Lecture 11D (Optional).
1. Quadratic programming (QP):
   Introducing Lagrange multipliers $\alpha_j$ and $\mu_j$
   (can be justified in QP for inequality as well as equality constraints) we define the Lagrangian
\[
L(a, b, \xi, \alpha, \mu) \equiv \frac{1}{n} \sum_{j=1}^{n} \xi_j + \lambda a^T K a
\]

\[
- \sum_{j=1}^{n} \alpha_j \left[ y_j \left( \sum_{i=1}^{n} a_i K(x_i, x_j) + b \right) - 1 + \xi_j \right]
\]

\[
- \sum_{j=1}^{n} \mu_j \xi_j
\]  

(4b)
By Lagrange multiplier theory for constraints with inequalities, the minimum of this in

\[ \mathbf{a}, b, \xi, \mathbf{\alpha} = (\alpha_1, \ldots, \alpha_n), \quad \mathbf{\mu} = (\mu_1, \ldots, \mu_n) \]

is a stationary point of this Lagrangian (derivatives vanish) is maximized wrt \( \mathbf{a}, b, \xi, \alpha \), and minimized wrt the Lagrange multipliers, \( \alpha, \mu \) subject to the constraints

\[ \alpha_i, \mu_i \geq 0. \]  

(5)
Derivatives:

\[ \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{j=1}^{n} \alpha_j y_j = 0; \quad (6a) \]

\[ \frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \frac{1}{n} - \alpha_j - \mu_j = 0. \quad (6b) \]

Plugging in get reduced Lagrangian

\[ L^* (a, \alpha) \]
Solving using quadratic programming

\[ = \lambda a^T K a - \sum_{j=1}^{n} \alpha_j \left( y_j \sum_{j=1}^{n} a_i K(x_i, x_j) - 1 \right) \]

\[ = \sum_{j=1}^{n} \alpha_j + \lambda a^T K a - \alpha^T Y K a \]
where

\[
Y = \begin{bmatrix}
y_1 & 0 & 0 & \ldots & 0 \\
0 & y_2 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & y_{n-1} & 0 \\
0 & 0 & 0 & \ldots & y_n
\end{bmatrix}
\]

(note (6) eliminates the $\xi_j$ terms) with same constraints (5).
Now:

\[ \frac{\partial L^*}{\partial a_j} = 0 \implies 2\lambda K a - KY \alpha = 0 \]  \hspace{1cm} (7)

\[ \implies a_i = \frac{\alpha_i y_i}{2\lambda}. \]
Plug in for $a_i$ using (7), replacing $K a$ by $\frac{1}{2\lambda} K Y \boldsymbol{\alpha}$ everywhere:

$$L^*(a, \alpha) = \sum_{j=1}^{n} \alpha_j - \frac{1}{4\lambda} \alpha^T Y K Y^T \alpha$$

$$= \sum_{j=1}^{n} \alpha_j - \frac{1}{4\lambda} \alpha^T P \alpha,$$

where $P = Y K Y^T$. 
Constraints: \( \alpha_j, \mu_j \geq 0; \) by (6b) this implies

\[ 0 \leq \alpha_j \leq \frac{1}{n}. \]

Define \( C = \frac{1}{2\lambda n}, \quad \bar{\alpha} = \frac{1}{2\lambda} \alpha. \)

[note this does not mean complex conjugate!]

Then want to minimize (division by constant \( 2\lambda \) OK - does not change minimizing \( \bar{\alpha} \))
\[
\frac{1}{2\lambda} \sum_{j=1}^{n} 2\lambda \bar{\alpha}_j - \frac{(2\lambda)^2}{2\lambda} \frac{1}{4\lambda} \bar{\alpha}^T P \bar{\alpha} = \sum_{j=1}^{n} \bar{\alpha}_j - \frac{1}{2} \bar{\alpha}^T P \bar{\alpha},
\]

subject to constraint \(0 \leq \bar{\alpha}_i \leq C\); also convenient to include \((6a)\) as constraint: \(\bar{\alpha} \cdot y = 0\). Thus constraints are:

\[
0 \leq \bar{\alpha} \leq C; \quad \bar{\alpha} \cdot y = 0.
\]
Summarizing above relationships:

\[ f(x) = \sum_{j=1}^{n} a_j K(x, x_j) + b, \]

where

\[ a_j = \frac{\alpha_j y_j}{2\lambda}, \]

\[ \alpha_j = 2\lambda \bar{\alpha}_j, \]
and \( \overline{\alpha}_j \) are the (unconstrained) minimizers of \( (8) \), with

\[
P = YY^T.
\]

After \( a_j \) are determined, \( b \) must be computed directly by plugging into \( (4b) \).
More briefly,

\[ f(x) = \sum_{j=1}^{n} \alpha_j y_j K(x, x_j) + b, \]

where \( \alpha_j \) minimize (8).

