Support Vector Machine

Wei-Ta Chu

Introduction – Linearly Separable Data

Consider the set of training data that consist of two classes $S = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, $y_i \in \{-1, 1\}$, $\boldsymbol{x}_i \in \boldsymbol{R}^d$. A linear classifier able to separate the positive from the negative examples will be a hyperline in $\boldsymbol{R}^d$ characterized by a normal $\boldsymbol{w}$ and an offset b:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$$

For a linearly separable data set *S*, there exists a hyperplane that satisfies all the points in *S*

$$\boldsymbol{w}^T\boldsymbol{x}_i + b \geq 0 \text{ for } y_i = +1$$

$$\boldsymbol{w}^T\boldsymbol{x}_i + b < 0 \text{ for } y_i = -1$$

$$\Rightarrow\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 0\ \forall i$$

Equation of a plane
$ax + by + cz + d = 0$
Normal vector: (a,b,c)

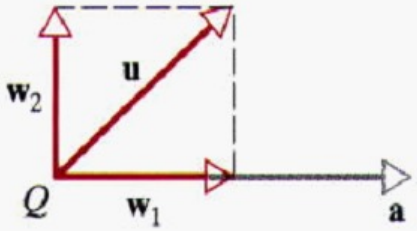
Y. Gong and W. Xu, Machine Learning for Multimedia Content Analysis, Springer, 2007

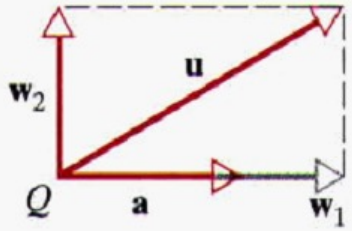
LA Recap: Orthogonal Projection

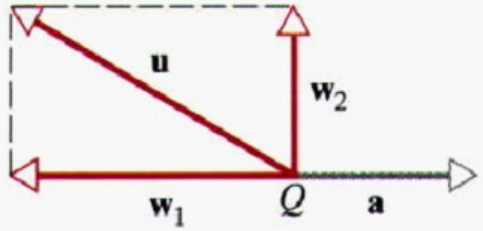
$$\text{proj}_{\mathbf{a}}\,\mathbf{u} = \frac{\mathbf{u}\cdot\mathbf{a}}{\|\mathbf{a}\|^2}\mathbf{a} \qquad \text{(vector component of } \mathbf{u} \text{ along } \mathbf{a})$$

$$\mathbf{u} - \text{proj}_{\mathbf{a}}\,\mathbf{u} = \mathbf{u} - \frac{\mathbf{u}\cdot\mathbf{a}}{\|\mathbf{a}\|^2}\mathbf{a} \qquad \text{(vector component of } \mathbf{u} \text{ orthogonal to } \mathbf{a})$$

$$\|\text{proj}_{\mathbf{a}}\,\mathbf{u}\| = \frac{|\mathbf{u}\cdot\mathbf{a}|}{\|\mathbf{a}\|} = \|\mathbf{u}\|\,|\cos\theta|$$



