Support vector machines

Václav Hlaváč

Czech Technical University in Prague

Czech Institute of Informatics, Robotics and Cybernetics

160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic

http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz

also Center for Machine Perception, http://cmp.felk.cvut.cz

Courtesy: V. Franc

Outline of the talk:

- Generative vs. discriminative classifier. Maximal margin classifier.
- Minimization of the structural risk.
- SVM, task formulation, solution: quadratic programming.
- Linearly separable case.
- Linearly non-separable case.

Generative / Discriminative classifier design

There are two principal approaches to supervised learning of a classifier. In final, both of them is predicting the conditional probability $p(y|x)$:

- Generative model learns the joint distribution $p(x, y)$. To learn it fully, all combinations of x, y have to be observed, which can be untractable. Having $p(x, y)$ estimate, it predicts the conditional probability $p(y|x)$ with the help of Bayes Theorem. A Generative model explicitly models the actual probability distribution of each class.

  Generative classifiers: Gaussian mixture models, Naïve Bayes, Bayesian networks, Linear discriminant analysis, Hidden Markov Models (e.g., chains), Markov random fields.

- Discriminative model learns the conditional probability $p(y|x)$ or (in SVMs) $\log \frac{p(y=+1|x)}{p(y=+1|x)} \lesseqgtr \Theta$. Both these tasks are much simpler than estimation of $p(x, y)$ in a generative fashion.

  Discriminative classifiers: Perceptron, Support Vector Machines, Logistic regression, k-nearest neighbor, Traditional neural networks.

Towards the support vector machine classifier

- So far in this course, we have used mainly the generative model. A known statistical model was assumed. It induced the appropriate decision rule.
- Since linear classifiers (Perceptron algorithm), we have started the discriminative approach.
- In Support vector machines, We will assume that the class of decision rules is known and we have to choose (discriminate) one of them.

  V. Vapnik: "Learning is the selection of one decision rule from the class of rules".

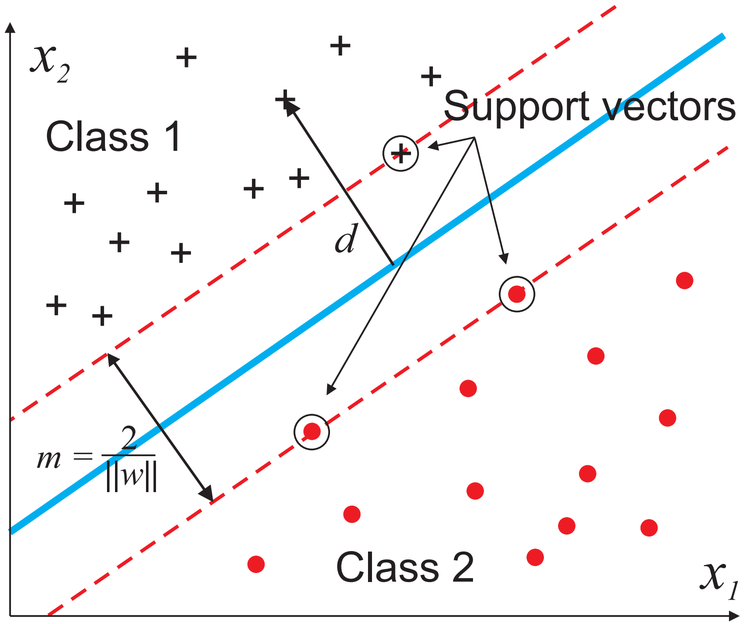
Maximal margin classifier

- We consider a linear classifier with the decision boundary $\langle w, x \rangle + b = 0$.
- We aim at maximizing the margin between classes, which increases generalization ability.
- V. Vapnik proved that this approach minimizes the structural risk. This is the core idea of Support Vector Machines.

- Support vectors are data points closest to the decision boundary.
- The distance of a data point x to the decision boundary is

$$d = \frac{|\langle w, x \rangle + b|}{\|w\|} .$$

- The margin $m = \frac{2}{\|w\|}$.