Finally, to find \( b \), must plug into original optimization problem: that is, we minimize
Solving using quadratic programming

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

\[ = \frac{1}{n} \sum_{j=1}^{n} \left( 1 - y_j \left[ \sum_{i=1}^{n} a_i K(x_j, x_i) + b \right] \right) + \lambda a^T K a. \]
2. The RKHS for SVM

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.
Consider a simple case (linear kernel):

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

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

where

\[ w \equiv \sum_j a_j x_j. \]

This gives the kernel. What class of functions is the corresponding space \( \mathcal{H} \)?
Claim it is the set of linear functions of $\mathbf{x}$:

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

with inner product

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

is the RKHS of $K(\mathbf{x}, \mathbf{y})$ above.
Indeed to show that \( K(x, y) = x \cdot y \) is the reproducing kernel for \( \mathcal{H} \), note that if \( f(x) = x \cdot w \in \mathcal{H} \), then recall \( K(x, y) = x \cdot y \).

So

\[
\langle f(\cdot), K(\cdot, y) \rangle_{\mathcal{H}} = w \cdot y = f(y),
\]

as desired.
Thus the 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 solving for $\mathbf{w}$ as before.

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

Note above inner product gives the norm

$$\| \mathbf{w} \cdot \mathbf{x} \|_H^2 = \| \mathbf{w} \|_{\mathbb{R}^n}^2 = \sum_{j=1}^{n} w_j^2.$$ 

Why use this norm? A priori information content.

Final classification rule: $$f(\mathbf{x}) > 0 \Rightarrow y = 1;$$
$$f(\mathbf{x}) < 0 \Rightarrow y = -1.$$
Learning from training data:

\[ \mathcal{N} f = (f(x_1), \ldots, f(x_n)) = (y_1, \ldots, y_n). \]

Thus

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

is set of linear separator functions (known as perceptrons in neural network theory).
Consider separating hyperplane $H : f(x) = 0$.
3. Toy example:
Information

\[ N f = \left\{ \left[ (1, 1), 1 \right], \left[ (1, -1), 1 \right], \right. \]

\[ \left. \left[ (-1, 1), -1 \right], \left[ (-1, -1), -1 \right] \right\} \]

(red = +1; blue = −1);
Example

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

\[ = \sum_i a_i (\mathbf{x}_i \cdot \mathbf{x}) + b \]

\[ K(\mathbf{x}_i, \mathbf{x}) \]

SO

\[ \mathbf{w} = \sum_i a_i \mathbf{x}_i. \]
\[ L(f) = \frac{1}{4} \sum_j (1 - f(x_j)y_j)_+ + \frac{1}{2} |\mathbf{w}|^2 \]  

(we let \( \lambda = 1/2 \); minimize wrt \( \mathbf{w}, b \)).
Equivalent:

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

$$y_j f(x_j) \geq 1 - \xi_j; \quad \xi_j \geq 0.$$  

[Note effectively $\xi_i = (1 - (w \cdot x_i + b)y_i)_+$]
Define kernel matrix

\[ K_{ij} = K(x_i, x_j) = x_i \cdot x_j = \begin{bmatrix}
2 & 0 & -2 & 0 \\
0 & 2 & 0 & -2 \\
-2 & 0 & 2 & 0 \\
0 & -2 & 0 & 2
\end{bmatrix} \]
\[ \| f \|_H = |w|^2 = a^T K a \]

\[
= 2 \left( \sum_{i=1}^{4} a_i^2 \right) - 4(a_1a_3 + a_2a_4). 
\]

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

\[ L(f) = L(a, b, \xi) = \frac{1}{4} \sum_{j=1}^{4} \xi_j + \frac{1}{2} a^T K a \]

\[ = \frac{1}{4} \sum_{j=1}^{4} \xi_i + \left( \sum_{j=1}^{4} a_i^2 \right) - 2(a_1 a_3 + a_2 a_4) \]
subject to (Eq. 4a):

\[ \xi_j \geq (1 - [y_j(Ka)_j + b]), \quad \xi_j \geq 0. \]

