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
SVM example: cancer classification

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 TCGA.
SVM example: cancer classification

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 - mRNA
- 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

<table>
<thead>
<tr>
<th></th>
<th>CA</th>
<th>Methyl</th>
<th>Exp-Exon</th>
<th>Exp-Gene</th>
<th>Exp-mRNA</th>
<th>SNP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L1</td>
<td>L2</td>
<td>L3</td>
<td>L1</td>
<td>L2</td>
<td>L3</td>
</tr>
<tr>
<td>GBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>458</td>
<td>460</td>
<td>460</td>
<td>20</td>
<td>247</td>
<td>249</td>
</tr>
<tr>
<td>OV</td>
<td>159</td>
<td>159</td>
<td>159</td>
<td>86</td>
<td>86</td>
<td>86</td>
</tr>
</tbody>
</table>

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 10/03/08 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 cancer... [Show More]
SVM example: cancer classification

2. Example: cancer classification


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

\[
\mathbb{R}^d = F = \text{feature space}
\]

We denote

\[
x = \text{feature vector } \in F
\]

With the data set \( D \), can we find the right function \( f: F \rightarrow B \) which generalizes the above examples, so that \( f(x) = y \) for all feature vectors?
SVM example: cancer classification

Easier: find a $f$ for which

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

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

4. Error function

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

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

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

$$= \begin{cases} 
\text{small} & \text{if } y, f(x) \text{ have same sign} \\
\text{large} & \text{otherwise}
\end{cases}.$$
This is the *hinge error function*.
Loss function

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

Thus data-based error (penalty) is

\[
ed = \frac{1}{n} \sum_{j=1}^{n} V(f(x_j), y_j)
\]

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

What other information do we have?
Loss function

Note surface $H: f(x) = 0$ will separate "positive" $x$ with $f(x) > 0$, and "negative" $x$ with $f(x) < 0$:
Fig. 1. Red points have $y = +1$ and blue have $y = -1$ in space $F$. $H: f(x) = 0$ is separating surface.
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 $f(x)$ 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 $V$:

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

5. Finding $f$: Introduction of slack variables

Define new variables $\xi_j$

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

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

(1a)

with the constraint
Slack variables

\[ y_j f(x_j) \geq 1 - \xi_j \]

\[ \xi_j \geq 0, \]

we get the same solution \( f \).

To see this, note the constraints are

\[ \xi_j \geq \max (0, 1 - y_j f(x_j)) = (1 - y_j f(x_j))_+, \quad (1b) \]

which yields the claim. (Clearly in fact in minimizing sum we will end up with \( \xi_j = (1 - y_j f(x_j))_+ \)).
Solving SVM

Summary: the $f$ which minimizes

$$f = \arg \min_{f \in \mathcal{H}} \frac{1}{n} \sum_{j=1}^{n} (1 - y_j f(x_j))^+ + \lambda \|f\|_K^2.$$  \hspace{1cm} (1)

is given by the \textit{quadratic programming} solution:

$$f(x) = \sum_{j=1}^{n} a_j K(x, x_j) + b.$$  \hspace{1cm} (4)

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

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

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

$$\vec{\alpha} = \arg \min_{\vec{\alpha}} \sum_{j=1}^{n} \vec{\alpha}_j - \frac{1}{2} \vec{\alpha}^T P \vec{\alpha}$$

(9)

with constraints

$$0 \leq \vec{\alpha} \leq \frac{1}{2\lambda n}; \quad \vec{\alpha} \cdot \vec{y} = 0$$
Solving SVM

We define

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

$$P = YKY^T,$$

and

$$K = (K_{ij}) = K(x_i, x_j)$$

with $$x_i = i^{th} \text{ sample (e.g. microarray)}.$$
Solving SVM

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

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

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

after finding $a$. 
2. The RKHS for support vector machine

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

\[ f(x) = \sum_{j} a_j K(x, x_j) + b, \]

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

A simple case (linear kernel):

\[ K(x, x_j) = x \cdot x_j. \]

Then we have
Right RKHS for SVM

\[ f(x) = \sum_j (a_j x_j) \cdot x + b \equiv w \cdot x + b, \]

where

\[ w \equiv \sum_j a_j x_j. \]  \hspace{1cm} (10)

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

\[ \mathcal{H} = \{w \cdot x | w \in \mathbb{R}^d\} \]

with inner product
Right RKHS for SVM

\[ \langle w_1 \cdot x, w_2 \cdot x \rangle = w_1 \cdot w_2 \]

is the RKHS of \( K(x, y) \) above.
Right RKHS for SVM

Thus matrix \( K_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j \), and we find the optimal separator

\[
  f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}
\]

by choosing \( \mathbf{w} \) 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

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

Why use this norm? A priori information content.