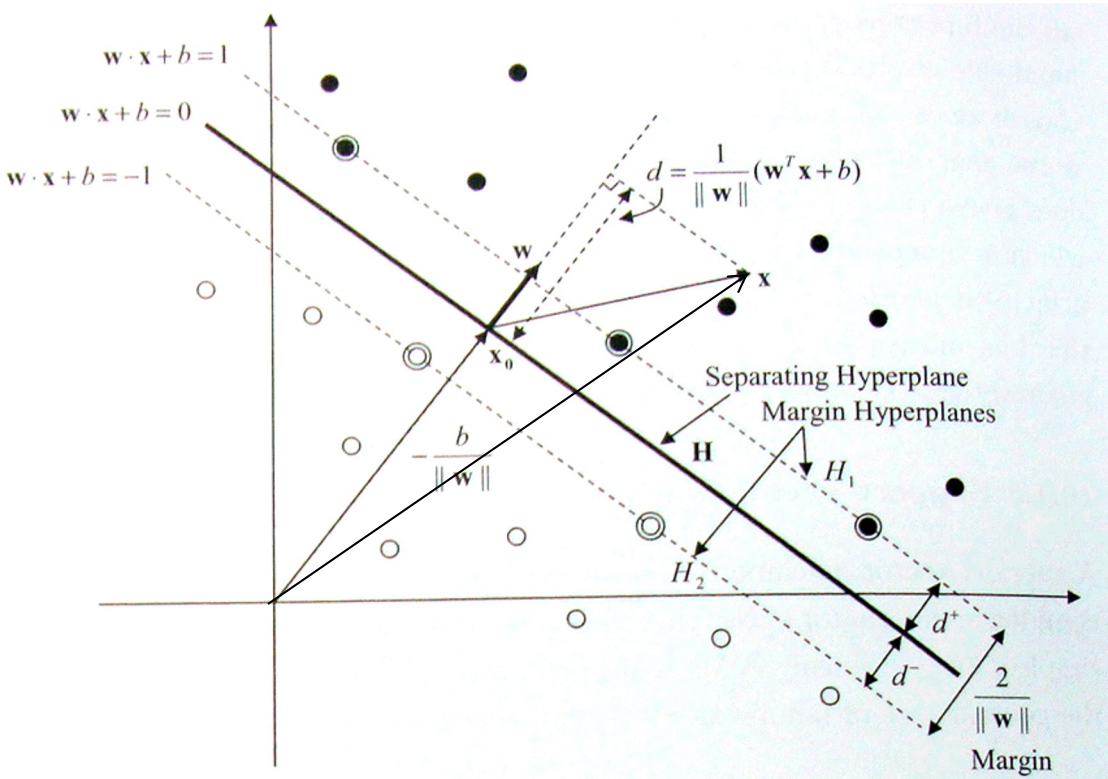




Introduction

- Rescale ($\boldsymbol{w}$,b) so that the closet points to the hyperplane satisfy $|\boldsymbol{w}^T\boldsymbol{x}_i + b| = 1$. This normalization leads to the canonical form for SVM

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \; \forall i$$



Properties

1. For any two points $\mathbf{x}_1$ and $\mathbf{x}_2$ lying on the hyperplane, $\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_2) = 0$. Therefore, $\mathbf{w}$ is the vector normal to the surface of the hyperplane.
2. For any point $\mathbf{x}_h$ on the hyperplane, $\mathbf{w}^T\mathbf{x}_h = -b$.
3. The signed distance from a point $\mathbf{x}$ to the hyperplane is given by

$$\vec{v} \cdot \vec{w} = \|\vec{v}\|\|\vec{w}\| \cos\theta$$

$$\|\vec{v}\| \cos\theta = \frac{\vec{v}\cdot\vec{w}}{\|\vec{w}\|}$$

$$\begin{aligned} d &= \frac{\mathbf{w}}{||\mathbf{w}||}(\mathbf{x} - \mathbf{x}_0) \\ &= \frac{1}{||\mathbf{w}||}(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\mathbf{x}_0) \\ &= \frac{1}{||\mathbf{w}||}(\mathbf{w}^T\mathbf{x} + b) \end{aligned} \tag{10.11}$$

where $\mathbf{x}_0$ is the intersection point between the normal vector $\mathbf{w}$ and the hyperplane. Since $\mathbf{x}_0$ lies on the hyperplane, it satisfies the equality

Properties

$\mathbf{w}^T\mathbf{x}_0 = -b$ in item 2, which leads to the last equality in the above derivation.

4. The perpendicular distance from the hyperplane to the origin equals $|b|/\|\mathbf{w}\|$ (see Problem 10.2 at the end of the chapter).
5. The points for which the equality in (10.10) holds are those points that lie on the hyperplanes $\mathbf{w}^T\mathbf{x} + b = \pm 1$ (denoted as H_1, H_2, respectively), and have the perpendicular distance $1/\|\mathbf{w}\|$ to the separating hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$.

$$d^+ + d^- = \frac{1}{\|\boldsymbol{w}\|} + \frac{1}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}$$

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \ \forall i \quad (10.10)$$

Substitute $\boldsymbol{x}$ as (0,0) in property 3

- The goal of SVM is to find the pair of hyperplanes H1, H2 that maximize the margin, subject to the constraints

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 \; \forall i$$

- This can be formulated as the constrained optimization problem:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w},$$

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

- This is a convex optimization problem for which we are guaranteed to obtain its global optimal solution.