Lagrange multipliers \( \alpha = (\alpha_1, \ldots, \alpha_n)^T \),

\( \mu = (\mu_1, \ldots, \mu_n)^T \) (see (4b)): 
optimize

\[ L(a, b, \xi, \alpha, \mu) \]

\[
= \frac{1}{4} \sum_{j=1}^{4} \xi_i + \frac{1}{2} a^T K a
\]

\[
- \sum_{j=1}^{4} \alpha_j \left[ ((K a)_j + b) y_j - 1 + \xi_j \right] - \sum_{j=1}^{4} \mu_j \xi_j
\]
\[
= \frac{1}{4} \sum_{j=1}^{4} \xi_j + \frac{1}{2} \mathbf{a}^T K \mathbf{a} - \mathbf{\alpha}^T \mathbf{Y} K \mathbf{a} - b \mathbf{\alpha}^T \mathbf{y} \\
+ \sum_{j=1}^{4} \mathbf{\alpha}_j - \mathbf{\alpha} \cdot \xi - \mu \cdot \xi \quad (10)
\]
with constraints

\[ \alpha_i, \mu_i \geq 0. \]

Solution has (see (7) above)

\[ \alpha = 2\lambda Y^{-1}a \]
(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 \((7a)\) above)

\[
\bar{\alpha} = \frac{1}{2\lambda} \alpha = \alpha.
\]
Finally optimize (8):

$$L_1 = \sum_{i=1}^{4} \overline{\alpha}_i - \frac{1}{2} \overline{\alpha}^T P \overline{\alpha},$$

where

$$P = YKY^T$$
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\begin{bmatrix}
2 & 0 & -2 & 0 \\
0 & 2 & 0 & -2 \\
-2 & 0 & 2 & 0 \\
0 & -2 & 0 & 2 \\
\end{bmatrix}
\times
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\begin{bmatrix}
2 & 0 & 2 & 0 \\
0 & 2 & 0 & 2 \\
-2 & 0 & -2 & 0 \\
0 & -2 & 0 & -2 \\
\end{bmatrix}
\]

= \[
\begin{bmatrix}
2 & 0 & 2 & 0 \\
0 & 2 & 0 & 2 \\
2 & 0 & 2 & 0 \\
0 & 2 & 0 & 2 \\
\end{bmatrix}
\].
constraint is

\[ 0 \leq \overline{\alpha} \leq C \equiv \frac{1}{2\lambda n} = \frac{1}{4}. \]  

(10a)

Thus optimize

\[ L_1 = \sum_{i=1}^{4} \overline{\alpha}_i - \left( \sum_{i=1}^{4} \overline{\alpha}_i^2 + 2\overline{\alpha}_1 \overline{\alpha}_3 + 2\overline{\alpha}_2 \overline{\alpha}_4 \right) \]
\[
\sum_{i=1}^{4} \bar{\alpha}_i - (\bar{\alpha}_1 + \bar{\alpha}_3)^2 - (\bar{\alpha}_2 + \bar{\alpha}_4)^2.
\]

\[
= u + v - u^2 - v^2,
\]

where

\[
u = \bar{\alpha}_2 + \bar{\alpha}_4.
\]
Minimizing:

\[ 1 - 2u = 0; \quad 1 - 2v = 0 \]

\[ \Rightarrow \]

\[ u = v = \frac{1}{2}. \]
Thus we have

\[ \overline{\alpha}_i = \frac{1}{4} \]

for all \( i \) (recall the constraint (10a)). Then

\[ \alpha = 2\lambda \overline{\alpha} = \overline{\alpha} = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}. \]
Thus

\[
a = \frac{Y \alpha}{2\lambda} = \begin{bmatrix} 1/4 \\ 1/4 \\ -1/4 \\ -1/4 \end{bmatrix}.
\]
Thus

$$w = \sum a_i x_i = \frac{1}{4}(x_1 + x_2 - x_3 - x_4)$$

$$= \frac{1}{2}((4, 0)) = (1, 0).$$

Margin = \frac{1}{|w|} = 1.$$
Now we find $b$ separately from original equation (9); we will minimize with respect to $b$ the original functional

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

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

Clearly the above is minimized when \( b = 0 \).

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

\[
f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = x_1
\]
4. Other choices of kernel

Recall in SVM we have used the kernel

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

There are many other choices of kernel, e.g.,

\[ K(x, y) = e^{-|x-y|} \quad \text{or} \quad K(x, y) = (1 + |x \cdot y|)^n \]

note - we must choose a kernel function which is positive definite.
How do these choices change the discrimination function $f(x)$ in SVM?
**Ex 1: Gaussian kernel**

\[ K_\sigma(x, y) = e^{-\frac{|x-y|^2}{2\sigma^2}} \]

[can show pos. def. Mercer kernel]

**SVM:** from (4) above have

\[ f(x) = \sum_j a_j K(x, x_j) + b = \sum_j a_j e^{-\frac{|x-x_j|^2}{2\sigma^2}} + b, \]

where examples \( x_j \) in \( F \) have known
classifications $y_j$, and $a_j$, $b$ are obtained by quadratic programming.

