PURDUE
UNIVERSITY

CS54701: Information Retrieval

Text Categorization (III)

8 March 2016

Prof. Chris Clifton

Indiana Center for Database Systems ™

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

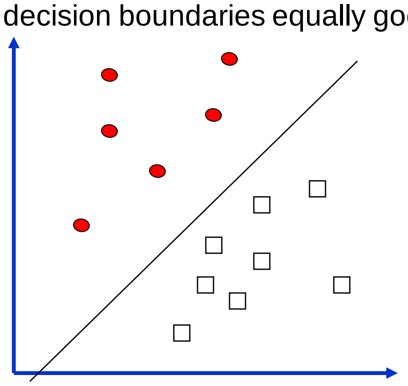
History of SVM

- 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 it success in handwritten digit recognition [2]
- SVM has been successfully utilized in
  - Image detection
  - Speaker identification
  - Text categorization
  - Many other problems…

[1] B.E. Boser *et al.* A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsburgh, 1992.
[2] L. Bottou *et al.* Comparison of classifier methods: a case study in handwritten digit recognition. Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 2, pp. 77-82, 1994.
[3] V. Vapnik. The Nature of Statistical Learning Theory. 2nd edition, Springer, 1999.

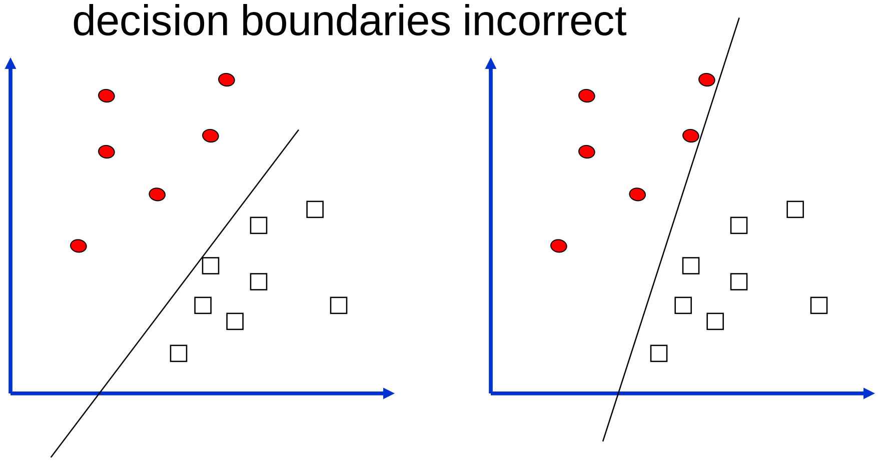
Support Vector Machine

- 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?