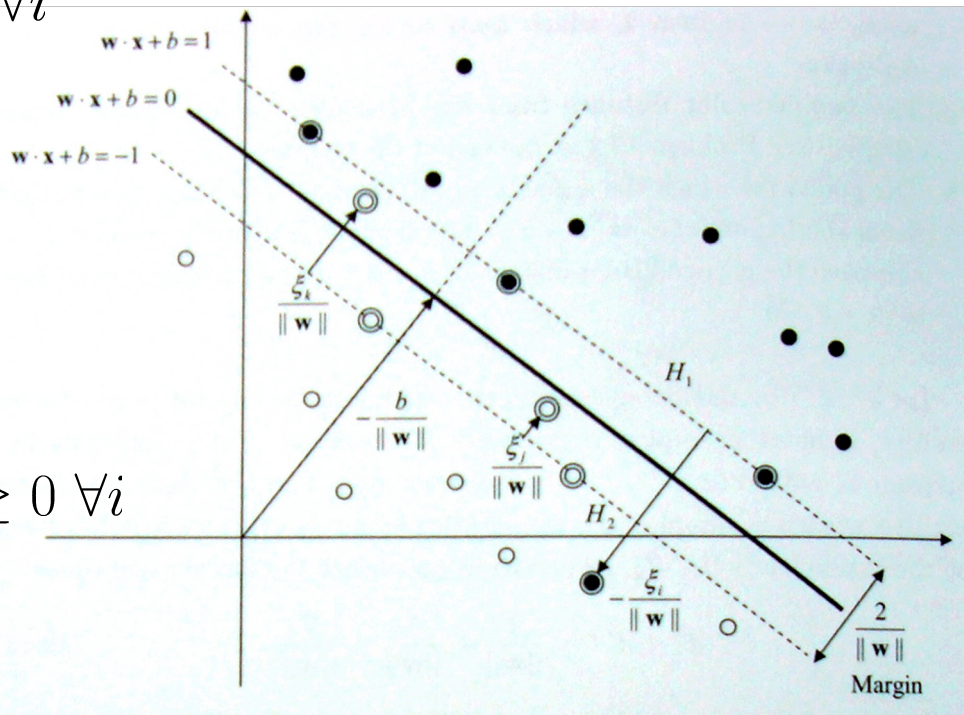
Introduction – Linearly Non-Separable Data

- When the SVM derived above is applied to non-separable data sets, some data points could be at a distance $\xi_i/\|\boldsymbol{w}\|$ on the wrong side.
- We transform the constraint to

  $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0\ \forall i$
- The SVM for non-separable case can be casted as the optimization problem:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{n}\sum_{i=1}^{n}\xi_i,$$

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0\ \forall i$$



Optimization Problem

- Use Lagrange multiplier method to find the optimal solution – transform the original problem to a dual problem

The Dual Problem

$$\min_{\boldsymbol{w},b} \tfrac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \tfrac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0\ \forall i$$

- The Lagrange multiplier method first defines a Lagrange function using a set of non-negative Lagrangian multipliers

$$\boldsymbol{\alpha} = \{\alpha_i\} \quad \boldsymbol{\beta} = \{\beta_i\}$$

$$L_P = \tfrac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \tfrac{C}{n}\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{b}\beta_i\xi_i\ ,$$

- The unconstrained minimum of the Lagrangian function is computed with respect to $\boldsymbol{w}, b,$ and ξ_i

$$\frac{\partial L_P}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \quad \Rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i = 0$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\frac{\partial L_P}{\partial \xi_i} = \frac{C}{n} - \alpha_i - \beta_i = 0 \quad \Rightarrow \quad \alpha_i = \frac{C}{n} - \beta_i$$

The Dual Problem

$$L_P = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{n}\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{b}\beta_i\xi_i \ ,$$

- Substituting these solutions, we obtain the dual objective function

$$L_D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j$$

- Since the primal problem is convex and strictly feasible, its minimum solution can be obtained by equivalently maximizing the dual objective function

$$\max_{\boldsymbol{\alpha}} L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j$$

