ĐẠI HỌC BÁCH KHOA SOICT

HUST
ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Support Vector and Kernel Methods

Dam Quang Tuan

Overview

What is an SVM?

- optimal hyperplane and soft-margin for inseparable data
- handling non-linear rules and non-standard data using kernels

How to use SVMs effectively and efficiently?

How to train SVMs?

- decomposition algorithms / primal vs. dual / shrinking

Why can SVMs learn?

- worst-case / average-case / relation to cross-validation

When do SVMs work well?

- properties of classification tasks - a case study in text classification

SVM-{ranking, novelty detection, regression, ...}?

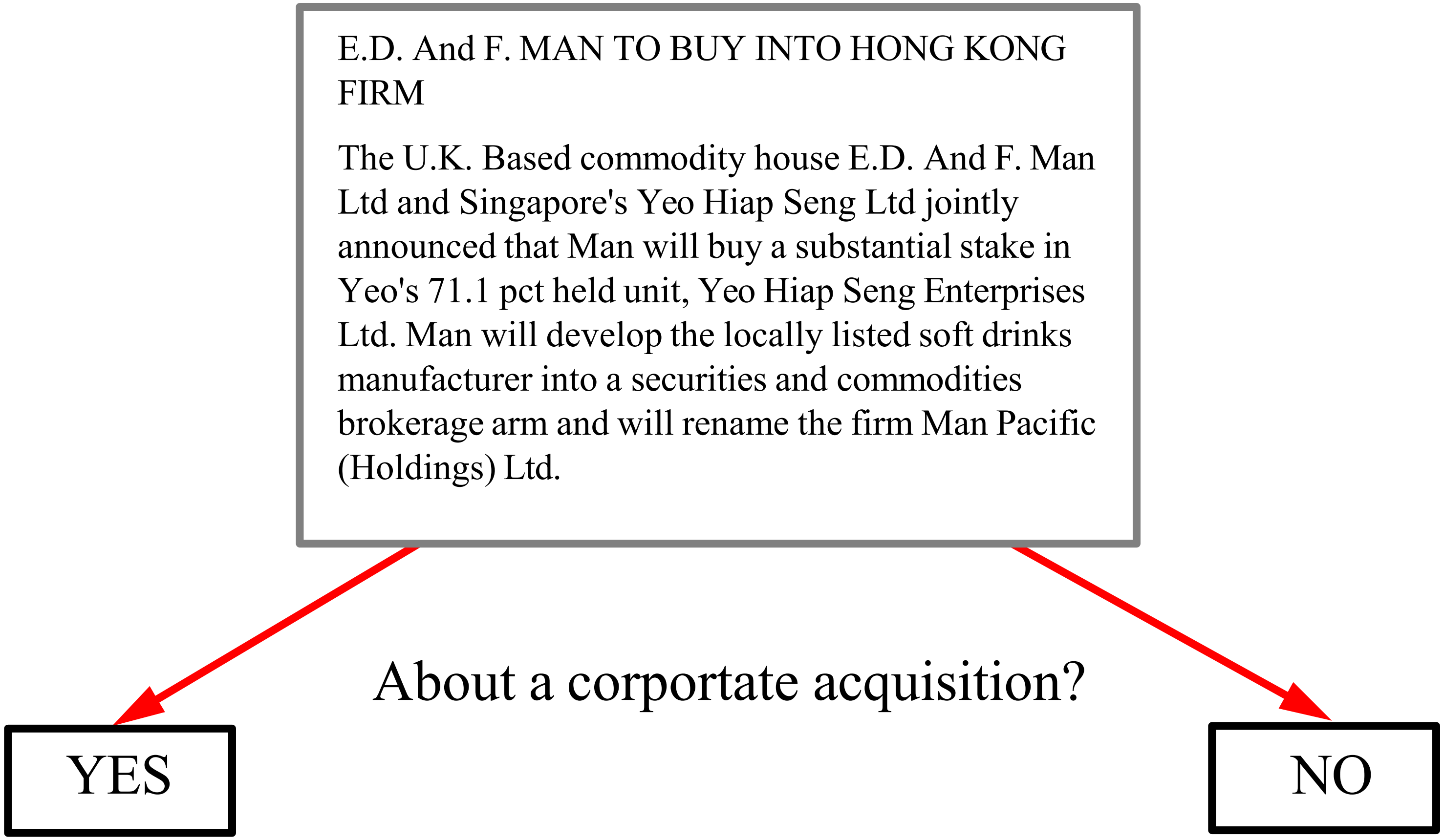
- ranking e.g. learning retrieval functions
- novelty detection: e.g. topic detection

What I will not (really) talk about...

- SVMs in the transductive setting [Vapnik, 1998][Joachims, 1999c][Bennet & Demiriz, 1999]
- Kernel Principal Component Analysis [Schoelkopf et al., 1998]
- connection to related methods (i.e. Gaussian Process Classifiers, Ridge Regression, Logistic Regression, Boosting) [Cristianini & Shawe-Tylor, 2000][MacKay, 1997][Schoelkopf & Smola, 2002]

Warning: At some points throughout this tutorial, precision is sacrificed for better intuition (e.g. uniform convergence bounds for SVMs).

Text Classification



E.D. And F. MAN TO BUY INTO HONG KONG FIRM

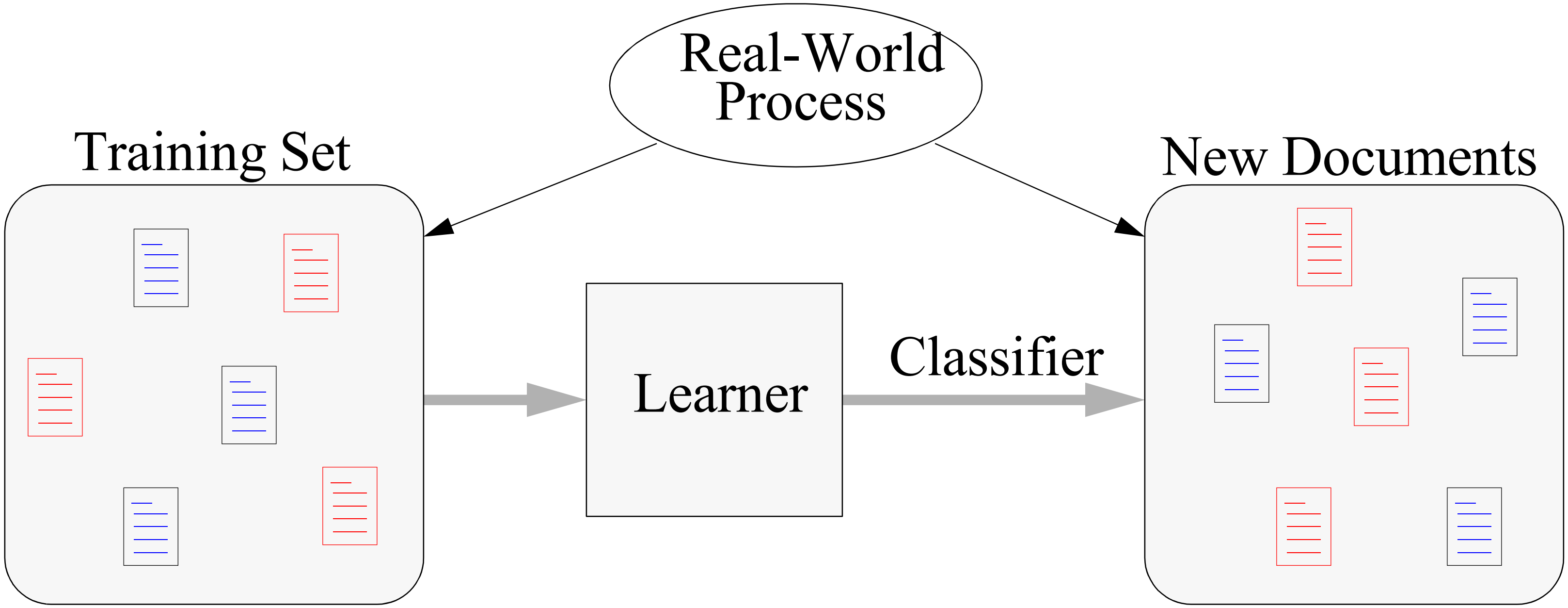
The U.K. Based commodity house E.D. And F. Man Ltd and Singapore's Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeo's 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

About a corportate acquisition?

YES

NO

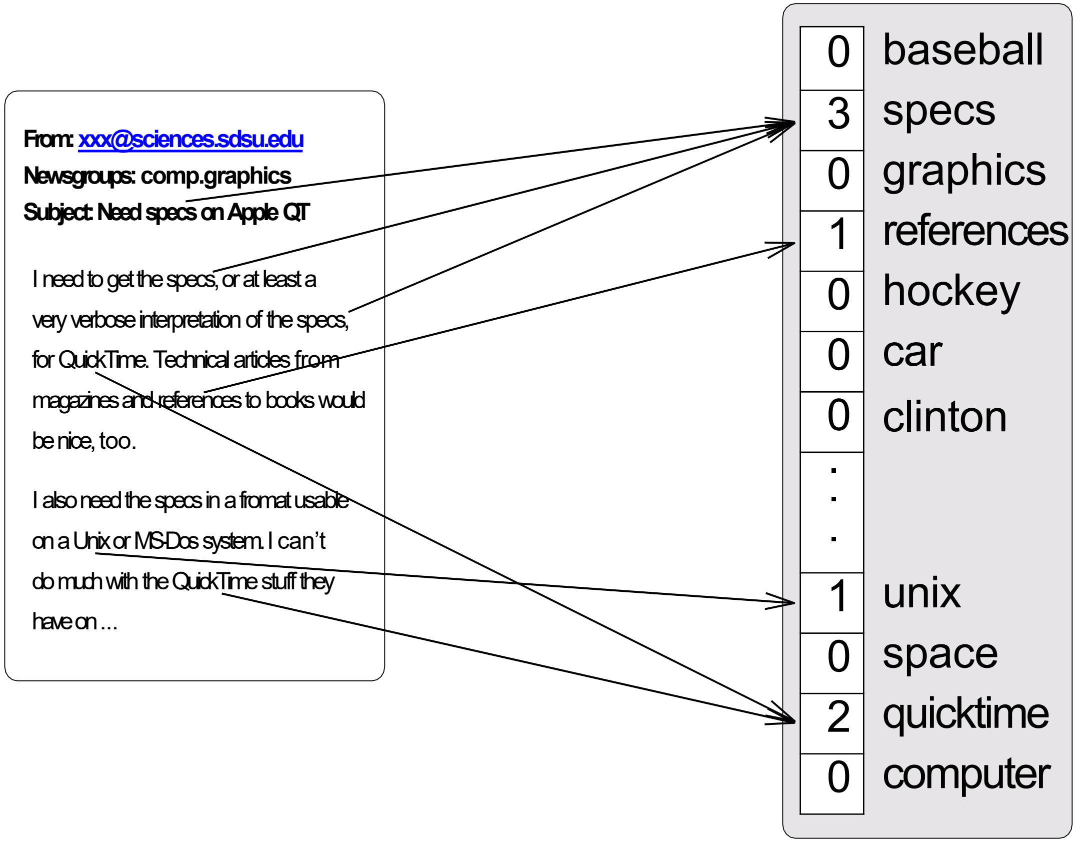
Learning Text Classifiers



Goal:

- Learner uses training set to find classifier with low prediction error.

Representing Text as Attribute Vectors



From: xxx@sciences.sdsu.edu
Newsgroups: comp.graphics
Subject: Need specs on Apple QT

I need to get the specs, or at least a very verbose interpretation of the specs, for QuickTime. Technical articles from magazines and references to books would be nice, too.

I also need the specs in a fromat usable on a Unix or MS-Dos system. I can't do much with the QuickTime stuff they have on ...

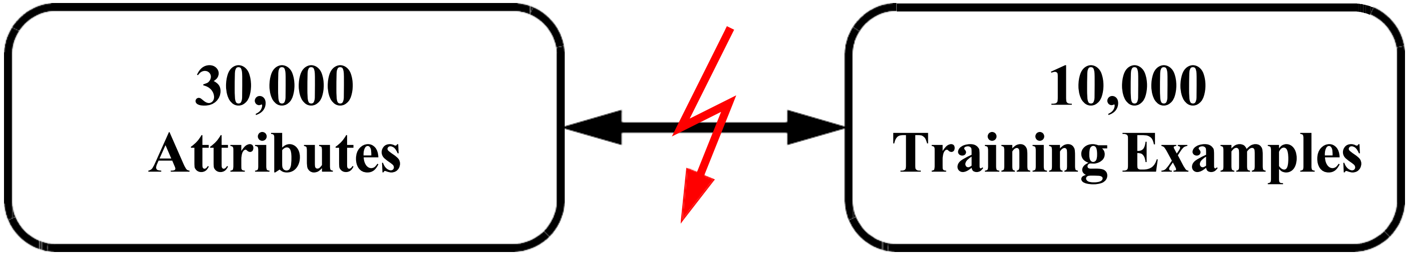
0	baseball
3	specs
0	graphics
1	references
0	hockey
0	car
0	clinton
. . .	
1	unix
0	space
2	quicktime
0	computer

Attributes: Words (Word-Stems)

Values: Occurrence-Frequencies

==> The ordering of words is ignored!

Paradoxon of Text Classification



30,000
Attributes

10,000
Training Examples

... but this is not necessarily a problem!

Good News: SVMs can overcome this problem!

Bad News: This does not hold for all high-dimensional problems!

Experimental Results

Reuters Newswire

- 90 categories
- 9603 training doc.
- 3299 test doc.
- ~27000 features

WebKB Collection

- 4 categories
- 4183 training doc.
- 226 test doc.
- ~38000 features

Ohsumed MeSH

- 20 categories
- 10000 training doc.
- 10000 test doc.
- ~38000 features

microaveraged precision/recall breakeven-point [0..100]	**Reuters**	**WebKB**	**Ohsumed**
Naive Bayes	72.3	82.0	62.4
Rocchio Algorithm	79.9	74.1	61.5
C4.5 Decision Tree	79.4	79.1	56.7
k-Nearest Neighbors	82.6	80.5	63.4
SVM	**87.5**	**90.3**	**71.6**

Table from [Joachims, 2002]

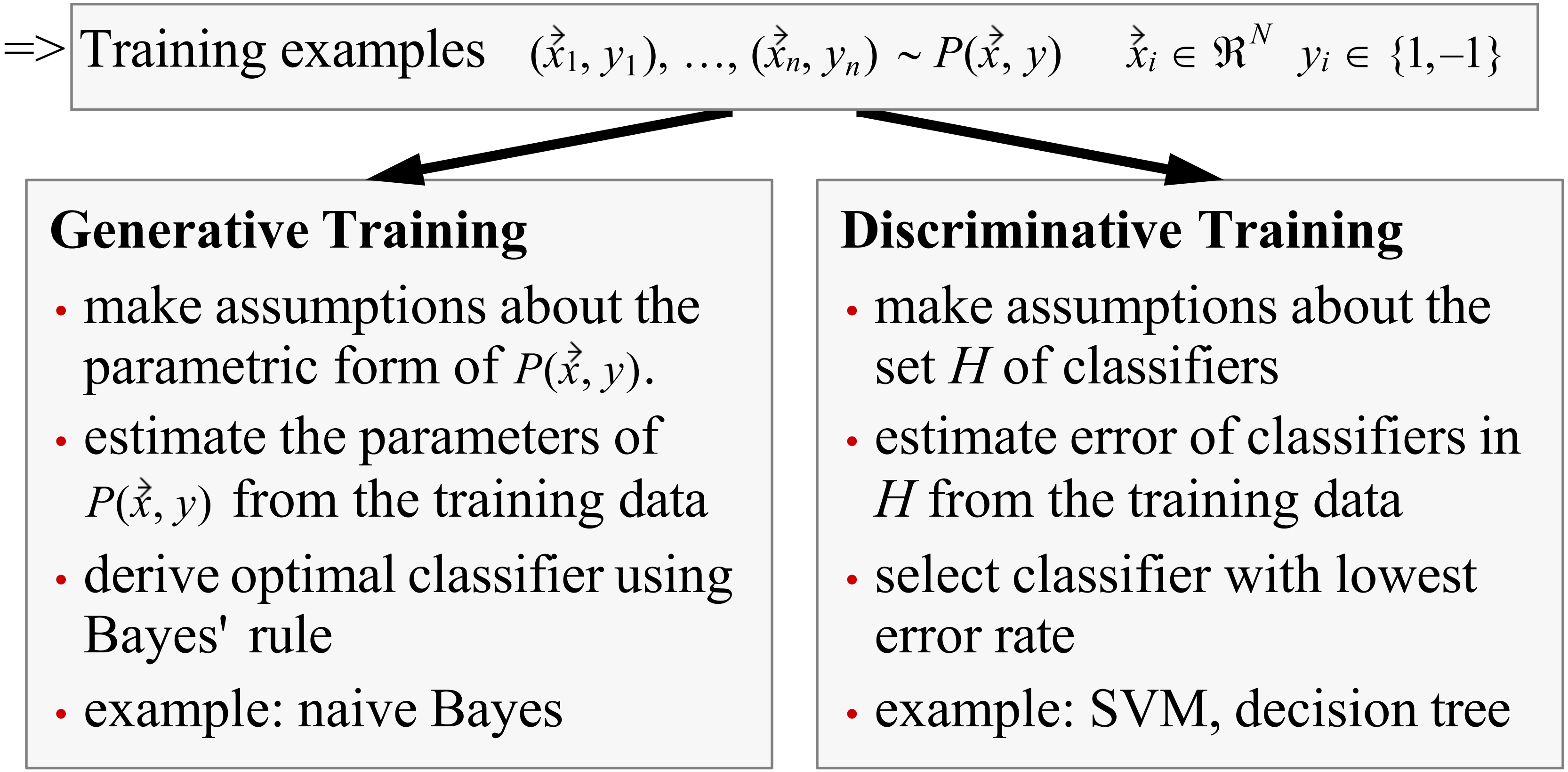
Part 1 (a): What is an SVM? (linear)

- prediction error vs. training error
- learning by empirical risk minimization
- VC-Dimension and learnability
- linear classification rules
- optimal hyperplane
- soft-margin separation

Generative vs. Discriminative Training

Process:

- Generator: Generates descriptions $\vec{x}$ according to distribution $P(\vec{x})$.
- Teacher: Assigns a value y to each description $\vec{x}$ based on $P(y|\vec{x})$.



=> Training examples $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(\vec{x}, y)$ $\quad \vec{x}_i \in \Re^N \quad y_i \in \{1,-1\}$

Generative Training

- make assumptions about the parametric form of $P(\vec{x}, y)$.
- estimate the parameters of $P(\vec{x}, y)$ from the training data
- derive optimal classifier using Bayes' rule
- example: naive Bayes

Discriminative Training

- make assumptions about the set H of classifiers
- estimate error of classifiers in H from the training data
- select classifier with lowest error rate
- example: SVM, decision tree

True (Prediction) Error

What is a "good" classification rule h?

$$P(h(\vec{x}) \neq y) = \int \Delta(h(\vec{x}) \neq y) dP(\vec{x}, y) = Err_P(h)$$

Loss function Δ:
- 1 if not equal
- 0 if equal

What is the "optimal" Learner L?

Finds the classification rule $h_{opt} \in H$ for which $Err_P(h)$ is minimal:

$$h_{opt} = \arg min_{h \in H} \{ Err_P(h) \}$$

Problem:

$P(\vec{x}, y)$ unknown. Known are training examples $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$.

Principle: Empirical Risk Minimization (ERM)

Learning Principle:

Find the decision rule $h^\circ \in H$ for which the training error is minimal:

$$h^\circ = \arg min_{h \in H}\{Err_S(h)\}$$

Training Error:

$$Err_S(h) = \frac{1}{n}\sum_{i=1}^{n}\Delta(y_i \neq h(\vec{x}_i))$$

==> Number of misclassifications on training examples.

Central Problem of Statistical Learning Theory:

When does a low training error lead to a low generalization error?

When is it Possible to Learn?

Definition [Consistency]: ERM is consistent for

- a hypothesis space *H* and
- independent of the distribution $P(\vec{x}, y)$

if and only if the sequence

$$\lim_{n \to \infty} Err_P(h^\circ) = inf_{h \in H} Err_P(h)$$

$$\lim_{n \to \infty} Err_S(h^\circ) = inf_{h \in H} Err_P(h)$$

converges in probability.

<==> one-sided uniform convergence [Vapnik]

$$\lim_{n \to \infty} P\{sup_{h \in H}(Err_P(h) - Err_S(h)) > \varepsilon\} = 0$$

<==> VC-dimension of *H* is finite [Vapnik].

Vapnik/Chervonenkis Dimension

Definition: The VC-dimension of *H* is equal to the maximal number *d* of examples that can be split into two sets in all 2^d ways using functions from *H* (shattering).

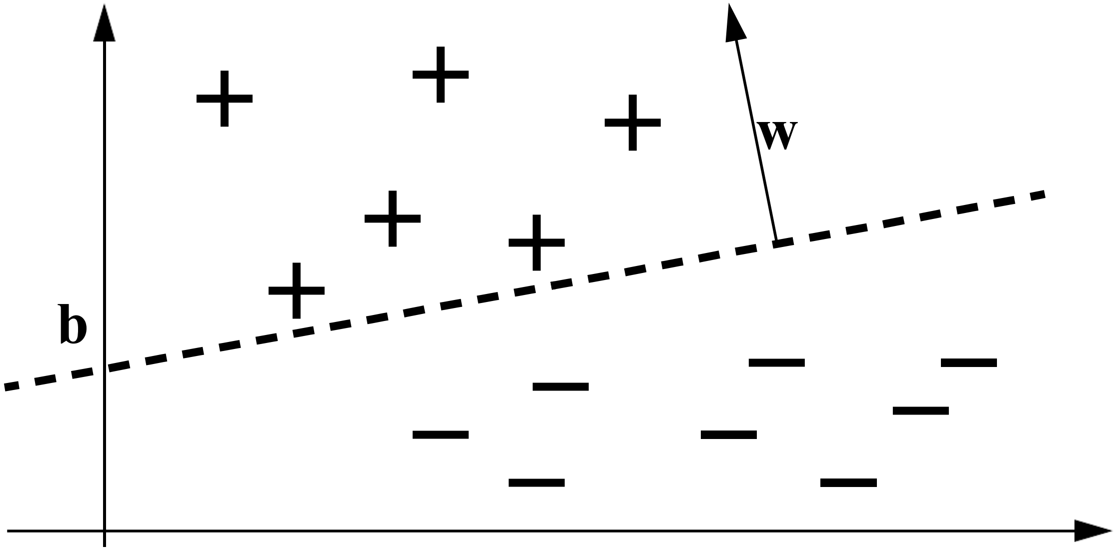
	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	**...**	$\mathbf{x_d}$
h_1	+	+	+	...	+
h_2	-	+	+	...	+
h_3	+	-	+	...	+
h_4	-	-	+	...	+
...	...	...	...	...	...
h_N	-	-	-	...	-

Linear Classifiers

Rules of the Form: weight vector $\vec{w}$, threshold b

$$h(\vec{x}) = sign\left[\sum_{i=1}^{N} \vec{w}_i \vec{x}_i + b\right] = \begin{cases} 1 & if \sum_{i=1}^{N} \vec{w}_i \vec{x}_i + b > 0 \\ -1 & else \end{cases}$$

Geometric Interpretation (Hyperplane):



Linear Classifiers (Example)

Text Classification: Physics (+1) versus Receipes (-1)

ID	nuclear (x_1)	atom (x_2)	salt (x_3)	pepper (x_4)	water (x_5)	heat (x_6)	and (x_7)	y
D1	1	2	0	0	2	0	2	+1
D2	0	0	0	3	0	1	1	-1
D3	0	2	1	0	0	0	3	+1
D4	0	0	1	1	1	1	1	-1

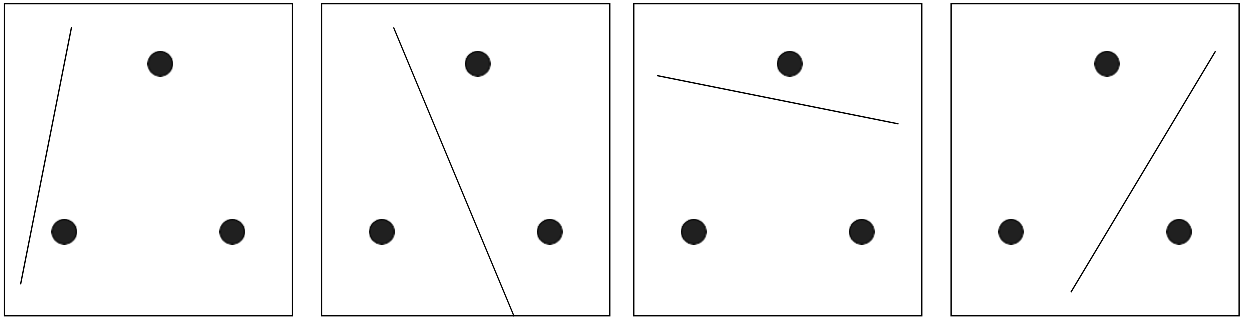
w,b	2	3	-1	-3	-1	-1	0	b=1

D1: $$\sum_{i=1}^{7} \vec{w}_i \vec{x}_i + b = [2 \cdot 1 + 3 \cdot 2 + (-1) \cdot 0 + (-3) \cdot 0 + (-1) \cdot 2 + (-1) \cdot 0 + 0 \cdot 2] + 1$$

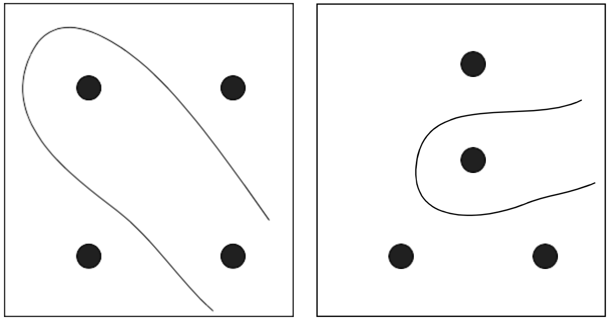
D2: $$\sum_{i=1}^{7} \vec{w}_i \vec{x}_i + b = [2 \cdot 0 + 3 \cdot 0 + (-1) \cdot 0 + (-3) \cdot 3 + (-1) \cdot 0 + (-1) \cdot 1 + 0 \cdot 1] + 1$$

VC-Dimension of Hyperplanes in $\Re^2$

- Three points in $\Re^2$ can be shattered with hyperplanes.



- Four points cannot be shattered.



=> Hyperplanes in $\Re^2$ -> *VCdim=3*

General: Hyperplanes in $\Re^N$ -> *VCdim=N+1*

Rate of Convergence

Question: After n training examples, how close is the training error to the true error?

With probablility η it hold for all $h \in H$:

$$Err_P(h) - Err_S(h) > \Phi(d, n, \eta)$$

$$\Phi(d, n) = \frac{1}{2}\sqrt{4\frac{d\left(\ln\frac{2n}{d} + 1\right) - \ln\frac{\eta}{4}}{n}}$$

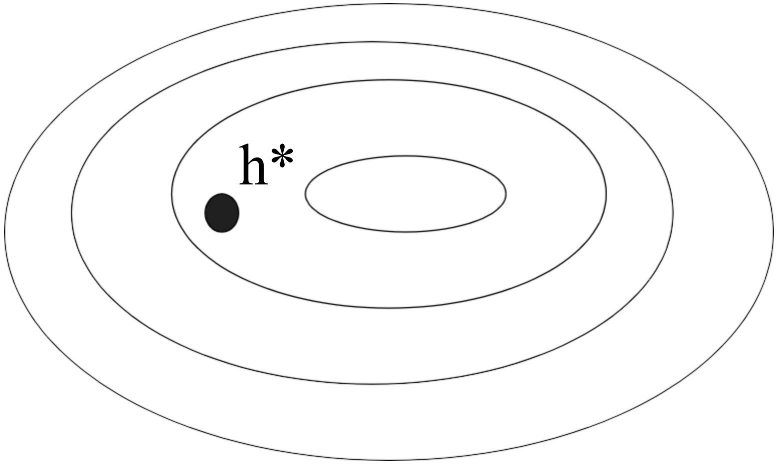
- n number of training examples
- d VC-dimension of hypothesis space H

==> $\boxed{Err_P(h) \leq Err_S(h) + \Phi(d,n, \eta)}$

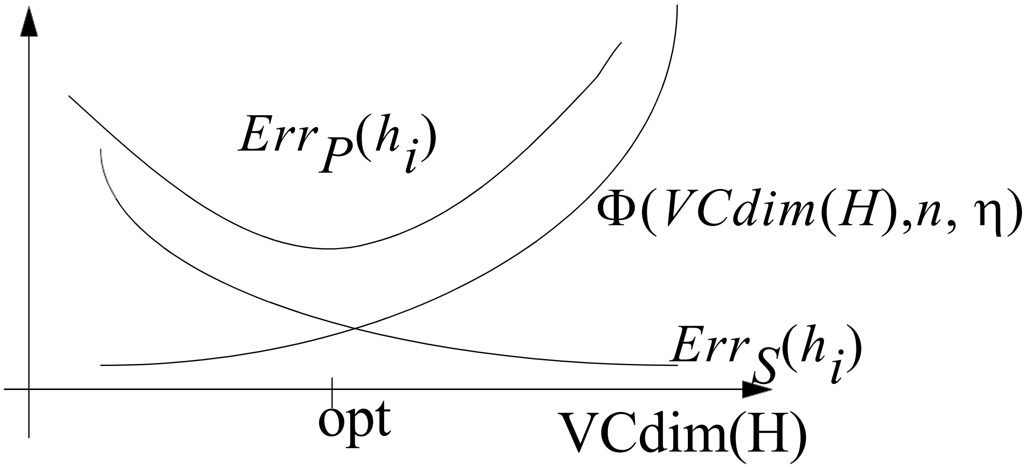
SVM Motivation: Structural Risk Minimization

$$\boxed{Err_P(h_i) \leq Err_S(h_i) + \Phi(VCdim(H), n, \eta)}$$

Idea: Structure on hypothesis space.

