SVM example: cancer classification

Support Vector Machines

1. Cancer genomics: TCGA

The cancer genome atlas (TCGA) will provide high-quality cancer data for large scale analysis by many groups:

SVM example: cancer classification

National Cancer Institute | National Human Genome Research Institute

THE CANCER GENOME ATLAS

Search [ ] GO

About TCGA | Program Components | Policies | Media Center | Launch Data Portal

Mission and Goal

The Cancer Genome Atlas (TCGA) is a comprehensive and coordinated effort to accelerate our understanding of the molecular basis of cancer through the application of genome analysis technologies, including large-scale genome sequencing.

Learn more >>

News from the Pilot Project

NEW* **NCI Announces New Funding to Support TCGA**

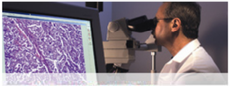
The National Cancer Institute (NCI) has announced a new funding opportunity to support TCGA. This funding opportunity announcement (FOA) is soliciting applications for Genome Characterizations Centers and Genome Data Analysis Centers. Presentations from the pre-application meeting held on January 29, 2009, are available for all interested prospective applicants to download.

Learn more >>

The Cancer Genome Atlas Reports First Results of Comprehensive Study of Brain Tumors: Large-Scale Effort Identifies New Genetic Mutations, Core Pathways

The Cancer Genome Atlas Research Network reported the first results of its large-scale, comprehensive study of the most common form of brain cancer, glioblastoma (GBM) in the Sept. 4, 2008 advance online edition of the journal *Nature*. Among the TCGA findings are the identification of many

TCGA Data Portal



Access TCGA Data Portal

View the phase two list of targets to be sequenced in glioblastoma multiforme (GBM)

TCGA: How Will It Work?

Click here for more information

Featured Articles

Comprehensive genomic characterization defines human glioblastoma genes and core pathways

TCGA Research Network

SVM example: cancer classification

About TCGA Data | Portal Help | Data Access | Browse Data | Analyze TCGA Data

Overview | Types of Data

TCGA Data Portal

Welcome to The Cancer Genome Atlas (TCGA) Data Portal.

TCGA Data Portal provides a platform for researchers to search, download, and analyze data sets generated by TCGA. This portal contains all TCGA data pertaining to clinical information associated with cancer tumors and human subjects, genomic characterization, and high-throughput sequencing analysis of the tumor genomes.

New data is derived on an ongoing basis from TCGA analyses and is deposited into databases. The Data Portal offers access to download these data sets.

Click here to access and download TCGA data.

In addition, the Cancer Molecular Analysis Portal provides the ability for researchers to use analytical tools designed to integrate, visualize, and explore genome characterization from TCGA data.

TCGA Data Portal

Application Help

For more information about how to search the Data Portal for TCGA data, click here.

TCGA Updates

Click here to read more about the latest progress of TCGA pilot project.

View the phase two list of targets to be sequenced in glioblastoma multiforme (GBM).

For more information about initiatives related to TCGA, click here.

Click here to learn more about

SVM example: cancer classification

About TCGA Data | Portal Help | Data Access | Browse Data | Analyze TCGA Data

▼ Get TCGA Data

The **Data Access Matrix** allows you to select results of individual samples from multiple centers, platforms and data types, thereby creating a custom archive with your customized data. Simply choose the disease type and data type(s) you would like to work with and proceed to the Data Access Matrix.

Disease Type

GBM – Glioblastoma multiforme

Data Types

- All
- Clinical
- Copy Number Results
- DNA Methylation
- Expression-Exon
- Expression-Genes
- Expression-miRNA
- SNP

Go to the Data Access Matrix

Alternatively, you can **search by archive** to search for and download complete data archives as submitted by the TCGA research centers.

If you prefer to access the downloads directly you may do so from either FTP (open access) or SFTP (controlled access).

TCGA Sample Counts

	CN			Methyl		Exp-Exon			Exp-Gene			Exp-miRNA			SNP		
	L1	L2	L3	L1	L2	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3
GBM	458	460	460	29	247	249	249	232	287	287	287	279	250	250	471	470	470
OV	159	159	159	86	86				49	49	49				86	86	86

▼ TCGA Related Resources

- **GBM Publication Site**
- **Somatic Mutation Data**
- **Analytical Views of TCGA data**
- **Sequence Data from NCBI Trace Archive**
- **TCGA-Data Listserv**

DCC Resources:

BCR Biospecimen Barcodes Table

Sample-to-file Association Matrix

+ Portal News

01/29/09 - Public Clinical Data File

All current public GBM clinical data is available in tab-delimited format here.

