Support Vector Machines: Brief Overview

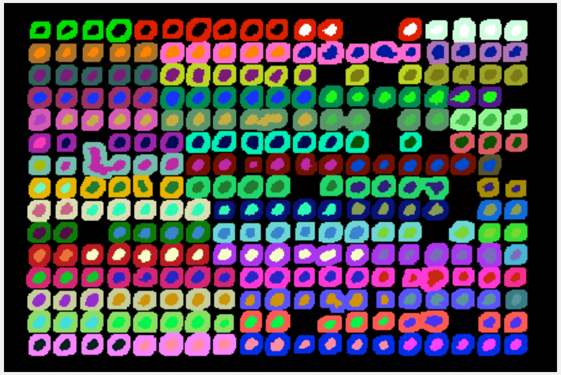
Outline

- Microarray Example
- Support Vector Machines (SVMs)
- Software: libsvm
- A Baseball Example with libsvm

Classifying Cancer Tissue: The ALL/AML Dataset

- Golub *et al*. (1999), Guyon *et al*. (2002): Affymetrix microarrays containing probes for 7,129 human genes.
- Scores on microarray represent intensity of gene expression after being re-scaled to make each chip equivalent.
- Training Data: 38 bone marrow samples, 27 acute lymphoblastic leukemia (ALL), 11 acute myeloid leukemia (AML).
- Test Data: 34 samples, 20 ALL and 14 AML.
- Our Experiment: Use LIBSVM to analyze the data set.

ML Experiment



Microarray Image File

training data

```
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:96 3:58 4:794 5:665 6:5328 7:1574 8:263 9:98 10:37
1.0   1:154 2:98 3:56 4:857 5:642 6:5196 7:1574 8:300 9:95 10:35
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:96 3:58 4:794 5:665 6:5328 7:1574 8:263 9:98 10:37
1.0   1:154 2:98 3:56 4:857 5:642 6:5196 7:1574 8:300 9:95 10:35
1.0   1:154 2:98 3:56 4:857 5:642 6:5196 7:1574 8:300 9:95 10:35
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:96 3:58 4:794 5:665 6:5328 7:1574 8:263 9:98 10:37
1.0   1:154 2:98 3:56 4:857 5:642 6:5196 7:1574 8:300 9:95 10:35
```

testing data

```
...
1.0   1:154 2:98 3:56 4:857 5:642 6:5196 7:1574 8:300 9:95 10:35
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0   1:154 2:96 3:58 4:794 5:665 6:5328 7:1574 8:263 9:98 10:37
1.0   1:154 2:98 3:56 4:857 5:642 6:5196 7:1574 8:300 9:95 10:35
```

Labeled Data File

ALL/AML	$gene_1$:$intensity_1$	$gene_2$:$intensity_2$	$gene_3$:$intensity_3$ …
0.0	1: 0.852272	2: 0.273378	3: 0.198784

Labeled Data

- **Training data**: Associates each ***feature vector*** of data ($\boldsymbol{X}_i$) with its known classification (y_i):

$$(\boldsymbol{X}_1, \mathrm{y}_1), (\boldsymbol{X}_2, \mathrm{y}_2), \ldots, (\boldsymbol{X}_\mathrm{p}, \mathrm{y}_\mathrm{p})$$

where each $\boldsymbol{X}_\mathrm{i}$ is a d-dimensional vector of real numbers and each y_i is classification label (1, -1) or (1, 0).

- Example (p=3):

```
0.0  1:154 2:72 3:81 4:650 5:698 6:5199 7:1397 8:216 9:71 10:22
0.0  1:154 2:96 3:58 4:794 5:665 6:5328 7:1574 8:263 9:98 10:37
1.0  1:154 2:98 3:56 4:857 5:642 6:5196 7:1574 8:300 9:95 10:35
```

Classification Labels

Feature Vectors (d=10 attribute:value pairs)

Training and Testing

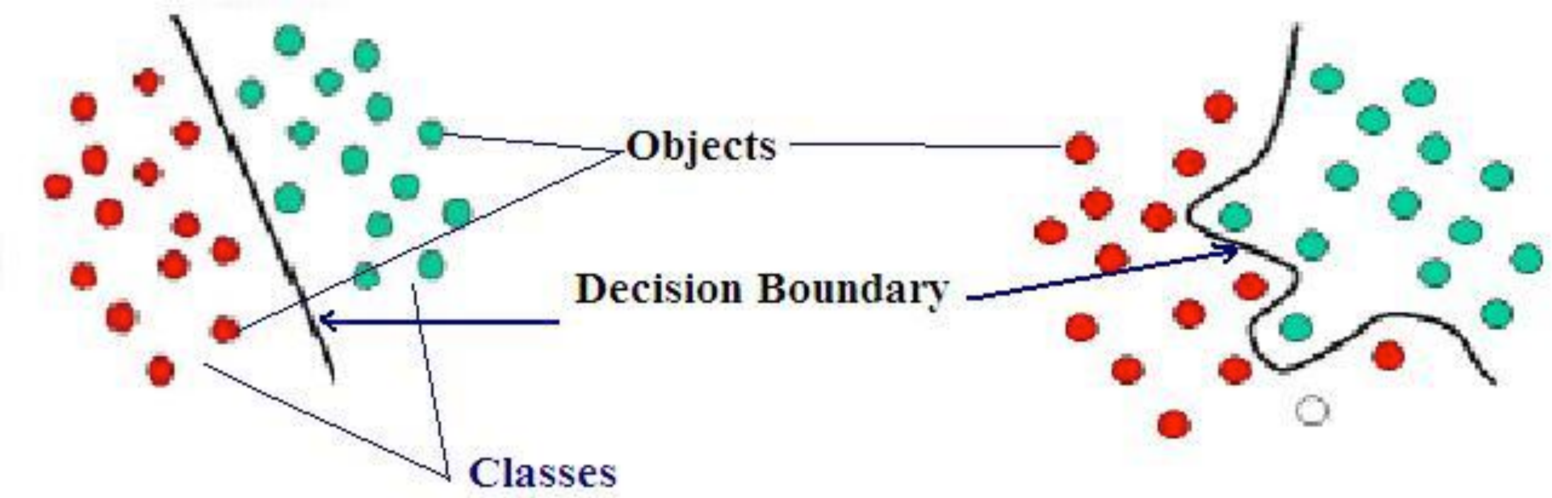
- **Scaling**: Data can be scaled, as needed to reduce the effect of variance among the features.
- **Five-fold Cross Validation (CV):**
  - Select a 4/5 subset of the training data.
  - Train a model and test on the remaining 1/5.
  - Repeat 5 times and choose the best model.
- **Test Data:** Same format as training data. Labels are used to calculate success rate of predictions.
- **Experimental Design:**
  - Divide it into training set and testing set.
  - Create the model on the training set.
  - Test the model on the test data.

