system of equations.

$$\frac{\partial F}{\partial x} = 2(x - 2) + 2\alpha x = 0$$
$$\frac{\partial F}{\partial y} = 2(y - 2) + 2\alpha y = 0$$
$$\frac{\partial F}{\partial \alpha} = x^2 + y^2 - 1 = 0$$

Solve for $x$ and $y$ in the 1st and 2nd, and plug in those to the 3rd equation

$$x = y = \frac{2}{1 + \alpha}, \text{ so } \left(\frac{2}{1 + \alpha}\right)^2 + \left(\frac{2}{1 + \alpha}\right)^2 = 1$$

from which we get $\alpha = 2\sqrt{2} - 1$. Thus, $(x, y) = (1/\sqrt{2}, 1/\sqrt{2})$.

**Primal Problem: Constrained Optimization (cont’d)**

Putting the constrained optimization problem into the Lagrangian form, we get (utilizing the Kuhn-Tucker theorem)

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left[ d_i (w^T x_i + b) - 1 \right].$$

- From $\frac{\partial J(w, b, \alpha)}{\partial w} = 0$:
  $$w = \sum_{i=1}^{N} \alpha_i d_i x_i.$$
- From $\frac{\partial J(w, b, \alpha)}{\partial b} = 0$:
  $$\sum_{i=1}^{N} \alpha_i d_i = 0.$$
Primal Problem: Constrained Optimization (cont’d)

- Note that when the optimal solution is reached, the following condition must hold (Karush-Kuhn-Tucker complementary condition)
  \[ \alpha_i \left[ d_i (w^T x_i + b) - 1 \right] = 0 \]
  for all \( i = 1, 2, \ldots, N \).
- Thus, non-zero \( \alpha_i \)s can be attained only when \( [d_i (w^T x_i + b) - 1] = 0 \), i.e., when the \( \alpha_i \) is associated with a support vector \( x_i \)!
- Other conditions include \( \alpha_i \geq 0 \).

Dual Problem

- Given the training sample \( \{ (x_i, d_i) \}_{i=1}^N \), find the Lagrange multipliers \( \{ \alpha_i \}_{i=1}^N \) that maximize the objective function:
  \[ Q(\alpha) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j x_i^T x_j + \sum_{i=1}^N \alpha_i \]
  subject to the constraints
  - \( \sum_{i=1}^N \alpha_i d_i = 0 \)
  - \( \alpha_i \geq 0 \) for all \( i = 1, 2, \ldots, N \).
- The problem is stated entirely in terms of the training data \( (x_i, d_i) \), and the dot products \( x_i^T x_j \) play a key role.

Solution to the Optimization Problem

Once all the optimal Lagrange multipliers \( \alpha_{o,i} \) are found, \( w_o \) and \( b_o \) can be found as follows:
\[
w_o = \sum_{i=1}^N \alpha_{o,i} d_i x_i \\
b_o = d(s) - w_o^T x(s)
\]
and from \( w_o^T x_i + b_o = d_i \) when \( x_i \) is a support vector:

Note: calculation of final estimated function does not need any explicit calculation of \( w_o \) since they can be calculated from the dot product between the input vectors:
\[
w_o^T x = \sum_{i=1}^N \alpha_{o,i} d_i x_i^T x
\]
Margin of Separation in SVM and VC Dimension

Statistical learning theory shows that it is desirable to reduce both the error (empirical risk) and the VC dimension of the classifier.

- Vapnik (1995, 1998) showed: Let $D$ be the diameter of the smallest ball containing all input vectors $x_i$. The set of optimal hyperplanes defined by $w_o^T x + b_o = 0$ has a VC dimension $h$ bounded from above as

$$h \leq \min \left\{ \left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0 \right\} + 1$$

where $\lceil \cdot \rceil$ is the ceiling, $\rho$ the margin of separation equal to $2/\|w_o\|$, and $m_0$ the dimensionality of the input space.

- The implication is that the VC dimension can be controlled independently of $m_0$, by choosing an appropriate (large) $\rho$!

Soft-Margin Classification (cont’d)

- We want to find a separating hyperplane that minimizes:

$$\Phi(\xi) = \sum_{i=1}^{N} I(\xi_i - 1)$$

where $I(\xi) = 0$ if $\xi \leq 0$ and 1 otherwise.