10/03/08 - Tier 1 Clinical Data Spreadsheet

The Tier 1 Clinical Data as of the 10/01/08 update of the BCR Data is available here

09/09/08 - GBM Publication Data Freeze

A list of the archives that comprise the GBM Publication Data Freeze is available here.

09/04/08 - TCGA Reports First Results

In a paper published Sept. 4, 2008, in the advance online edition of the journal Nature, the TCGA team describes the discovery of new genetic mutations and other types of DNA alterations with potential implications for the diagnosis and treatment of

... Show More

SVM example: cancer classification

2. Example: cancer classification

Source: T. Furey, N. Cristianini, et al. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics* **16**, 906-914.

Consider a set of 40 samples of colon cancer tissue, and 22 samples of normal colon tissue (62 all together).

SVM example: cancer classification

For each sample s compute

$$\mathbf{x} = (x_1, \ldots, x_d) = \text{microarray profile of sample } s$$

Let

$$D = \{\mathbf{x}_i, y_i\}_{i=1}^{62}$$

be collection of samples and correct classifications:

$$y_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ cancerous} \\ -1 & \text{if } \mathbf{x}_i \text{ non-cancerous} \end{cases}.$$

We want function $f(\mathbf{x}) = y$ which for a *new (test)* sample $\mathbf{x}$ predicts its $y = \pm 1$.

SVM example: cancer classification

Note the set of all possible **x** $= (x_1, \ldots, x_d)$ of microarray profiles is

$$\mathbb{R}^d = F = \textit{feature space}$$

We denote

$$\mathbf{x} = \textit{feature vector} \in F$$

With the data set D, can we find the right function f: $F \to \mathcal{B}$ which generalizes the above examples, so that $f(\mathbf{x}) = y$ for all feature vectors?

SVM example: cancer classification

Easier: find a f for which

$$f(\mathbf{x}) > 0 \text{ if } y = 1; \;\; f(\mathbf{x}) < 0 \;\text{ if }\; y = -1$$

(and $f(\mathbf{x})$ >> 1 indicates we are more certain $y = 1$).

Loss function

4. Error function

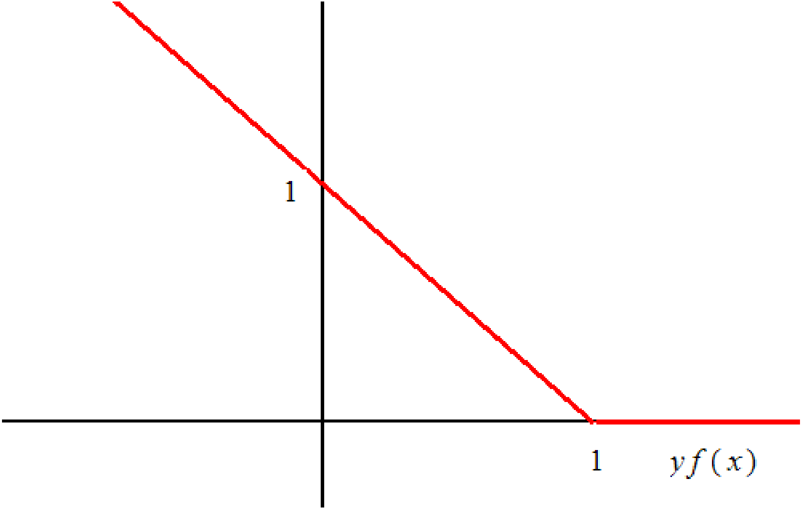
Consider the error measure: we want $f(\mathbf{x}) > 0$ whenever $y = 1$ and want $f(\mathbf{x}) < 0$ whenever $y = -1$

Measure the error (or penalty) for bad choice of y by

$$V(f(\mathbf{x}), y) = (1 - yf(\mathbf{x}))_+ \equiv \max(1 - yf(\mathbf{x}), 0).$$

$$= \begin{cases} \text{small} & \text{if } y, f(\mathbf{x}) \text{ have same sign} \\ \text{large} & \text{otherwise} \end{cases} .$$

Loss function



This is the *hinge error function*.

Loss function

Notice a *margin* is built in: error is 0 only if $yf(\mathbf{x}) \geq 1$ (more stringent requirement than just $yf(\mathbf{x}) \geq 0$)

Thus data-based error (penalty) is

$$e_d = \frac{1}{n}\sum_{j=1}^{n} V(f(\mathbf{x}_j), y_j)$$

Not enough to determine f! As usual need *a priori* (prior) information.