Support vector machines, the task formulation

- Two hidden states (classes) only, $\{y_1, y_2\}$.
- Task: Find a separable hyperplane (specified by parameters w, b), which maximizes the margin for all $\{x_i, y_i\}$, $i = 1 \dots L$.
- The task expresses as a quadratic programming task

$$(w^*, b^*) = \operatorname*{argmin}_{w,b} \frac{1}{2} \|w\|^2$$

under the constraints

$$\begin{aligned} \langle w, x_j \rangle + b &\geq \quad 1 \qquad \text{for} \quad y_j = 1 \\ \langle w, x_j \rangle + b &< \ -1 \qquad \text{for} \quad y_j = -1 \end{aligned}$$

Support vector machines, a road map

Minimization of the structural risk
Theory how to learn classifier

↓

Linear classifier, a maximal margin classifier
Seek for a maximal margin, respectively. a soft margin for nonseparable data

↓

Learning expressed as quadratic optimization
Primal task

↓

Transformation of the primal task to the dual task
In dual task, data is expressed as scalar products

↓

Support Vector Machines

← ***Extension***

Straightening of the feature space by embedding into higher-dim linear space, typically by use of kernel functions

Minimization of the structural risk (1)

Introduction

- The classifier is learnt from a finite training (multi-)set.
- The statistical model $p(x, y)$ is unknown. Chervonenkis and Vapnik derived an upper bound on the risk

$$\sum_x \sum_y p(x, y)(y \neq Q(x)) \; ,$$

which does not involve $p(x, y)$.
- The upper bound is provided which sums errors on the training (multi-)set and the generalization error. When learning is performed, it should minimize training error and also the complexity of the classifier has to be controlled.

Minimization of the structural risk (2)

Assumptions

- $x \in \mathbb{R}^n$ ... observation of the object (a vector of measurements).
- $y \in \{-1, 1\}$ ... hidden states. This notation leads to more compact derivations and formulas.
- There is a training (multi-)set available
  $\{(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L)\}$,
  which is drawn randomly and generated by an unknown probability distribution $p(x, y)$.

Minimization of the structural risk (3)

The aim is to find a classifier (decision strategy) $q(x, \Theta)$,

where Θ is a parameter (usually vector of parameters) with the minimal expected classification error

$$R_{exp}(q(x, \Theta)) = \int \frac{1}{2} \left| y - q(x, \Theta) \right| \, \mathrm{d}\, p(x, y)$$

The simple approximation of R_{exp} is the empirical risk R_{emp},

$$R_{emp}(q(x, \Theta)) = \frac{1}{L} \sum_{i=1}^{L} \frac{1}{2} \left| y_i - q(x_i, \Theta) \right| .$$

Note: a 1/0 loss (penalty) function is used, i.e., $\frac{1}{2} \left| y - q(x, \Theta) \right| = \begin{cases} 0 & \text{if } y = q(x, \Theta) \, , \\ 1 & \text{if } y \neq q(x, \Theta) \, . \end{cases}$

Minimization of the structural risk (4)

Complications

The expected risk $R_{exp}(q(x, \Theta))$ cannot be calculated because the joint probability distribution $p(x, y)$ is unknown.

Solution

Use the upper bound called guaranteed or structural risk $J(\Theta)$ as proposed by Chervonenkis-Vapnik.

$$R(\Theta) \leq J(\Theta) = R_{\mathrm{emp}}(\Theta) + \sqrt{\frac{h\left(\log\left(\frac{2L}{h}\right) + 1\right) - \log\left(\frac{\eta}{4}\right)}{L}}\,.$$

Minimization of the structural risk (5)

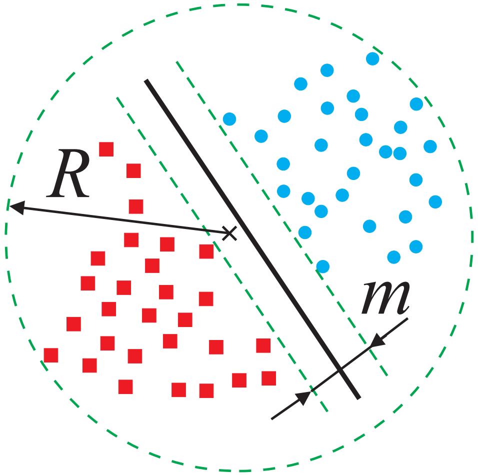
- $R_{emp} = \frac{1}{L} \sum_{i=1}^{L} \frac{1}{2}|y_i - f(x_i, \Theta)|$ is the empirical risk.
- h is a VC dimension characterizing the class of decision functions $q(x, \Theta) \in Q$.
- L is the length of the training multi-set.
- η is the degree of belief into the bound $R(q(x, \Theta))$, i.e., $0 \leq \eta \leq 1$.