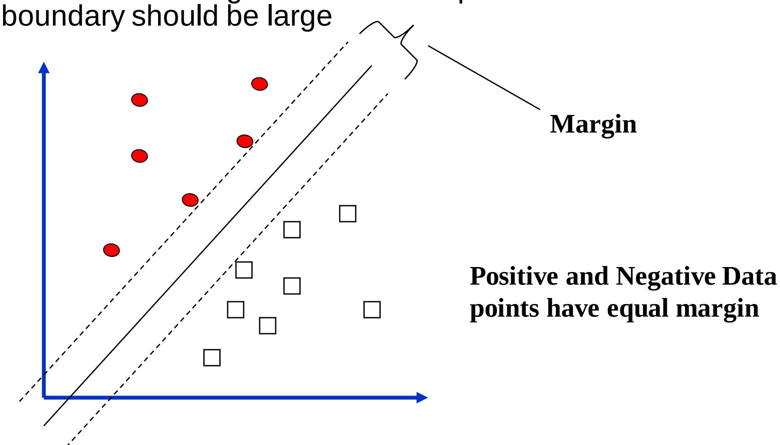
Support Vector Machine

- A slight variation of the data makes some decision boundaries incorrect

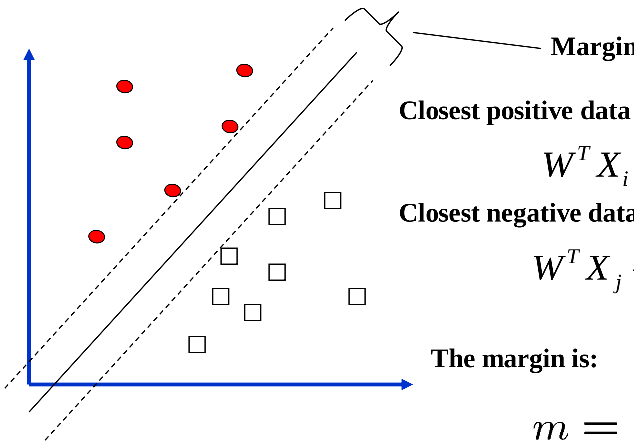


Large-Margin Decision Criterion

- 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



Large-Margin Decision Criterion



Closest positive data point to boundary

$$W^T X_i + b = 1$$

Closest negative data point to boundary

$$W^T X_j + b = -1$$

The margin is:

$$m = \frac{2}{||\mathbf{w}||}$$

Linear SVM

•Let $\{x_1, \ldots, x_n\}$ 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 c or not

The decision boundary should classify all points correctly

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad \forall i$$

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

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

More about Optimization

- Primal optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p, \end{array}$$

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 corresponding Lagrangian dual 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

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0. \end{array}$$

Linear SVM

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

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \ge 1 \qquad \forall i$$

The corresponding Lagrangian of this optimization problem is

$$\mathcal{L} = \frac{1}{2}||\mathbf{w}||^2 - \sum_i \alpha_i \left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right) \qquad \alpha_i \ge 0 \quad \forall i$$

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)

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

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

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

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

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

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

$$\Downarrow$$

Support Vectors

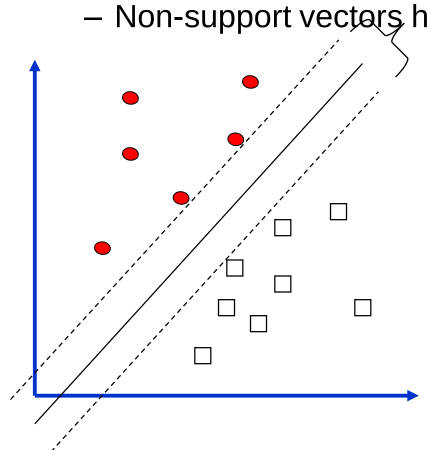
$$or \begin{cases} \alpha_i = 0 \\ (\alpha_i > 0) \wedge \left(1 - y_i (W^T X_i + b) = 0\right) \end{cases}$$

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

68

The Karush-Kuhn-Tucker Condition

- The optimal solution of model parameter satisfies
  - Each support vector xi has positive weight
  - Non-support vectors have a zero weight



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

Hard Margin Linear SVM Solution

•The optimal parameters are

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

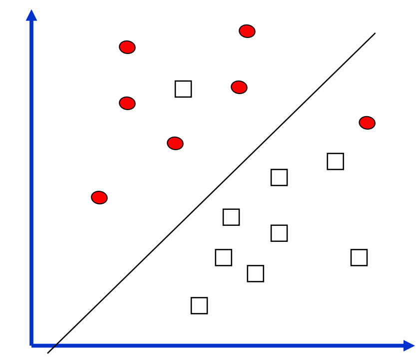
$$y_i(W^* X_i - b) = 1 \quad \forall i \in SV$$

Prediction is made by:

$$sign(WX - b) = sign(\sum_{i \in SV} \alpha_i y_i (X_i \bullet X) - b)$$

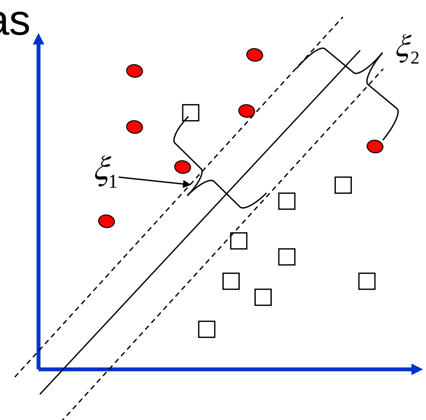
The Karush-Kuhn-Tucker Condition

- What about linearly non-separable data?