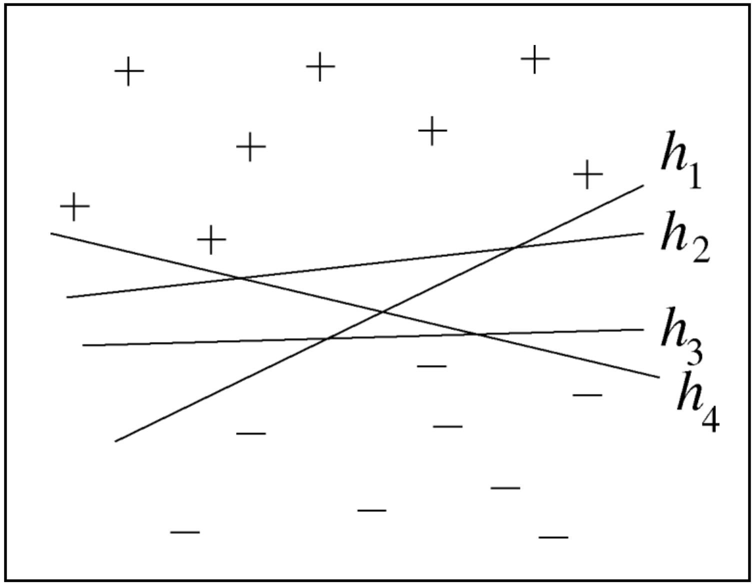


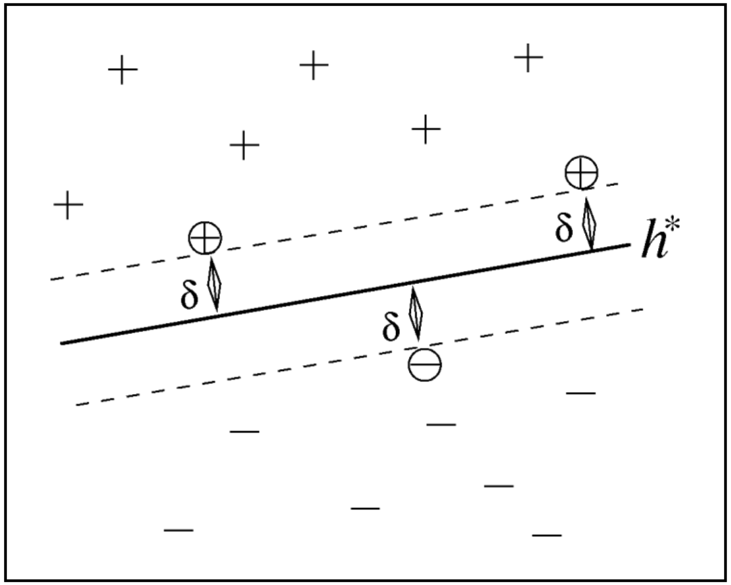
Goal: Minimize upper bound on true error rate.



Optimal Hyperplane (SVM Type 1)

Assumption: The training examples are linearly separable.

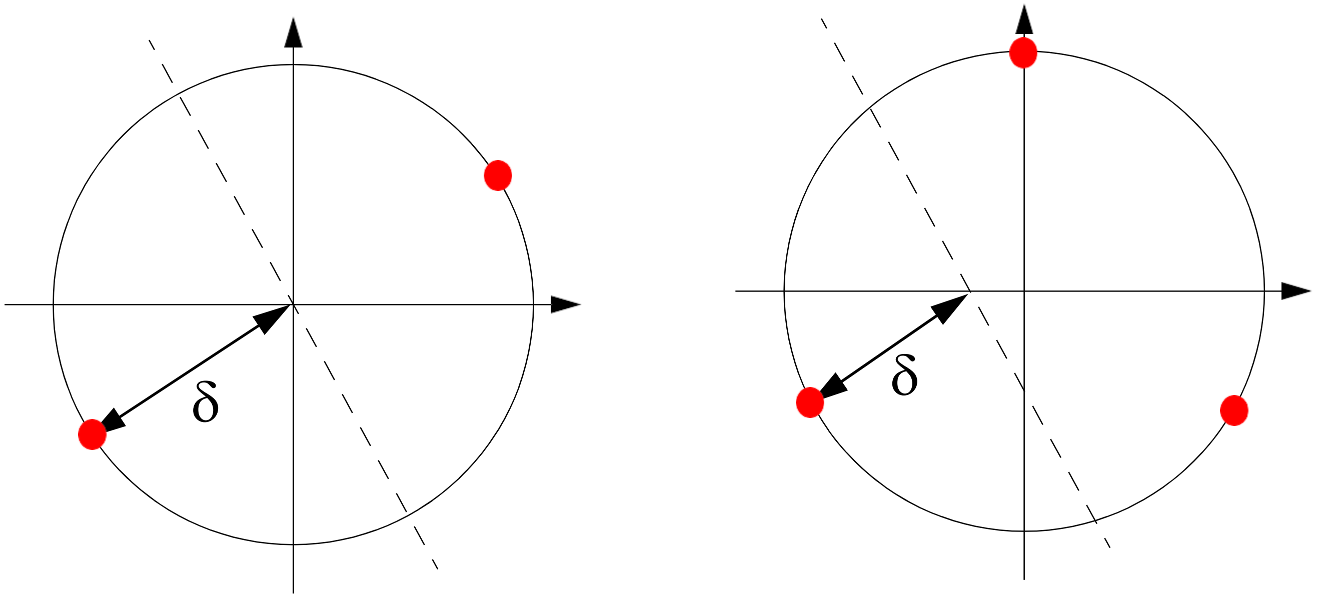




VC-Dimension of “thick” Hyperplanes

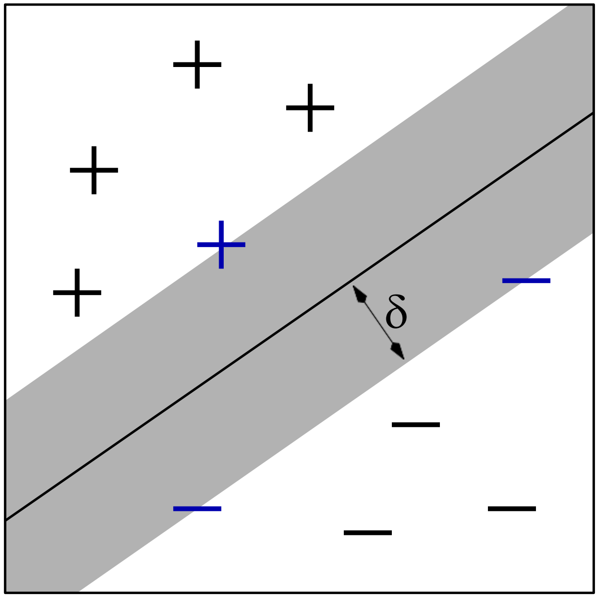
Lemma: The VCdim of hyperplanes $\langle \vec{w}, b \rangle$ with margin δ and description vectors $\|\vec{x}_i\| \leq R$ is bounded by

$$VCdim \leq \frac{R^2}{\delta^2} + 1$$



The VC-dimension does not necessarily depend on the number of attributes or the number of parameters!

Maximizing the Margin



The hyperplane with maximum margin

<~ (roughly, see later) ~>

The hypothesis space with minimal VC-dimension according to SRM

Support Vectors: Examples with minimal distance.

Computing the Optimal Hyperplane

Training Examples: $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$ $\quad \vec{x}_i \in \Re^N \quad y_i \in \{1,-1\}$

Requirement 1: Zero training error!

$$\left.\begin{array}{r} (y = -1) \Rightarrow [\vec{w} \cdot \vec{x}_i + b] < 0 \\ (y = 1) \Rightarrow [\vec{w} \cdot \vec{x}_i + b] > 0 \end{array}\right\} \Leftrightarrow y_i[\vec{w} \cdot \vec{x}_i + b] > 0$$

Requirement 2: Maximum margin!

maximize δ, with $\delta = min_i \left| \frac{1}{\sqrt{\vec{w} \cdot \vec{w}}} [\vec{w} \cdot \vec{x}_i + b] \right|$

Distance δ of point x from hyperplane <w,b>:

$$\delta = \left| \frac{1}{\sqrt{\vec{w} \cdot \vec{w}}} [\vec{w} \cdot \vec{x} + b] \right|$$

=> Requirement 1 & Requirement 2:

maximize δ, with $\forall i \in [1 \ldots n] \left[y_i \left(\frac{1}{\sqrt{\vec{w} \cdot \vec{w}}} [\vec{w} \cdot \vec{x}_i + b] \right) \geq \delta \right]$

Primal Optimization Problem

maximize δ, with $\forall i \in [1 \ldots n]\left[y_i\left(\frac{1}{\sqrt{\vec{w} \cdot \vec{w}}}[\vec{w} \cdot \vec{x}_i + b]\right) \geq \delta\right]$

Set $\frac{1}{\sqrt{\vec{w} \cdot \vec{w}}} = \delta$:

=> maximize $\frac{1}{\sqrt{\vec{w} \cdot \vec{w}}}$, with $\forall i \in [1 \ldots n]\left[y_i\left(\frac{1}{\sqrt{\vec{w} \cdot \vec{w}}}[\vec{w} \cdot \vec{x}_i + b]\right) \geq \frac{1}{\sqrt{\vec{w} \cdot \vec{w}}}\right]$

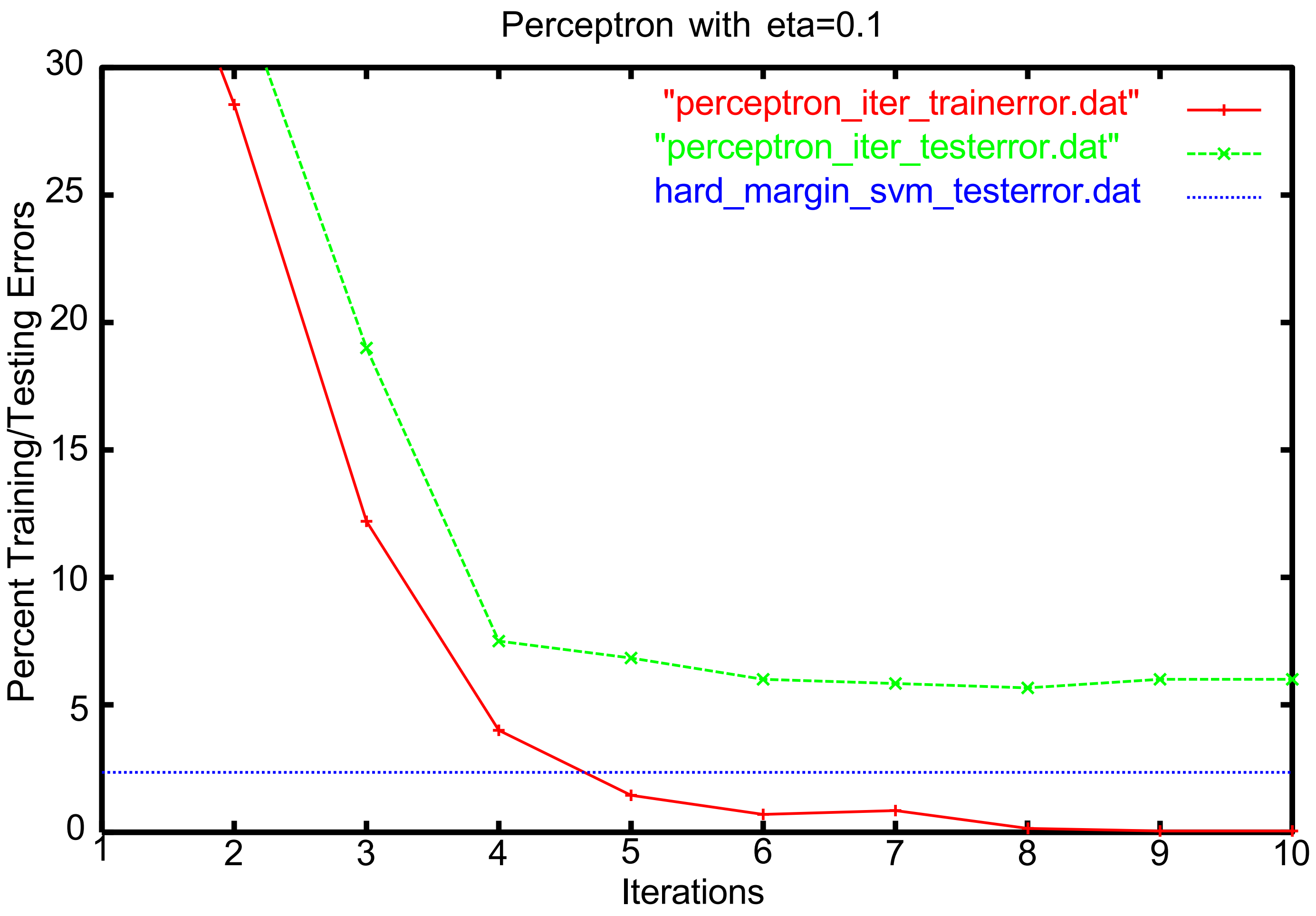
Cancel:

=> maximize $\frac{1}{\sqrt{\vec{w} \cdot \vec{w}}}$, with $\forall i \in [1 \ldots n][y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1]$

Minimize inverse and take square:

=> minimize $P(\vec{w}, b) = \frac{1}{2}\vec{w} \cdot \vec{w}$, with $\begin{bmatrix} y_1[\vec{w} \cdot \vec{x}_1 + b] \geq 1 \\ \ldots \\ y_n[\vec{w} \cdot \vec{x}_n + b] \geq 1 \end{bmatrix}$

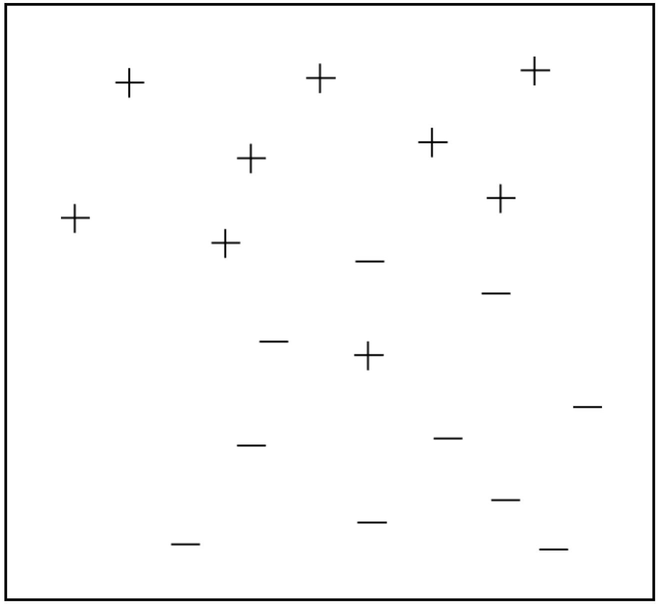
Example: Optimal Hyperplane vs. Perceptron

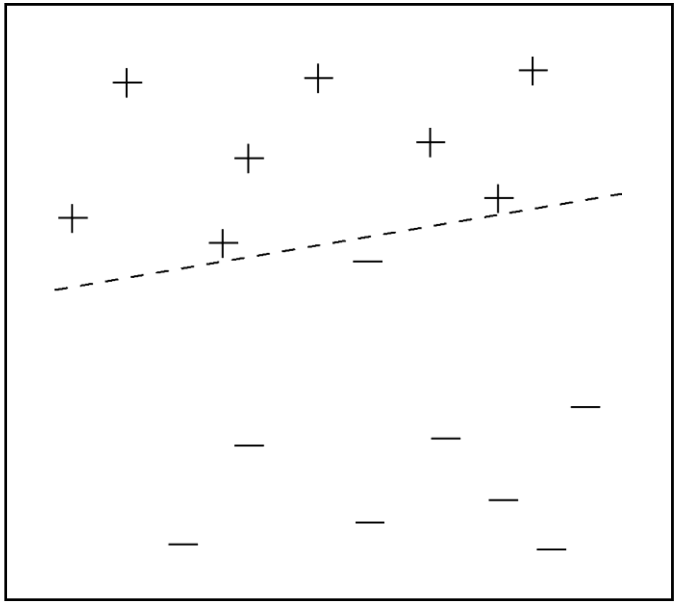


Train on 1000 pos / 1000 neg examples for “acq” (Reuters-21578).

Non-Separable Training Samples

- For some training samples there is no separating hyperplane!
- Complete separation is suboptimal for many training samples!





=> minimize trade-off between margin and training error.

Soft-Margin Separation

Idea: Maximize margin and minimize training error simultanously.

Hard Margin:

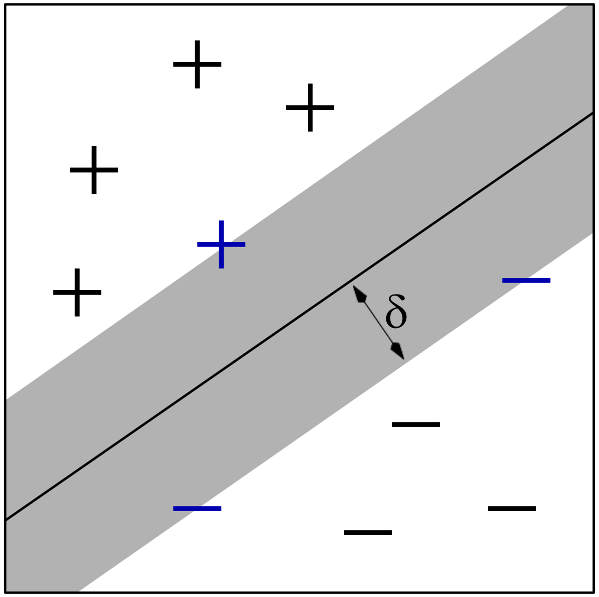
minimize $P(\vec{w}, b) = \frac{1}{2}\vec{w} \cdot \vec{w}$

s. t. $y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1$

Soft Margin:

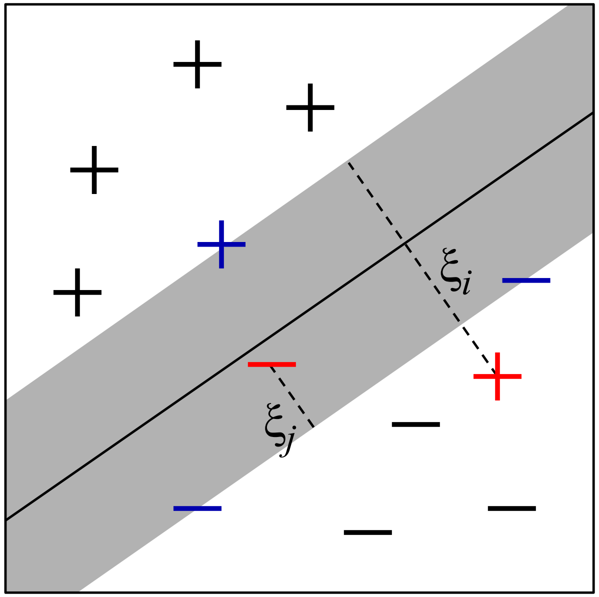
minimize $P(\vec{w}, b, \vec{\xi}) = \frac{1}{2}\vec{w} \cdot \vec{w} + C \sum_{i=1}^{n} \xi_i$

s. t. $y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 - \xi_i$ *and* $\xi_i \geq 0$



Hard Margin
(separable)

Soft Margin
(training error)

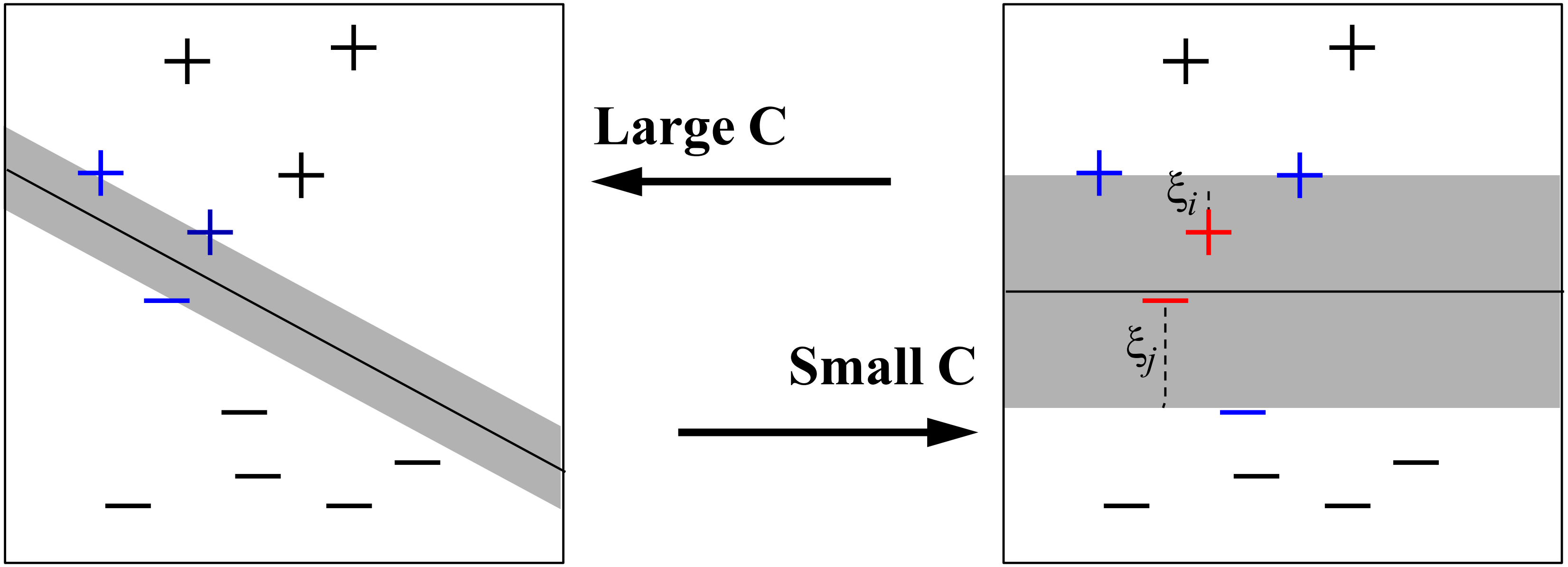


Controlling Soft-Margin Separation

Soft Margin: minimize $P(\vec{w}, b, \vec{\xi}) = \frac{}{2}\vec{w} \cdot \vec{w} + C \sum_{i=1}^{n} \xi_i$

s. t. $y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 - \xi_i$ *and* $\xi_i \geq 0$

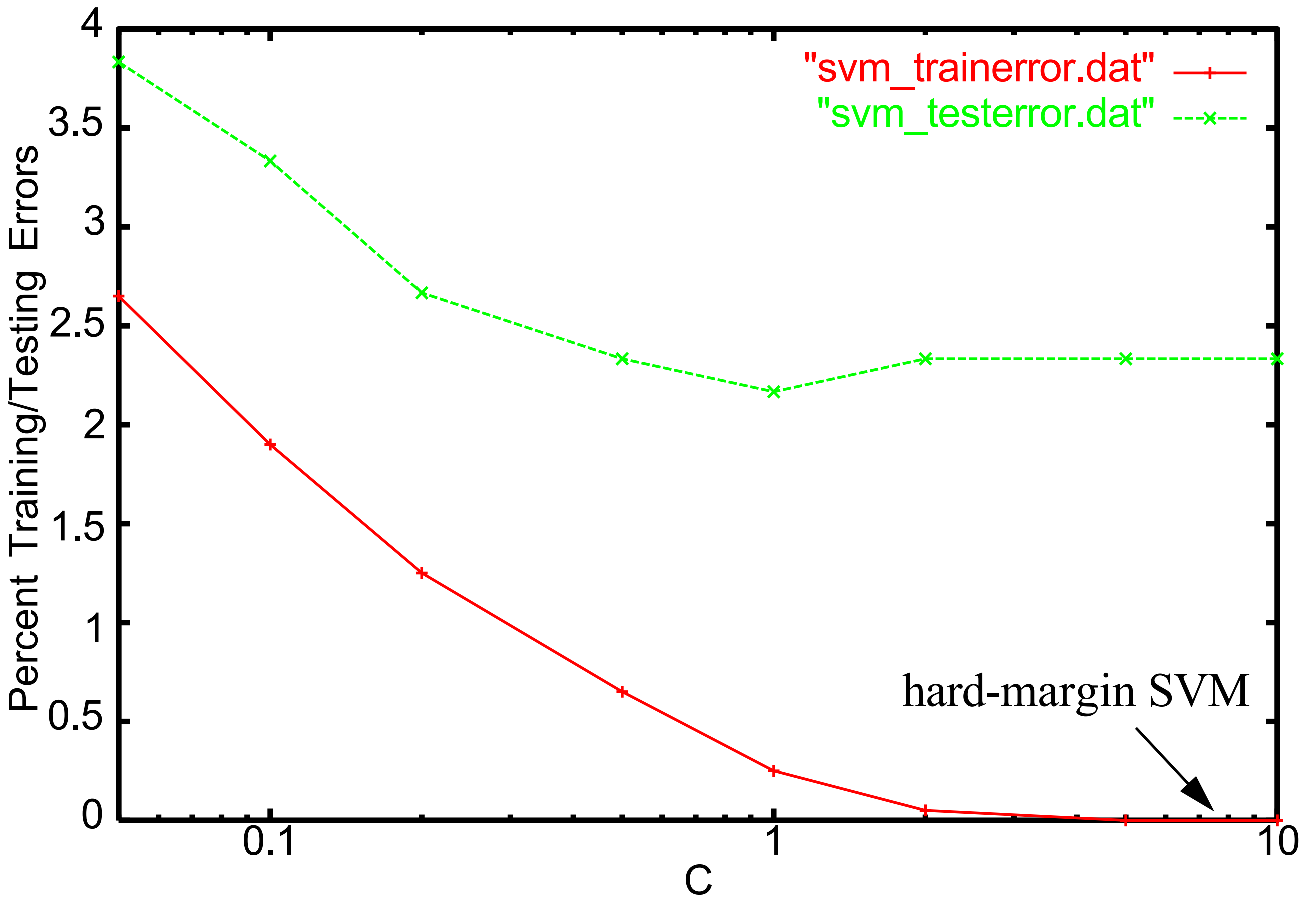
- $\sum \xi_i$ is an upper bound on the number of training errors.
- C is a parameter that controls trade-off between margin and error.



Large C

Small C

Example Reuters "acq": Varying C



Observation: Typically no local optima, but not necessarily...