- The structural risk minimization principle means a selection of a classifier based on a minimization of the guaranteed risk $J(\Theta)$.
- Support Vector Machines implement an instance of the structural risk minimization principle.

Linearly separable SVM (1)

The aim is to find a linear discriminant function

$$q(x, w, b) = \operatorname{sign}(\langle w, x \rangle + b) = \operatorname{sign}\left(w^\top x + b\right)$$



- VC dimension (capacity) depends on the margin m

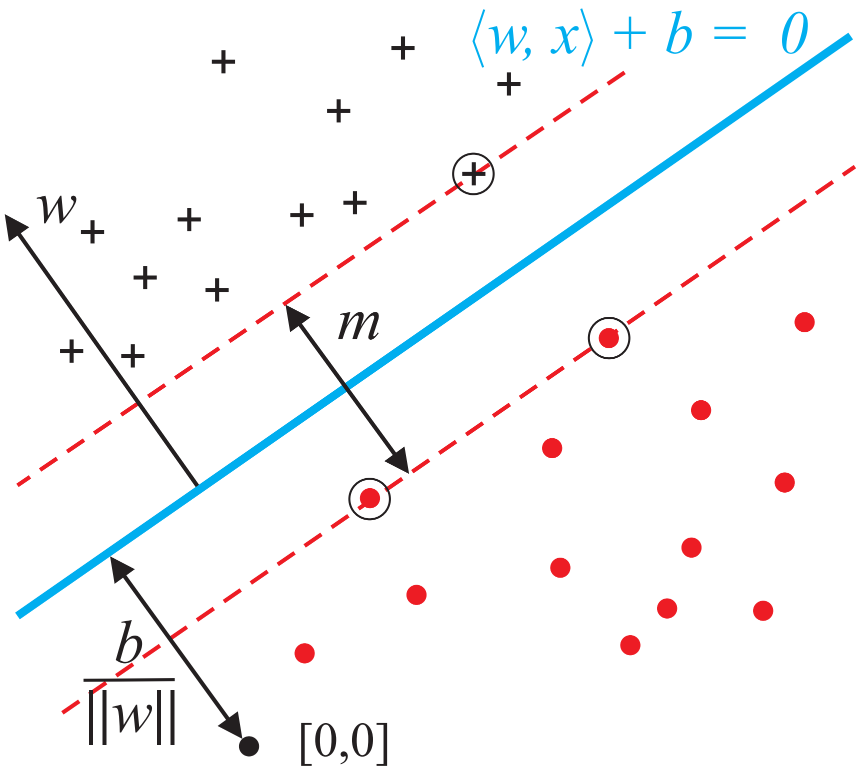
$$h \leq \frac{R^2}{m^2} + 1$$

- R is given by the data itself.
- Margin m can be optimized in the classifier design.

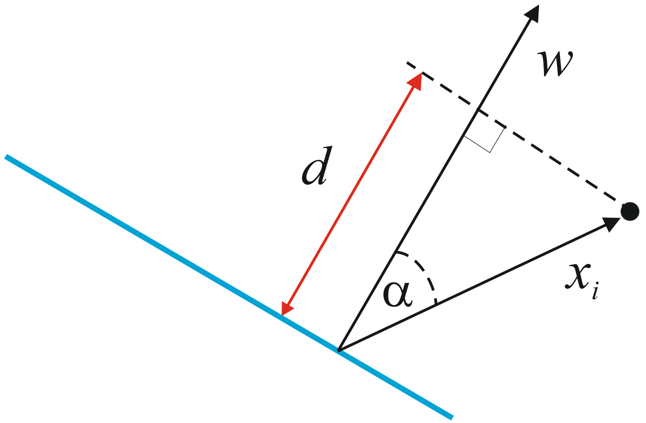
Conclusion: separation hyperplanes with a larger margin have a lower VC dimension $\Leftrightarrow$ lower value of the upper bound.

Linearly separable SVM (2)

The separating hyperplane is sought which maximizes distance to the data (margin m).



Linearly separable SVM (3)



Derivation of the distance d between the observation x_i and the separating hyperplane $w^\top x_i + b = 0$

$$\cos\alpha = \frac{w^\top x_i}{\|w\|\|x_i\|}, \quad \cos\alpha = \frac{d}{\|x_i\|} \quad \Rightarrow \quad d = \frac{w^\top x_i + b}{\|w\|}$$

The parameter b gives the distance from the origin of coordinates.