Final classification rule:

\[ f(x) > 0 \Rightarrow y = 1; \]
\[ f(x) < 0 \Rightarrow y = -1. \]
Right RKHS for SVM

Learning from training data:

\[ Df = (f(x_1), \ldots, f(x_n)) = (y_1, \ldots, y_n). \]

Thus can show RKHS here is

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

is set of linear separator functions (known as \textit{perceptrons} in neural network theory).

Consider separating hyperplane \( H : f(x) = 0 \):
3. Toy example:
Toy example

Information

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

(red = +1; blue = −1);

\[ f = \mathbf{w} \cdot \mathbf{x} + b \]

\[ = \sum_{i} a_{i}(\mathbf{x}_{i} \cdot \mathbf{x}) + b \]

\[ K(\mathbf{x}_{i}, \mathbf{x}) \]
Toy example

so

\[ w = \sum_{i} a_i x_i. \]

Recall \( \| f \|_{H}^2 = \| w \|^2 \), so

\[ L(f) = \frac{1}{4} \sum_{j} (1 - f(x_j) y_j)^+ + \frac{1}{2} \| w \|^2 \]

(\( \lambda = 1/2; \) minimize wrt \( w, b \)).
Toy example

Equivalent:

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

\[ y_j f(\mathbf{x}_j) \geq 1 - \xi_j; \quad \xi_j \geq 0. \]

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

\[ K_{ij} = K(x_i, x_j) = x_i \cdot x_j = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 \end{bmatrix} \]

\[ \|f\|_H = |w|^2 = a^T Ka = 2 \left( \sum_{i=1}^{4} a_i^2 \right) - 4(a_1a_3 + a_2a_4). \]
Toy example

where \( \mathbf{a} = \begin{bmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_d
\end{bmatrix} \).
Toy example

Solution has (see (8a) above)

\[ \alpha = 2\lambda Y^{-1}a = Y^{-1}a \]

(recall \( Y = \begin{bmatrix} y_1 & 0 & \ldots & 0 \\ 0 & y_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & y_n \end{bmatrix} \) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \)
and (8a above)

\[ \overline{\alpha} = \frac{1}{2\lambda} \alpha = \alpha. \]

Finally optimize (8)

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

where
Toy example

\[ P = YKY^T \]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
2 & 0 & -2 & 0 \\
0 & 2 & 0 & -2 \\
-2 & 0 & 2 & 0 \\
0 & -2 & 0 & 2
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1
\end{bmatrix}
\]
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

\[ 0 \leq \alpha_j \leq C \equiv \frac{1}{2\lambda n} = \frac{1}{4}. \] (11)

\[ 0 = \bar{\alpha} \cdot y = \bar{\alpha}_1 + \bar{\alpha}_2 - \bar{\alpha}_3 - \bar{\alpha}_4. \]
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

\[ \bar{\alpha} = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}. \]
Toy example

Thus

\[
a = Y\overline{\alpha} = \begin{bmatrix}
1/4 \\
1/4 \\
-1/4 \\
-1/4
\end{bmatrix}.
\]

Thus

\[
w = \sum a_i x_i = \frac{1}{4}(x_1 + x_2 - x_3 - x_4) = \frac{1}{4}((4, 0)) = (1, 0).
\]

Margin = \(\frac{1}{|w|} = 1\) (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

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

\[ = \frac{1}{4} \left\{ [1 - (1 + b)(1)]_+ + [1 - (1 + b)(1)]_+ \\
+ [1 - (-1 + b)(-1)]_+ + [(1 - (-1 + b)(-1)]_+ \right\} + 1 \]
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 $\lambda$ is small enough; see Theorem below].
1. Basics

Recall: if

\[ f(x) = w \cdot x + b \]

for some \( w \in F \), we have defined:

\[ \|f\|_H = |w| \]

(independent of \( b \)).
SVM: Geometric interpretation

Fig 2: SVM geometry (2 dimensions)
SVM: Geometric interpretation

Recall Lagrangian (full loss function) to be minimized:

\[ \mathcal{L}(f) = \frac{1}{n} \sum_{j=1}^{n} (1 - y_j f(x_j))^+ + \lambda |w|^2 \equiv \mathcal{L}_d + \mathcal{L}_p \tag{8a} \]

(minimization over \((w, b)\)).

Why was this a good choice for \(\mathcal{L}\)? What should \(\lambda\) be?