Part 1 (b): What is an SVM? (non-linear)

- quadratic programs and duality
- properties of the dual
- non-linear classification rules
- kernels and their properties
- kernels for vectorial data
- kernels for non-vectorial data

Quadratic Program

$$\text{minimize} \quad P(\vec{w}) = -\left(\sum_{i=1}^{n} k_i w_i\right) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j H_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i g_i^{(1)} \le 0 \qquad \dots \qquad \sum_{i=1}^{n} w_i g_i^{(k)} \le 0$$

$$\sum_{i=1}^{n} w_i h_i^{(1)} = 0 \qquad \dots \qquad \sum_{i=1}^{n} w_i h_i^{(m)} = 0$$

- k linear inequality constraints
- m linear equality constraints
- Hessian $H = H_{(i,j)}$ is pos. semi-definite $\forall \alpha_1 \dots \alpha_n; \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j H_{ij} \ge 0$
  => convex, no local optima
- $\vec{\alpha}$ is feasible, if it fulfills constraints

Fermat Theorem

Given an unconstrained optimization problem

$$\boxed{\text{minimize } P(\vec{w})}$$

with $P(\vec{w})$ convex and differentiable, a necessary and sufficient conditions for a point $\vec{w}^{\circ}$ to be an optimum is that

$$\frac{\delta P(\vec{w}^{\circ})}{\delta \vec{w}} = 0$$

Lagrange Function

Given an optimization problem

$$\begin{aligned} &\text{minimize } P(\vec{w}) \\ &\textbf{s.t. } \; g_1(\vec{w}) \le 0 \quad \ldots \quad g_k(\vec{w}) \le 0 \\ &\qquad\; h_1(\vec{w}) = 0 \quad \ldots \quad h_m(\vec{w}) = 0 \end{aligned}$$

the Lagrangian function is defined as

$$L(\vec{w}, \vec{\alpha}, \vec{\beta}) = P(\vec{w}) + \sum_{i=1}^{k} \alpha_i g_i(\vec{w}) + \sum_{i=1}^{m} \beta_i h_i(\vec{w})$$

- $\vec{\alpha}$ and $\vec{\beta}$ are called Lagrange Multipliers

Lagrange Theorem

Given an optimization problem

$$\boxed{\begin{array}{l} \text{minimize } P(\vec{w}) \\ \qquad \textbf{s.t.}\ \ h_1(\vec{w}) = 0 \qquad \ldots \qquad h_m(\vec{w}) = 0 \end{array}}$$

with $P(\vec{w})$ convex and differentiable and all h affine (w*x+b), necessary and sufficient conditions for a point $\vec{w}°$ to be an optimum are the existence of $\vec{\beta}°$ such that

$$\frac{\delta L(\vec{w}°, \vec{\beta}°)}{\delta \vec{w}} = 0 \qquad \frac{\delta L(\vec{w}°, \vec{\beta}°)}{\delta \vec{\beta}} = 0 \qquad L(\vec{w}, \vec{\beta}) = P(\vec{w}) + \sum_{i=1}^{m} \beta_i h_i(\vec{w})$$

=> $L(\vec{w}°, \vec{\beta}) \le L(\vec{w}°, \vec{\beta}°) \le L(\vec{w}, \vec{\beta}°)$

Karush-Kuhn-Tucker Theorem

Given an optimization problem

$$\boxed{\begin{array}{l} \text{minimize } P(\vec{w}) \\ \quad \textbf{s.t.} \quad g_1(\vec{w}) \le 0 \quad \dots \quad g_k(\vec{w}) \le 0 \\ \qquad\qquad h_1(\vec{w}) = 0 \quad \dots \quad h_m(\vec{w}) = 0 \end{array}}$$

with $P(\vec{w})$ convex and differentiable and all g and h affine, necessary and sufficient conditions for a point $\vec{w}^\circ$ to be an optimum are the existence of $\vec{\alpha}^\circ$ and $\vec{\beta}^\circ$ such that

$$\frac{\delta L(\vec{w}^\circ, \vec{\alpha}^\circ, \vec{\beta}^\circ)}{\delta \vec{w}} = 0 \qquad \frac{\delta L(\vec{w}^\circ, \vec{\alpha}^\circ, \vec{\beta}^\circ)}{\delta \vec{\beta}} = 0$$

$$\alpha_i^\circ g_i(\vec{w}^\circ) = 0, i = 1, \dots, k$$

$$g_i(\vec{w}^\circ) \le 0, i = 1, \dots, k$$

$$\alpha_i^\circ \ge 0, i = 1, \dots, k$$

Sufficient for convex QP: $max_{\vec{\beta}, \vec{\alpha} \ge 0}[min_{\vec{w}} L(\vec{w}, \vec{\alpha}, \vec{\beta})]$

Dual Optimization Problem

Primal OP: minimize $P(\vec{w}, b, \vec{\xi}) = \frac{1}{2}\vec{w} \cdot \vec{w} + C \sum_{i=1}^{n} \xi_i$

s. t. $y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 - \xi_i$ *and* $\xi_i \geq 0$

Lemma: The solution $w°$ can always be written as a linear combination

$$\vec{w}° = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i \qquad \alpha_i \geq 0$$

of the training data.

Dual OP: maximize $D(\vec{\alpha}) = \left(\sum_{i=1}^{n} \alpha_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$

s.t. $\sum_{i=1}^{n} \alpha_i y_i = 0$ *and* $0 \leq \alpha_i \leq C$

==> positive semi-definite quadratic program

Primal <=> Dual

Theorem: The primal OP and the dual OP have the same solution. Given the solution α_i° of the dual OP,

$$\vec{w}^\circ = \sum_{i=1}^{n} \alpha_i^\circ y_i \vec{x}_i \qquad b^\circ = \frac{1}{2}(\vec{w}_0 \cdot \vec{x}^{pos} + \vec{w}_0 \cdot \vec{x}^{neg})$$

is the solution of the primal OP.

Theorem: For any set of feasible points $P(\vec{w}, b) \geq D(\vec{\alpha})$.

=> two alternative ways to represent the learning result

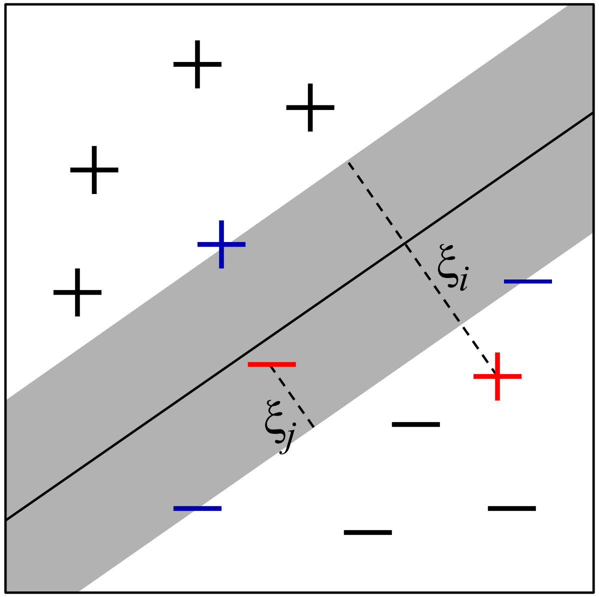
- weight vector and threshold $\langle \vec{w}, b \rangle$
- vector of “influences” $\alpha_1, \ldots, \alpha_n$

Properties of the Soft-Margin Dual OP

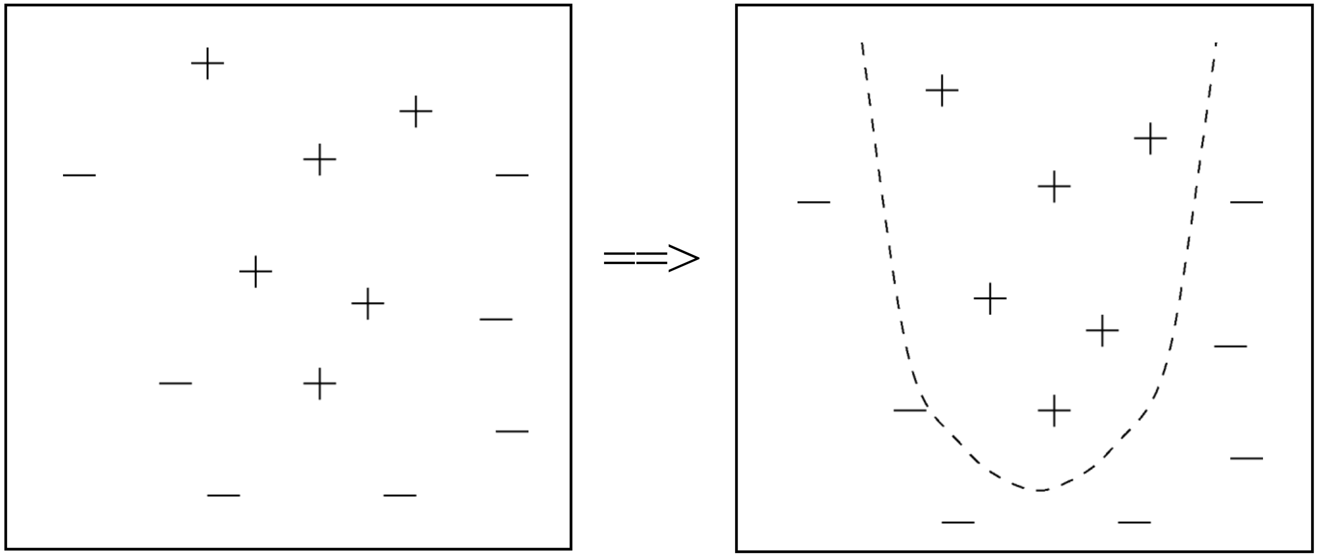
Dual OP: maximize $$D(\vec{\alpha}) = \left(\sum_{i=1}^{n} \alpha_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

s. t. $$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad und \qquad 0 \le \alpha_i \le C$$

- typically single solution (i. e. $\langle \vec{w}, b \rangle$ is unique)
- one factor α_i for each training example
  - “influence” of single training example limited by C
  - $0 < \alpha_i < C$ <=> SV with $\xi_i = 0$
  - $\alpha_i = C$ <=> SV with $\xi_i > 0$
  - $\alpha_i = 0$ else
- based exclusively on inner product between training examples



Non-Linear Problems



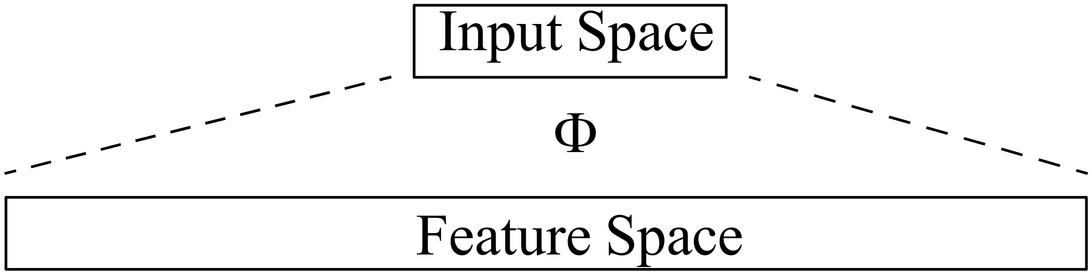
Problem:

- some tasks have non-linear structure
- no hyperplane is sufficiently accurate

How can SVMs learn non-linear classification rules?

Extending the Hypothesis Space

Idea:



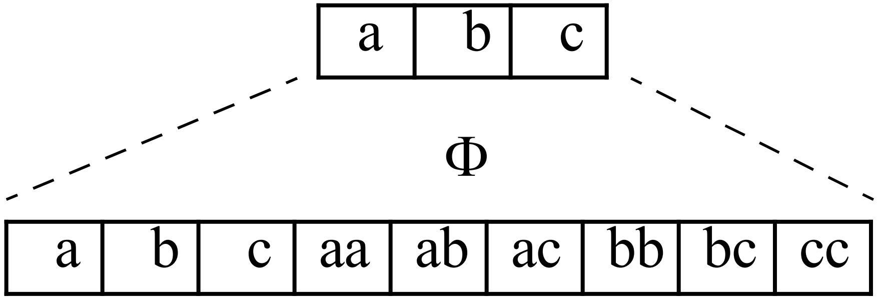
Input Space

Φ

Feature Space

==> Find hyperplane in feature space!

Example:



a	b	c

Φ

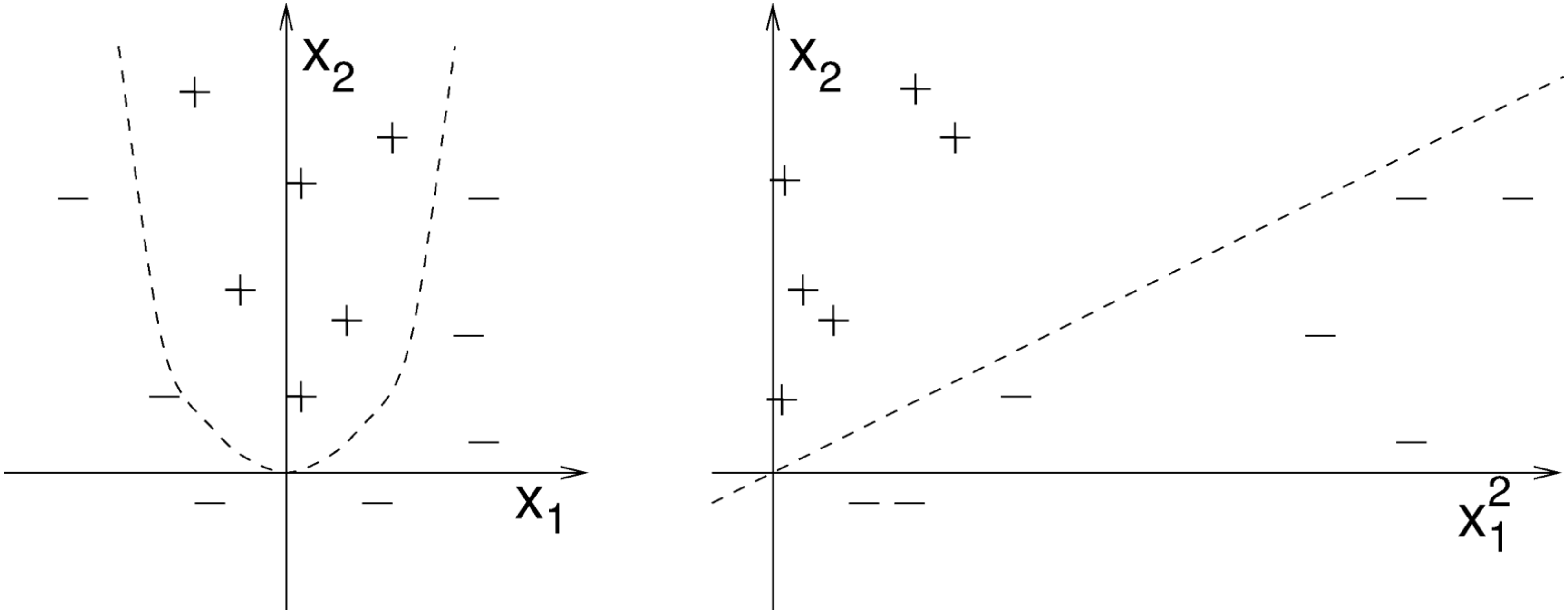
a	b	c	aa	ab	ac	bb	bc	cc

==> The separating hyperplane in features space is a degree two polynomial in input space.

Example

Input Space: $\vec{x} = (x_1, x_2)$ (2 Attributes)

Feature Space: $\Phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$ (6 Attributes)



Kernels

Problem: Very many Parameters! Polynomials of degree *p* over *N* attributes in input space lead to $O(N^p)$ attributes in feature space!

Solution: [Boser et al., 1992] The dual OP need only inner products => Kernel Functions

$$\boxed{K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)}$$

Example: For $\Phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$ calculating

$$K(\vec{x}_i, \vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^2 = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$$

gives inner product in feature space.

We do not need to represent the feature space explicitly!

SVM with Kernels

Training: maximize $D(\vec{\alpha}) = \left(\sum_{i=1}^{n} \alpha_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j)$

s. t. $\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad und \qquad 0 \le \alpha_i \le C$

Classification: For new example x $\quad h(\vec{x}) = sign\left(\sum_{x_i \in SV} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b\right)$

New hypotheses spaces through new Kernels:

Linear: $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$

Polynomial: $K(\vec{x}_i, \vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^d$

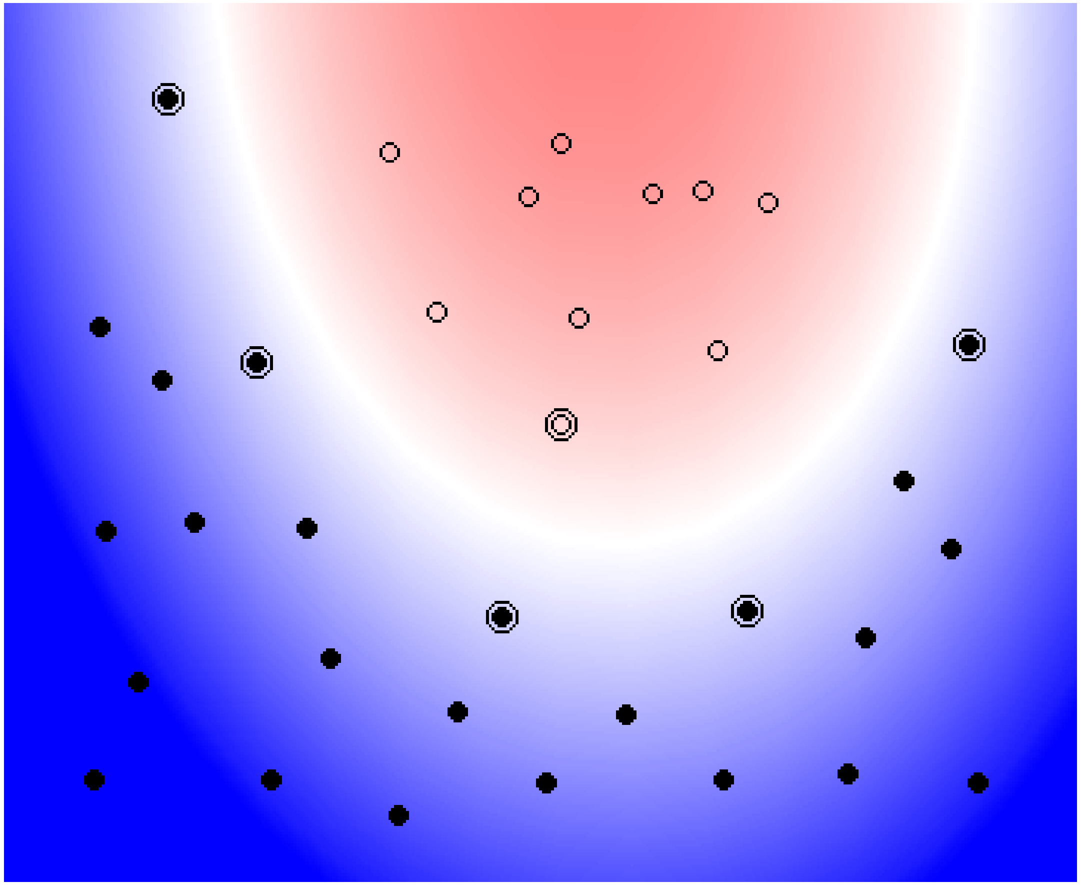
Radial Basis Functions: $K(\vec{x}_i, \vec{x}_j) = \exp(-|\vec{x}_i - \vec{x}_j|^2 / \sigma^2)$

Sigmoid: $K(\vec{x}_i, \vec{x}_j) = \tanh(\gamma(\vec{x}_i - \vec{x}_j) + c)$

Example: SVM with Polynomial of Degree 2

Kernel: $K(\vec{x}_i, \vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^2$

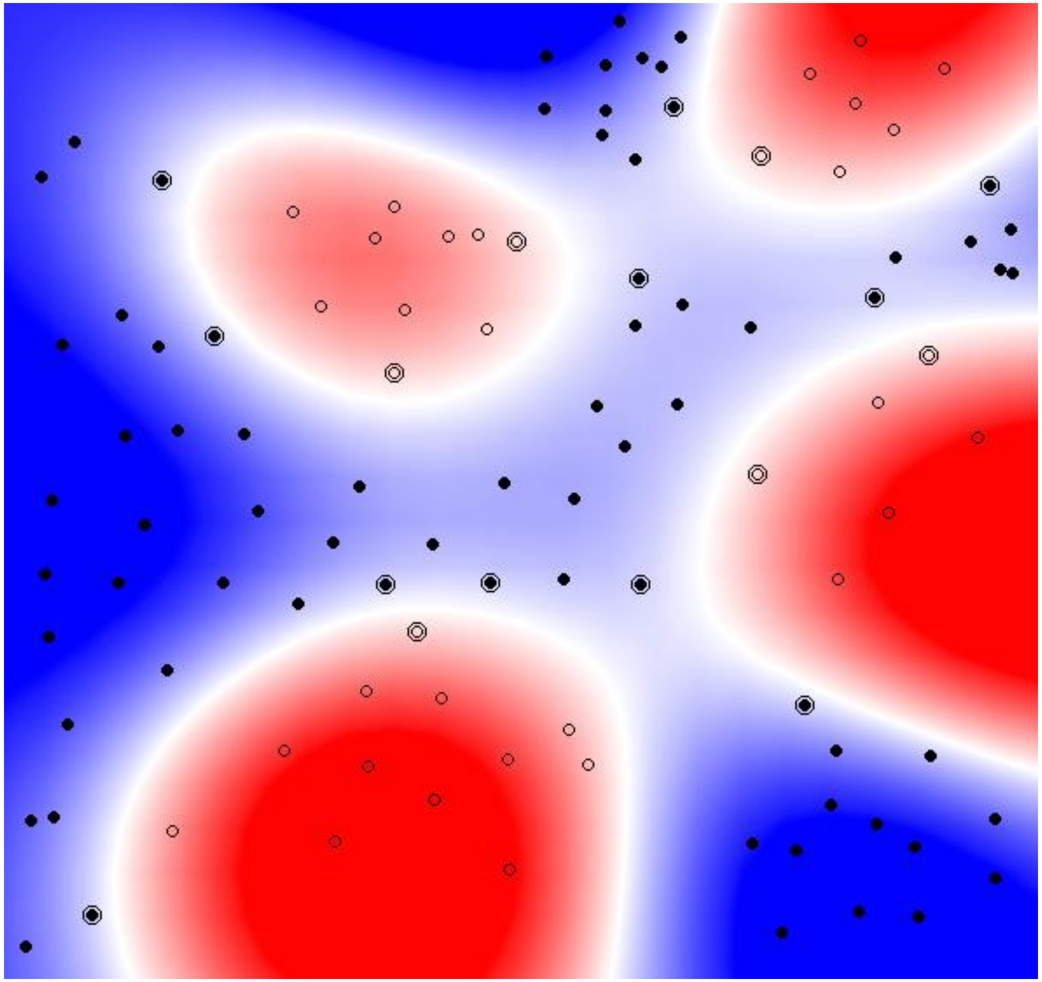
plot by Bell SVM applet



Example: SVM with RBF-Kernel

Kernel: $K(\vec{x}_i, \vec{x}_j) = \exp(-|\vec{x}_i - \vec{x}_j|^2 / \sigma^2)$

plot by Bell SVM applet



What is a Valid Kernel?

Mercer's Theorem (see [Cristianini & Shawe-Taylor, 2000])

Theorem [Saitoh]: Let X be a finite input space of n points $(\vec{x}_1, \ldots, \vec{x}_n)$. A function $K(\vec{x}_i, \vec{x}_j)$ is a valid kernel in X iff it produces a Gram matrix

$$G_{ij} = K(\vec{x}_i, \vec{x}_j)$$

that is symmetric

$$G = G^T$$

and positive semi-definite

$$\forall \vec{\alpha} \left(\vec{\alpha}^T G \vec{\alpha} = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j) \geq 0 \right)$$

How to Construct Valid Kernels?

Theorem: Let K_1 and K_2 be valid Kernels over $X \times X$, $X \subseteq \Re^N$, $a \geq 0$, $0 \leq \lambda \leq 1$, f a real-valued function on X, $\phi;X \to \Re^m$ with K_3 a kernel over $\Re^m \times \Re^m$, and K a summetric positive semi-definite matrix. Then the following functions are valid Kernels

$$K(\vec{x}, \vec{z}) = \lambda K_1(\vec{x}, \vec{z}) + (1 - \lambda) K_2(\vec{x}, \vec{z})$$

$$K(\vec{x}, \vec{z}) = a K_1(\vec{x}, \vec{z})$$

$$K(\vec{x}, \vec{z}) = K_1(\vec{x}, \vec{z}) K_2(\vec{x}, \vec{z})$$

$$K(\vec{x}, \vec{z}) = f(\vec{x}) f(\vec{z})$$

$$K(\vec{x}, \vec{z}) = K_3(\phi(\vec{x}), \phi(\vec{z}))$$

$$K(\vec{x}, \vec{z}) = \vec{x}^T K \vec{z}$$

=> Construct complex Kernels from simple Kernels.

Other Kernels for Complex Data

General information on Kernels:

- Introduction to Kernels [Cristianini & Shawe-Taylor, 2000]
- All the details on Kernels + Background [Schoelkopf & Smola, 2002]

Kernels for specific structures:

- Diffusion Kernels for graphs [Kondor & Lafferty, 2002]
- Kernels for grammars [Collins & Duffy, 2002]
- Kernels for trees, lists, etc. [Gaertner et al., 2002]

Two Reasons for Using a Kernel

(1) Turn a linear learner into a non-linear learner

(e.g. RBF, polynomial, sigmoid)

(2) Make non-vectorial data accessible to learner

(e.g. string kernels for sequences)

Summary
What is an SVM?

Given:

- Training examples $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \quad \vec{x}_i \in \Re^N \quad y_i \in \{1,-1\}$
- Hypothesis space according to kernel $K(\vec{x}_i, \vec{x}_j)$
- Parameter C for trading-off training error and margin size

Training:

- Finds hyperplane in feature space generated by kernel.
- The hyperplane has maximum margin in feature space with minimal training error (upper bound $\sum \xi_i$) given C.
- The result of training are $\alpha_1, \ldots, \alpha_n$. They determine $\langle \vec{w}, b \rangle$.

Classification: For new example $h(\vec{x}) = sign\left(\sum_{x_i \in SV} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b \right)$

Part 2: How to use an SVM effectively and efficiently?

- normalization of the input vectors
- selecting C
- handling unbalanced datasets
- selecting a kernel
- multi-class and multi-label classification
- selecting a training algorithm

Design Decisions in Working with SVMs

Setting up the learning task

- multi-class problems
- multi-label problems

Representation of the data (efficiency and effectiveness)

- selecting features
- selecting feature values
- normalizing the data (directional vs. non-directional data)
- selecting a kernel

Selecting a good value for the parameter C and kernel parameters

Selecting a training algorithm that is efficient for the particular QP

- kernel SVM vs. linear SVM
- many sparse features vs. few dense features

Handling Multi-Class / Multi-Label Problems

Standard classification SVM addresses binary problems $y \in \{1, -1\}$

Multi-class classification: $y \in \{1, \ldots, k\}$

- one-against-rest decomposition into k binary problems
  - learn one binary SVM $h^{(i)}$ per class with $y^{(i)} = \begin{cases} 1 & if(y = i) \\ -1 & else \end{cases}$
  - assign new example to $y = \arg max[h^{(i)}(\vec{x})]$
- pairwise decomposition into $k(k-1)$ binary problems
  - learn one binary SVM $h^{(i)}$ per class pair $y^{(i,j)} = \begin{cases} 1 & if(y = i) \\ -1 & if(y = j) \end{cases}$
  - assign new example by majority vote
  - reducing number of classifications [Platt et al., 2000]
- multi-class SVM [Weston & Watkins, 1998]
- multi-class SVM via ranking [Crammer & Singer, 2001]

Multi-label classification: $y \subseteq \{1, \ldots, k\}$

- learn one binary SVM $h^{(i)}$ per class with $y^{(i)} = \begin{cases} 1 & if(i \in y) \\ -1 & else \end{cases}$

Which Features to Choose?

Things to take into consideration:

- if features sparse, then dimensionality of space no efficiency problem
  - computations based on inner product between vectors
- consider frequency distribution of features (e.g. many rare features)
  - Zipf distribution of words
  - see TCat-model
- SVMs can handle redundancy in features
  - bag-of-words representation redundant for topic classification
  - see TCat-model
- as few irrelevant features as possible
  - stopword removal often helps in text classification
  - see TCat-model

How to Assign Feature Values?