Linearly separable SVM, the primal task

The optimization task, i.e. seeking the optimal weight vector w^* and optimal bias b^*

$$(w^*, b^*) = \underset{w,b}{\operatorname{argmax}} \min_{i=1,\dots,L} \frac{w^\top x_i + b}{\|w\|} y_i$$

can be converted in to a standard quadratic programming task, which is called the primal task

$$(w^*, b^*) = \operatorname{argmin} \frac{1}{2} \|w\|^2$$

$$w^\top x_i + b \geq +1 \,, \quad y_i = +1$$

$$w^\top x_i + b \leq -1 \,, \quad y_i = -1$$

Properties:

- Convex optimization, strictly convex.
- Unique solution for a linearly separable sample.

Towards the dual task

- The aim is to convert the primal task into its dual formulation, which allows to use kernel functions.
- Lagrange function $\mathcal{L}$ is introduced, α_i are Lagrange multipliers,

$$\mathcal{L}(w, b, \alpha_i) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i \left(w^\mathsf{T} x_i + b\right) \; y_i + \sum_{i=1}^{L} \alpha_i \, . \quad \text{(Eq. 1)}$$

- Let formulate the dual task,

$$(w^*, b^*, \alpha^*) = \underset{w,b}{\operatorname{argmin}} \; \max_{\alpha \geq 0} \; \mathcal{L}(w, b, \alpha) \qquad \text{Primal task.}$$

$$(w^*, b^*, \alpha^*) = \underset{\alpha \geq 0}{\operatorname{argmax}} \; \min_{w,b} \mathcal{L}(w, b, \alpha) \qquad \text{Dual task.}$$

- For convex problems, both formulations lead to the same optimum.

The solution to the dual task

$$\min_{w,b} \max_{\alpha_i > 0} \mathcal{L}(w, b, \alpha_i) = \max_{\alpha_i > 0} \min_{w,b} \mathcal{L}(w, b, \alpha_i)$$

- Seek the optimum, i.e., 1st partial derivatives = 0,

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{L} \alpha_i y_i x_i \,, \qquad \frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{L} \alpha_i y_i = 0 \,.$$

- Substitute to (Eq. 1), slide 16, get rid off w, b and get

$$\alpha_i = \operatorname*{argmax}_{\alpha_i} \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j x_i^\top x_j \,, \quad \alpha_i \geq 0 \,, \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \,.$$

SVM decision strategy

$$w = \sum_{i=1}^{L} \alpha_i \, y_i \, x_i \, .$$

$$q(x) = w^\top x + b = \sum_{i=1}^{L} \alpha_i \, y_i \, x_i^\top x + b \, .$$

- Support vectors are vectors x_i such that

$$\alpha_i \neq 0 \quad \text{and} \quad y_i(w^\top x + b) = 1$$

- Note: Support vectors are not unique.

SVM – the primal and the dual tasks

Primal task

- Optimized according to vector $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$.
- Number of variables is $L + 1$.
- Number of linear constraints is $2L$.

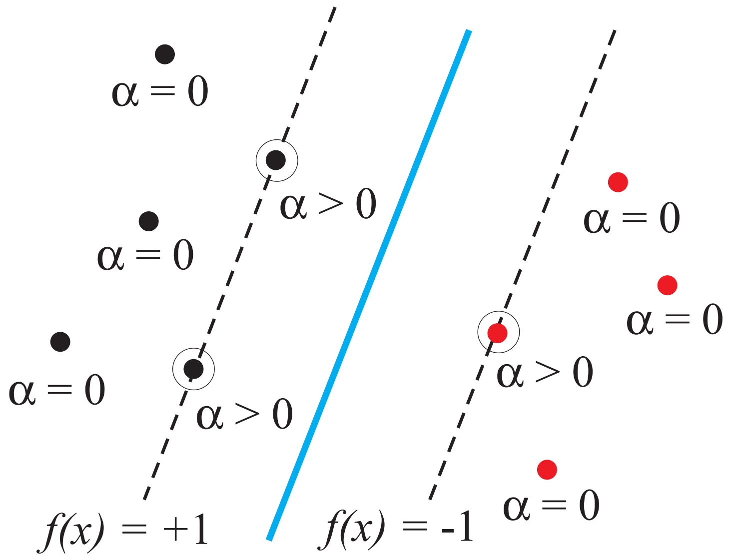
Dual task

- Optimized according to $\alpha_1, \alpha_2, \ldots, \alpha_L$, $\alpha_i \in \mathbb{R}$.
- Number of variables is L.
- Number of linear constraints is $L + 1$.
- Data appear as scalar products only, i.e., $x_i^\top x_j$.