The Karush-Kuhn-Tucker Condition

- We tolerate some error for specific data points as



Soft Margin Linear SVM

Introduction "slack variables", slack variables are always positive

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

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

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$$

The optimization problem becomes

$$\begin{array}{l} \text{Minimize } \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{array}$$

Soft Margin Linear SVM

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

$$\begin{array}{l} \text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ \text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \end{array}$$

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

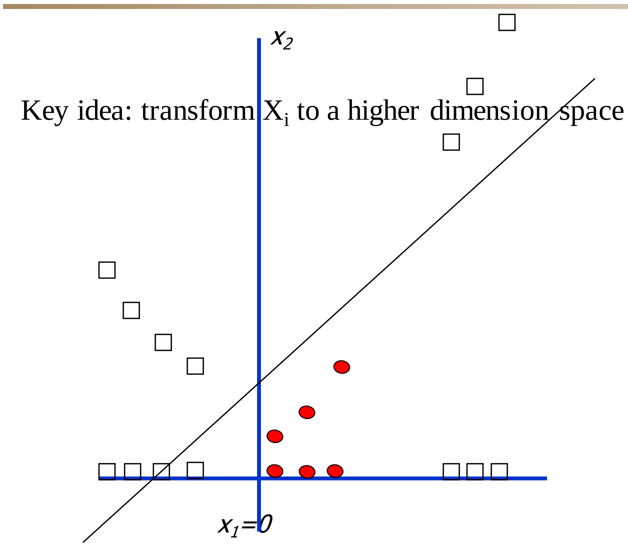
This is very similar to the optimization problem in the linear separable case, except that there is an upper bound *C* on a_i now

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



Non-linear SVM

Key idea: transform X_i to a higher dimension space

- Input space: the space the point $\mathbf{x}_i$ are located
- Feature space: the space after transformation

Use $\Phi(\mathbf{x}_i)$ to transform low level feature to high level feature

Sometimes, the $\Phi(\mathbf{x}_i)$ 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

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

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

Only need inner product

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 $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

Examples for the Kernel trick

- Suppose f(.) is given as follows

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

- An inner product in the feature space is

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

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

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

More Kernel Functions

- Polynomial kernel with degree d

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Gaussian Radial basis function kernel with width σ

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

- Two-layer sigmoid neural network

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

Kernlized SVM Solution

•The optimal parameters are

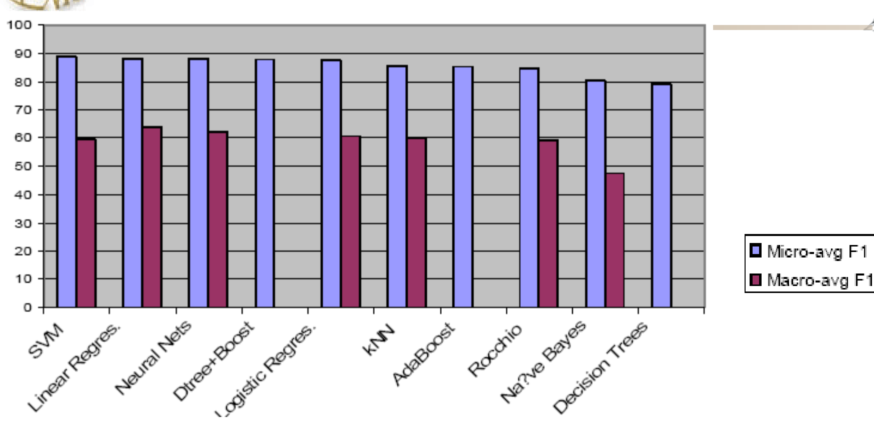
$$w^* = \sum_{i \in SV} \alpha_i y_i \phi(X_i)$$

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

Prediction is made by:

$$\begin{aligned} sign(WX - b) &= sign(\sum_{i \in SV} \alpha_i y_i (\phi(X_i) \bullet \phi(X)) - b) \\ &= sign(\sum_{i \in SV} \alpha_i y_i (K(X_i, X) - b)) \end{aligned}$$

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