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

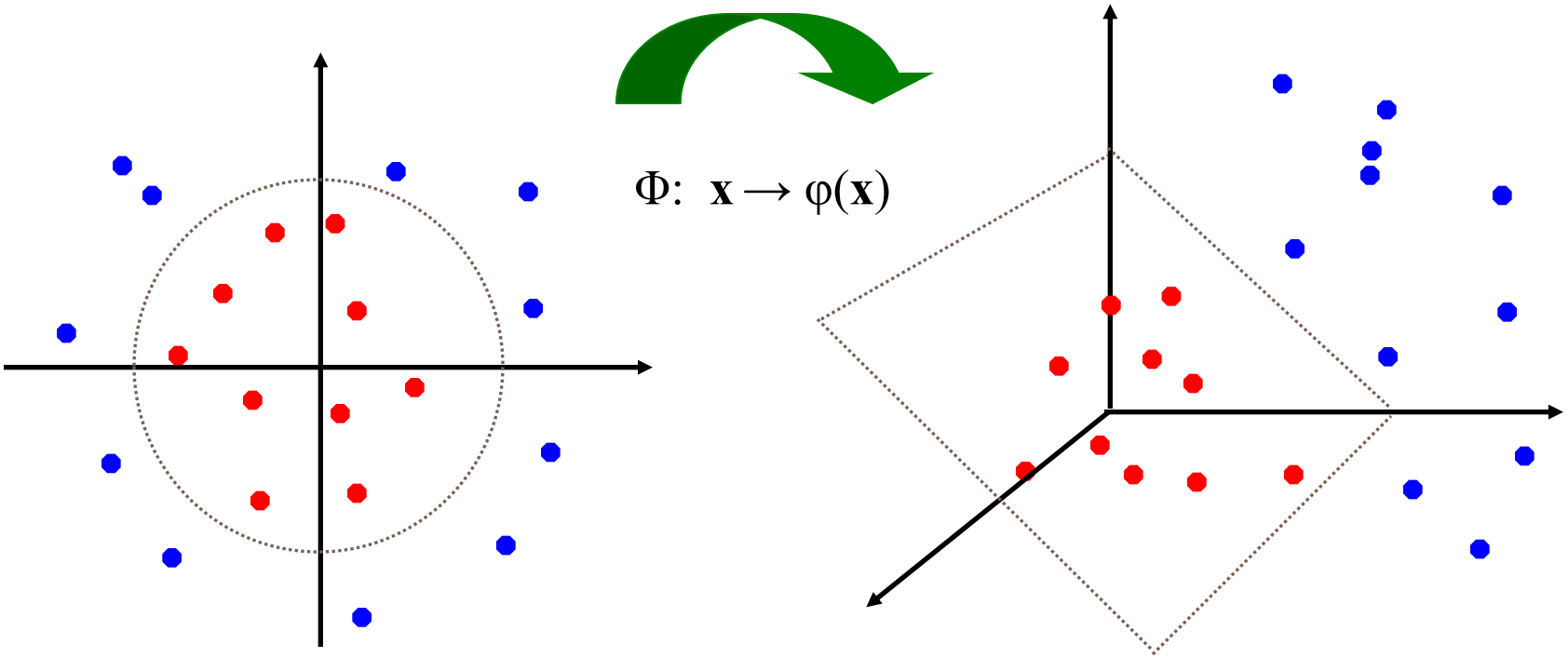
- Once the optimal $\boldsymbol{\alpha}$ set is obtained, the optimal classifier is then given by $f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = sign\left(\sum_{i=1}^{n}\alpha_i y_i(\boldsymbol{x}_i \cdot \boldsymbol{x}) + b\right)$

Kernel Trick

- So far, we discuss the case that finds a linear boundary in the input feature space which divides data into two classes.
- We now extend the ideas of linear SVM to enable the generation of a nonlinear classification boundary using the kernel trick techniques.
  - Implicitly map the original feature space into an enlarged feature space using a kernel function, and to use a linear classifier to conduct data classification in this enlarged space.