Things to take into consideration:

- importance of feature is monotonic in its absolute value
    - the larger the absolute value, the more influence the feature gets
    - typical problem: number of doors [0-5], price [0-100000]
    - want relevant features large / irrelevant features low (e.g. IDF)
- normalization to make features equally important
    - by mean and variance: $x_{norm} = \frac{x - mean(X)}{\sqrt{var(X)}}$
    - by other distribution
- normalization to bring feature vectors onto the same scale
    - directional data: text classification
    - by normalizing the length of the vector $\vec{x}_{norm} = \frac{\vec{x}}{\|\vec{x}\|}$ according to some norm
    - changes whether a problem is (linearly) separable or not
- scale all vectors to a length that allows numerically stable training

Selecting a Kernel

Things to take into consideration:

- kernel can be thought of as a similarity measure
  - examples in the same class should have high kernel value
  - examples in different classes should have low kernel value
  - ideal kernel: equivalence relation $K(\vec{x}_i, \vec{x}_j) = sign(y_i y_j)$
- normalization also applies to kernel
  - relative weight for implicit features
  - normalize per example for directional data

$$K(\vec{x}_i, \vec{x}_j) = \frac{K(\vec{x}_i, \vec{x}_j)}{\sqrt{K(\vec{x}_i, \vec{x}_i)}\sqrt{K(\vec{x}_j, \vec{x}_j)}}$$

  - potential problems with large numbers, for example polynomial kernel $K(\vec{x}_i, \vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^d$ for large d

Selecting Regularization Parameter C

Common Method

- a reasonable starting point and/or default value is $C_{def} = \dfrac{1}{\sum K(\vec{x}_i, \vec{x}_i)}$
- search for C on a log-scale, for example

$$C \in [10^{-4} C_{def}, \ldots, 10^{4} C_{def}]$$

- selection via cross-validation or via approximation of leave-one-out [Jaakkola&Haussler,1999][Vapnik&Chapelle,2000][Joachims,2000]

Note

- optimal value of C scales with the feature values
- implicit slack variables via infrequent features
  - if every example has one unique feature x_i, then always separable
  - unique features x_i act like squared slack variable

$$\boxed{\text{minimize } P(\vec{w}, b, \vec{\xi}) = \frac{1}{2}\vec{w} \cdot \vec{w} + \frac{1}{2}\sum_{i=1} w_i^2 \text{ s. t. } y_i[\vec{w} \cdot \vec{x}_i + b] \geq 1 - w_i x_i}$$

Selecting Kernel Parameters

Problem

- results often very sensitive to kernel parameters (e.g. variance γ in RBF kernel)
- need to simultaneously optimize C, since optimal C typically depends on kernel parameters

Common Method

- search for combination of parameters via exhaustive search
- selection of kernel parameters typically via cross-validation

Advanced Approach

- avoiding exhaustive search for improved search efficiency [Chapelle et al, 2002]

Handling Unbalanced Datasets

Problem

- often the number of positive examples is much lower than the number of negative examples
- SVM minimizes error rate
  => always say "no" gives great error rate, but poor recall

Common Methods

- cost model that makes errors on positive examples more expensive

$$\boxed{\min\ P(\vec{w},b,\vec{\xi}) = \frac{1}{2}\vec{w}\cdot\vec{w} + JC\sum_{y_i=1}\xi_i + C\sum_{y_i=-1}\xi_i \quad \text{s.t.} \quad y_i[\vec{w}\cdot\vec{x}_i + b] \geq 1 - \xi_i \ \ and\ \ \xi_i \geq 0}$$

- change threshold b after training to some higher value b'

$$h(\vec{x}) = sign\left(\sum_{x_i \in SV} \alpha_i y_i K(\vec{x}_i,\vec{x}) + b'\right)$$

Selecting an SVM Training Algorithm

SVM*light* (also SVMtorch, mySVM, BSVM, etc.) [Joachims, 1999b]

- solve dual QP to obtain hyperplane from α-coefficients
- iteratively decompose large QP into a sequence of small QPs
- handles kernels and treats linear SVM as special case

SMO [Platt, 1999]

- special case of working sets of size two
- simple analytical solution of QP subproblems

ASVM [Mangasarian & Musicant, 2000]

- restricted to linear SVMs with quadratic loss
- fast for low dimensional data

Nearest Point Algorithm [Keerthi et al., 1999]

- restricted to quadratic loss
- compute distance between convex hulls

Part 3: How to Train SVMs?

- efficiency of primal vs. dual
- decomposition algorithm
- working set selection
- optimality criteria
- caching
- shrinking

How can One Train SVMs Efficiently?

Solve one of the following quadratic optimization problems:

$$\min\ P(\vec{w}, b, \vec{\xi}) = \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i=1}^{n}\xi_i$$

$$\text{s. t.}\quad y_i[\vec{w}\cdot\vec{x}_i + b] \ge 1 - \xi_i \ \mathit{and}\ \xi_i \ge 0$$

- $n+N+1$ variables
- n linear inequality constraints
- no direct use of kernels
- size scales $O(nN)$

<= DUAL =>

$$\max\ D(\vec{\alpha}) = \left(\sum_{i=1}^{n}\alpha_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j)$$

$$\text{s. t.}\quad \sum_{i=1}^{n}\alpha_i y_i = 0 \qquad \mathit{and} \qquad 0 \le \alpha_i \le C$$

- n variables
- 1 linear equality, $2n$ box constraints
- use of kernels natural
- size scales $O(n^2)$

=> positive semi-definite quadratic program with n variables

Decomposition

Idea: Solve small subproblems until convergence (Osuna, et al.)!

$$max \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^T \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{bmatrix}^T \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} & k_{37} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} & k_{47} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} & k_{57} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} & k_{67} \\ k_{71} & k_{73} & k_{73} & k_{74} & k_{75} & k_{76} & k_{77} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{bmatrix}$$

Decomposition

Idea: Solve small subproblems until convergence (Osuna, et al.)!

$$max \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\end{bmatrix}^T \begin{bmatrix}\alpha_1\\\alpha_2\\\alpha_3\\\alpha_4\\\alpha_5\\\alpha_6\\\alpha_7\end{bmatrix} - \frac{1}{2} \begin{bmatrix}\alpha_1\\\alpha_2\\\alpha_3\\\alpha_4\\\alpha_5\\\alpha_6\\\alpha_7\end{bmatrix}^T \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17}\\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27}\\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} & k_{37}\\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} & k_{47}\\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} & k_{57}\\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} & k_{67}\\ k_{71} & k_{73} & k_{73} & k_{74} & k_{75} & k_{76} & k_{77} \end{bmatrix} \begin{bmatrix}\alpha_1\\\alpha_2\\\alpha_3\\\alpha_4\\\alpha_5\\\alpha_6\\\alpha_7\end{bmatrix}$$

Time complexity: working set of size $2 \le q \le 100$ and f nonzero features:

- extracting subproblem: $O(q^2 f)$
- solving subproblem: $O(q^3)$
- updating large problem with result of subproblem: $O(nqf)$

What Working Set to Select Next?

Solution: Select subproblem with *q* variables that minimizes

$$V(\vec{d}) = g(\vec{\alpha})^T \vec{d}$$

$$\textit{subject to} \quad \begin{array}{c} \vec{y}^T \vec{d} = 0 \\ d_i \geq 0, if(\alpha_i = 0) \\ d_i \leq 0, if(\alpha_i = C) \\ -1 \leq \vec{d} \leq 1 \\ \left|\{d_i \neq 0\}\right| = q \end{array}$$

Efficiency: Selection linear in number of examples.

Convergence: Proofs by Chi-Chen Lin / Keerthi under mild assumptions.

How to Tell that we Found the Optimal Solution?

Karush-Kuhn-Tucker conditions lead to the following criterion:

$$\boxed{\begin{array}{l} \text{maximize} \quad D(\vec{\alpha}) = \left(\sum_{i=1}^{n} \alpha_i\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j) \\ \text{s. t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \mathit{and} \qquad 0 \le \alpha_i \le C \end{array}} \quad \text{is optimal}$$

<=>

$$\forall i \left\{ \begin{array}{l} (\alpha_i = 0) \Rightarrow y_i \left[\sum_{j=1}^{n} \alpha_j y_j K(\vec{x}_i, \vec{x}_j) + b \right] \ge 1 \\ (0 < \alpha_i < C) \Rightarrow y_i \left[\sum_{j=1}^{n} \alpha_j y_j K(\vec{x}_i, \vec{x}_j) + b \right] = 1 \\ (\alpha_i = C) \Rightarrow y_i \left[\sum_{j=1}^{n} \alpha_j y_j K(\vec{x}_i, \vec{x}_j) + b \right] \le 1 \end{array} \right.$$

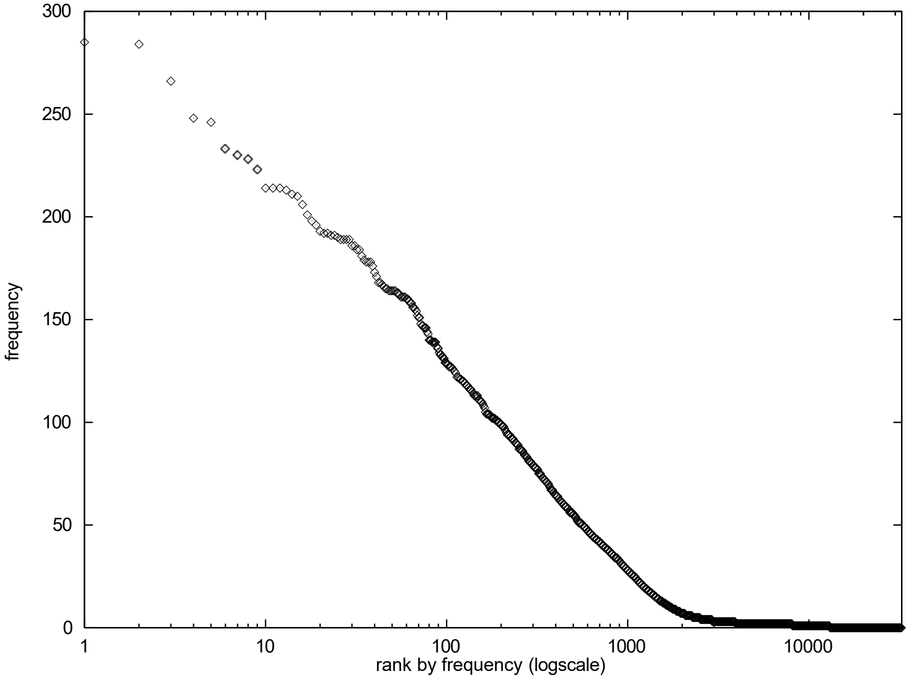
Demo

The Steps of Solving a 2-d Problem.

Caching

Observation: Most CPU-time is spent on computing the Hessian!

Idea: Cache kernel evaluations.



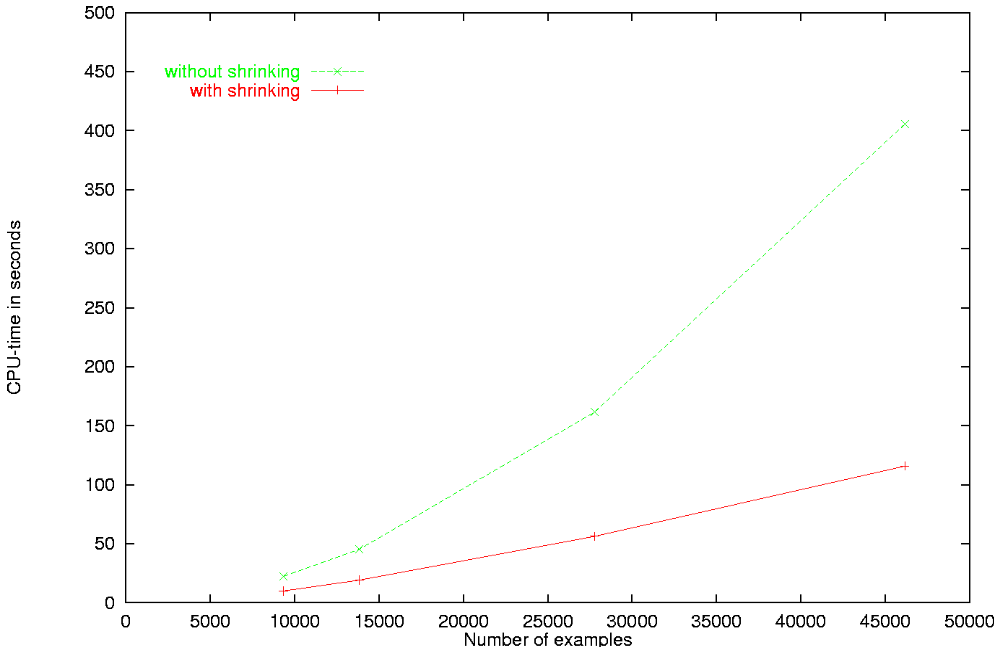
Result: A small cache leads to a large improvement.

Shrinking

Idea: If we knew the set of SVs, we could solve a smaller problem!
(complexity per iteration from $O(nqf)$ to $O(sqf)$)

Algorithm:

- monitor the KKT-conditions in each iteration
- if a variable is "stuck at bound", remove it
- do final optimality check



Summary
How can One Train SVMs Efficiently?

SVM*light* (also SVMtorch, mySVM, BSVM, etc.)

- solve dual QP to obtain hyperplane from α-coefficients
- iteratively decompose large QP into a sequence of small QPs
- select working set according to steepest feasible descent criterion
- check optimality using Karush-Kuhn-Tucker conditions

Other training algorithms:

- SMO requires working set of size two => simple analytical solution of QP subproblems [Platt, 1999]
- ASVM restricted to linear SVMs with quadratic loss => fast for low dimensional data [Mangasarian & Musicant, 2000]
- Nearest Point Algorithm restricted to quadratic loss => compute distance between convex hulls [Keerthi et al., 1999]

Part 3: Why do SVMs Work?

- worst-case bounds
- bounds on the expected generalization error
- leave-one-out estimation
- necessary criteria for leave-one-out

...classifies as well as possible!?

What is a "good" classification rule h?

$$P(h(x) \neq y) = \int \Delta(h(x) \neq y) dP(x, y) = Err_P(h)$$

What is a "good" learner L?

"Worst-Case" Learner:

$$P(Err_P(h_L) > \varepsilon) < \eta$$

"Average-Case" Learner:

$$E(Err_P(h_L)) = \int Err_P(h_L) dP(x_1, y_1) \dots P(x_n, y_n)$$

SVMs as Worst-Case Learner

Goal: Guarantee of the form

$$P(Err_P(h_L) > \varepsilon) < \eta$$

Theorem: $P\left(Err_P(h) - Err_S(h) \geq \Phi\left(\frac{R^2}{\delta^2}, n, \eta\right)\right) < \eta$ [Shawe-Taylor et al,1996]

So, if

- the training error $Err_S(h)$ on sample S is low and
- the margin δ is large,

then with probablility η the SVM will output a classification rule with true error

$$Err_P(h_i) \leq Err_S(h_i) + \Phi\left(\frac{R^2}{\delta^2}, n, \eta\right) \quad .$$

Problem: For most practical problems this bound is vacuous, i.e. $Err_P(h_i) \leq 1$.

SVMs as Average-Case Learner

Theorem: The expected error of an SVM is bounded by

$$E(Err_P(h_{SVM})) \leq \frac{2E\left(\frac{R^2}{\delta^2}\right) + 2CR^2E\left(\sum_{n=1}^{n} \xi_i\right)}{n} \qquad C \geq \frac{1}{2R^2}$$

$$E(Err_P(h_{SVM})) \leq \frac{2E\left(\frac{R^2}{\delta^2}\right) + 2(CR^2+1)E\left(\sum_{n=1}^{n} \xi_i\right)}{n} \qquad C < \frac{1}{2R^2}$$

with $E\left(\frac{R^2}{\delta^2}\right)$ the expected soft margin and $E\left(\sum_{n=1}^{n} \xi_i\right)$ the expected training error bound [Joachims, 2001] [Vapnik, 1998].

Problem: The expectations are unknown.

Leave-One-Out

Training set: $(\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_3, y_3), \ldots, (\vec{x}_n, y_n)$

Approach: Repeatedly leave one example out for testing.

train on	test on
$(\vec{x}_2, y_2), (\vec{x}_3, y_3), (\vec{x}_4, y_4), \ldots, (\vec{x}_n, y_n)$	$(\vec{x}_1, y_1)$
$(\vec{x}_1, y_1), (\vec{x}_3, y_3), (\vec{x}_4, y_4), \ldots, (\vec{x}_n, y_n)$	$(\vec{x}_2, y_2)$
$(\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_4, y_4), \ldots, (\vec{x}_n, y_n)$	$(\vec{x}_3, y_3)$
...	...
$(\vec{x}_1, y_1),(\vec{x}_2, y_2),(\vec{x}_3, y_3), \ldots,(\vec{x}_{n-1},y_{n-1})$	$(\vec{x}_n, y_n)$

=> **Error estimate:**

$$Err_{loo}(h) = \frac{1}{n} \sum_{i=1}^{n} \left| h_i(\vec{x}_i) = y_i \right|$$

Question: Is there a connection between margin and the estimate?

Necessary Cond. for Leave-One-Out Error of SVM

Lemma: SVM $\left[h_i(\vec{x}_i) \neq y_i\right] \Rightarrow \left[2\alpha_i R^2 + \xi_i \geq 1\right]$ [Joachims, 2000] [Jaakkola & Haussler, 1999] [Vapnik & Chapelle, 2000]

Input:

- α_i dual variable of example *i*
- ξ_i slack variable of example *i*
- $\|\vec{x}\| \leq R$ bound on length

$$\Downarrow$$

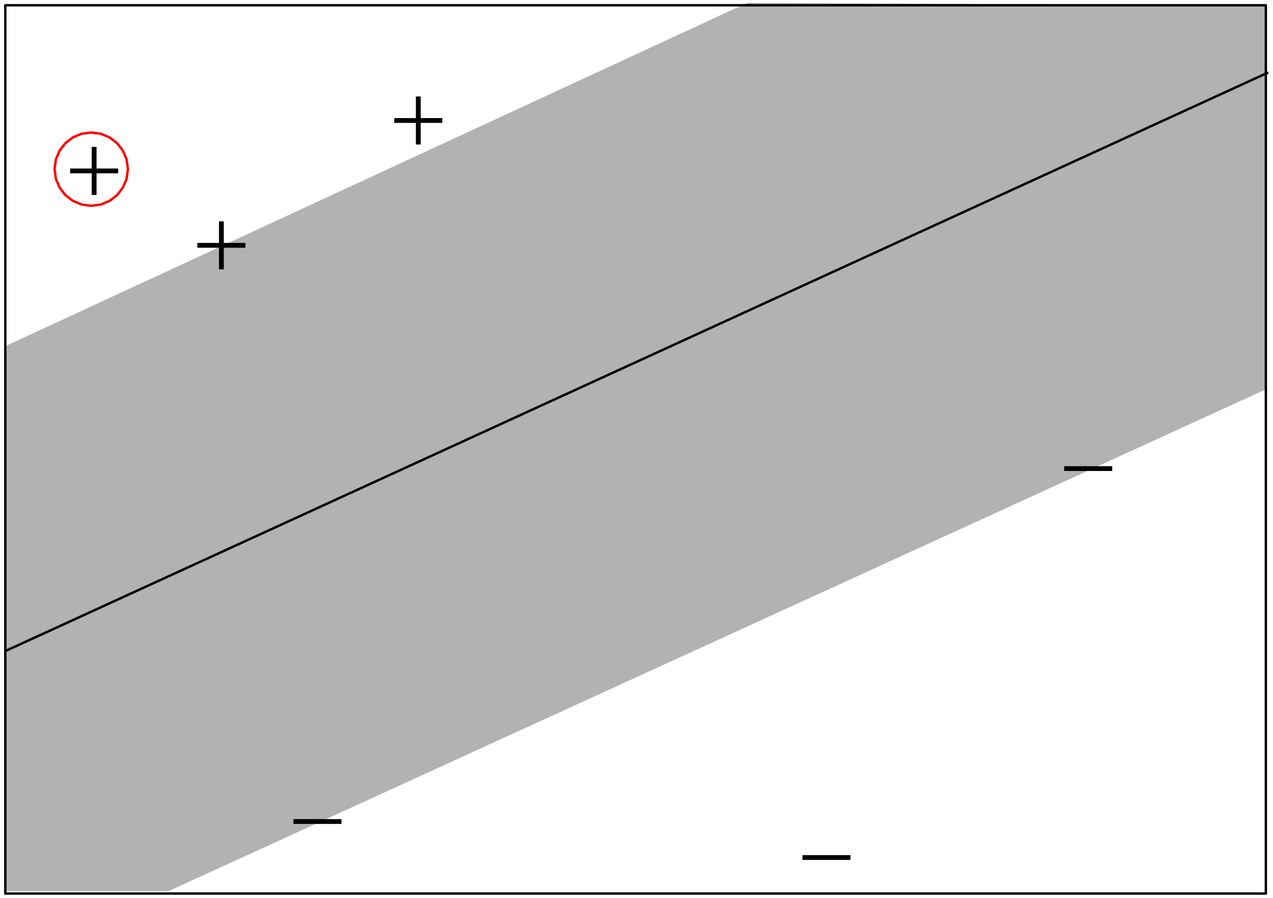
Available after training SVM on the full training data

Example:

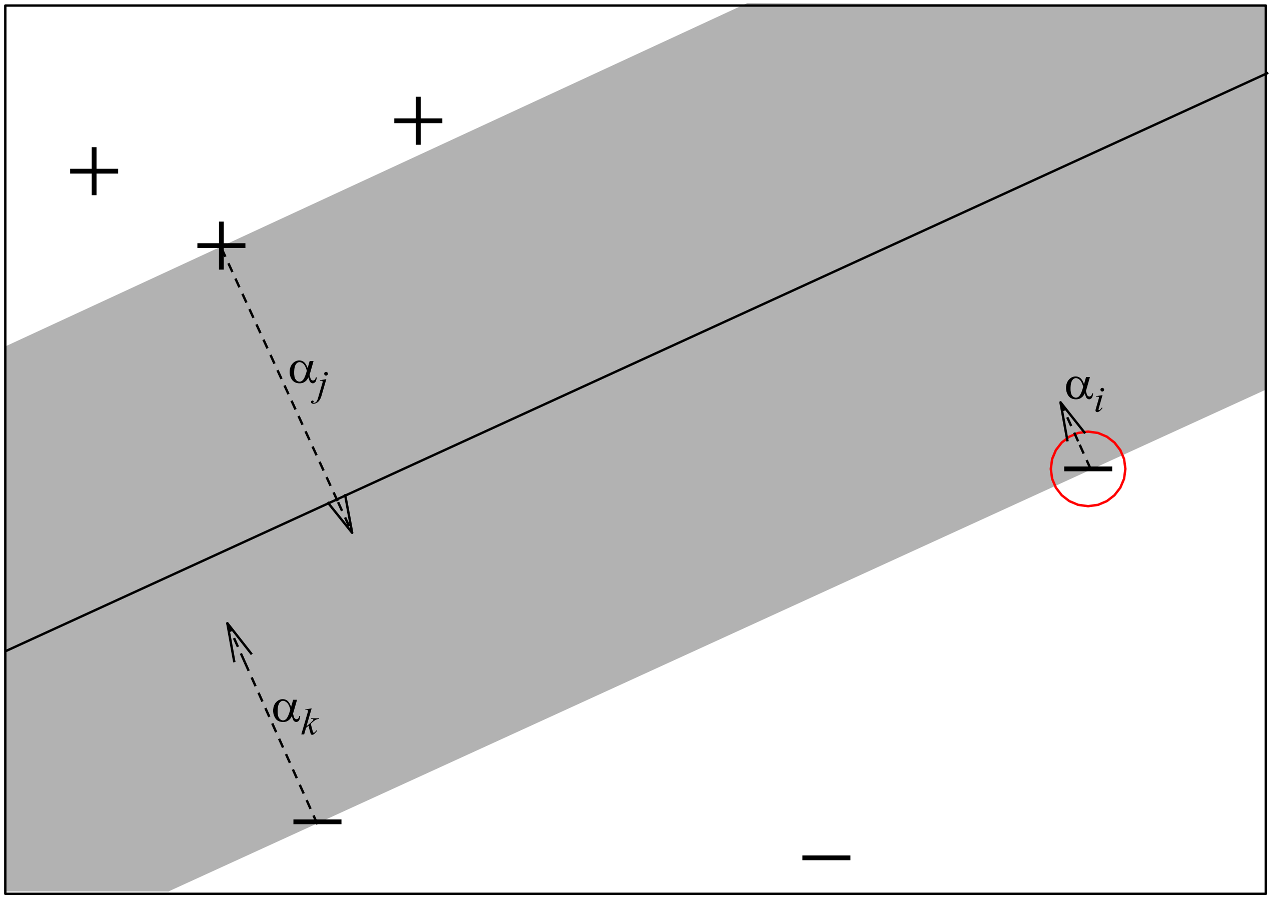
$\rho\alpha_i R^2 + \xi_i$	**leave-one-out error**
0.0	OK
0.7	OK
3.5	ERROR
0.1	OK
1.3	OK
0.0	OK
0.0	OK
...	...

Case 1: Example is no SV

$$(\alpha_i = 0) \Rightarrow (\xi_i = 0) \Rightarrow (2\alpha_i R^2 + \xi_i < 1) \Rightarrow \textit{no leave-one-out error}$$

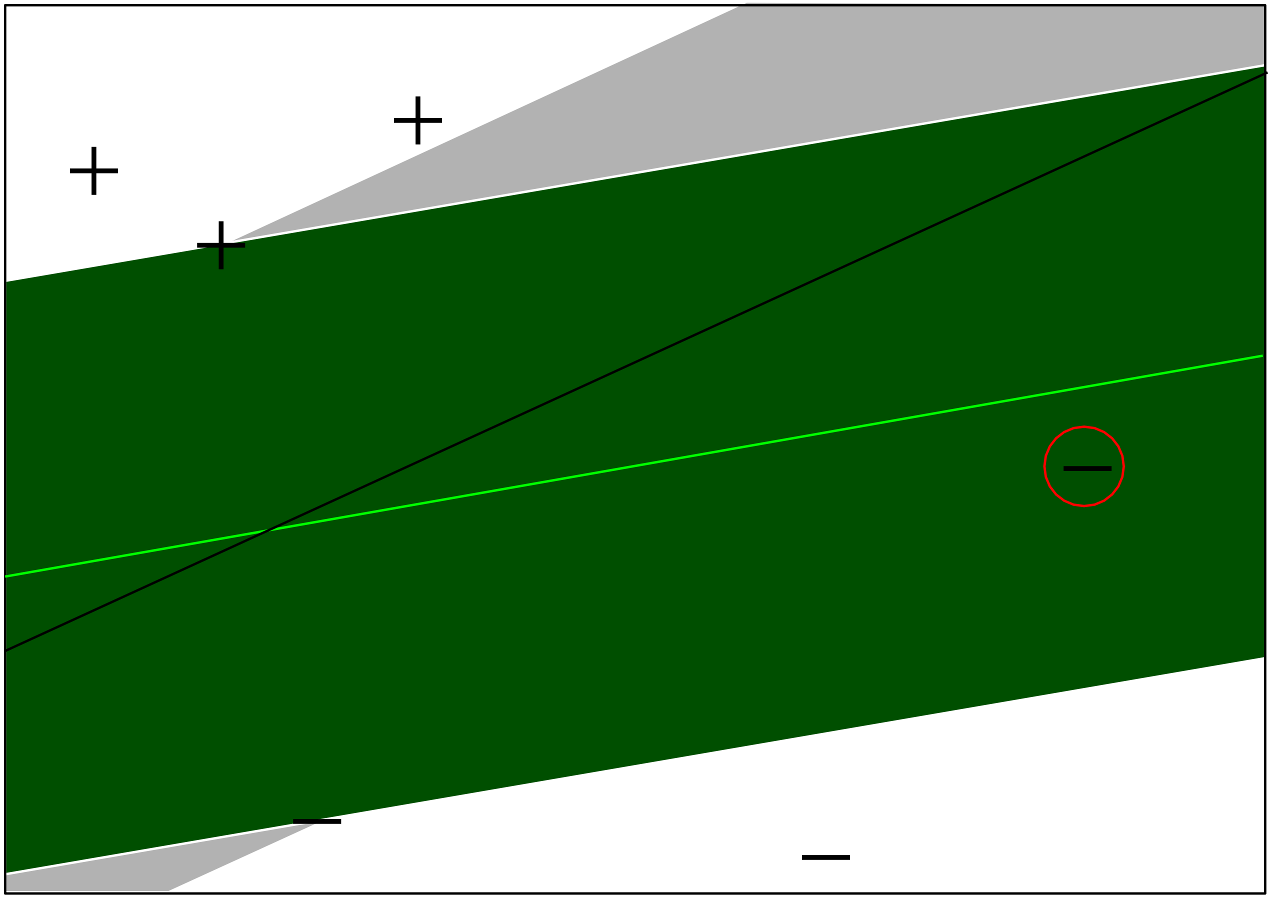


Case 2: Example is SV with Low Influence



$$\left(\alpha_i < \frac{0.5}{R^2} < C\right) \Rightarrow (\xi_i = 0) \Rightarrow (2\alpha_i R^2 + \xi_i < 1) \Rightarrow \textit{no leave-one-out error}$$