What kind of classifier is this? It depends on $\sigma$ (see Vert movie).

Note Movie1 varies $\sigma$ in the Gaussian ($\sigma = \infty$ corresponds to a linear SVM); then movie2 varies the margin $\frac{1}{|w|}$ (in linear feature space $F_2$) as determined by changing $\lambda$ or equivalently $C = \frac{1}{2\lambda n}$. 
5. Software available

Software which implements the quadratic programming algorithm above includes:

- SVMLight: http://svmlight.joachims.org
- SVMTorch: http://www.idiap.ch/learning/SVMTorch.html
- LIBSVM: http://wws.csie.ntu.edu.tw/~cjlin/libsvm
A Matlab package which implements most of these is Spider:

http://www.kyb.mpg.de/bs/people/spider/what_isit.html
6. Example application: handwritten digit recognition - USPS (Scholkopf, Burges, Vapnik)

Handwritten digits:
Other kernels

0 0 0 0 0
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
6 6 6 6 6
7 7 7 7 7
8 8 8 8 8
9 9 9 9 9
Training set: 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

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

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

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:
### Polynomial

**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

**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

**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>
Computational Biology Applications

References:


7. Gene expression arrays for cancer classification

Goal: infer cancer genetics by looking at microarray.

Gene expression array reveals expression patterns and can hopefully be used to discriminate similar cancers, and thus lead to better treatments.
Usual problem: small sample size (e.g., 50 cancer tissue samples), high dimensionality (e.g., 20-30,000). *Curse of dimensionality.*

**Example 1:** Myeloid vs. Lymphoblastic leukemias

ALL: acute lymphoblastic leukemia
AML: acute myeloblastic leukemia

SVM training: leave one out cross-validation
### Applications

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Algorithm</th>
<th>Total Samples</th>
<th>Total errors</th>
<th>Class 1 errors</th>
<th>Class 0 errors</th>
<th>Number Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leukemia Morphology (trest)</td>
<td>SVM</td>
<td>35</td>
<td>0/35</td>
<td>0/21</td>
<td>0/14</td>
<td>40</td>
</tr>
<tr>
<td>AML vs ALL</td>
<td>WW</td>
<td>35</td>
<td>2/35</td>
<td>1/21</td>
<td>1/14</td>
<td>50</td>
</tr>
<tr>
<td>k-NN</td>
<td>35</td>
<td>3/35</td>
<td>1/21</td>
<td>2/14</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Leukemia Lineage (ALL)</td>
<td>SVM</td>
<td>23</td>
<td>0/23</td>
<td>0/15</td>
<td>0/8</td>
<td>10</td>
</tr>
<tr>
<td>B vs T</td>
<td>WW</td>
<td>23</td>
<td>0/23</td>
<td>0/15</td>
<td>0/8</td>
<td>9</td>
</tr>
<tr>
<td>k-NN</td>
<td>23</td>
<td>0/23</td>
<td>0/15</td>
<td>0/8</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Lymphoma FS vs DLCL</td>
<td>SVM</td>
<td>77</td>
<td>4/77</td>
<td>2/32</td>
<td>2/35</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>WW</td>
<td>77</td>
<td>6/77</td>
<td>1/32</td>
<td>5/35</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>k-NN</td>
<td>77</td>
<td>3/77</td>
<td>1/32</td>
<td>2/35</td>
<td>250</td>
</tr>
<tr>
<td>Brain MD vs Glioma</td>
<td>SVM</td>
<td>41</td>
<td>1/41</td>
<td>1/27</td>
<td>0/14</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>WW</td>
<td>41</td>
<td>1/41</td>
<td>1/27</td>
<td>0/14</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>k-NN</td>
<td>41</td>
<td>0/41</td>
<td>0/27</td>
<td>0/14</td>
<td>5</td>
</tr>
</tbody>
</table>

fig. 1: Myeloid and Lymphoblastic Leukemia classification by SVM

S. Mukherjee
Fig 2: AML vs. ALL error rates with increasing sample size
In above figure the curves represent error rates with split between training and test sets. Red dot represents leave one out cross-validation.