Consider variables (see (1b) earlier)

\[ \xi_j = (1 - y_j f(x_j))^+ . \]
SVM: Geometric interpretation

Then

$$\mathcal{L} = \frac{1}{n} \sum_{j=1}^{n} \xi_j + \lambda |w|^2$$  \hspace{1cm} (8b)

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

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

$$H_0 : f(x) = 0.$$
SVM: Geometric interpretation

Define margin hyperplane (see diagram)

\[ H_1: f(x) = 1. \]

We assume \( d(x) \) positive in *positive* direction (parallel to \( w \)), negative in negative direction (antiparallel to \( w \)).
SVM: Geometric interpretation

Specifically

\[ f(x) = |w|d(x) \]

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

Note if \( y_j = 1 \) (i.e., \( x_j \) is in positive class),

\[ \xi_j = (1 - |w|d(x_j))_+ = \begin{cases} 0 & \text{if } d(x_j) \geq \frac{1}{|w|} \\ 1 - |w|d(x_j) & \text{if } d(x_j) < \frac{1}{|w|} \end{cases} \]

If \( x \) on positive side of \( H_1 \) \( (d(x) \geq \frac{1}{|w|}) \):
SVM: Geometric interpretation

\[ \xi_j = 0, \]

if \( x \) on negative side of \( H_1 \):

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

$$\sum_j \xi_j = |w| \cdot D$$

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 $H_1 = \frac{1}{|w|}$. 
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 \(\frac{1}{|w|}\), the balance determined by the regularization parameter \(\lambda\).
1. Special case: Perfect separability

If classes perfectly separable:
Minimizing

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

involves maximizing margin \( \frac{1}{\| \mathbf{w} \|} \) and minimizing the total error \( \sum_{j} \xi_j \) with the balance determined by \( \lambda \).

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

Then still have

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

while margin $\frac{1}{|w|}$ is as large as possible.
We thus have in perfectly separable case:

**Theorem:** The $w, b$ which minimize (1) give $f(x) = w \cdot x + b$ whose separating hyperplane $H : f(x) = 0$ gives the widest margin, if $\lambda$ is sufficiently small.
Summary: In the general case we choose $\|f\|_{\mathcal{H}} = |w|$, and we minimize

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

subject to

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

$$\xi_j \geq 0.$$ 

This is the basic SVM algorithm for finding $f(x)$; see earlier for the QP algorithm leads to this.
2. The reproducing kernel

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

\[
K(x, y) = x \cdot y.
\]
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

\[ x = (x_1, \ldots, x_d) = \text{microarray profile} \]

Let

\[ D = \{x_i, y_i\}_{i=1}^{62} \]
Colon cancer application
be collection of samples and correct classifications:

\[ y_i = \begin{cases} 
1 & \text{if } x_i \text{ cancerous} \\
-1 & \text{if } x_i \text{ non-cancerous} 
\end{cases} \]

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.
Handwritten digit recognition

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

Handwritten digits:
Handwritten digit recognition

0 0 0 0
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
5 5 5 5
6 6 6 6
7 7 7 7
8 8 8 8
9 9 9 9
Handwritten digit recognition

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

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

\[
f_i(x) = w_i \cdot x + b_i
\]

Chosen class is

\[
\text{Class} = \arg\max_{i \in \{0, \ldots, 9\}} f_i(x).
\]

\(\Phi: \text{digit } g \rightarrow \text{feature vector } \Phi(g) = x \in F\)
Handwritten digit recognition

Kernels in feature space $F$:

RBF: $K(x_i, x_j) = e^{-\frac{|x_i - x_j|^2}{2\sigma^2}}$

Polynomial: $K = (x_i \cdot x_j + \theta)^d$

Sigmoidal: $K = \tanh(\kappa(x_i \cdot x_j + \theta))$

Results:
### Handwritten digit recognition

**Polynomial:** \( K(x, y) = \left(\frac{x \cdot y}{256}\right)^{\text{degree}} \)

<table>
<thead>
<tr>
<th>degree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw error/%</td>
<td>8.9</td>
<td>4.7</td>
<td>4.0</td>
<td>4.2</td>
<td>4.5</td>
<td>4.5</td>
</tr>
<tr>
<td>av. # of SVs</td>
<td>282</td>
<td>237</td>
<td>274</td>
<td>321</td>
<td>374</td>
<td>422</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>( \sigma^2 )</th>
<th>1.0</th>
<th>0.8</th>
<th>0.5</th>
<th>0.2</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw error/%</td>
<td>4.7</td>
<td>4.3</td>
<td>4.4</td>
<td>4.4</td>
<td>4.5</td>
</tr>
<tr>
<td>av. # of SVs</td>
<td>234</td>
<td>235</td>
<td>251</td>
<td>366</td>
<td>722</td>
</tr>
</tbody>
</table>

**Sigmoid:** \( K(x, y) = 1.04 \tanh\left(2(x \cdot y)/256 - \Theta\right) \)

<table>
<thead>
<tr>
<th>( \Theta )</th>
<th>0.9</th>
<th>1.0</th>
<th>1.2</th>
<th>1.3</th>
<th>1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw error/%</td>
<td>4.8</td>
<td>4.1</td>
<td>4.3</td>
<td>4.4</td>
<td>4.8</td>
</tr>
<tr>
<td>av. # of SVs</td>
<td>242</td>
<td>254</td>
<td>278</td>
<td>289</td>
<td>296</td>
</tr>
</tbody>
</table>