What other information do we have?

Loss function

Note surface H: $f(\mathbf{x}) = 0$ will separate "positive" **x** with $f(\mathbf{x}) > 0$, and "negative" **x** with $f(\mathbf{x}) < 0$:

Loss function

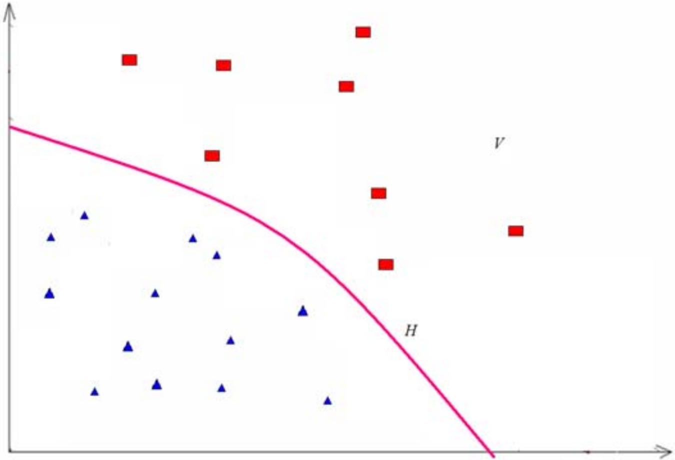


Fig. 1. Red points have $y =$ +1 and blue have $y = -1$ in space F. H: $f(\mathbf{x}) = 0$ is separating surface.

Loss function

Additional information: introduce penalty (loss) functional $L(f)$ which is large when f is 'bad'.

E.G., bad maybe non-smooth, etc.

Form of $L(f)$: assume $f(\mathbf{x})$ is allowed to range over collection $\mathcal{H}$ of functions.

Assume form of $\mathcal{H}$ is an RKHS. Thus e.g.

$$L(f) = \|f\|_K^2.$$

Will specify desirable norm $\|\cdot\|_K$ later -- but for now:

Loss function

Solve regularization problem for the above norm and loss V:

$$f_0 = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (1 - y_j f(\mathbf{x}_j))_+ + \lambda \|f\|_K^2. \quad (1)$$

5. Finding f: Introduction of slack variables

Define new variables ξ_j

Note if we find the min over $f \in \mathcal{H}$ and ξ_j of

$$\underset{f \in \mathcal{H}, \xi_j}{\arg\min} \frac{1}{n} \sum_{j=1}^{n} \xi_j + \lambda \| f \|_K^2 \tag{1a}$$

with the constraint

Slack variables

$$y_j f(\mathbf{x}_j) \geq 1 - \xi_j$$

$$\xi_j \geq 0,$$

we get the same solution f.

To see this, note the constraints are

$$\xi_j \geq \max\left(0, 1 - y_j f(\mathbf{x}_j)\right) = \left(1 - y_j f(\mathbf{x}_j)\right)_+, \tag{1b}$$

which yields the claim. (Clearly in fact in minimizing sum we will end up with $\xi_j = (1 - y_j f(\mathbf{x}_j))_+$).

Solving SVM

Summary: the f which minimizes

$$f = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (1 - y_j f(\mathbf{x}_j))_+ + \lambda \|f\|_K^2. \tag{1}$$

is given by the *quadratic programming* solution:

$$f(\mathbf{x}) = \sum_{j=1}^{n} a_j K(\mathbf{x}, \mathbf{x}_j) + b. \tag{4}$$

We find $\mathbf{a} = [a_1, \ldots, a_n]^T$ from

$$a_j = \overline{\alpha}_j y_j.$$

Solving SVM

Here vector $\overline{\alpha} = (\overline{\alpha}_1, \ldots, \overline{\alpha}_n)$ is defined by

$$\overline{\alpha} = \arg\min_{\overline{\alpha}} \sum_{j=1}^{n} \overline{\alpha}_j - \frac{1}{2}\overline{\boldsymbol{\alpha}}^T P \overline{\boldsymbol{\alpha}} \tag{9}$$

with constraints

$$0 \leq \overline{\boldsymbol{\alpha}} \leq \frac{1}{2\lambda n}; \;\; \overline{\boldsymbol{\alpha}} \cdot \mathbf{y} = 0$$

Solving SVM

We define

$\mathbf{y} = (y_1, \ldots, y_n) = D =$ classifications of known samples,

$$P = \mathbf{YKY}^T,$$

and

$$\mathbf{K} = (\mathbf{K}_{ij}) = K(\mathbf{x}_i, \mathbf{x}_j)$$

with $\mathbf{x}_i = i^{th}$ sample (e.g. microarray).

