Text Categorization (III)

Luo Si
Department of Computer Science
Purdue University

Outline
- Support Vector Machine (SVM)
  - A Large-Margin Classifier
    - Introduction to SVM
    - Linear, hard margin
    - Linear, Soft margin
    - Non-Linear SVM (kernel functions)
    - Discussion
**History of SVM**

A brief history of SVM

- SVM is inspired from statistical learning theory by Vapnik (1979) [3]
- Put into practical application as “Large Margin Classifiers” in (1992) [1]
- SVM became famous for its success in handwritten digit recognition [2]
- SVM has been successfully utilized in
  - Image detection
  - Speaker identification
  - Text categorization
  - Many other problems…

---

**Support Vector Machine**

Consider a two-class (binary classification problem like text categorization), find a line to separate data points in two classes.

There are many possible solutions!

Are those decision boundaries equally good?
A slight variation of the data makes some decision boundaries incorrect.

Support Vector Machine

Large-Margin Decision Criterion

The decision boundary should be far away from the data points of two classes as much as possible

Indicates the margin between data points and the decision boundary should be large

Positive and Negative Data points have equal margin
**Large-Margin Decision Criterion**

The margin is:

\[ m = \frac{2}{||w||} \]

**Linear SVM**

Let \{x_1, ..., x_n\} denote input data. For example, vector representation of all documents.

Let \( y_i \) be the binary indicator 1 or -1 that indicates whether \( x_i \) belongs to a particular category or not.

The decision boundary should classify all points correctly:

\[ y_i(w^T x_i + b) \geq 1, \quad \forall i \]

The decision boundary can be found by solving the following constrained optimization problem:

Minimize \[ \frac{1}{2}||w||^2 \]

subject to \[ y_i(w^T x_i + b) \geq 1 \quad \forall i \]
More about Optimization

Primal optimization problem

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad h_i(x) = 0, \quad i = 1, \ldots, p,
\end{align*}
\]

The corresponding Lagrangian of this optimization problem is

\[
L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)
\]

The dual optimization problem function

\[
g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left( f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right)
\]

The dual optimization problem function

\[
\begin{align*}
\text{maximize} & \quad g(\lambda, \nu) \\
\text{subject to} & \quad \lambda \geq 0.
\end{align*}
\]

Linear SVM

The decision boundary can be found by solving the following constrained optimization problem

\[
\text{Minimize} \quad \frac{1}{2} ||w||^2
\]

\[
\text{subject to} \quad y_i(w^T x_i + b) \geq 1 \quad \forall i
\]

The corresponding Lagrangian of this optimization problem is

\[
\mathcal{L} = \frac{1}{2} ||w||^2 - \sum_i \alpha_i \left( y_i(w^T x_i + b) - 1 \right) \quad \alpha_i \geq 0 \quad \forall i
\]
**Linear SVM**

Set the derivative of the Lagrangian to be zero and calculate $W$ by $a_i$, plug new form of $w$ into the Lagrangian, the optimization problem can be written in terms of $a_i$ (the dual problem)

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

Plug new form of $w$ into the Lagrangian, the optimization problem can be written in terms of $a_i$ (the dual problem)

$$\max \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$

The above optimization problem is a quadratic program problem, which means there is a global maximum of $a_i$ can always be found

**The Karush-Kuhn-Tucker Condition**

The optimal solution of model parameter satisfies

$$\alpha_i (1 - y_i (W^T X_i + b)) = 0 \quad \forall i$$

or

$$\left\{ \begin{array}{l}
\alpha_i = 0 \\
(\alpha_i > 0) \wedge (1 - y_i (W^T X_i + b) = 0)
\end{array} \right.$$  

- Each support vector $x_i$ has positive weight
- Non-support vectors have a zero weight
The Karush-Kuhn-Tucker Condition

The optimal solution of model parameter satisfies

- Each support vector $x_i$ has positive weight
- Non-support vectors have a zero weight

![Diagram showing support vectors]

Prediction only needs to consider support vectors; save storage and computation.