General Idea

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Kernel Trick

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = sign\left(\sum_{i=1}^{n} \alpha_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{x}) + b\right)$$

- Note that the input feature vectors appear only in the form of dot products. We do not need to care about individual components of the input vectors. All we need to know is dot products between the input vectors.
- Assume that we enlarge the original feature space by mapping the data into some high dimensional Euclidean space $\mathcal{N}$ using a mapping function $\Phi : R_d \longmapsto \mathcal{N}$
- In the enlarged feature space, because the SVM depends on the data only through dot products $\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$, if there is a "kernel function" K such that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$, then we only need K in computation rather than Φ

Kernel Trick

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = sign\left(\sum_{i=1}^{n} \alpha_i y_i \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}) + b\right)$$
$$= sign\left(\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right)$$

Example: Consider a feature space with two inputs $\boldsymbol{x} = [x_1, x_2]$. Applying the degree-2 polynomial kernel $K(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x} \cdot \boldsymbol{x}')^2$

$$K(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x} \cdot \boldsymbol{x}')^2$$
$$= (1 + x_1 x_1' + x_2 x_2')^2$$
$$= 1 + 2x_1 x_1' + 2x_2 x_2' + (x_1 x_1')^2 + (x_2 x_2')^2 + 2x_1 x_1' x_2 x_2'$$

If we choose the mapping function as

$$\Phi(\boldsymbol{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]^T$$

Which is the function that maps the two dimensional feature space into a six dimensional one, then $K(\boldsymbol{x}, \boldsymbol{x}') = \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x}')$

Kernel Trick

- This example reveals that the degree-2 polynomial kernel function serves to map the original two dimensional feature space into a six dimensional one, and to conduct dot products in the it without the need to explicitly compute $\Phi(\boldsymbol{x})$ for each input data.
- Using kernel trick, we can generalize the linear SVM to a nonlinear one with a minimal change in the framework and a minimal computational cost.
- Examples of kernel functions:

  Degree-d polynomial: $K(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x} \cdot \boldsymbol{x}')^d$

  Radial basis: $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{c})$

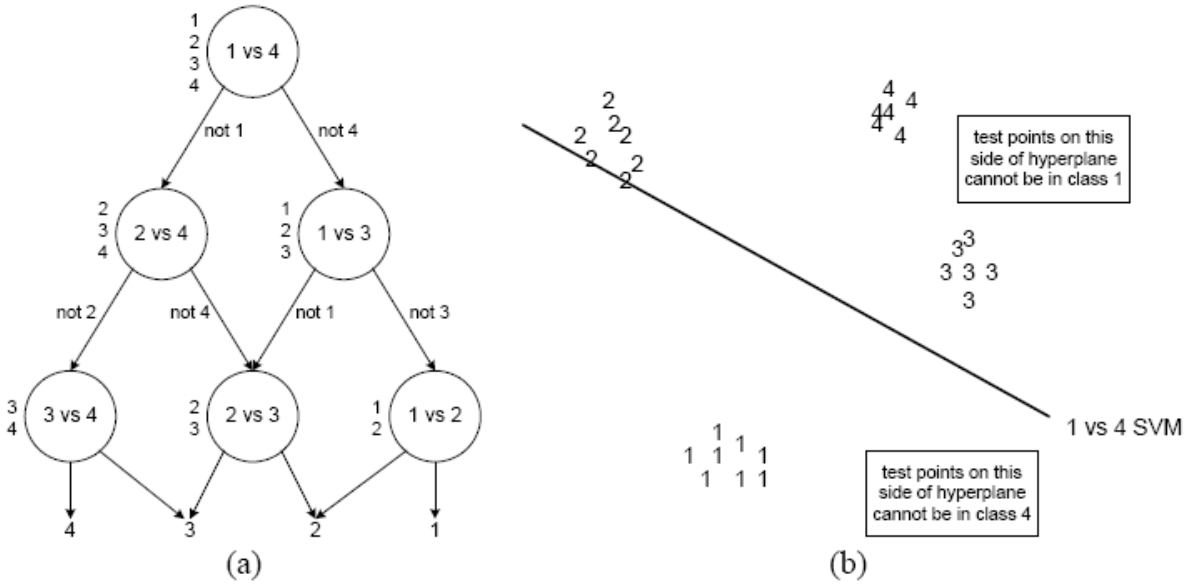
  Neural network: $K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\kappa_1 \boldsymbol{x} \cdot \boldsymbol{x}' + \kappa_2)$

SVM Training/Testing

- Training with labeled data
- Training error vs. testing error
- k-fold cross validation
  - The original sample is partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model, and the remaining K-1 subsamples are used as training data.
  - The cross-validation process is then repeated K times, with each of the K subsamples used exactly once as the validation data.
  - The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation.