Solving SVM

Finally, to find b, must plug into original optimization problem: that is, we minimize with respect to b

$$\frac{1}{n}\sum_{j=1}^{n}(1 - y_j f(\mathbf{x}_j))_+ + \lambda\|f\|_K^2$$

$$= \frac{1}{n}\sum_{j=1}^{n}\left(1 - y_j\left[\sum_{i=1}^{n} a_i K(\mathbf{x}_j, \mathbf{x}_i) + b\right]\right)_+ + \lambda \mathbf{a}^T \mathbf{K} \mathbf{a}$$

after finding **a.**

2. The RKHS for support vector machine

General SVM: solution function is (see (4) above)

$$f(\mathbf{x}) = \sum_j a_j K(\mathbf{x}, \mathbf{x}_j) + b,$$

with sol'n for a_j given by quadratic programming as above.

A simple case (linear kernel):

$$K(\mathbf{x}, \mathbf{x}_j) = \mathbf{x} \cdot \mathbf{x}_j.$$

Then we have

Right RKHS for SVM

$$f(\mathbf{x}) = \sum_j (a_j \mathbf{x}_j) \cdot \mathbf{x} + b \equiv \mathbf{w} \cdot \mathbf{x} + b,$$

where

$$\mathbf{w} \equiv \sum_j a_j \mathbf{x}_j. \qquad (10)$$

What class of RKHS $\mathcal{H}$ does this correspond to? Claim the set of linear functions of $\mathbf{x}$

$$\mathcal{H} = \{\mathbf{w} \cdot \mathbf{x} | \mathbf{w} \in \mathbb{R}^d\}$$

with inner product

Right RKHS for SVM

$$\langle \mathbf{w}_1 \cdot \mathbf{x}, \mathbf{w}_2 \cdot \mathbf{x} \rangle = \mathbf{w}_1 \cdot \mathbf{w}_2$$

is the RKHS of $K(\mathbf{x}, \mathbf{y})$ above.

Right RKHS for SVM

Thus matrix $\mathbf{K}_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$, and we find the optimal separator

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$$

by choosing **w** as in (10).

Note add b to $f(\mathbf{x})$ (as earlier), so have all separator functions $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$.

Right RKHS for SVM

Note above inner product gives the norm

$$\|f(\mathbf{x})\|_{\mathcal{H}}^2 = \|\mathbf{w}\cdot\mathbf{x}\|_{\mathcal{H}}^2 = |\mathbf{w}|_{\mathbb{R}^n}^2 = \sum_{j=1}^{n} w_j^2.$$

Why use this norm? A priori information content.

Final classification rule:

$$f(\mathbf{x}) > 0 \Rightarrow y = 1;$$

$$f(\mathbf{x}) < 0 \Rightarrow y = -1.$$

Right RKHS for SVM

Learning from training data:

$$Df = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) = (y_1, \ldots, y_n).$$

Thus can show RKHS here is

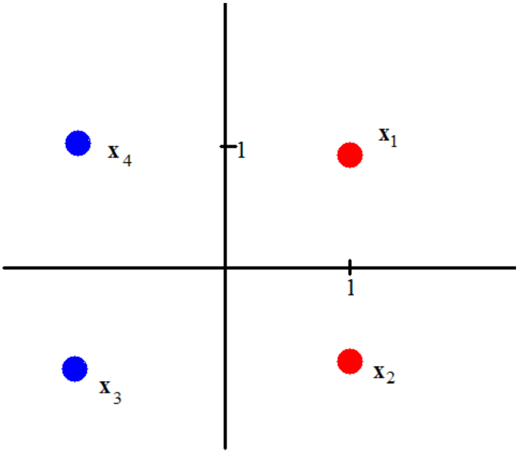
$$\mathcal{H} = \{f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} : \mathbf{w} \in \mathbb{R}^n\}$$

is set of linear separator functions (known as *perceptrons* in neural network theory).