- Solving the above is NP-complete, so we instead solve an approximation:

$$\Phi(\xi) = \sum_{i=1}^{N} \xi_i$$

- Furthermore, the weight vector can be factored in:

$$\Phi(x, \xi) = \frac{1}{2} w^T w \underbrace{+ C \sum_{i=1}^{N} \xi_i}_{\text{Controls error}}$$

with a control parameter $C$.  

Soft-Margin Classification

- Some problems can violate the condition:

$$d_i (w^T x_i + b) \geq 1$$

- We can introduce a new set of variables $\{\xi_i\}_{i=1}^{N}$:

$$d_i (w^T x_i + b) \geq 1 - \xi_i$$

where $\xi_i$ is called the slack variable.

Soft-Margin Classification: Solution

- Following a similar route involving Lagrange multipliers, and a more restrictive condition of $0 \leq \alpha_i \leq C$, we get the solution:

$$w_o = \sum_{i=1}^{N_s} \alpha_{o,i} d_i x_i$$

$$b_o = d_i (1 - \xi_i) - w_o^T x_i$$
Nonlinear SVM

- Nonlinear mapping of an input vector to a high-dimensional feature space (exploit Cover’s theorem)
- Construction of an optimal hyperplane for separating the features identified in the above step.

Inner-Product Kernel

- Input $\mathbf{x}$ is mapped to $\varphi(\mathbf{x})$.
- With the weight $\mathbf{w}$ (including the bias $b$), the decision surface in the feature space becomes (assume $\varphi(\mathbf{x}) = 1$):
  \[ \mathbf{w}^T \varphi(\mathbf{x}) = 0 \]
- Using the steps in linear SVM, we get
  \[ \mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \varphi(\mathbf{x}_i) \]
- Combining the above two, we get the decision surface
  \[ \sum_{i=1}^{N} \alpha_i d_i \varphi^T(\mathbf{x}_i) \varphi(\mathbf{x}) = 0. \]

Inner-Product Kernel (cont’d)

- Mercer’s theorem states that $K(\mathbf{x}, \mathbf{x}_i)$ that follow certain conditions (continuous, symmetric, positive semi-definite) can be expressed in terms of an inner-product in a nonlinearly mapped feature space.
- Kernel function $K(\mathbf{x}, \mathbf{x}_i)$ allows us to calculate the inner product $\varphi^T(\mathbf{x}) \varphi(\mathbf{x}_i)$ in the mapped feature space without any explicit calculation of the mapping function $\varphi(\cdot)$.
Examples of Kernel Functions

- Linear: $K(x, x_i) = x^T x_i$.
- Polynomial: $K(x, x_i) = (x^T x_i + 1)^p$.
- RBF: $K(x, x_i) = \exp \left(-\frac{1}{2\sigma^2} \|x - x_i\|^2\right)$.
- Two-layer perceptron: $K(x, x_i) = \tanh(\beta_0 x^T x_i + \beta_1)$ (for some $\beta_0$ and $\beta_1$).

Kernel Example

- Expanding $K(x, x_i) = (1 + x^T x_i)^2$ with $x = [x_1, x_2]^T, x_i = [x_{i1}, x_{i2}]^T$.
  
  $K(x, x_i) = 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_2^2 x_{i2}^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$
  
  $= [1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2] \cdot [1, x_{i1}^2, \sqrt{2}x_{i1} x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]^T$
  
  $= \varphi(x)^T \varphi(x_i)$,

  where $\varphi(x) = [1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]^T$.

Nonlinear SVM: Solution

- The solution is basically the same as the linear case, where $x^T x_i$ is replaced with $K(x, x_i)$, and an additional constraint that $\alpha \leq C$ is added.

Nonlinear SVM Summary

Project input to high-dimensional space to turn the problem into a linearly separable problem.

Issues with a projection to higher dimensional feature space:

- **Statistical problem**: Danger of invoking curse of dimensionality and higher chance of overfitting
  - Use large margins to reduce VC dimension

- **Computational problem**: Computational overhead for calculating the mapping $\varphi(\cdot)$:
  - Solve by using the kernel trick.