The dual task properties, cont.

- The solution is sparse. Many α_i equal to 0.
  $\alpha_i = 0 \Rightarrow y_i(w^\top x_i + b) \geq 1.$
  $\alpha_i > 0 \Rightarrow y_i(w^\top x_i + b) = 1.$

- Data x_i for which $\alpha_i > 0$ are called Support Vectors. $w = \sum_{i=1}^{L} \alpha_i y_i x_i = \sum_{i \in \mathrm{SV}} \alpha_i y_i x_i$



Calculation of b for $i \in$ SV:

$y_i(w^\top x_i + b) = 1 \Rightarrow$

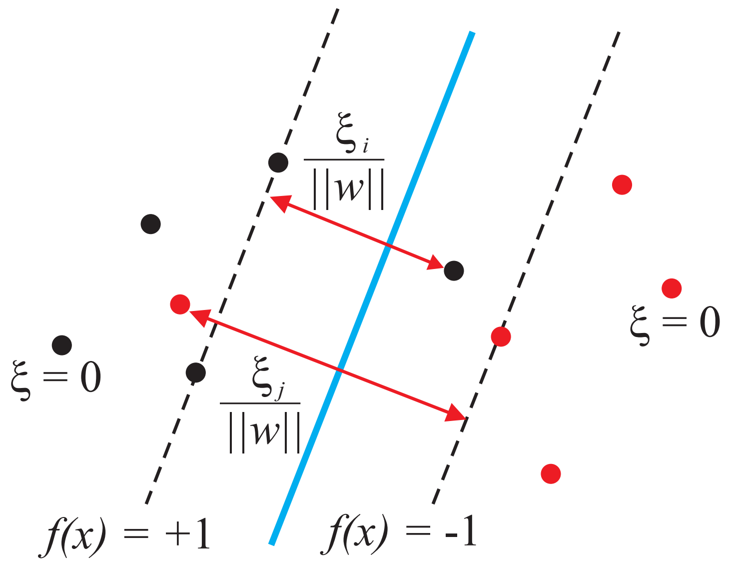
$b = \frac{1 - y_i w^\top x_i}{y_i} = y_i \langle w, x \rangle$

- One support vector should be enough.
- Practically, many support vectors are considered. The mean of corresponding b is used.

SVM linearly non-separable. Soft-margin SVM

Nonseparable data $\Leftrightarrow$ It is not possible to find a separable hyperplane without errors, i.e., $\alpha_i = 0$ $\Rightarrow y_i(w^\top x_i + b) \ngeq 1$.

- Solution: Regularization, i.e., introduction of non-negative slack variables ξ_i, $\alpha_i = 0 \Rightarrow$ $y_i(w^\top x_i + b) \geq 1 - \xi_i$.
- Slack variables measure and penalize the degree of misclassification of the data point x_i in the optimization.
- Suggested by Corina Cortes and Vladimir Vapnik in 1995.



Soft margin SVM with the linear penalty function

$$(w^*, b^*, \xi^*) = \operatorname*{argmin}_{w,b,\xi} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{L} \xi_i^y \,, \quad \text{where}$$

C is a regularization constant. Large C penalizes errors; small C penalizes the complexity of the decision function; $C = \infty$ represents the separable case.

$$w^\top x_i + b \geq +1 - \xi_i \,, \quad y_i = +1$$

$$w^\top x_i + b \leq -1 + \xi_i \,, \quad y_i = -1$$

Optimization criterion, marginal behavior

- $\min \|w\|^2$ – maximization of the margin.
- $\sum_{i=1}^{L} \xi_i^y$ – number misclassified training points (upper bound on the empirical error).

Quadratic programming for the dichotomic task, i.e., $y = -1, 1$ or $|\mathrm{Y}| = 2$.

SVM linearly non-separable, cont.