Consider separating hyperplane $H : f(\mathbf{x}) = 0$:

Toy example

3. Toy example:



Toy example

Information

$$Df = \{[(1,1),1],\ [(1,-1),1],[(-1,1),-1],[(-1,-1),-1]\}$$

(red $=$ +1; blue $= -1$);

$$f = \mathbf{w} \cdot \mathbf{x} + b$$

$$= \sum_i a_i \underbrace{(\mathbf{x}_i \cdot \mathbf{x})}_{K(\mathbf{x}_i, \mathbf{x})} + b$$

Toy example

so

$$\mathbf{w} = \sum_i a_i \mathbf{x}_i.$$

Recall $\|f\|^2_{\mathcal{H}} = |\mathbf{w}|^2$, so

$$L(f) = \frac{1}{4}\sum_j (1 - f(\mathbf{x}_j)y_j)_+ + \frac{1}{2}|\mathbf{w}|^2$$

($\lambda = 1/2;$ minimize wrt $\mathbf{w}$, b).

Toy example

Equivalent:

$$L(f) = \frac{1}{4}\sum_{j=1}^{4}\xi_j + \frac{1}{2}|\mathbf{w}|^2$$

$$y_j f(\mathbf{x}_j) \geq 1 - \xi_j; \qquad \xi_j \geq 0.$$

[Note effectively $\xi_i = (1 - (\mathbf{w} \cdot \mathbf{x}_i + b)y_i)_+$]

Toy example

Define kernel matrix

$$\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 \end{bmatrix}$$

$$\|f\|_{\mathcal{H}} = |\mathbf{w}|^2 = \mathbf{a}^T \mathbf{K} \mathbf{a} = 2\left(\sum_{i=1}^{4} a_i^2\right) - 4(a_1 a_3 + a_2 a_4).$$

Toy example

where **a** $= \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix}$.

Toy example

Solution has (see (8a) above)

$$\boldsymbol{\alpha} = 2\lambda Y^{-1}\mathbf{a} = Y^{-1}\mathbf{a}$$

$$\left(\text{recall } \mathbf{Y} = \begin{bmatrix} y_1 & 0 & \dots & 0 \\ 0 & y_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & y_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\right)$$

Toy example

and (8a above)

$$\overline{\boldsymbol{\alpha}} = \frac{1}{2\lambda}\boldsymbol{\alpha} = \boldsymbol{\alpha}.$$

Finally optimize (8)

$$\sum_{j=1}^{4} \overline{\alpha}_j - \frac{1}{2}\overline{\boldsymbol{\alpha}}^T P \overline{\boldsymbol{\alpha}},$$

where

Toy example

$$\mathbf{P} = \mathbf{YKY}^T$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ -2 & 0 & -2 & 0 \\ 0 & -2 & 0 & -2 \end{bmatrix}$$

Toy example

$$= \begin{bmatrix} 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \end{bmatrix}.$$

Toy example

constraints are

$$0 \le \overline{\alpha}_j \le C \equiv \frac{1}{2\lambda n} = \frac{1}{4}. \tag{11}$$

$$0 = \overline{\alpha} \cdot \mathbf{y} = \overline{\alpha}_1 + \overline{\alpha}_2 - \overline{\alpha}_3 - \overline{\alpha}_4.$$

Toy example

Thus optimize

$$\mathcal{L}_1 = \sum_{j=1}^{4} \overline{\alpha}_j - \left(\sum_{j=1}^{4} \overline{\alpha}_j^2 + 2\overline{\alpha}_1\overline{\alpha}_3 + 2\overline{\alpha}_2\overline{\alpha}_4 \right)$$

$$= \sum_{i=1}^{4} \overline{\alpha}_i - (\overline{\alpha}_1 + \overline{\alpha}_3)^2 - (\overline{\alpha}_2 + \overline{\alpha}_4)^2.$$

$$= u + v - u^2 - v^2,$$

Toy example

where

$$u = \overline{\alpha}_1 + \overline{\alpha}_3; \quad v = \overline{\alpha}_2 + \overline{\alpha}_4.$$

Minimizing:

$$1 - 2u = 0; \quad 1 - 2v = 0$$

$\Rightarrow$

$$u = v = \frac{1}{2}.$$

Clearly this is largest if we make $u = v = \frac{1}{2}$; this can only happen (see constraint (10)) if $\overline{\alpha}_j = \frac{1}{4} \; \forall \; j$.