Hard Margin Linear SVM Solution

The optimal parameters are

$$w^* = \sum_{i \in SV} \alpha_i y_i X_i$$

$$y_i (W^* X_i - b) = 1 \quad \forall i \in SV$$

Prediction is made by:

$$\text{sign}(WX - b) = \text{sign}(\sum_{i \in SV} \alpha_i y_i (X_i \cdot X) - b)$$
The Karush-Kuhn-Tucker Condition

What about linearly non-separable data?

We tolerate some error for specific data points as
Soft Margin Linear SVM

Introduction “slack variables”, slack variables are always positive

\[
\begin{cases}
  w^T x_i + b \geq 1 - \xi_i & y_i = 1 \\
  w^T x_i + b \leq 1 + \xi_i & y_i = 1 \\
  \xi_i \geq 0 & \forall i
\end{cases}
\]

Introduce const C to balance error for linear boundary and the margin

\[
\frac{1}{2}||w||^2 + C \sum_i \xi_i
\]

The optimization problem becomes

\[
\begin{align*}
\text{Minimize} & \quad \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to} & \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0
\end{align*}
\]

Soft Margin Linear SVM

The dual of the problem for soft margin linear SVM is:

\[
\max \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

subject to \( C \geq \alpha_i \geq 0 \), \( \sum_{i=1}^{n} \alpha_i y_i = 0 \)

\( w \) is calculated as \( w^* = \sum_{i\in S} \alpha_i y_i x_i \)

This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on a_i now.

Once again, a QP solver can be used to find a_i.
Non-linear SVM

Linear SVM only uses a line to separate data points, how to generalize it to non-linear case?

Key idea: transform $X_i$ to a higher dimension space

- Input space: the space the point $x_i$ are located
- Feature space: the space of $f(x_i)$ after transformation
Non-linear SVM

Key idea: transform $X_i$ to a higher dimension space
- Input space: the space the point $x_i$ are located
- Feature space: the space after transformation

Use $\Phi(x_i)$ to transform low level feature to high level feature

Sometimes, the $\Phi(x_i)$ transformation maps to very high dimensional space or even infinite dimensional space

How can we calculate the high dimensional representation for all data points?

The Kernel Trick

Recall the SVM optimization problem

$$
\max W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i, j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j)
$$

subject to $C \geq \alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$

The data points only appear as inner product
As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly
Many common geometric operations (angles, distances) can be expressed by inner products
Define the kernel function $K$ by $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$
Examples for the Kernel trick

Suppose \( f(.) \) is given as follows

\[
\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)
\]

An inner product in the feature space is

\[
\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) \rangle = (1 + x_1y_1 + x_2y_2)^2
\]

So, if we define the kernel function as follows, there is no need to carry out \( f(.) \) explicitly

\[
K(x, y) = (1 + x_1y_1 + x_2y_2)^2
\]

More Kernel Functions

Polynomial kernel with degree \( d \)

\[
K(x, y) = (x^T y + 1)^d
\]

Gaussian Radial basis function kernel with width \( \sigma \)

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

Two-layer sigmoid neural network

\[
K(x, y) = \tanh(nx^T y + \theta)
\]
Kernlized SVM Solution

The optimal parameters are

\[ w^* = \sum_{i \in SV} \alpha_i y_i \phi(X_i) \]

\[ y_i (W^* X_i - b) = 1 \quad \forall i \in SV \]

Prediction is made by:

\[ \text{sign}(W X - b) = \text{sign}(\sum_{i \in SV} \alpha_i y_i (\phi(X_i) \cdot \phi(X)) - b) \]

\[ = \text{sign}(\sum_{i \in SV} \alpha_i y_i (K(X_i, X) - b)) \]

---

Text Categorization: Evaluation

Performance of different algorithms on Reuters-21578 corpus: 90 categories, 7769 Training docs, 3019 test docs, (Yang, JIR 1999)
SVM Toolkit

SMO: Sequential Minimal Optimization
SVM-Light
LibSVM
BSVM
......

Text Categorization (III)

Outline
- Support Vector Machine (SVM)
  A Large-Margin Classifier
  - Introduction to SVM
  - Linear, hard margin
  - Linear, Soft margin
  - Non-Linear SVM (kernel functions)
  - Discussion