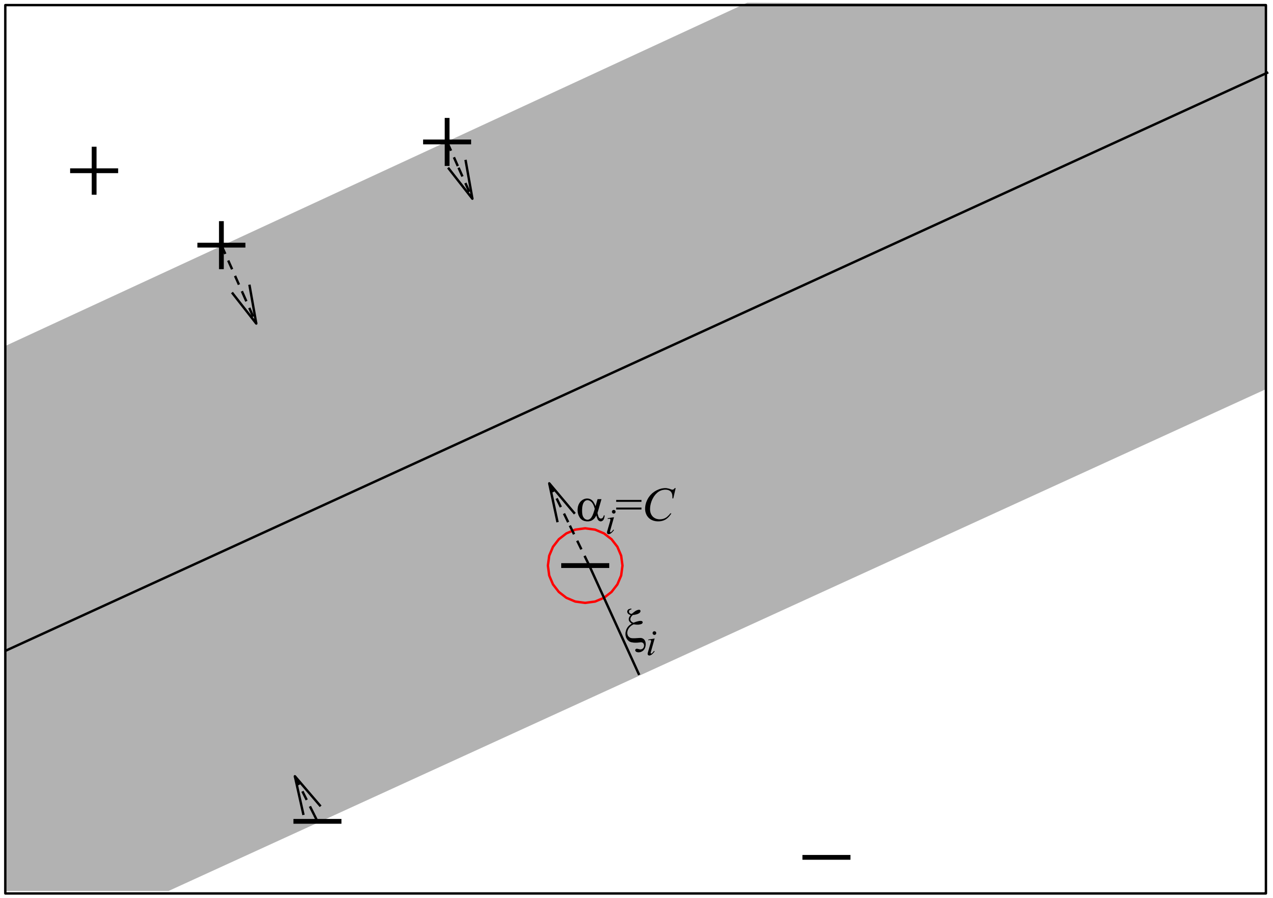
Case 2: Example is SV with Low Influence



$$\left(\alpha_i < \frac{0.5}{R^2} < C\right) \Rightarrow (\xi_i = 0) \Rightarrow (2\alpha_i R^2 + \xi_i < 1) \Rightarrow \textit{no leave-one-out error}$$

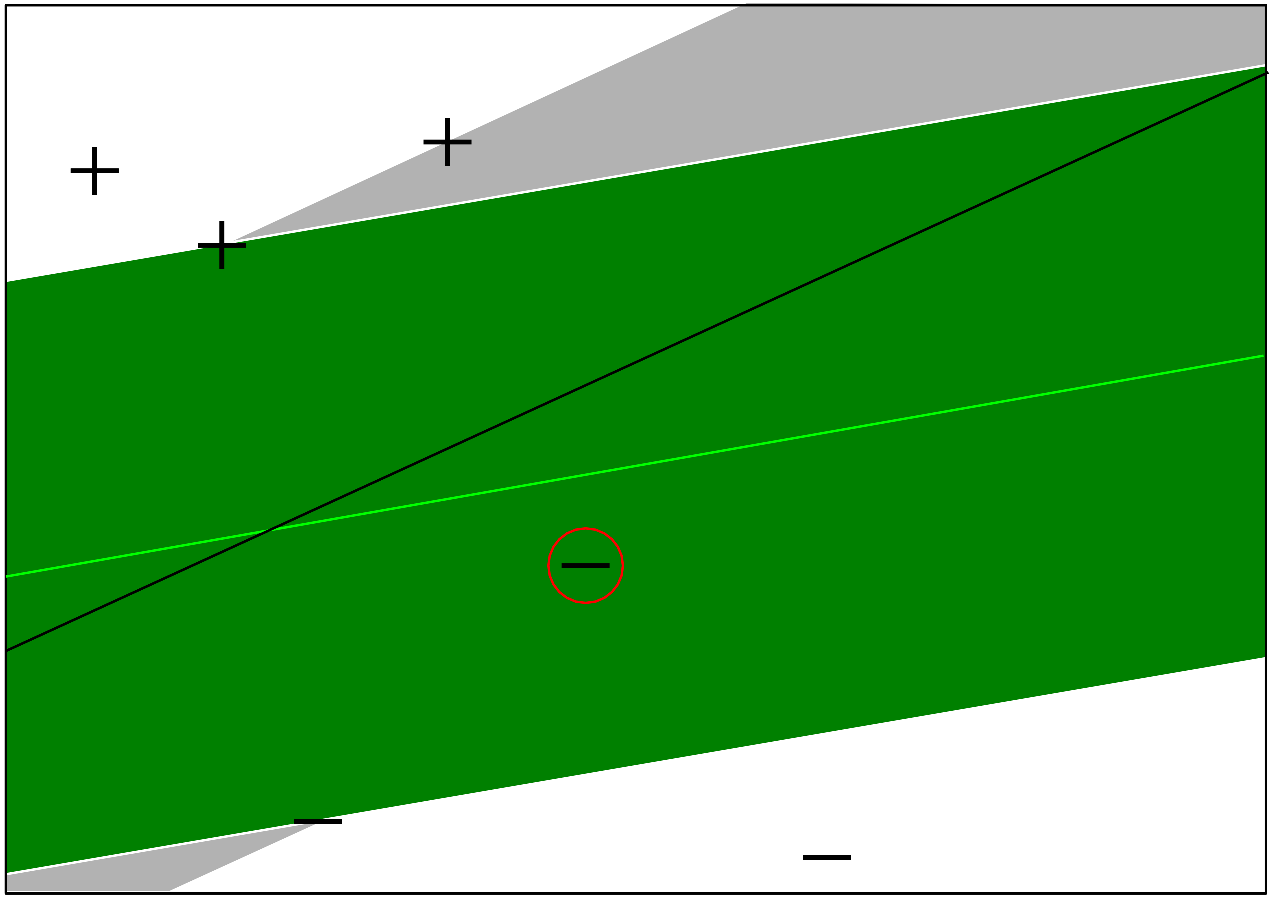
Case 3: Example has Small Training Error

$$(\alpha_i = C) \wedge (\xi_i < 1 - 2CR^2) \Rightarrow (2\alpha_i R^2 + \xi_i < 1) \Rightarrow \textit{no leave-one-out error}$$



Case 3: Example has Small Training Error

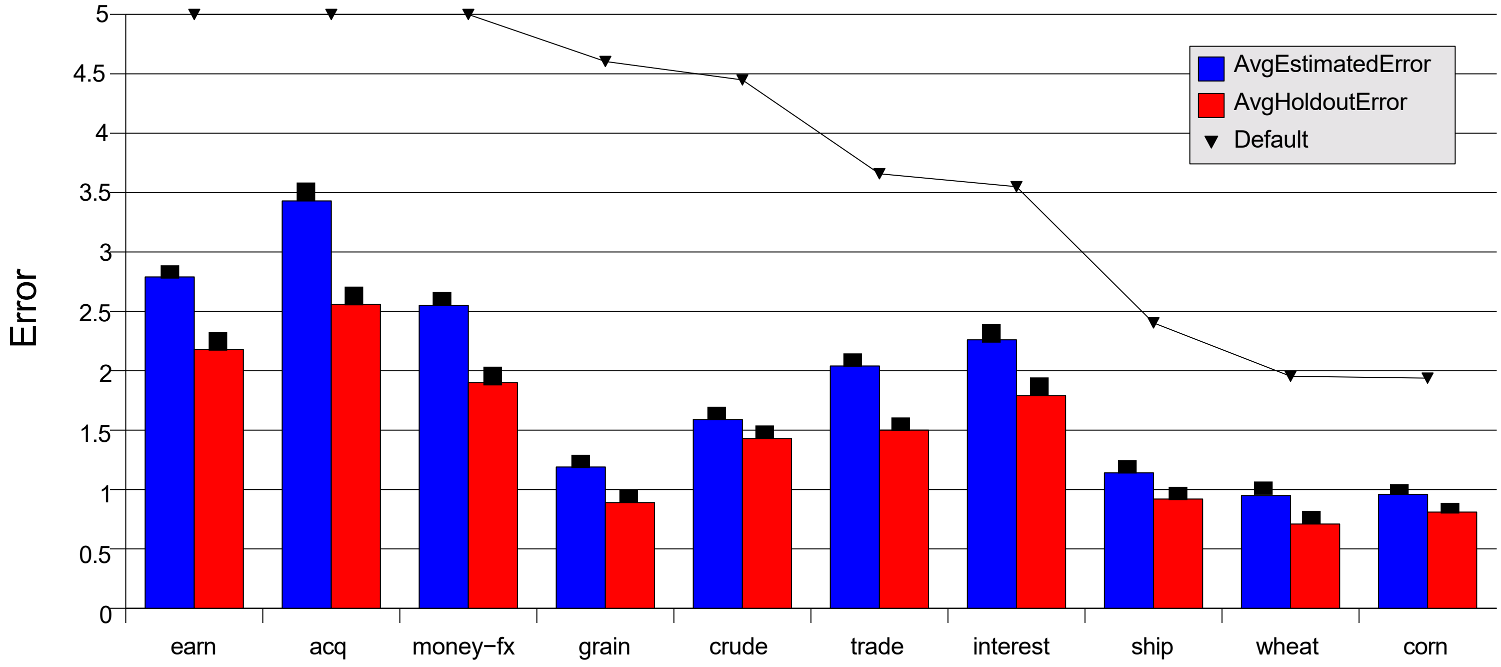
$$(\alpha_i = C) \wedge (\xi_i < 1 - 2CR^2) \Rightarrow (2\alpha_i R^2 + \xi_i < 1) \Rightarrow \textit{no leave-one-out error}$$



Experiment: Reuters-21578

- 6451 training examples
- 6451 test examples for holdout testing
- ~27,000 features

Average error estimate over 10 random training/test splits:



=> small bias, variance of estimators is approximately equal

Fast Leave-One-Out Estimation for SVMs

Lemma: Training errors are always leave-out-out errors.

Algorithm:

```
• $(R, \alpha, \xi) = train\_SVM(X,O,O)$;
• for all training examples, do
    • if $\xi_i > 1$ then loo++;
    • else if ( $\rho\alpha_i R^2 + \xi_i < 1$ ) then loo=loo;
    • else train_SVM( $X_i, \alpha, \xi$ );
```

Experiment:

	Training	Retraining Steps (%)		CPU-Time (sec)	
	Examples	$\rho = 1$	$\rho = 2$	$\rho = 1$	$\rho = 2$
Reuters	6451	0.20%	0.58%	11.1	32.3
WebKB	2092	6.78%	20.42%	78.5	235.4
Ohsumed	10000	1.07%	2.56%	433.0	1132.3

Estimated Error of SVM

Leave-One-Out Error Estimate: $Err_{loo}(h) = \frac{1}{n}\sum_{i=1}^{n} \left| h_i(\vec{x}_i) = y_i \right|$

For general SVMs:

$$\left[h_i(\vec{x}_i) \neq y_i \right] \Rightarrow \left[2\alpha_i R^2 + \xi_i \geq 1 \right] \qquad \left\| \vec{x} \right\| \leq R$$

$$\Rightarrow Err_{loo}(h) \leq \frac{1}{n} \left| \left\{ i \;\middle|\; 2\alpha_i R^2 + \xi_i \geq 1 \right\} \right| \leq \frac{1}{n} \sum_{i=1}^{n} 2\alpha_i R^2 + \xi_i$$

For separable problems:

$$\left[h_i(\vec{x}_i) \neq y_i \right] \Rightarrow \left[\alpha_i R^2 \geq 1 \right] \qquad \left\| \vec{x} \right\| \leq R$$

$$\Rightarrow Err_{loo}(h) \leq \frac{1}{n} \left| \left\{ i \;\middle|\; \alpha_i R^2 \geq 1 \right\} \right| \leq \frac{1}{n} \sum_{i=1}^{n} \alpha_i R^2 \leq \frac{R^2}{\delta^2}$$

Summary
Why do SVMs Work?

If

- the training error $Err_S(h)$ (on the sample S / on average) is low and
- the margin δ/R (on the sample S / on average) is large

then

- the SVM has learned a classification rule with low error rate with high probablility (worst-case).
- the SVM learns classification rules that have low error rate on average.
- the SVM has learned a classification rule for which the (leave-one-out) estimated error rate is low.

Part 4: When do SVMs Work Well?

Successful Use:

- Optical Character Recognition (OCR) [Vapnik, 1998]
- Face Recognition, etc. [Osuna et al., 1997]
- Text Classification [Joachims, 1997] [Dumais et al., 1998]
- ...

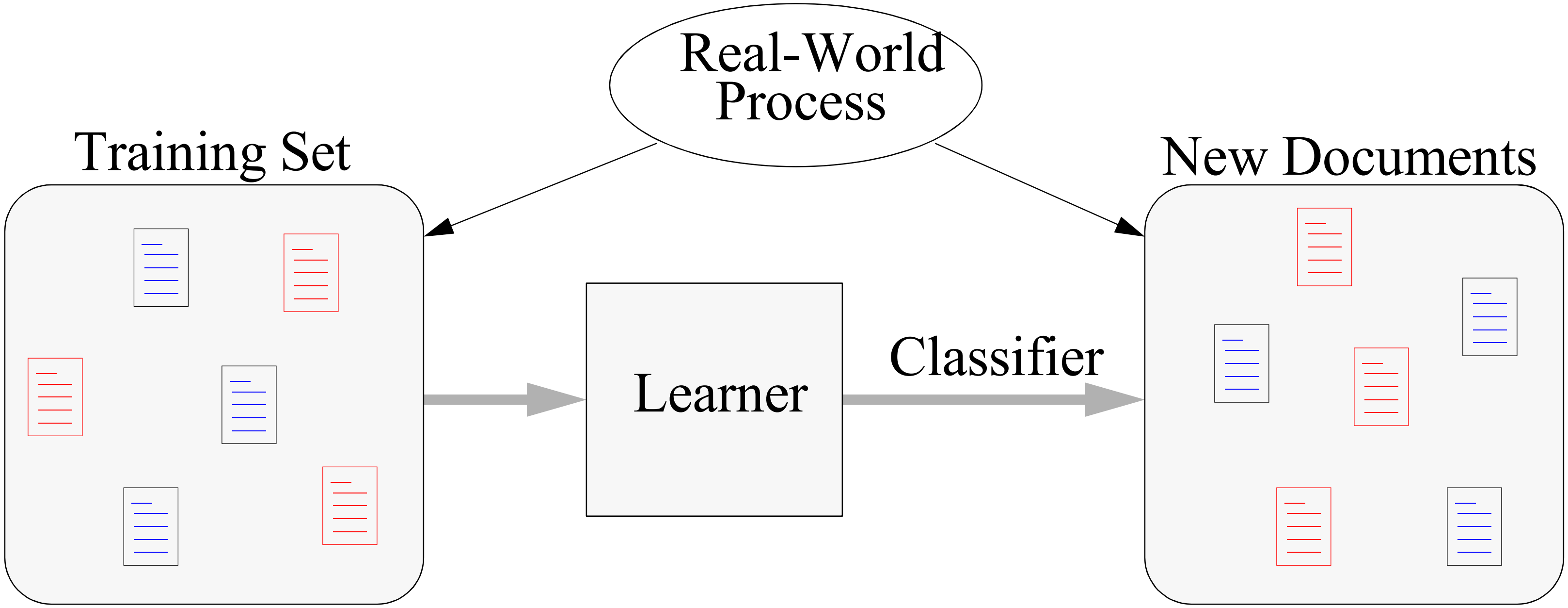
Open Questions:

What characterizes these problems?

How can the good performance be explained?

What are “sufficient conditions” for using (linear) SVMs successfully?

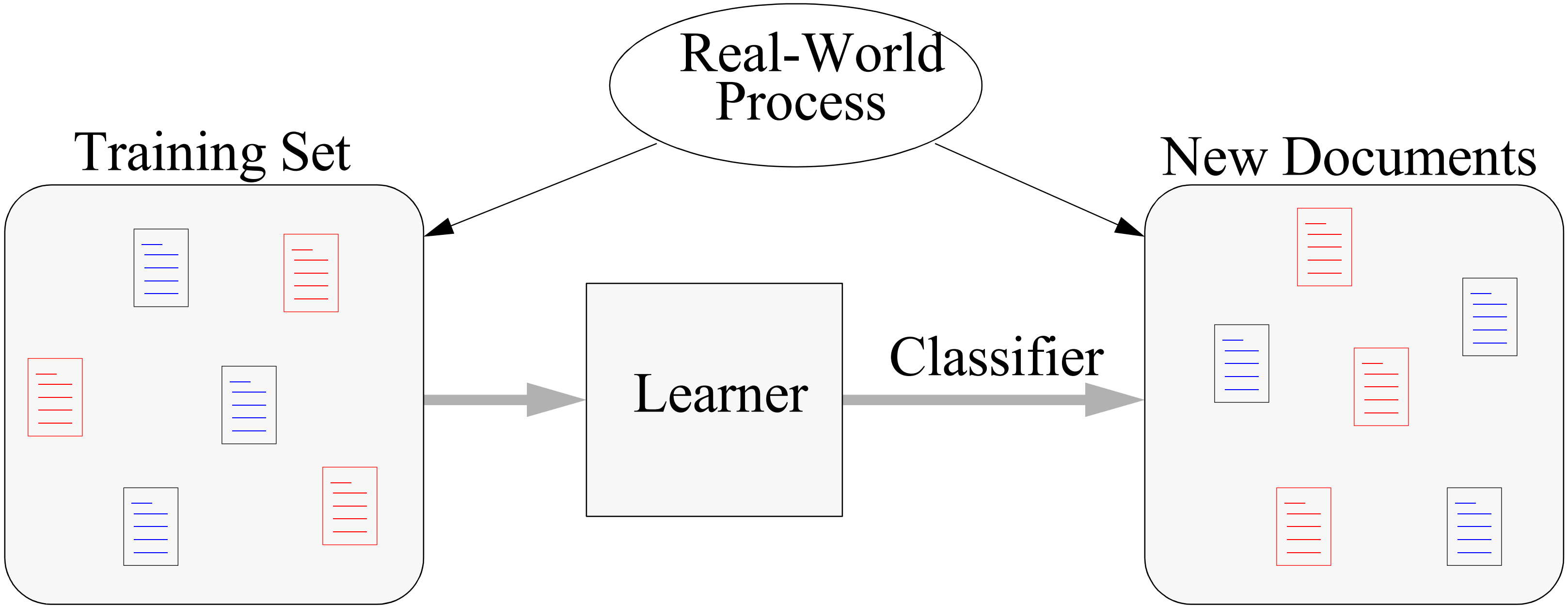
Learning Text Classifiers



Goal:

- Learner uses training set to find classifier with low prediction error.

Learning Text Classifiers



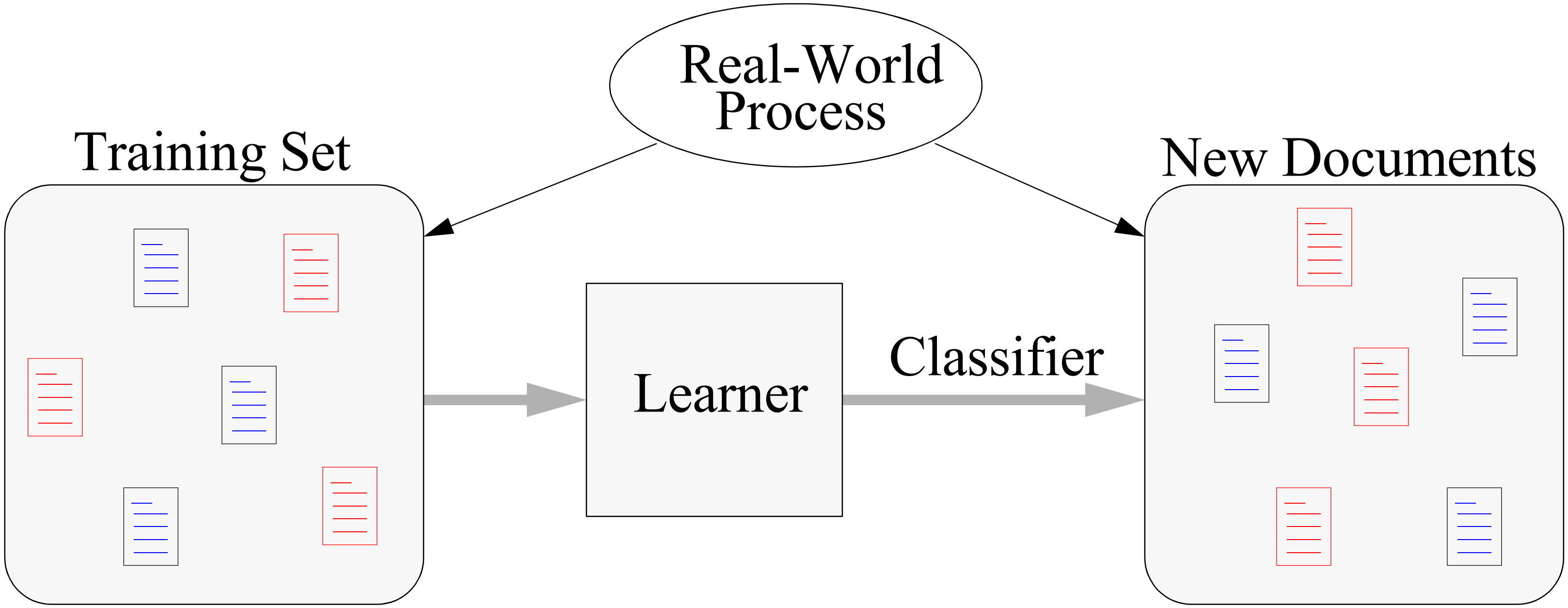
Goal:

- Learner uses training set to find classifier with low prediction error.

Obstacle:

- No Free Lunch: There is no learner that does well on every task.

Learning Text Classifiers Successfully

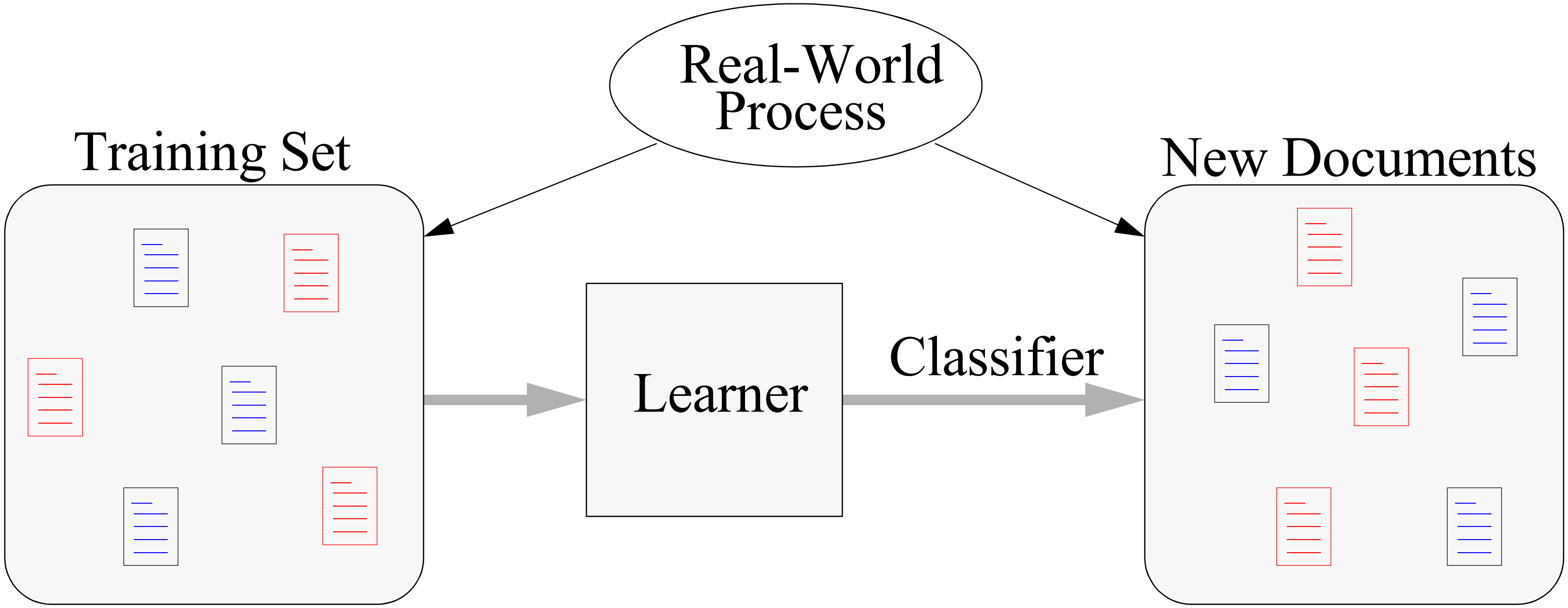


The learner produces a classifier with low error rate

<=>

The properties of the learner fit the properties of the process.

Learning SVM Text Classifiers Successfully

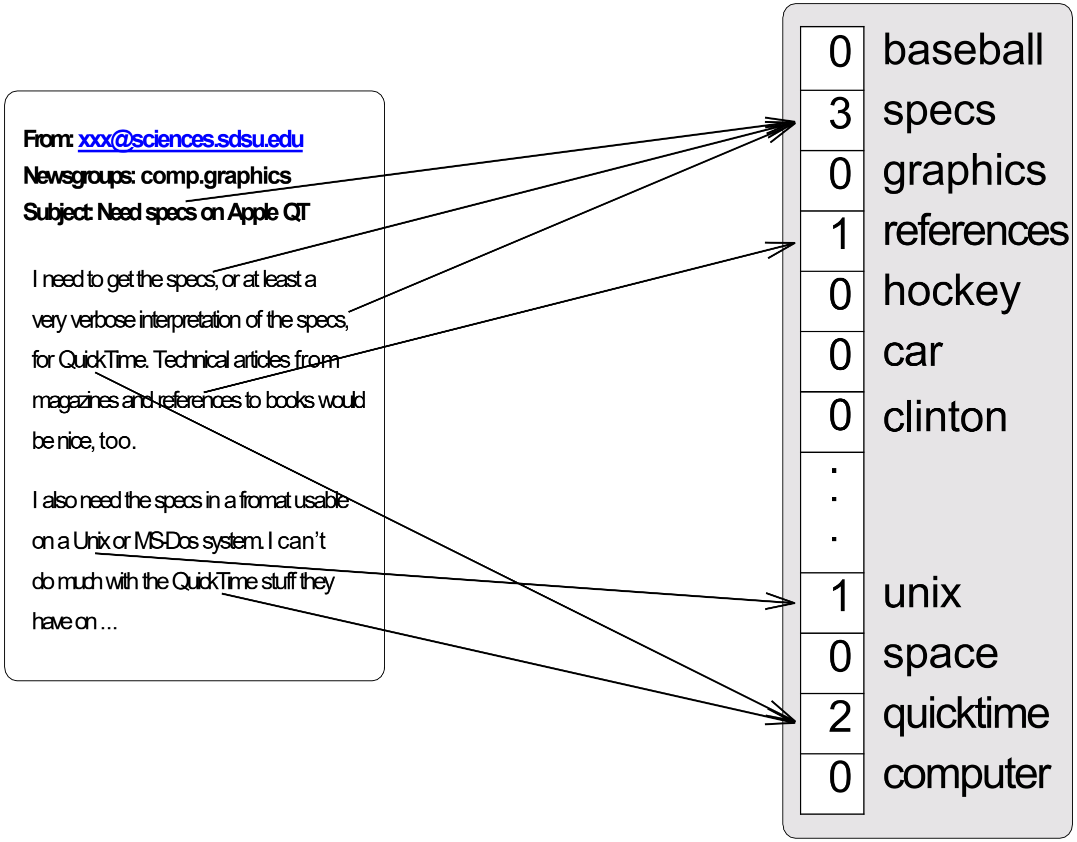


SVM
The ~~learner~~ produces a classifier with low error rate

<=>

SVM
The properties of the ~~learner~~ fit the properties of the process.

Representing Text As Feature Vectors



From: xxx@sciences.sdsu.edu
Newsgroups: comp.graphics
Subject: Need specs on Apple QT

I need to get the specs, or at least a very verbose interpretation of the specs, for QuickTime. Technical articles from magazines and references to books would be nice, too.

I also need the specs in a fromat usable on a Unix or MS-Dos system. I can't do much with the QuickTime stuff they have on ...