- Transform to the dual task, analogically to the separable case.

$$\alpha_i = \operatorname*{argmax}_{\alpha_i} \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j x_i^\top x_j \,,$$

$$0 \le \alpha_i \le C \,, \qquad \sum_{i=1}^{L} \alpha_i y_i = 0 \,.$$

Note: $\le C$ above is the only difference when comparing to the linearly separable case.

- The decision strategy is

$$q(x) = w^\top x + b = \sum_{i=1}^{L} \alpha_i \, y_i \, x_i^\top x + b \,.$$

Soft margin SVM, the theoretic backing

$$\text{Risk} = \frac{C}{L}\left(\frac{R^2 + \left(\sum_{i=1}^{L}\xi_i\right)\log\left(\frac{1}{L}\right)}{m^2}\log^2 L + \log\left(\frac{1}{\eta}\right)\right)$$

is minimized when

$$\|w\|^2 R + \left(\sum_{i=1}^{L}\xi_i\right)\log\left(\frac{1}{\sqrt{(\|w\|)}}\right)$$

is minimal.

This matches to Soft Margin SVM criterion with exception to the last term on the right side.

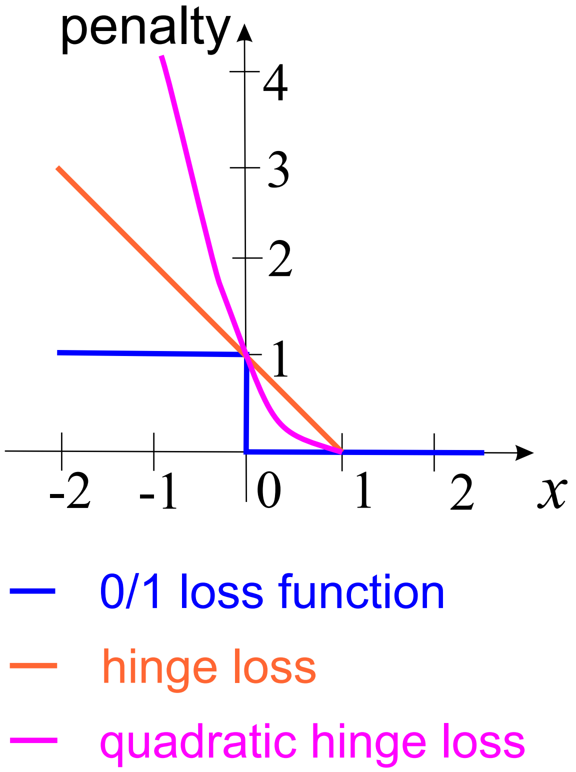
Interpretation of the constant C

- Parameter C represents a trade-off between the misclassification (maximizing the margin) and the classifier complexity (given by the VC-dimension; minimizing the training error).
  - Large values of C favor solutions with few misclassifications.
  - Small values of C express a preference towards low-complexity solutions.
- Parameter C can be viewed as a regularization parameter.
- A suitable value for C is typically determined by trying several values of $C = C_1, \ldots, C_m$. The best value is selected by the cross-validation.
- The general problem of determining a hyperplane minimizing the error on the training set is NP-complete (as a function of dimension).

Courtesy: Mehryar Mohri

Loss functions, hinge loss

- Other convex functions of the slack variables could be used.
- Our choice and similar ones, e.g., with squared slack variables lead to a convenient formulation and solution.



A high-dimensional feature space

- Observations:
  - Generalization bound does not depend on the dimension but on the margin.
  - It this suggests seeking a large-margin separating hyperplane in a higher-dimensional feature space.
- Computational difficulties:
  - Computing dot products in a high-dimensional feature space can be very costly.
  - The solution is based on kernel functions (next lecture).

Courtesy: Mehryar Mohri

Multi-class SVMs

Several approaches are used:

- Direct multi-class formulation.
- One-against all.
- One against one.
- DAG, Directed Acyclic Graphs.

Multi-class SVMs, one-against-all

- So far, we have considered only SVMs handling two-class problems, i.e, dichotomic classification.
- If the task is to classify into N classes then then learn N independent SVMs such that
  - SVM 1: learns $y = 1$ vs. $y \neq 1$.
  - SVM 2: learns $y = 2$ vs. $y \neq 2$.
  - $\ldots$
  - SVM N: learns $y = N$ vs. $y \neq N$.
- When deciding about new observation in a run mode, apply all N SVMs and select the class by looking which SVM puts the prediction the furthest into the positive region.