Toy example

So

$$\overline{\boldsymbol{\alpha}} = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}.$$

Toy example

Thus

$$\mathbf{a} = Y\overline{\boldsymbol{\alpha}} = \begin{bmatrix} 1/4 \\ 1/4 \\ -1/4 \\ -1/4 \end{bmatrix}.$$

Thus

$$\mathbf{w} = \sum a_i \mathbf{x}_i = \frac{1}{4}(\mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 - \mathbf{x}_4) = \frac{1}{4}((4,0)) = (1,0) \text{ .}$$

Margin = $\frac{1}{|\mathbf{w}|} = 1$ (we'll revisit this--).

Toy example

Now plug in **a** find b separately from original equation (9); we will minimize with respect to b the original functional

Toy example

$$\mathcal{L}(f) = \frac{1}{4}\sum_j (1 - (\mathbf{w}\cdot\mathbf{x}_j + b)y_j)_+ + |\mathbf{w}|^2$$

$$= \frac{1}{4}\Big\{[1-(1+b)(1)]_+ + [1-(1+b)(1)]_+$$

$$+ [1-(-1+b)(-1)]_+ + [(1-(-1+b)(-1)]_+\Big\} + 1$$

Toy example

$$= \frac{1}{4}\left\{[-b]_+ + [-b]_+ + [b]_+ + [b]_+\right\} + 1$$

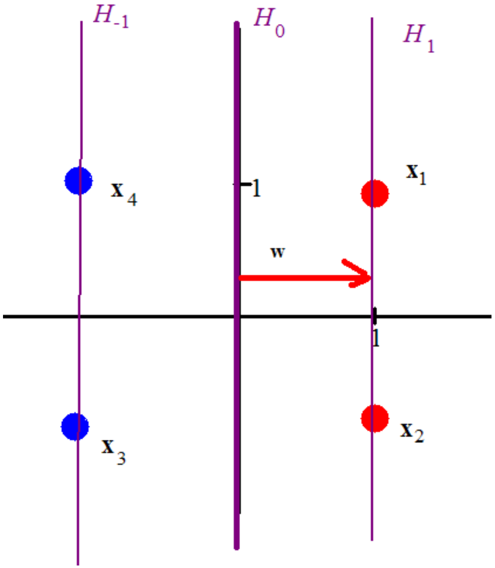
$$= \frac{1}{2}\{[-b]_+ + [b]_+\} + 1.$$

Clearly the above is minimized when $b = 0$.

Thus $\mathbf{w} = (1, 0)$; $b = 0 \ \Rightarrow$

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = x_1$$

Toy example



Toy example

[note in this case the margins reach just out to the closest data vectors; this always happens if λ is small enough; see Theorem below].

SVM: Geometric interpretation

SVM: Geometric interpretation

1. Basics

Recall: if

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

for some $\mathbf{w} \in F$, we have defined:

$$\|f\|_{\mathcal{H}} = |\mathbf{w}|$$

(independent of b).

SVM: Geometric interpretation

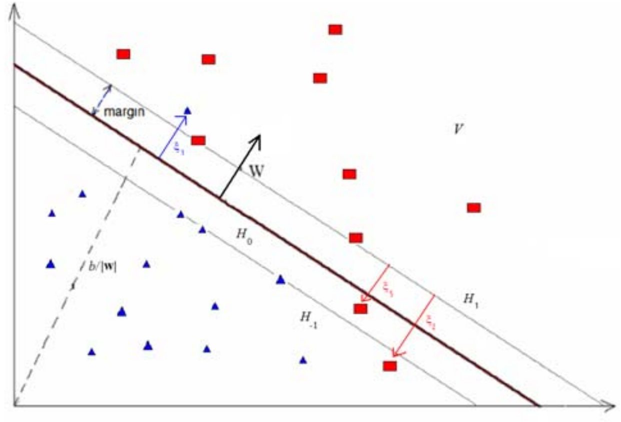


Fig 2: SVM geometry (2 dimensions)

SVM: Geometric interpretation

Recall Lagrangian (full loss function) to be minimized:

$$\mathcal{L}(f) = \frac{1}{n}\sum_{j=1}^{n}(1 - y_j f(\mathbf{x}_j))_+ + \lambda|\mathbf{w}|^2 \equiv \mathcal{L}_d + \mathcal{L}_p \qquad \text{(8a)}$$

(minimization over $(\mathbf{w}, b)$).

Why was this a good choice for $\mathcal{L}$? What should λ be?