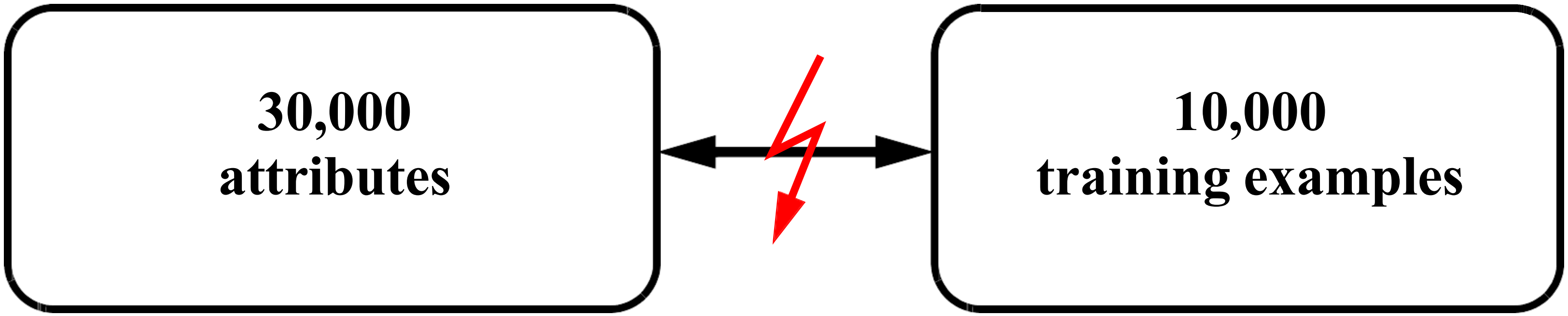
0	baseball
3	specs
0	graphics
1	references
0	hockey
0	car
0	clinton
. . .	
1	unix
0	space
2	quicktime
0	computer

Features: words (wordstems)

Values: occurrence frequency

==> Ignore ordering of words.

Paradox of Text Classification



30,000
attributes

10,000
training examples

Experimental Results

Reuters Newswire

- 90 categories
- 9603 training doc.
- 3299 test doc.
- ~27000 features

WebKB Collection

- 4 categories
- 4183 training doc.
- 226 test doc.
- ~38000 features

Ohsumed MeSH

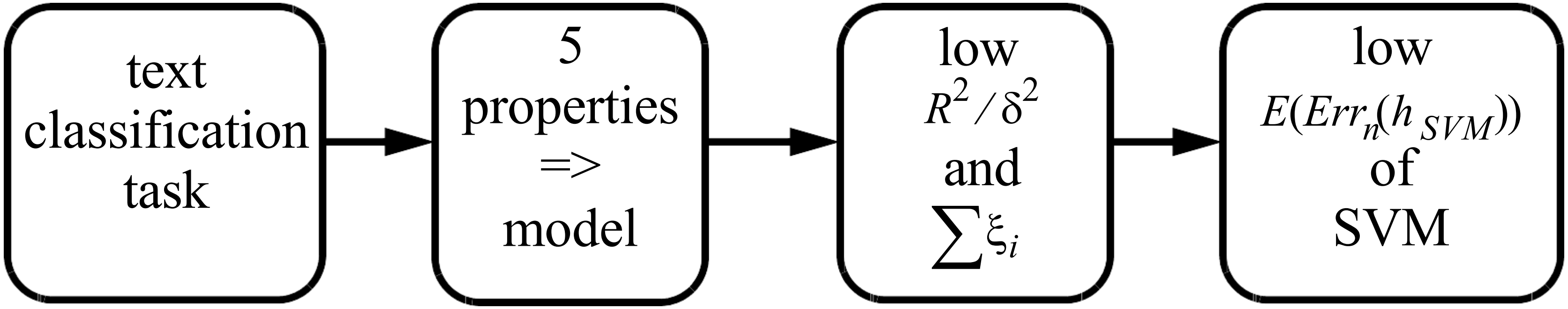
- 20 categories
- 10000 training doc.
- 10000 test doc.
- ~38000 features

microaveraged precision/recall breakeven-point [0..100]	**Reuters**	**WebKB**	**Ohsumed**
Naive Bayes	72.3	82.0	62.4
Rocchio Algorithm	79.9	74.1	61.5
C4.5 Decision Tree	79.4	79.1	56.7
k-Nearest Neighbors	82.6	80.5	63.4
SVM	**87.5**	**90.3**	**71.6**

[Joachims, 2002]

Why Do SVMs Work Well for Text Classification?

A statistical learning model of text classification with SVMs:



text classification task

5 properties => model

low R^2/δ^2 and $\sum \xi_i$

low $E(Err_n(h_{SVM}))$ of SVM

Margin/Loss Based Bound on the Expected Error

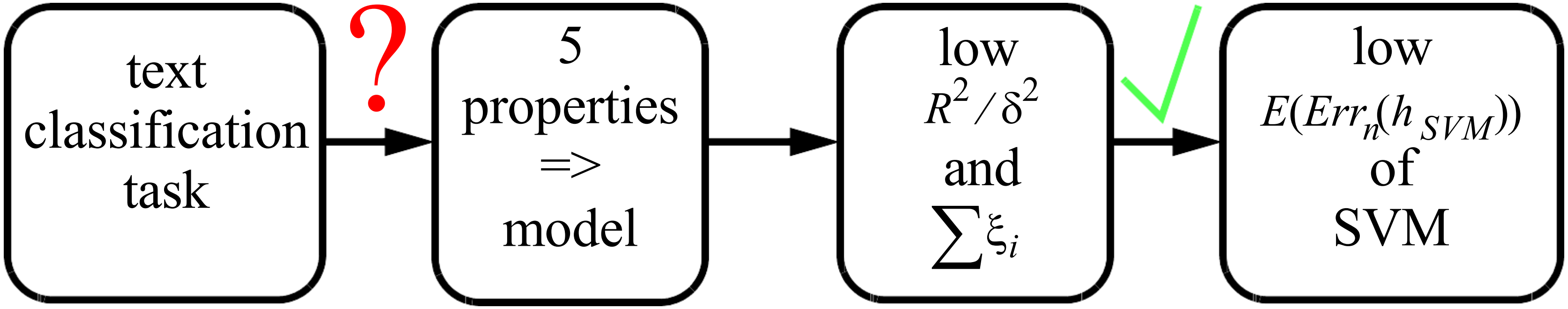
Theorem: The expected error of a soft margin SVM is bounded by

$$E(Err_n(h_{SVM})) \le \frac{\rho E\left(\frac{R^2}{\delta^2}\right) + \rho C R^2 E\left(\sum_{n=1}^{n+1} \xi_i\right)}{n+1} \qquad C \ge \frac{1}{\rho R^2}$$

$$E(Err_n(h_{SVM})) \le \frac{\rho E\left(\frac{R^2}{\delta^2}\right) + \rho (C R^2 + 1) E\left(\sum_{n=1}^{n+1} \xi_i\right)}{n+1} \qquad C < \frac{1}{\rho R^2}$$

Where $E\left(\frac{R^2}{\delta^2}\right)$ is the expected soft margin and $E\left(\sum_{n=1}^{n+1} \xi_i\right)$ is the expected training loss on training sets of size $n+1$.

First Step Completed



text classification task

?

5 properties => model

low R^2/δ^2 and $\sum \xi_i$

✓

low $E(Err_n(h_{SVM}))$ of SVM

Properties 1+2: Sparse Examples in High Dimension

- High dimensional feature vectors (30,000 features)
- Sparse document vectors: only a few words of the whole language occur in each document

	Training Examples	**Number of Features**	**Distinct Words (Sparsity)**
Reuters Newswire Articles	9,603	27,658	74 (0.27%)
Ohsumed MeSH Abstracts	10,000	38,679	100 (0.26%)
WebKB WWW-Pages	3,957	38,359	130 (0.34%)

Property 3: Heterogeneous Use Of Words

MODULAIRE BUYS BOISE HOMES PROPERTY

Modulaire Industries said it acquired the design library and manufacturing rights of privately-owned Boise Homes for an undisclosed amount of cash. Boise Homes sold commercial and residential prefabricated structures, Modulaire said.

JUSTICE ASKS U.S. DISMISSAL OF TWA FILING

The Justice Department told the Transportation Department it supported a request by USAir Group that the DOT dismiss an application by Trans World Airlines Inc for approval to take control of USAir. ``Our rationale is that we reviewed the application for control filed by TWA with the DOT and ascertained that it did not contain sufficient information upon which to base a competitive review," James Weiss, an official in Justice's Antitrust Division, told Reuters.

USX, CONS. NATURAL END TALKS

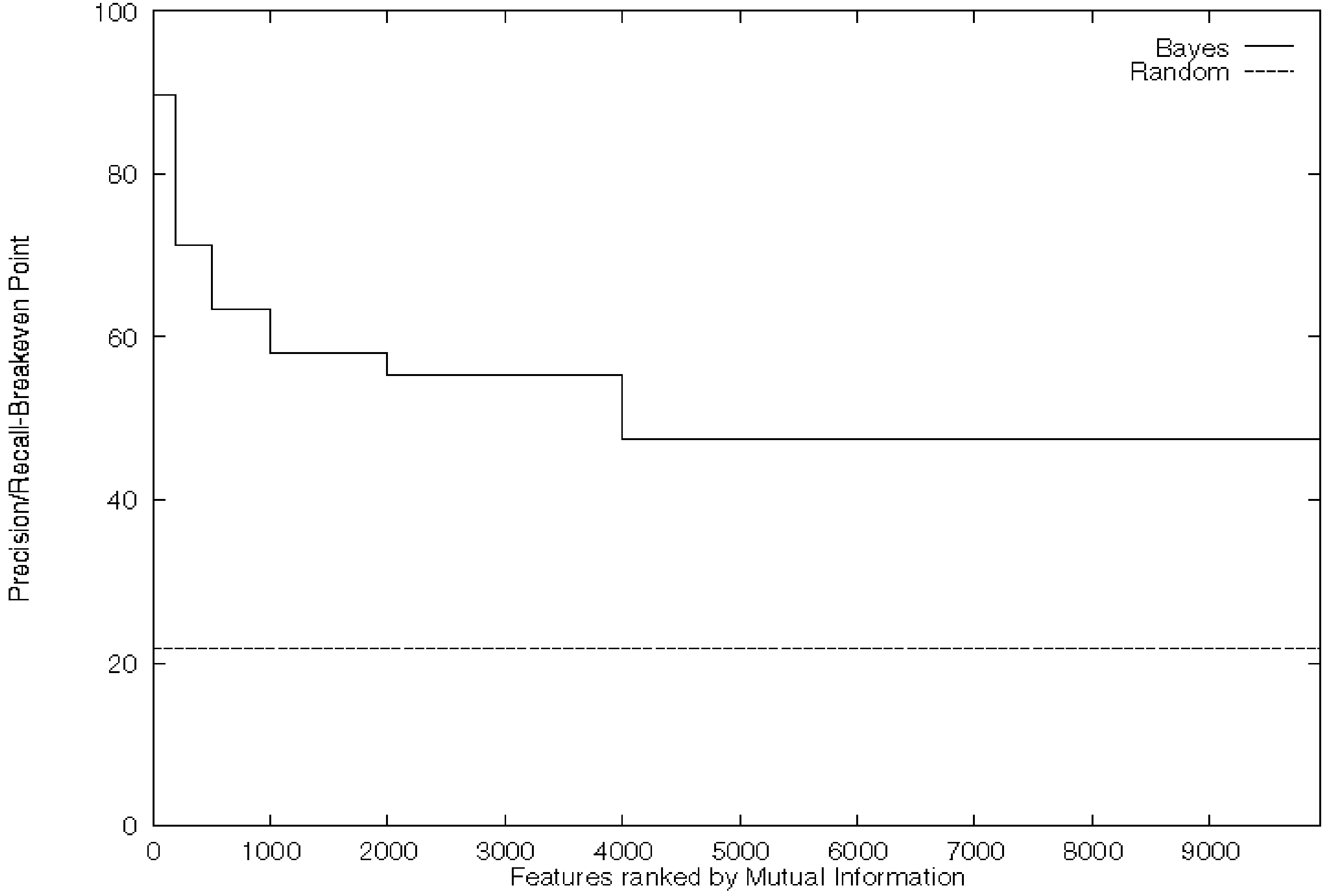
USX Corp's Texas Oil and Gas Corp subsidiary and Consolidated Natural Gas Co have mutually agreed not to pursue further their talks on Consolidated's possible purchase of Apollo Gas Co from Texas Oil. No details were given.

E.D. And F. MAN TO BUY INTO HONG KONG FIRM

The U.K. Based commodity house E.D. And F. Man Ltd and Singapore's Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeo's 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

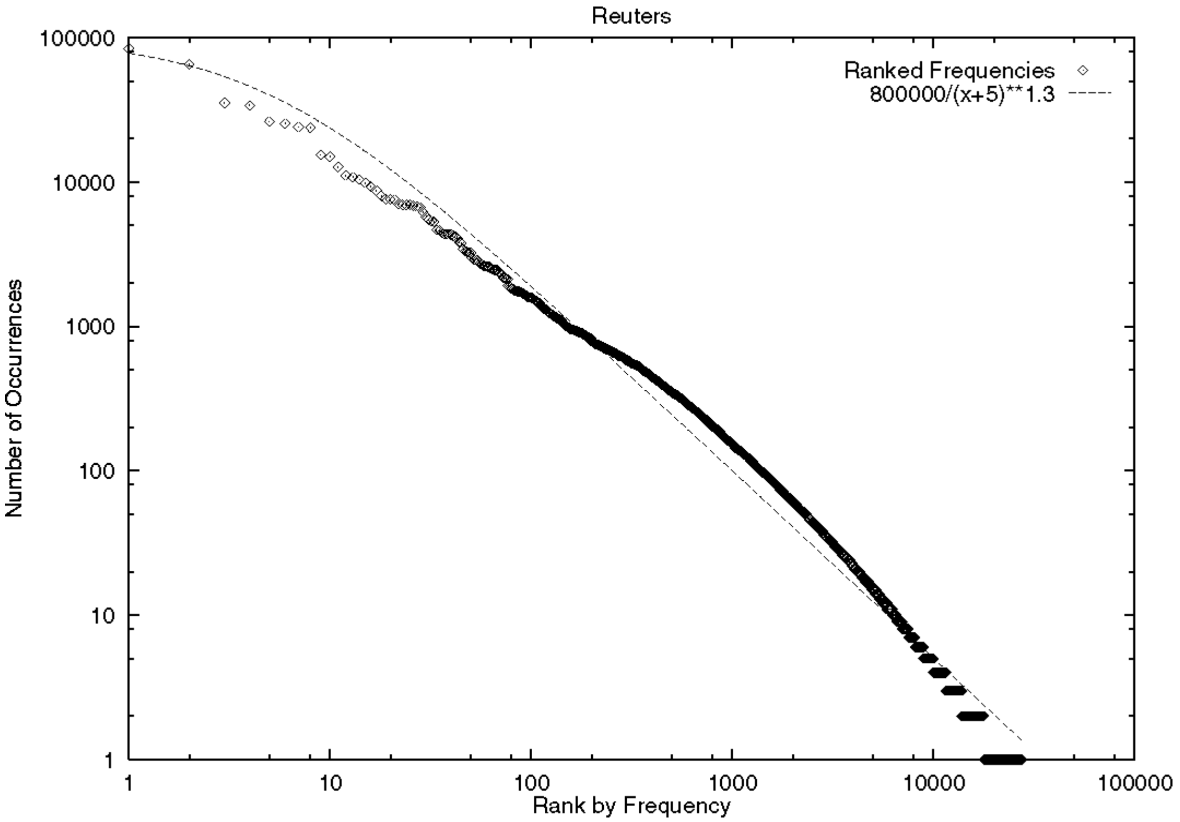
No pair of documents shares any words, but “it”, “the”, “and”, “of”, “for”, “an”, “a”, “not”, “that”, “in”.

Property 4: High Level Of Redundancy



=> Few features are irrelevant!

Property 5: “Zipf’s Law”



Zipf’s Law: In text, the *i*-th frequent word occurs $f_i = \frac{k}{(c+i)^{\Theta}}$ times.

=> Most words occur very infrequently!

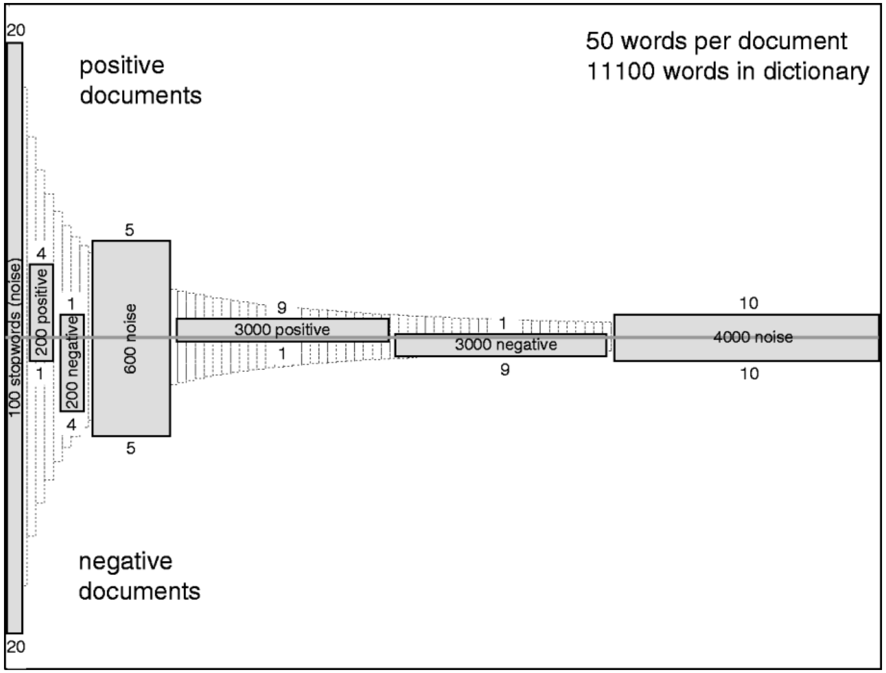
Text Classification Model

Definition: For the TCat-concept there are s disjoint sets of features.

$$TCat([p_1|n_1|f_1], \ldots, [p_s|n_s|f_s])$$

Each positive (negative) example contains p_i (n_i) occurrences from the f_i features in set i.

Example: $TCat\begin{pmatrix}[20|20|100]\\ [4|1|200]\\ [1|4|200]\\ [5|5|600]\\ [9|1|3000]\\ [1|9|3000]\\ [10|10|4000]\end{pmatrix}$



TCat-Concept for WebKB “Course”

$$TCat\left(\begin{array}{c}[77|29|98],\ [4|21|52]\\ [16|2|431],\ [1|12|341]\\ [9|1|5045],\ [1|21|24276]\\ [169|191|8116]\end{array}\right)\begin{array}{l}\textit{high frequency}\\ \textit{medium frequency}\\ \textit{low frequency}\\ \textit{rest}\end{array}$$

	high frequency	medium frequency	low frequency
pos	**98 words** all any assignment assignments available be book c chapter class code course cse description discussion document due each eecs exam exams fall final ... section set should solution solutions spring structures students syllabus ta text textbook there thursday topics tuesday unix use wednesday week will you your	**431 words** account acrobat adapted addison adt ahead aho allowed alternate announced announcement announcements answers appointment approximately ... tuesdays turing turn turned tuth txt uidaho uiowa ullman understand ungraded units unless upenn usr vectors vi walter weaver wed wednesdays weekly weeks weights wesley yurttas	**5045 words** 002cc 009a 00a 00om 01oct 01pm 02pm 03oct 03pm 03sep 04dec ... gradable gradebook gradebooks gradefreq1 gradefreq2 gradefreq3 graders gradesheet gradients grafica grafik ... zimmermann zinc zipi zipser zj zlocate znol zoran zp zwatch zwhere zwiener zyda
neg	acm address am austin ca california center college computational conference contact current currently d department dr faculty fax graduate group he ... me member my our parallel performance ph pp proceedings professor publications recent research sciences support technical technology university vision was working **52 words**	aaai academy accesses accurate adaptation advisor advisory affiliated affiliations agent agents alberta album alumni amanda america amherst annual ... victoria virginia visiting visitors visualization vita vitae voice wa watson weather webster went west wi wife wireless wisconsin worked workshop workshops wrote yale york **341 words**	0a 0b 0b1 0e 0f 0r 0software 0x82d4ff 100k 100mhz 100th 1020x620 102k 103k ... lunar lunches lunchtime lund lundberg lunedi lung luniewski luo luong lupin lupton lure lurker lus ... zuo zuowei zurich zvi zw zwaenepoel zwarico zwickau zwilling zygmunt zzhen00 **24276 words**
	high frequency	**medium frequency**	**low frequency**

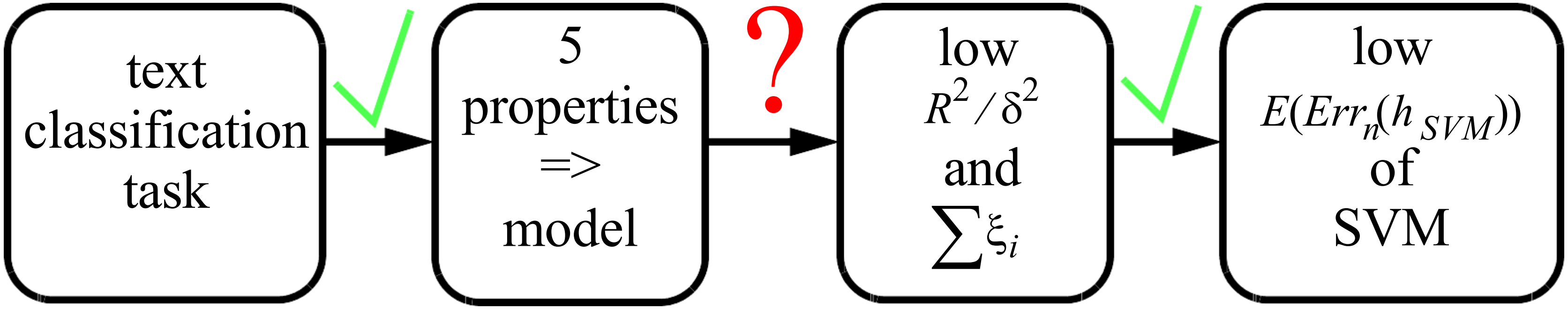
Real Text Classification Tasks as TCat-Concepts

$$\text{Reuters “Earn”: } TCat\left(\begin{array}{c} [33|2|65], [32|65|152] \\ [2|1|171], [3|21|974] \\ [3|1|3455], [1|10|17020] \\ [78|52|5821] \end{array}\right) \begin{array}{l} \textit{high frequency} \\ \textit{medium frequency} \\ \textit{low frequency} \\ \textit{rest} \end{array}$$

$$\text{Webkb “Course”: } TCat\left(\begin{array}{c} [77|29|98], [4|21|52] \\ [16|2|431], [1|12|341] \\ [9|1|5045], [1|21|24276] \\ [169|191|8116] \end{array}\right) \begin{array}{l} \textit{high frequency} \\ \textit{medium frequency} \\ \textit{low frequency} \\ \textit{rest} \end{array}$$

$$\text{Ohsumed “Pathology”: } TCat\left(\begin{array}{c} [2|1|10], [1|4|22] \\ [2|1|92], [1|2|94] \\ [5|1|4080], [1|10|20922] \\ [197|190|13459] \end{array}\right) \begin{array}{l} \textit{high frequency} \\ \textit{medium frequency} \\ \textit{low frequency} \\ \textit{rest} \end{array}$$

Second Step Completed



text classification task
✓
5 properties => model
?
low R^2/δ^2 and $\sum \xi_i$
✓
low $E(Err_n(h_{SVM}))$ of SVM

The Margin δ^2 of TCat-Concepts

Lemma 1: For $TCat([p_1|n_1|f_1], \ldots, [p_s|n_s|f_s])$ -concepts there is always a hyperplane passing through the origin with margin δ^2 at least

$$\delta^2 \geq \frac{ad - b^2}{a + 2b + d} \qquad with \qquad a = \sum_{i=1}^{s} \frac{p_i^2}{f_i} \qquad d = \sum_{i=1}^{s} \frac{n_i^2}{f_i} \qquad b = \sum_{i=1}^{s} \frac{n_i p_i}{f_i}$$

Example: The previous example WebKB "course" has a margin of at least

$$\delta^2 \geq 0.23$$

The Length R^2 of Document Vectors

Lemma 2: If the ranked term frequencies f_i in a document with l words have the form of the generalized Zipf's Law

$$f_i = \frac{k}{(c+i)^{\Theta}}$$

based on their frequency rank i, then the Euclidean length of the document vector $\vec{x}$ is bounded by

$$\|\vec{x}\| \leq \sqrt{\sum_{i=1}^{d} \left(\frac{k}{(c+i)^{\Theta}}\right)^2} \qquad \text{with} \qquad \sum_{i=1}^{d} \frac{k}{(c+i)^{\Theta}} = l$$

Example: For WebKB “course” with

$$f_i = \frac{470000}{(5+i)^{1.25}}$$

follows that $R^2 \leq 1900$.

R^2, δ^2, and $\sum\xi_i$ for Text Classification

Reuters Newswire Stories

- 10 most frequent categories
- 9603 training examples
- 27658 attributes

$$E(Err_P(h_{SVM})) \le \frac{E\left(\frac{R^2}{\delta^2}\right) + CR^2 E\left(\sum_{i=1}^{n+1} \xi_i\right)}{n+1}$$

	R^2/δ^2	$\sum\xi_i$
earn	1143	0
acq	1848	0
money-fx	1489	27
grain	585	0
crude	810	4

	R^2/δ^2	$\sum\xi_i$
trade	869	9
interest	2082	33
ship	458	0
wheat	405	2
corn	378	0

Learnability of TCat-Concepts

Theorem: For $TCat([p_1|n_1|f_1], \ldots, [p_s|n_s|f_s])$ -concepts and documents with l words that follow the generalized Zipf's Law $f_i = k/(c+i)^\Theta$ the expected generalization error of an unbiased SVM after training on n examples is bounded by

$$E(Err_n(h_{SVM})) \leq \frac{R^2}{n+1} \frac{ad - b^2}{a + 2b + d} \quad with \quad \begin{aligned} a &= \sum_{i=1}^{s} \frac{p_i^2}{f_i} \\ d &= \sum_{i=1}^{s} \frac{n_i^2}{f_i} \\ b &= \sum_{i=1}^{s} \frac{n_i p_i}{f_i} \\ R^2 &\leq \sum_{i=1}^{d} \left(\frac{k}{(c+i)^\Theta}\right)^2 \end{aligned}$$

Comparison Theory vs. Experiments

	Learning Curve Bound	Predicted Bound on Error Rate	Error Rate in Experiment
Reuters "earn"	$E(Err_n(h_{SVM})) \leq \frac{138}{n+1}$	1.5%	1.3%
WebKB "course"	$E(Err_n(h_{SVM})) \leq \frac{443}{n+1}$	11.2%	4.4%
Ohsumed "pathology"	$E(Err_n(h_{SVM})) \leq \frac{9457}{n+1}$	94.6%	23.1%

- Model can differentiate between "difficult" and "easy" tasks
- Predicts and reproduces the effect of information retrieval heuristics (e.g. TFIDF-weighting)

Sensitivity Analysis

What makes a text classification problem suitable for a linear SVM?

High Redundancy:

$$TCat\left(\begin{array}{c}{[40|40|50]}\\ {[25|5|1000],\ [5|25|1000]}\\ {[30|30|30000]}\end{array}\right)\quad\begin{array}{l}\textit{high frequency}\\ \textit{medium frequency}\\ \textit{low frequency}\end{array}$$

High Discriminatory Power:

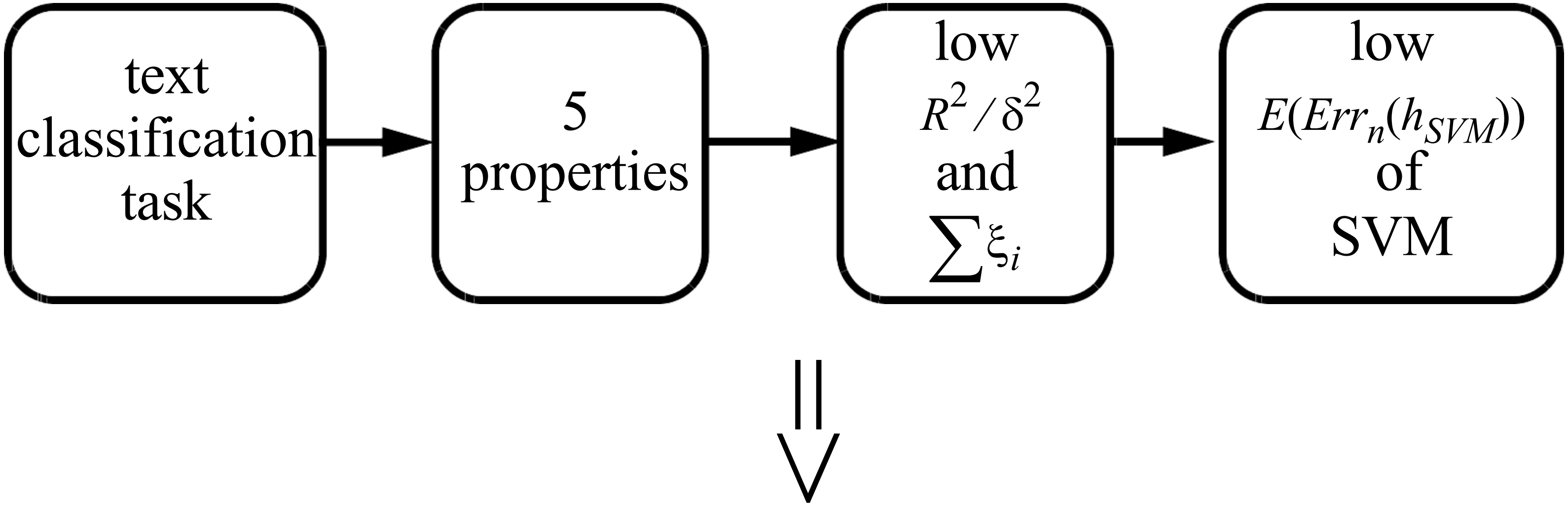
$$TCat\left(\begin{array}{c}{[40|40|50]}\\ {[15|0|500],\ [0|15|500],\ [15|15|1000]}\\ {[30|30|30000]}\end{array}\right)\quad\begin{array}{l}\textit{high frequency}\\ \textit{medium frequency}\\ \textit{low frequency}\end{array}$$

High Frequency:

$$TCat\left(\begin{array}{c}{[16|4|10],\ [4|16|10],\ [20|20|30]}\\ {[30|30|2000]}\\ {[30|30|30000]}\end{array}\right)\quad\begin{array}{l}\textit{high frequency}\\ \textit{medium frequency}\\ \textit{low frequency}\end{array}$$

What does this Model Provide?