Multiclass SVMs

- The support vector machine is fundamentally a two-class classifier.
- For multiclass problems, combine multiple two-class SVMs.
  - "One against one" vs. "one against the rest"
  - E.g. DAGSVM (Decision Acyclic Graph SVM)

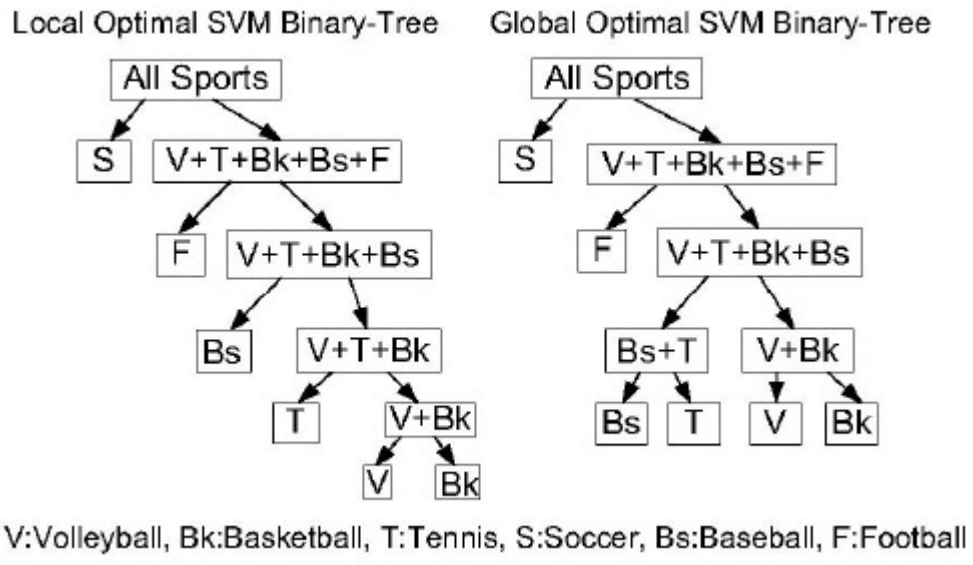


Platt et al. "Large margin DAGs for multiclass classification," Advances in Neural Information Processing Systems, vol. 12, pp. 547-553, 2000

Figure 1: (a) The decision DAG for finding the best class out of four classes. The equivalent list state for each node is shown next to that node. (b) A diagram of the input space of a four-class problem. A 1-v-1 SVM can only exclude one class from consideration.

Case Study

- Automatic video genre categorization
- Temporal features
  - Shot length, cut percentage, average color diff., …
- Spatial features
  - Face frames ratio, average brightness, average color entropy, …



Yuan, et al. "Automatic video genre categorization using hierarchical SVM," Proc. of ICIP, pp. 2905-2908, 2006.

Related Resources

- LIBSVM
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html

References

- L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of IEEE, vol. 77, no. 2, pp. 257-286, 1989.
- Statistical Data Mining Tutorials, Tutorial Slides by Andrew Moore: http://www.autonlab.org/tutorials/index.html
- Jiang, et al., “A new method to segment playfield and its applications in match analysis in sports video,” In Proc. of ACM MM, pp. 292-295, 2004.
- Peng, et al., “Extract highlights from baseball game video with hidden Markov models,” In Proc. of ICIP, vol. 1, pp. 609-612, 2002.

References

- Platt et al. “Large margin DAGs for multiclass classification,” Advances in Neural Information Processing Systems, vol. 12, pp. 547-553, 2000
- Yuan, et al. “Automatic video genre categorization using hierarchical SVM,” Proc. of ICIP, pp. 2905-2908, 2006.
- LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html
- Hidden Markov Model (HMM) Toolbox for Matlab
  - http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html
- The General Hidden Markov Model library (GHMM)
  - http://ghmm.sourceforge.net/
- HTK Speech Recognition Toolkit
  - http://htk.eng.cam.ac.uk