Consider variables (see (1b) earlier)

$$\xi_j = (1 - y_j f(\mathbf{x}_j))_+ .$$

SVM: Geometric interpretation

Then

$$\mathcal{L} = \frac{1}{n}\sum_{j=1}^{n}\xi_j + \lambda|\mathbf{w}|^2 \tag{8b}$$

In feature space F, define *positive* direction be parallel to **w**, *negative* direction antiparallel to **w**.

For $\mathbf{x} \in F$, value of $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ determined by $d(\mathbf{x}) =$ distance of **x** from the separating hyperplane

$$H_0 : f(\mathbf{x}) = 0.$$

SVM: Geometric interpretation

Define margin hyperplane (see diagram)

$$H_1 \colon f(\mathbf{x}) = 1.$$

We assume $d(\mathbf{x})$ positive in *positive* direction (parallel to **w**), negative in negative direction (antiparallel to **w**).

SVM: Geometric interpretation

Specifically

$$f(\mathbf{x}) = |\mathbf{w}| d(\mathbf{x})$$

since gradient $\nabla f(\mathbf{x}) = \mathbf{w}$, so f increases along **w** rate $|\mathbf{w}|$ per unit change of **x** in **w** direction.

Note if $y_j = 1$ (i.e., $\mathbf{x}_j$ is in positive class),

$$\xi_j = (1 - |\mathbf{w}| d(\mathbf{x}_j))_+ \; = \begin{cases} 0 & \text{if } d(\mathbf{x}_j) \geq \frac{1}{|\mathbf{w}|} \\ 1 - |\mathbf{w}| d(\mathbf{x}_j) & \text{if } d(\mathbf{x}_j) < \frac{1}{|\mathbf{w}|} \end{cases}.$$

If **x** on *positive* side of H_1 ($d(\mathbf{x}) \geq \frac{1}{|\mathbf{w}|}$):

SVM: Geometric interpretation

$$\xi_j = 0,$$

if $\mathbf{x}$ on *negative* side of H_1:

$$\xi_j = 1 - |\mathbf{w}| d(\mathbf{x}) = +|\mathbf{w}|(\text{distance from } H_1).$$

SVM: Geometric interpretation

Thus if $y_j = 1$

$$\xi_j = \begin{cases} 0 & \text{if } \mathbf{x}_j \text{ on "correct" side of margin } H_1 \\ |\mathbf{w}| \cdot (\text{distance from } H_1) & \text{if } \mathbf{x}_j \text{ on "wrong" side of } H_1 \end{cases}.$$

Similarly, defining the "negative margin" hyperplane

$$H_{-1} : f(\mathbf{x}) = -1,$$

we have if $y_j = -1$ ($\mathbf{x}_j$ in negative class)

$$\xi_j = \begin{cases} 0 & \text{if } \mathbf{x}_j \text{ on "correct" side of margin } H_{-1} \\ |\mathbf{w}| \cdot \text{distance from } H_{-1} & \text{if } \mathbf{x}_j \text{ on "wrong" side of } H_{-1} \end{cases}.$$

Therefore (see above figure)

SVM: Geometric interpretation

$$\sum_j \xi_j = |\mathbf{w}| \cdot D$$

with D the total distance of points on the "wrong" sides of their respective margin hyperplanes $H_{\pm 1}$, i.e., $D =$ "total error".

Also:

distance from separating hyperplane H_0 to margin hyperplane $H_1 = \frac{1}{|\mathbf{w}|}$.

SVM: Geometric interpretation

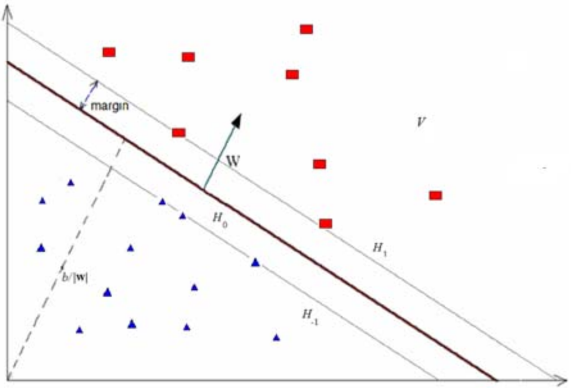
[note: vectors on wrong side of margins are only ones needed for quadratic programming calculation; these are the *support vectors*]

[fewer support vectors $\Rightarrow$ easier calculation $\Rightarrow$ *sparse* machine]

Conclusion: Minimization of full Lagrangian (1) involves a balance between minimizing total error $\sum_j \xi_j$ and the margin width $\frac{1}{|\mathbf{w}|}$, the balance determined by the regularization parameter λ.