Connects the statistical properties of text classification tasks with generalization error of SVM!



text classification task

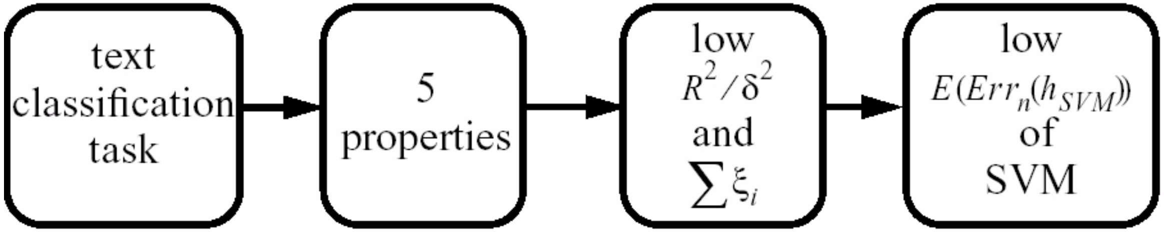
5 properties

low R^2/δ^2 and $\sum \xi_i$

low $E(Err_n(h_{SVM}))$ of SVM

- Explains the behavior of (linear) SVMs on text classification tasks
- Gives guideline for when to apply (linear) SVMs
- Provides formal basis for developing new methods

Summary
When do (Linear) SVMs Work Well?



text classification task

5 properties

low R^2/δ^2 and $\sum \xi_i$

low $E(Err_n(h_{SVM}))$ of SVM

Intuition: If the problem can be cast as a TCat-concept with

- high redundancy,
- strongly discriminating features
- particularly in the high frequency region

then linear SVMs achieve a low generalization error [Joachims, 2002].

Assumptions and Restrictions:

- no noise (attribute and classification)
- no variance (only “average” examples)
- only upper bounds, no lower bounds

Part 3: SVM-X?

- common elements of SVMs for other problem
- learning ranking functions from preferences
- novelty and outlier detection
- regression

The Receipe for Cooking an SVMs

Ingredients:

- linear prediction rules $h(\vec{x}) = \vec{w} \cdot \vec{x} + b$
- training problem with objective a la $min \quad w \cdot w + C \sum \xi_i$ and with linear constraints (=> quadratic program)

Stirr and add flavor:

- Classification
- Ranking [Herbrich et al., 2000][Joachims, 2002c]
- Novelty Detection [Schoelkopf et al., 2000]
- Regression [Vapnik, 1998][Smola & Schoelkopf, 1998]

That makes:

- nice SVM with global optimal solution and duality
- often sparse solution (#SVs < n)
- Hint: garnish the dual with kernel to get non-linear prediction rules

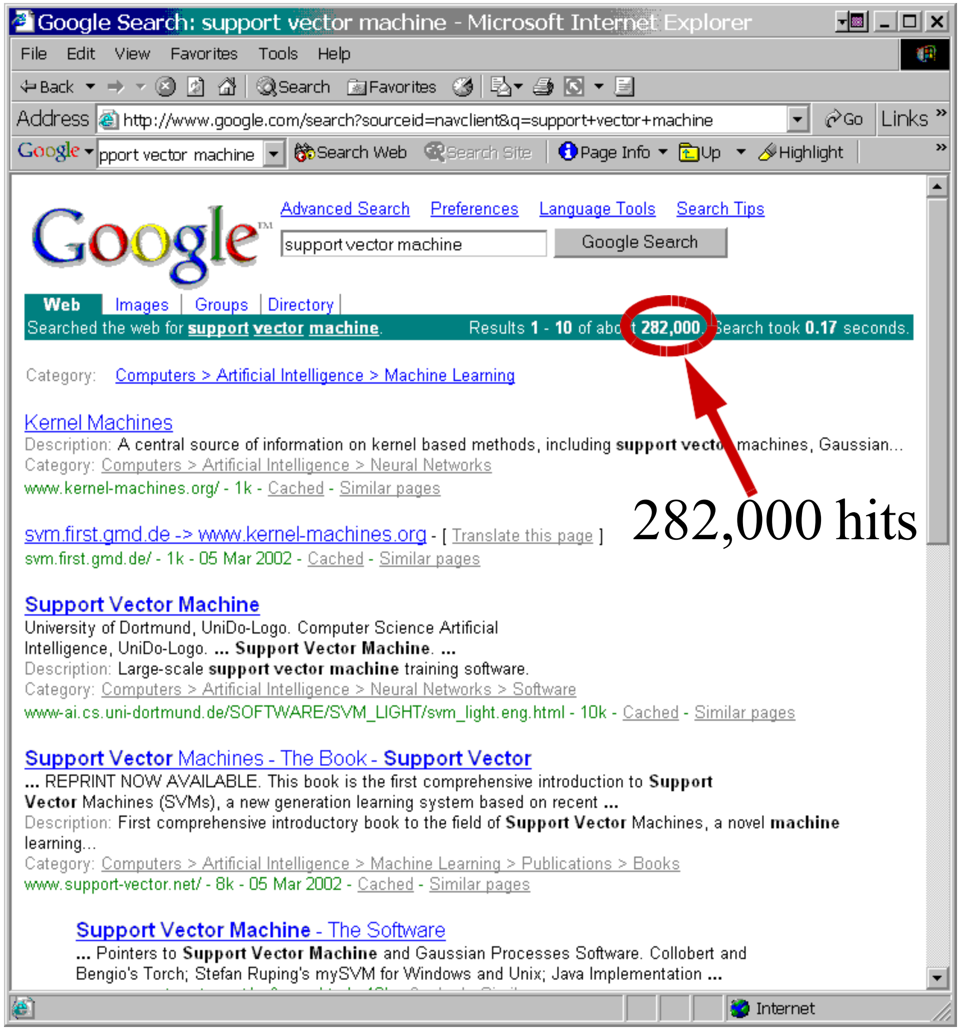
SVM Ranking

Query:

- "Support Vector Machine"

Goal:

- "rank the document I want high in the list"



Searched the web for **support vector machine**. Results **1** - **10** of about **282,000**. Search took **0.17** seconds.

Category: Computers > Artificial Intelligence > Machine Learning

Kernel Machines
Description: A central source of information on kernel based methods, including **support vector** machines, Gaussian...
Category: Computers > Artificial Intelligence > Neural Networks
www.kernel-machines.org/ - 1k - Cached - Similar pages

svm.first.gmd.de -> www.kernel-machines.org - [Translate this page]
svm.first.gmd.de/ - 1k - 05 Mar 2002 - Cached - Similar pages

Support Vector Machine
University of Dortmund, UniDo-Logo. Computer Science Artificial Intelligence, UniDo-Logo. ... **Support Vector Machine**. ...
Description: Large-scale **support vector machine** training software.
Category: Computers > Artificial Intelligence > Neural Networks > Software
www-ai.cs.uni-dortmund.de/SOFTWARE/SVM_LIGHT/svm_light.eng.html - 10k - Cached - Similar pages

Support Vector Machines - The Book - **Support Vector**
... REPRINT NOW AVAILABLE. This book is the first comprehensive introduction to **Support Vector** Machines (SVMs), a new generation learning system based on recent ...
Description: First comprehensive introductory book to the field of **Support Vector** Machines, a novel **machine** learning...
Category: Computers > Artificial Intelligence > Machine Learning > Publications > Books
www.support-vector.net/ - 8k - 05 Mar 2002 - Cached - Similar pages

Support Vector Machine - The Software
... Pointers to **Support Vector Machine** and Gaussian Processes Software. Collobert and Bengio's Torch; Stefan Ruping's mySVM for Windows and Unix; Java Implementation ...

282,000 hits

Training Examples from Clickthrough

Assumption: If a user skips a link *a* and clicks on a link *b* ranked lower, then the user preference reflects *rank(b) < rank(a)*.

Example: *(3 < 2) and (7 < 2), (7 < 4), (7 < 5), (7 < 6)*

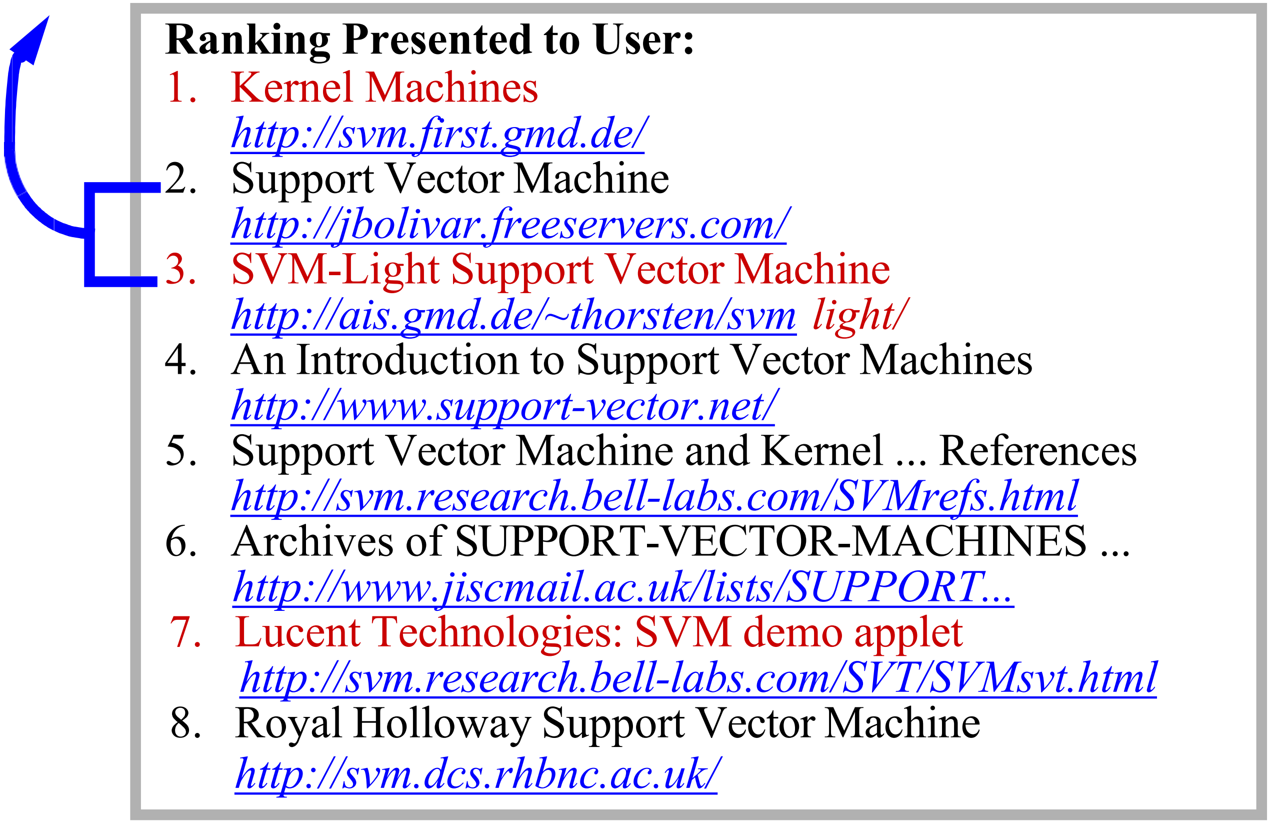
Ranking Presented to User:

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm light/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk/

Training Examples from Clickthrough

Assumption: If a user skips a link *a* and clicks on a link *b* ranked lower, then the user preference reflects *rank(b) < rank(a)*.

Example: *(3 < 2) and (7 < 2), (7 < 4), (7 < 5), (7 < 6)*



Ranking Presented to User:

1. Kernel Machines
   http://svm.first.gmd.de/
2. Support Vector Machine
   http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
   http://ais.gmd.de/~thorsten/svm light/
4. An Introduction to Support Vector Machines
   http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References
   http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES ...
   http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
   http://svm.research.bell-labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
   http://svm.dcs.rhbnc.ac.uk/

Learning to Rank

Assume:

- distribution of queries P(Q)
- distribution of target rankings for query P(R | Q)

Given:

- collection *D* of *m* documents
- i.i.d. training sample $(q_1, r_1), \ldots, (q_n, r_n)$

Design:

- set of ranking functions *F*, with elements f:$Q \rightarrow P^{D \times D}$ (weak ordering)
- loss function $l(r_a, r_b)$
- learning algorithm

Goal:

- find $f^\circ \in F$ with minimal $R_P(f) = \int l(f(q), r) dP(q, r)$

A Loss Function for Rankings

For two orderings r_a and r_b, a pair $d_i \neq d_j$ is

- *concordant*, if r_a and r_b agree in their ordering
  P = number of concordant pairs
- *discordant*, if r_a and r_b disagree in their ordering
  Q = number of discordant pairs

Loss function: [Wong et al., 88], [Cohen et al., 1999], [Crammer & Singer, 01], [Herbrich et al., 98] ...

$$l(r_a, r_b) = Q$$

Example:

$$r_a = (a, c, d, b, e, f, g, h)$$

$$r_b = (a, b, c, d, e, f, g, h)$$

=> discordant pairs *(c,b), (d,b)* => $l(r_a, r_b) = 2$

A Loss Function for Rankings

For two orderings r_a and r_b, a pair $d_i \neq d_j$ is

- *concordant*, if r_a and r_b agree in their ordering
  P = number of concordant pairs
- *discordant*, if r_a and r_b disagree in their ordering
  Q = number of discordant pairs

Loss function: [Wong et al., 88], [Cohen et al., 1999], [Crammer & Singer, 01], [Herbrich et al., 98] ...

$$l(r_a, r_b) = Q$$

Example:

$$r_a = (a, \underbrace{c, d, b}, e, f, g, h)$$

$$r_b = (a, \underbrace{b, c}, d, e, f, g, h)$$

=> discordant pairs *(c,b), (d,b)* => $l(r_a, r_b) = 2$

A Loss Function for Rankings

For two orderings r_a and r_b, a pair $d_i \neq d_j$ is

- *concordant*, if r_a and r_b agree in their ordering
  P = number of concordant pairs
- *discordant*, if r_a and r_b disagree in their ordering
  Q = number of discordant pairs

Loss function: [Wong et al., 88], [Cohen et al., 1999], [Crammer & Singer, 01], [Herbrich et al., 98] ...

$$l(r_a, r_b) = Q$$

Example:

$$r_a = (a, c, \underbrace{d, b}, e, f, g, h)$$

$$r_b = (a, \underbrace{b, c, d}, e, f, g, h)$$

=> discordant pairs *(c,b), (d,b)* => $l(r_a, r_b) = 2$

Interpretation of Loss Function

Notation:

- P concordant pairs
- Q discordant pairs

Kendall's Tau: $r°$ total ordering, uniform sampling of document pairs

$$\tau(r, r°) = \frac{P-Q}{P+Q} = 1 - \frac{2Q}{\binom{m}{2}} = 1 - \frac{2l(r, r°)}{\binom{m}{2}}$$

Average Precision: $r°$ ordering with two ranks

$$AvgPrec(r, r°) \geq \left[l(r, r°) + \binom{R+1}{2} \right]^{-1} \left(\sum_{i=1}^{R} \sqrt{i} \right)^2$$

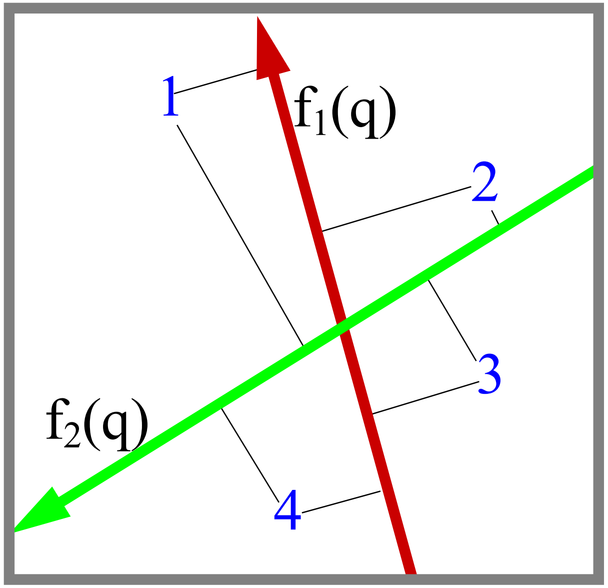
What does the Ranking Function Look Like?

Sort documents d_i by their “retrieval status value” $\mathrm{rsv}(q, d_i)$ with query q [Fuhr, 89]:

$$
\begin{aligned}
\mathrm{rsv}(q, d_i) = \quad & w_1 * \#(\text{of query words in title of } d_i) \\
& + w_2 * \#(\text{of query words in H1 headlines of } d_i) \\
& \ldots \\
& + w_N * \mathrm{PageRank}(d_i) \\
= {} & \vec{w}\, \Phi(q, d_i).
\end{aligned}
$$

Select F as:

$$
\begin{gathered}
d_i > d_j \\
\Leftrightarrow \\
(d_i, d_j) \in f_{\vec{w}}(q) \\
\Leftrightarrow \\
\vec{w}\Phi(q, d_i) > \vec{w}\Phi(q, d_j)
\end{gathered}
$$



Minimizing Training Loss For Linear Ranking Functions

Given:

- training sample $S = (q_1, r_1), \ldots, (q_n, r_n)$

Zero training loss on S:

$$\forall (d_i, d_j) \in r_1; \quad \vec{w}\Phi(q_1, d_i) > \vec{w}\Phi(q_1, d_j)$$

$$\ldots$$

$$\forall (d_i, d_j) \in r_n; \quad \vec{w}\Phi(q_n, d_i) > \vec{w}\Phi(q_n, d_j)$$

Minimize (bound on) training loss (total ordering) on S:

$$min \sum \xi_{l,\, i, j}$$

$$\forall (d_i, d_j) \in r_1; (\vec{w}\Phi(q_1, d_i) \geq \vec{w}\Phi(q_1, d_j) + 1 - \xi_{1,\, i, j})$$

$$\ldots$$

$$\forall (d_i, d_j) \in r_n; (\vec{w}\Phi(q_n, d_i) \geq \vec{w}\Phi(q_n, d_j) + 1 - \xi_{n,\, i, j})$$

Ranking Support Vector Machine

Optimization Problem (primal):

$$min \quad \frac{1}{2}w \cdot w + C\sum \xi_{l,\,i,\,j}$$

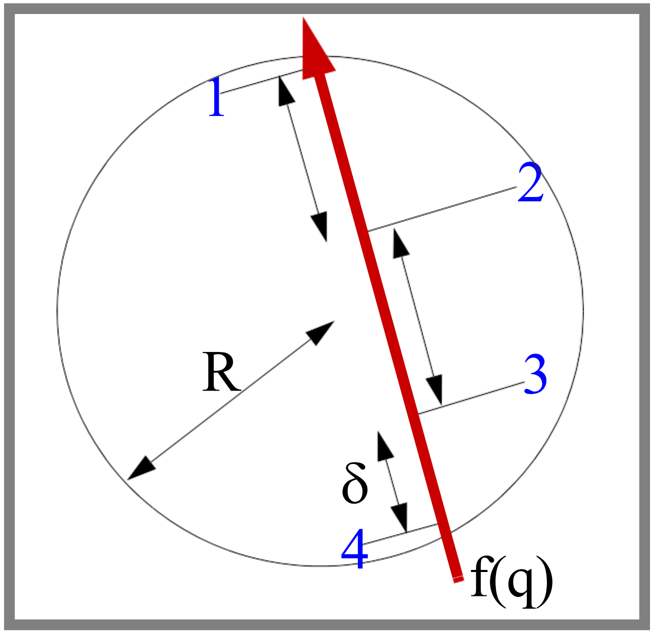
$$\forall(d_i, d_j) \in r_1; (\vec{w}\Phi(q_1, d_i) \geq \vec{w}\Phi(q_1, d_j) + 1 - \xi_{1,\,i,\,j})$$

$$\dots$$

$$\forall(d_i, d_j) \in r_n; (\vec{w}\Phi(q_n, d_i) \geq \vec{w}\Phi(q_n, d_j) + 1 - \xi_{n,\,i,\,j})$$

Properties:

- minimize trade-off between training loss and margin size δ = 1 / ||w||
- quadratic program, similar to classification SVM (=> SVMlight)
- convex => unique global optimum
- radius of ball containing the training points R



How is this different from ...

... classification?

$f_1(q)$: - - + -

$f_2(q)$: - +

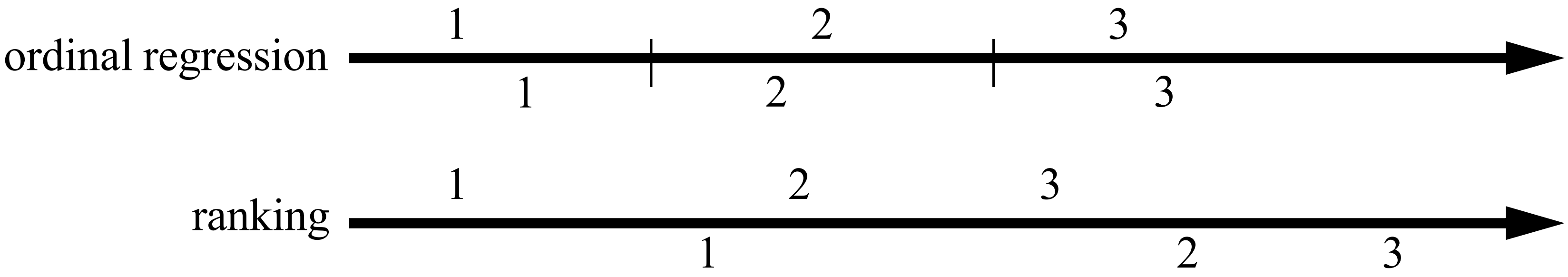
=> both have same error rate (always classify as non-relevant)

=> very different rank loss

... ordinal regression?

Training set $S = (x_1, y_1), \ldots, (x_n, y_n)$, with Y ordinal (and finite)

=> ranks need to be comparable between examples



Experiment Setup

Collected training examples with partial feedback about ordering from

- user skipping links

Ranking Presented to User:

1. Kernel Machines
http://svm.first.gmd.de/
2. Support Vector Machine
http://jbolivar.freeservers.com/
3. SVM-Light Support Vector Machine
http://ais.gmd.de/~thorsten/svm light/
4. An Introduction to Support Vector Machines
http://www.support-vector.net/
5. Support Vector Machine and Kernel ... References
http://svm.research.bell-labs.com/SVMrefs.html
6. Archives of SUPPORT-VECTOR-MACHINES ...
http://www.jiscmail.ac.uk/lists/SUPPORT...
7. Lucent Technologies: SVM demo applet
http://svm.research.bell-labs.com/SVT/SVMsvt.html
8. Royal Holloway Support Vector Machine
http://svm.dcs.rhbnc.ac.uk/

=> $(3 < 2)$ and $(7 < 2)$, $(7 < 4)$, $(7 < 5)$, $(7 < 6)$

- clicked on document should be ranked higher than 50 random documents
=> $S = (q_1, r'_1), \ldots, (q_n, r'_n)$

Query/Document Match Features $\Phi(q,d)$

Rank in other search engine:

- Google, MSNSearch, Altavista, Hotbot, Excite

Query/Content Match:

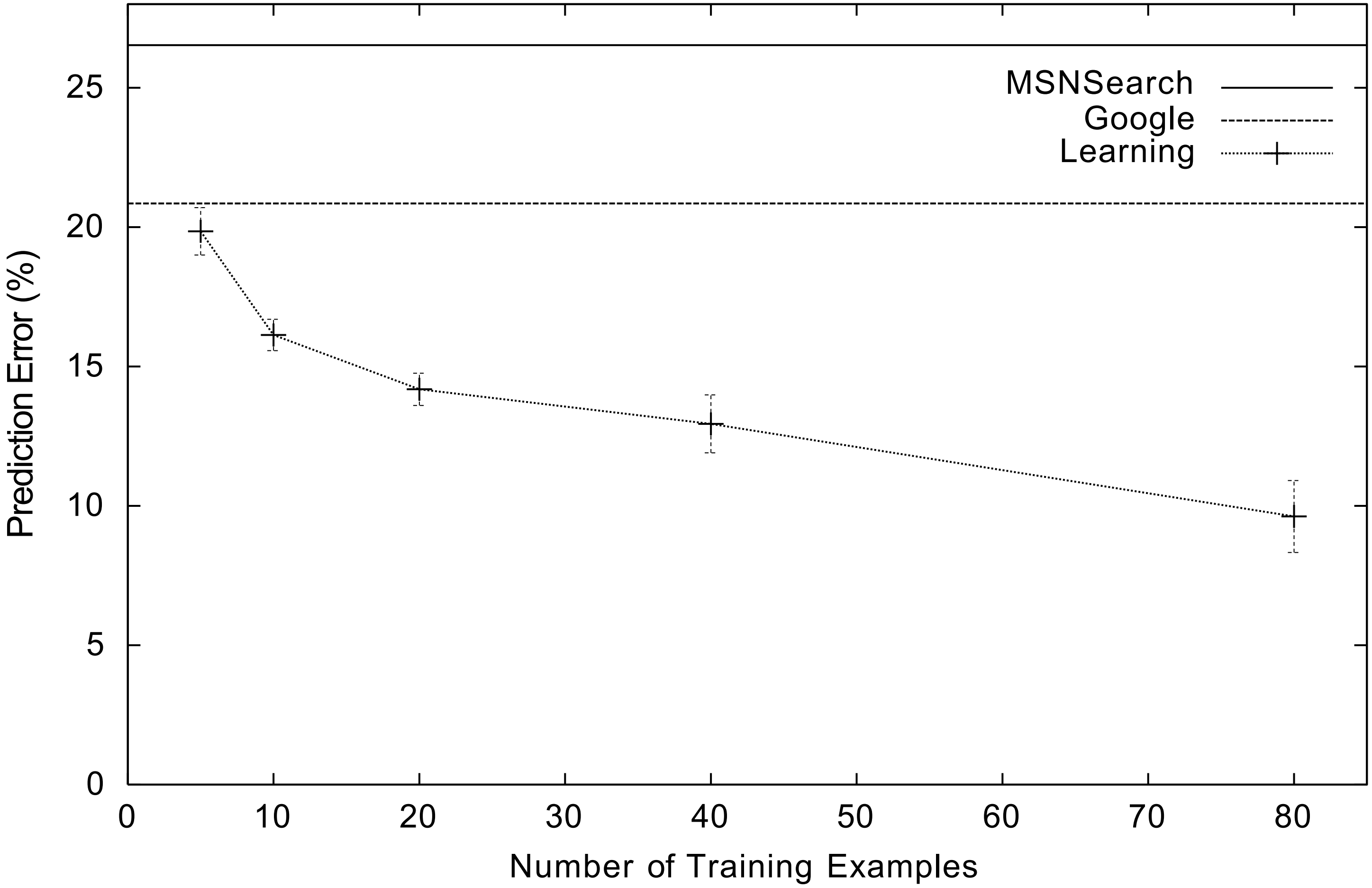
- cosine between URL-words and query
- cosine between title-words and query
- query contains domain-name

Popularity Attributes:

- length of URL in characters
- country code of URL
- domain of URL
- word “home” appears in title
- URL contains “tilde”
- URL as an atom

Experiment I: Learning Curve

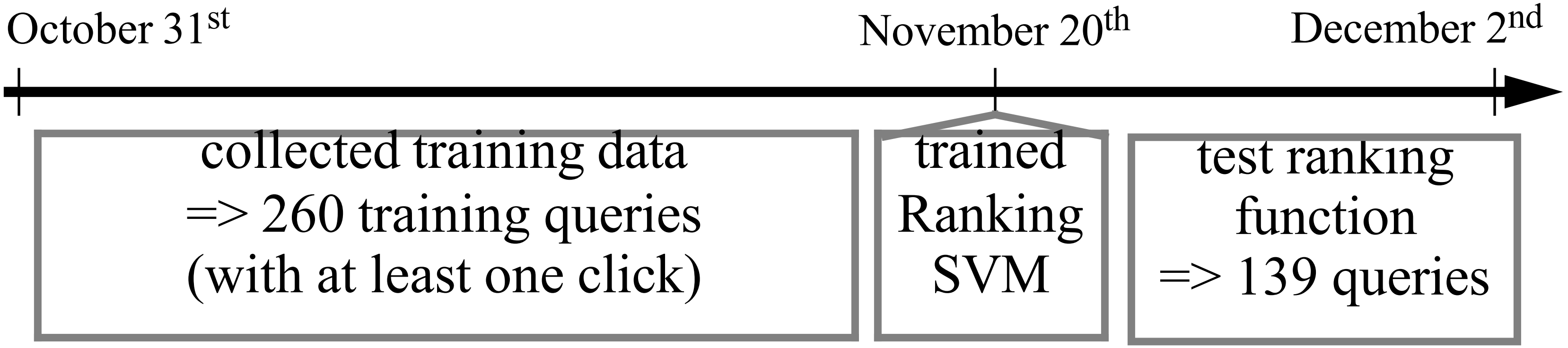
Training examples: preferences from 112 queries



Experiment II

Experiment Setup:

- meta-search engine (Google, MSNSearch, Altavista, Hotbot, Excite)
- approx. 20 users
- machine learning students and researchers from University of Dortmund AI Unit (Prof. Morik)
- asked to use system as any other search engine
- display title and URL of document



October 31st

November 20th

December 2nd

collected training data
=> 260 training queries
(with at least one click)

trained
Ranking
SVM

test ranking
function
=> 139 queries

Experiment: Learning vs. Google/MSNSearch

Ranking A	Ranking B	A better	B better	Tie	Total
Learned	Google	29	13	27	69
Learned	MSNSearch	18	4	7	29
Learned	Toprank	21	9	11	41

~20 users, as of 2nd of December

Toprank: rank by increasing mimium rank over all 5 search engines

=> **Result**: Learned > Google
Learned > MSNSearch
Learned > Toprank

Learned Weights

weight	feature
0.60	cosine between query and abstract
0.48	ranked in top 10 from Google
0.24	cosine between query and the words in the URL
0.24	document was ranked at rank 1 by exactly one of the 5 search engines
...	
0.17	country code of URL is ".de"
0.16	ranked top 1 by HotBot
...	
-0.15	country code of URL is ".fi"
-0.17	length of URL in characters
-0.32	not ranked in top 10 by any of the 5 search engines
-0.38	not ranked top 1 by any of the 5 search engines

Summary: SVM Ranking

- An SVM method for learning ranking functions
- Training examples are rankings
  => pairwise preferences like "A should be ranked higher than B"
- Turn training examples into linear inequality constraints
- Results in quadratic program similar to classification
- Rank new examples by sorting according to distance from hyperplane

Applications:

- personalizing search engines
- tuning retrieval functions in XML intranets
- recommender systems
- betting on horses

SVM Novelty/Outlier Detection

Assume:

- distribution of feature vectors *P(X)*

Goal: [Schölkopf et al., 1995] [Tax & Duin, 2001]

- find the region *R* of $(1 - \varepsilon)$-support for the distribution *P(X)*, i.e.

$$P(x \in R) \geq 1 - \varepsilon$$

- keep the volume of *R* as small as possible

=> new points falling outside of *R* are either outliers, or the distribution must have changed.