1. Special case: Perfect separability

If classes perfectly separable:



Minimizing

$$L = \underbrace{\frac{1}{n}\sum_{j=1}^{n}\xi_j}_{L_d} + \underbrace{\lambda|\mathbf{w}|^2}_{L_p} = L_d + L_p$$

involves maximizing margin $\frac{1}{|\mathbf{w}|}$ and minimizing the total error $\sum_j \xi_j$ with the balance determined by λ.

Choose **w** and b so H_0 bisects the two groups with the maximum "margin" (see diagram above), and the hyperplanes $H_{\pm 1}$ touch closest $\mathbf{x}_j$ to H_0 (such $\mathbf{x}_j$ are *support vectors*).

Then still have

$$\sum_j \xi_j = \text{total error} = 0,$$

while margin $\frac{1}{|\mathbf{w}|}$ is as large as possible.

We thus have in perfectly separable case:

Theorem: The **w**, b which minimize (1) give $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ whose separating hyperplane $H : f(\mathbf{x}) = 0$ gives the widest margin, if λ is sufficiently small.

Summary: In the general case we choose $\|f\|_{\mathcal{H}} = |\mathbf{w}|$, and we minimize

$$\sum_{j=1}^{n} \xi_j + \lambda |\mathbf{w}|^2$$

subject to

$$y_j(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi_j$$

$$\xi_j \geq 0.$$

This is the basic SVM algorithm for finding $f(\mathbf{x})$;
see earlier for the QP algorithm leads to this.

2. The reproducing kernel

As shown earlier the reproducing kernel $K(\mathbf{x}, \mathbf{y})$ for $\mathcal{H}$ above is ordinary dot product of vectors:

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}.$$

Colon cancer application

4. Result: SVM on cancer

Recall: 40 samples colon cancer tissue
22 samples of normal colon tissue (62 total).

For each sample computed

$$\mathbf{x} = (x_1, \ldots, x_d) = \text{microarray profile}$$

Let

$$D = \{\mathbf{x}_i, y_i\}_{i=1}^{62}$$

Colon cancer application

be collection of samples and correct classifications:

$$y_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ cancerous} \\ -1 & \text{if } \mathbf{x}_i \text{ non-cancerous} \end{cases}.$$

Result: using leave one out cross validation obtained:

Feature space F is 6,500 dimensional (6,500 genes)

Misclassification of 6/62 tissues using leave one out cross validation.

5. Example application: handwritten digit recognition - USPS (Scholkopf, Burges, Vapnik)

Handwritten digits:

Handwritten digit recognition

Handwritten digit recognition

Training set (sample size): 7300; Test set: 2000

10 class classifier; i^{th} class has a separating SVM function

$$f_i(\mathbf{x}) = \mathbf{w}_i \cdot \mathbf{x} + b_i$$

Chosen class is

$$\text{Class} = \operatorname*{argmax}_{i \in \{0,\ldots,9\}} f_i(\mathbf{x}).$$

$$\Phi : \text{digit } g \to \text{feature vector } \Phi(g) = \mathbf{x} \in F$$

Handwritten digit recognition

Kernels in feature space F:

RBF: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\sigma^2}}$

Polynomial: $K = (\mathbf{x}_i \cdot \mathbf{x}_j + \theta)^d$

Sigmoidal: $K = \tanh(\kappa(\mathbf{x}_i \cdot \mathbf{x}_j) + \theta)$

Results:

Handwritten digit recognition

polynomial: $K(\mathbf{x},\mathbf{y}) = ((\mathbf{x}\cdot\mathbf{y})/256)^{\text{degree}}$

degree	1	2	3	4	5	6
raw error/%	8.9	4.7	4.0	4.2	4.5	4.5
av. # of SVs	282	237	274	321	374	422

RBF: $K(\mathbf{x},\mathbf{y}) = \exp\left(-\|\mathbf{x}-\mathbf{y}\|^2/(256\,\sigma^2)\right)$

σ^2		1.0	0.8	0.5	0.2	0.1
raw error/%		4.7	4.3	4.4	4.4	4.5
av. # of SVs		234	235	251	366	722

sigmoid: $K(\mathbf{x},\mathbf{y}) = 1.04\tanh(2(\mathbf{x}\cdot\mathbf{y})/256-\Theta)$

Θ		0.9	1.0	1.2	1.3	1.4
raw error/%		4.8	4.1	4.3	4.4	4.8
av. # of SVs		242	254	278	289	296