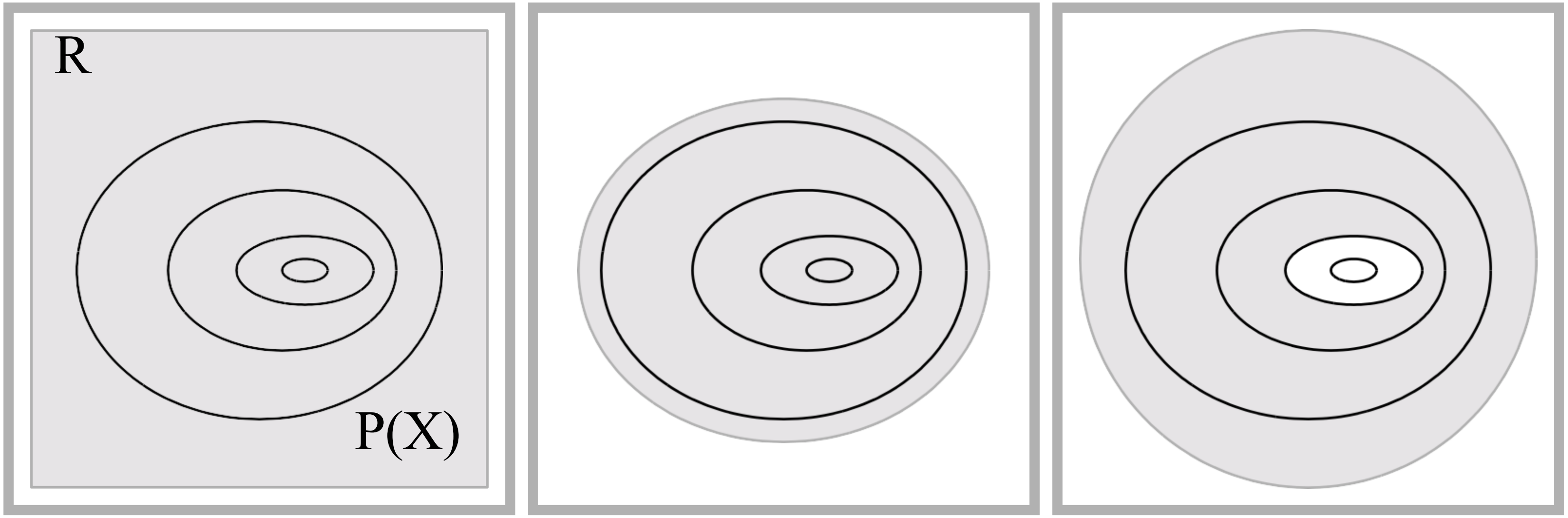
Problem:

- estimate *R* from unlabeled oberservations $x_1, \ldots, x_n$

Example: Small and Large Volume Regions

Assume that we know the distribution *P(X)*.

All following are regions with $P(x \in R) \geq 1 - \varepsilon$:



trivial

optimal

sub-optimal

Find Region using Examples

Problem:

- *P(X)* cannot be observed directly.

Given:

- training oberservations $x_1, \ldots, x_n$ drawn according to *P(X)*.

Approach: [Schoelkopf et al., 1995][Tax & Duin, 2001]

- find smallest ball that includes (most) training observations

Primal:

$$\min \; P(\vec{c}, r, \vec{\xi}) = r^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{s. t.} \quad [\vec{c} - \vec{x}_i]^2 \le r^2 + \xi_i$$

$$\xi_i \ge 0$$

Dual:

$$\max \; D(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i K(\vec{x}_i, \vec{x}_i) - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j)$$

$$\text{s. t.} \quad \sum_{i=1}^{n} \alpha_i = 1 \quad \text{and} \quad 0 \le \alpha_i \le C$$

- $\vec{c}$ is the center of the ball, r is its radius.

Properties of the Primal/Dual

Primal:

$$\min \; P(\vec{c}, r, \vec{\xi}) = r^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{s. t.} \quad [\vec{c} - \vec{x}_i]^2 \le r^2 + \xi_i$$

$$\xi_i \ge 0$$

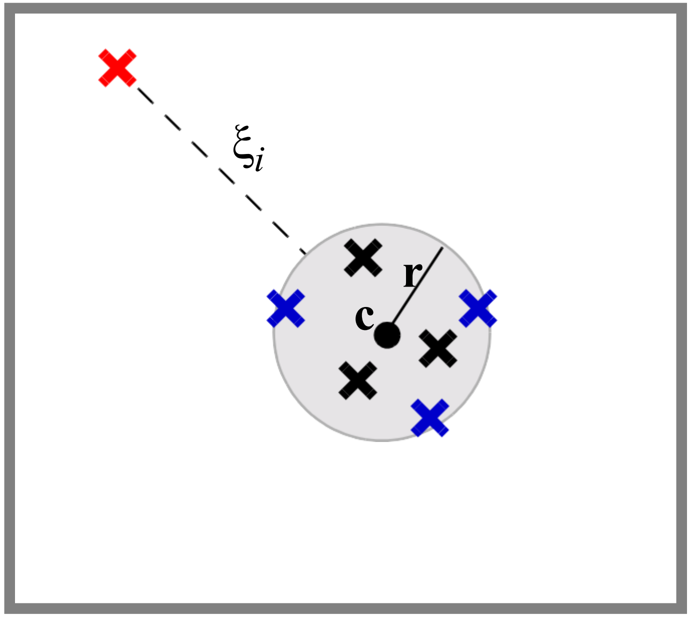
Dual:

$$\max \; D(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i K(\vec{x}_i, \vec{x}_i) - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j)$$

$$\text{s. t.} \quad \sum_{i=1}^{n} \alpha_i = 1 \quad \textit{and} \quad 0 \le \alpha_i \le C$$

Properties:

- convex => global optimum
- ξ_i measures distance from ball
- $\alpha_i = 0$: example lies inside the ball
- $0 < \alpha_i < C$: example on hull of ball
- $\alpha_i = C$: example is training error



One-Class SVM: Separating from the Origin

Observation: [Schölkopf et al., 2000][Schölkopf et al., 2001]

- For kernels depending on the distance between points, the dual is the same as for classification SVM with
  - all training observations in the positive class (with slack)
  - one virtual negative example with $\alpha = -1$ and $K(\vec{x}_i, \vec{x}_j) = 0$.

Dual:

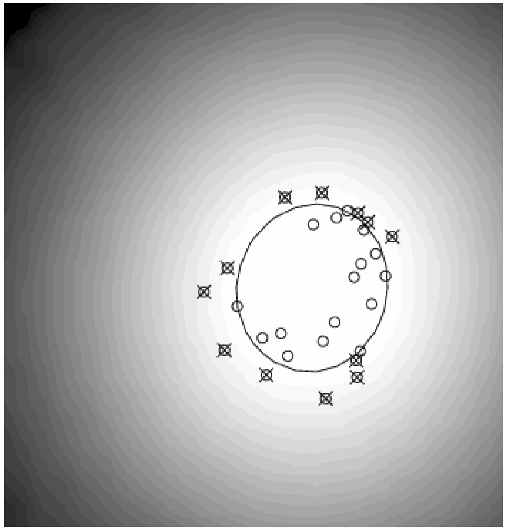
$$\max \; D(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i K(\vec{x}_i, \vec{x}_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j)$$

$$\text{s. t.} \quad \sum_{i=1}^{n} \alpha_i = 1 \quad \textit{and} \quad 0 \le \alpha_i \le C$$

Constant for kernels depending on distance between points (e.g. RBF)

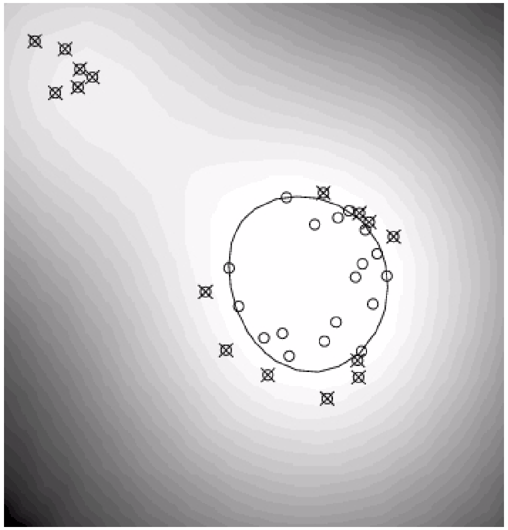
=> Equivalent for RBF kernel $K(\vec{x}_i, \vec{x}_j) = \exp(-|\vec{x}_i - \vec{x}_j|^2 / \sigma^2)$!

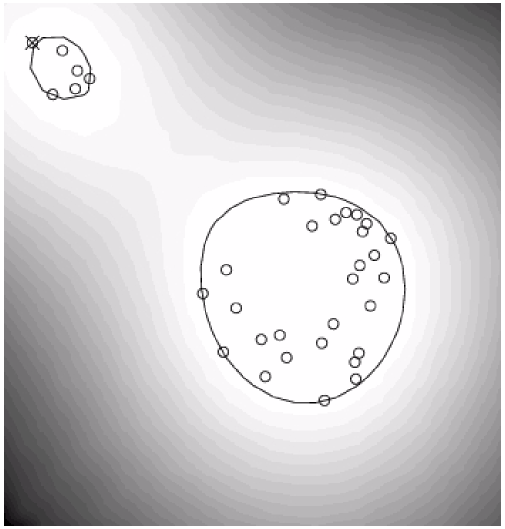
Influence of C and RBF-Width σ^2



small C
large width σ^2
no outliers

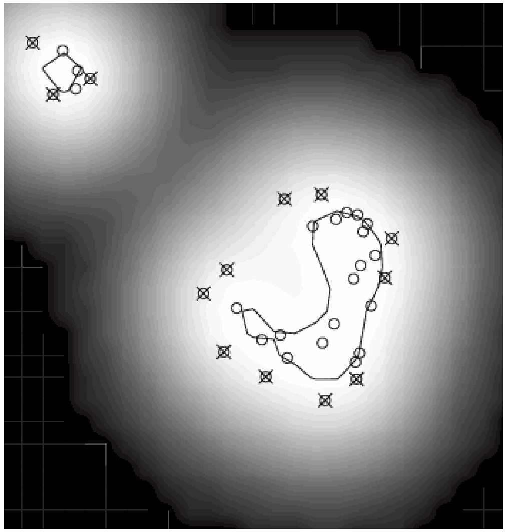
small C
large width σ^2
some ouliers





large C
large width σ^2

small C
small width σ^2



(plots courtesy of B. Schoelkopf)

Summary: SVM Novelty Detection

- Find small region where most observations fall
- One-Class SVM: separate observations from origin
- Outliers (or new observations after shift in distribution) lie outside of region
- Training problem similar to classification SVM

Further work:

- Extension to ν-SVMs and error bounds [Schölkopf et al., 2001] [Schölkopf et al., 2001]
- SVM clustering [Ben-Hur et al., 2001]

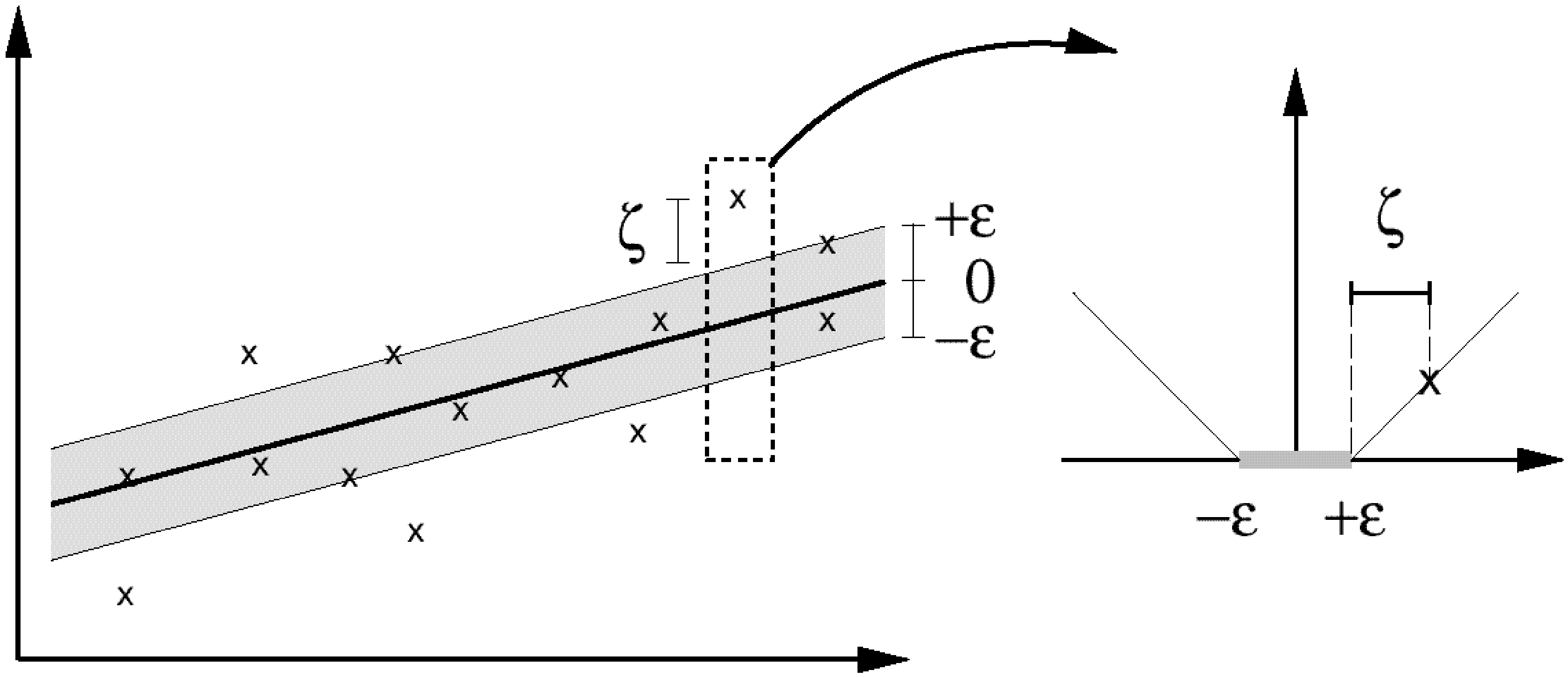
Applications:

- Text classification [Manevitz & Yousef, 2001]
- Topic detection

SVM Regression

Loss function:

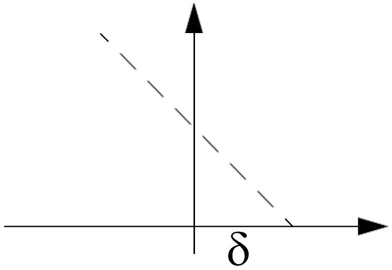
- ε-insensitive region with zero loss
- linear loss beyond the “tube”



Graph taken from [Smola & Schoelkopf, 1998]

Primal SVM Optimization Problems

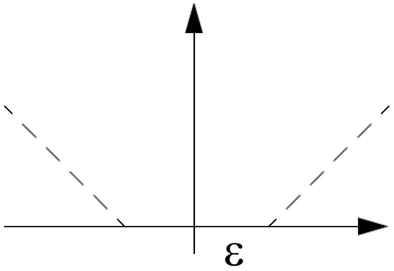
Classification:



$$\text{minimize } J(\vec{w}, b, \vec{\xi}) \;=\; \frac{}{2}\vec{w}\cdot\vec{w} + C\sum_{i=1}^{n}\xi_i$$

$$\text{s. t. } \; y_i[\vec{w}\cdot\vec{x}_i + b] \ge 1 - \xi_i \;\; \mathit{and} \;\; \xi_i \ge 0$$

Regression:



$$\text{minimize } R(\vec{w}, b, \vec{\xi}, \vec{\xi^\circ}) \;=\; \frac{1}{2}\vec{w}\cdot\vec{w} + C\sum_{i=1}^{n}(\xi_i + \vec{\xi^\circ_i})$$

$$\text{s. t. } \; y_i - [\vec{w}\cdot\vec{x}_i + b] \le \varepsilon + \xi_i \;\; \mathit{and} \;\; \xi_i \ge 0$$

$$-y_i + [\vec{w}\cdot\vec{x}_i + b] \le \varepsilon + \xi^\circ{}_i \;\; \mathit{and} \;\; \xi^\circ{}_i \ge 0$$

Dual SVM Optimization Problems

$$\text{maximize} \quad L(\vec{\alpha}) = \left(\sum_{i=1}^{m} p_i\alpha_i\right) - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j u_i u_j K(\vec{v}_i, \vec{v}_j)$$

$$\text{s.t.} \quad \sum_{i=1}^{m}\alpha_i u_i = 0 \qquad \textit{and} \qquad 0 \le \alpha_i \le C$$

Classification: $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(\vec{x}, y) \qquad \vec{x}_i \in \Re^N \quad y_i \in \{1,-1\} \qquad m = n$

- $p_i = 1$ for $1 \le i \le n$
- $u_i = y_i$ for $1 \le i \le n$
- $\vec{v}_i = \vec{x}_i$ for $1 \le i \le n$

Regression: $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \sim P(\vec{x}, y) \qquad \vec{x}_i \in \Re^N \quad y_i \in \Re \qquad m = 2n$

- $p_i = \varepsilon + y_i$ for $1 \le i \le n$ and $p_i = \varepsilon - y_i$ for $n + 1 \le i \le 2n$
- $u_i = 1$ for $1 \le i \le n$ and $u_i = -1$ for $n + 1 \le i \le 2n$
- $\vec{v}_i = \vec{x}_i$ for $1 \le i \le n$ and $v_i = x_i$ for $n + 1 \le i \le 2n$

Conclusions

- What! How! Why! When! ...and that SVMs solve any other problem!

Info

- Chris Burges' tutorial (Classification)
  http://www.kernel-machines.org/papers/Burges98.ps.gz
- Smola & Schölkopf's tutorial (Regression)
  http://www.kernel-machines.org/papers/tr-30-1998.ps.gz
- Cristianini & Shawe-Taylor book: Introduction to SVMs, Cambridge University Press, 2000.
- Schölkopf' & Smola book: Learning with Kernels, MIT Press, 2002.
- My dissertation: Learning to Classify Text Using Support Vector Machines, Kluwer.
- Software: SVM*light* for Classification, Regression, and Ranking
  http://svmlight.joachims.org/
- General: http://www.kernel-machines.org

Support Vector and Kernel Methods

Thorsten Joachims
Cornell University Computer Science Department
tj@cs.cornell.edu http://www.joachims.org

Bibliography

- [Ben-Hur et al., 2001] Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V. (2001). Support Vector Clustering. Journal of Machine Learning Research, 2:125-137.
- [Bennet & Demiriz, 1999] K. Bennet and A. Demiriz (1999). Semi-supervised support vector machines, Advances in Neural Information Processing Systems, 12, pages 368-374, MIT Press.
- [Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A traininig algorithm for optimal margin classifiers. In Haussler, D., editor, Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, pages 144-152.
- [Burges, 1998] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Knowledge Discovery and Data Mining, 2(2), 1998.

- [Chapelle et al., 2002] Chapelle, O., V. Vapnik, O. Bousquet and S. Mukherjee (2002). Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46(1), pages 131-159.
- [Cohen et al., 1999] W. Cohen, R. Shapire, and Y. Singer (1999). Learning to order things. Journal of Artificial Intelligence Research, 10:243-270.
- [Collins & Duffy, 2001] Collins, M., and Duffy, N. (2001). Convolution kernels for natural language. Proceedings of Advances in Neural Information Processing Systems (NIPS).
- [Cortes & Vapnik, 1995] Cortes, C. and Vapnik, V. N. (1995). Support-vector networks. Machine Learning Journal, 20:273-297.
- [Crammer & Singer, 2001] K. Crammer and Y. Singer (2001). On the algorithmic implementation of multiclass kernel-based vector machines, Journal of Machine Learning Research, 2:265-292.
- [Cristianini & Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press.

- [Dumais et al., 1998] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In Proceedings of ACM-CIKM98.
- [Gaertner et al., 2002] Gaertner, T., Lloyd, J., Flach, P. (2002). Kernels for structured data. Twelfth International Conference on Inductive Logic Programming (ILP).
- [Herbrich et al., 2000] R. Herbrich, T. Graepel, K. Obermayer (2000). Large Margin Rank Boundaries for Ordinal Regression, In Advances in Large Margin Classifiers, pages 115-132, MIT Press.
- [Jaakkola, and Haussler, 1998] Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In Advances in Neural Information Processing Systems 11, MIT Press.
- [Jaakkola and Haussler, 1999] Jaakkola, T. and Haussler, D. (1999). Probabilistic kernel regression models. In Conference on AI and Statistics.
- [Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In

Proceedings of the European Conference on Machine Learning, pages 137 - 142, Berlin. Springer.

- [Joachims, 1999a] Joachims, T. (1999a). Aktuelles Schlagwort: Support Vector Machines. Künstliche Intelligenz, 4.
- [Joachims, 1999b] Joachims, T. (1999b). Making Large-Scale SVM Learning Practical. In Schölkopf, B., Burges, C., and Smola, A., editors, Advances in Kernel Methods - Support Vector Learning, pages 169 - 184. MIT Press, Cambridge.
- [Joachims, 1999c] Joachims, T. (1999c) Transductive Inference for Text Classification using Support Vector Machines. International Conference on Machine Learning (ICML).
- [Joachims, 2002] Joachims, T. (2002). Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms, Kluwer.
- [Joachims, 2002c] Joachims, T. (2002). Optimizing Search Engines using Clickthrough Data, Conference on Knowledge Discovery and Data Mining (KDD), ACM.

- [Keerthi et al., 1999] Keerthi, S., Shevade, S., Bhattacharyya, C., and Murthy, K. (1999). A fast iterative nearest point algorithm for support vector machine classifier design. Technical Report TR-ISL-99-03, Indian Institute of Science, Bangalore, India.
- [Kindermann & Paass, 2000] Kindermann, Jörg and Paass, Gerhard: Multiclass Classification with Error Correcting Codes, FGML, 2000.
- [Kondor & Lafferty, 2002] Kondor, I., and Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces, International Conference on Machine Learning (ICML).
- [Lodhi et al., 2002] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text Classification using String Kernels. Journal of Machine Learning Research, 2:419-444.
- [MacKay, 1997] D.J.C. MacKay. Introduction to gaussian processes, 1997. Tutorial at ICANN'97. ftp://wol.ra.phy.cam.ac.uk/pub/mackay/gpB.ps.gz

- [Manevitz & Yousef, 2001] L. Manevitz and M. Yousef (2001). One-Class SVMs for Document Classification, Journal of Machine Learning Research, 2:139-154.
- [Mangasarian and Musicant, 2000] Mangasarian, O. L. and Musicant, D. R. (2000). Active support vector machine classification. Technical Report 00-04, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. ftp://ftp.cs.wisc.edu/ pub/dmi/tech-reports/00-04.ps.
- [Osuna et al., 1997] Osuna, E., Freund, R., and Girosi, F. (1997b). Training support vector machines: An application to face detection. In Proceedings CVPR'97.
- [Platt, 1999] Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C., and Smola, A., editors, Advances in Kernel Methods - Support Vector Learning, chapter 12. MIT-Press.

- [Platt et al., 2000] Platt, J., Cristianini, N., and Shawe-Taylor, J. (2000). Large margin dags for multiclass classification. In Advances in Neural Information Processing Systems 12.
- [Schölkopf et al., 1995] Schölkopf, B., Burges, C., Vapnik, V. (1995). Extracting support data for a given task. Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD).
- [Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Mueller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10:1299-1319.
- [Schölkopf et al., 2000] Schölkopf, B., R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt (2000). Support vector method for novelty detection. In S. Solla, T. Leen, and K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, pp. 582-588. MIT Press.

- [Schölkopf et al., 2001] Schölkopf, B., J. Platt, J. Shawe-Taylor, A.J. Smola and R.C. Williamson (2001). Estimating the support of a high-dimensional distribution. Neural Computation 13(7);1443-1471.
- [Schölkopf & Smola, 2002] Schölkopf, B., Smola, A. (2002). Learning with Kernels, MIT Press.
- [Shawe-Taylor et al., 1996] Shawe-Taylor, J., Bartlett, P., Williamson, R., and Anthony, M. (1996). Structural risk minimization over data-dependent hierarchies. Technical Report NC-TR-96-053, NeuroCOLT.
- [Smola & Schölkopf, 1998] A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College of London, UK, 1998.
- [Tax & Duin, 1999] D. Tax and R. Duin (1999). Support vector domain description. Patter Recognition Letters, 20:1991-1999.
- [Tax & Duin, 2001] D. Tax and R. Duin (2001). Uniform Object Generation for Optimizing One-Class Classifiers, Journal of Machine Learning Research, 2:155-173.

- [Vapnik, 1998] Vapnik, V. (1998). Statistical Learning Theory. Wiley, Chichester, GB.
- [Vapnik & Chapelle, 2000] V. Vapnik and O. Chapelle. Bounds on Error Expectation for Support Vector Machines. In Neural Computation, 2000, vol 12, 9.
- [Wapnik & Tscherwonenkis, 1979] Wapnik, W. and Tscherwonenkis, A. (1979). Theorie der Zeichenerkennung. Akademie Verlag, Berlin.
- [Weston & Watkins, 1998] Weston, J. and Watkins, C. (1998). Multi-class support vector machines. Technical Report CSD-TR-98-04, Royal Holloway University of London.