ALL/AML Results

Approach	Training/Testing Details	Training Accuracy	Testing Accuracy
LIBSVM Saroj & Morelli	• 5-fold cross validation • RBF Kernel • All 7129 features.	36/38 (94.7 %)	28/34 (82.4 %)
Weighted Voting Golub *et al*. (1999)	• Hold-out-one cross validation • Informative genes cast weighted votes • 50 informative genes	36/38 (94.7 %)	29/34 (85.3 %) (prediction strength > 0.3)
Weighted Voting Slonim *et al*. (2000)	• 50 gene predictor • cross-validation with prediction strength > 0.3 cutoff at 0.3	36/38 (94.7 %)	29/34 (85.3%)
SVM Furey *et al*. 2000	• Hold-out-one cross validation • Top ranked 25, 250, 500, 1000 features • Linear Kernel plus Diagonal Factor	100 %	From 30/34 to 32/34 (88 % - 94 %)

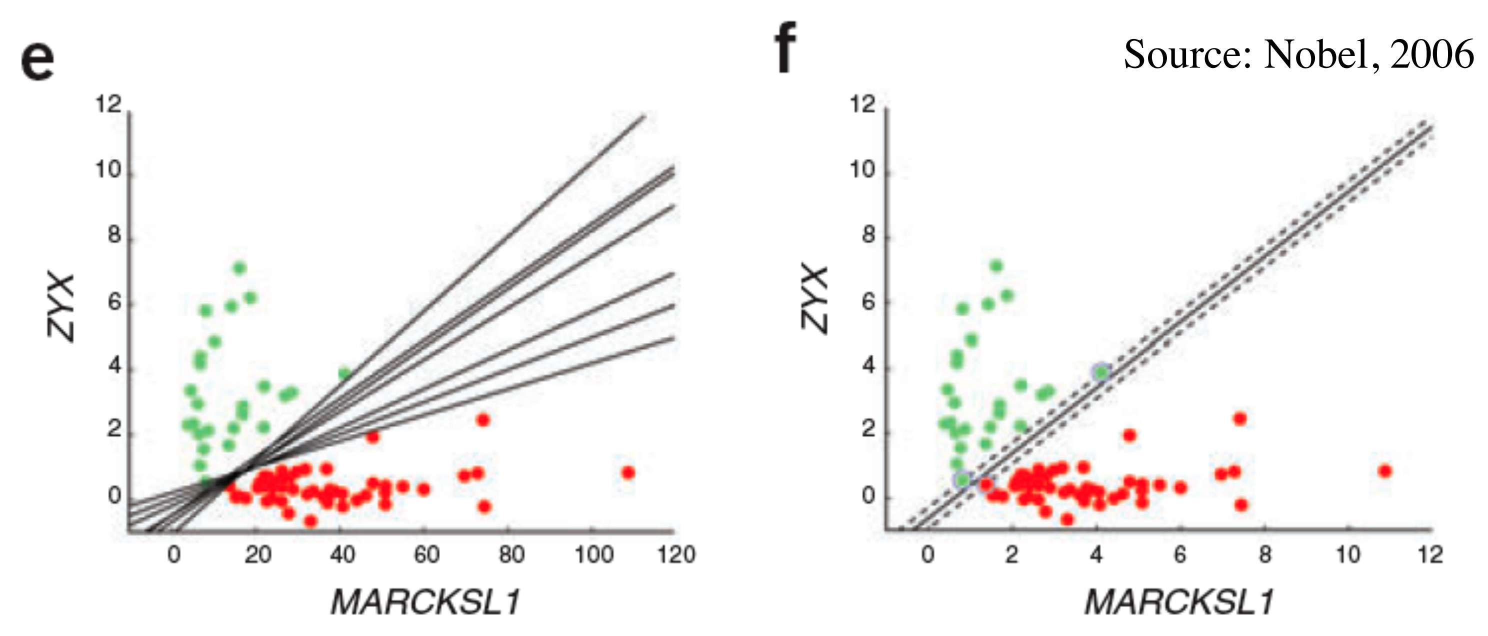
Support Vector Machine (SVM)

- SVM: Uses (***supervised***) machine learning to solve classification and regression problems.
- Classification Problem: Train a model that will classify input data into two or more distinct classes.
- Training: Find a decision boundary (a ***hyperplane***) that divides the data into two or more classes.



Maximum-Margin Hyperplane

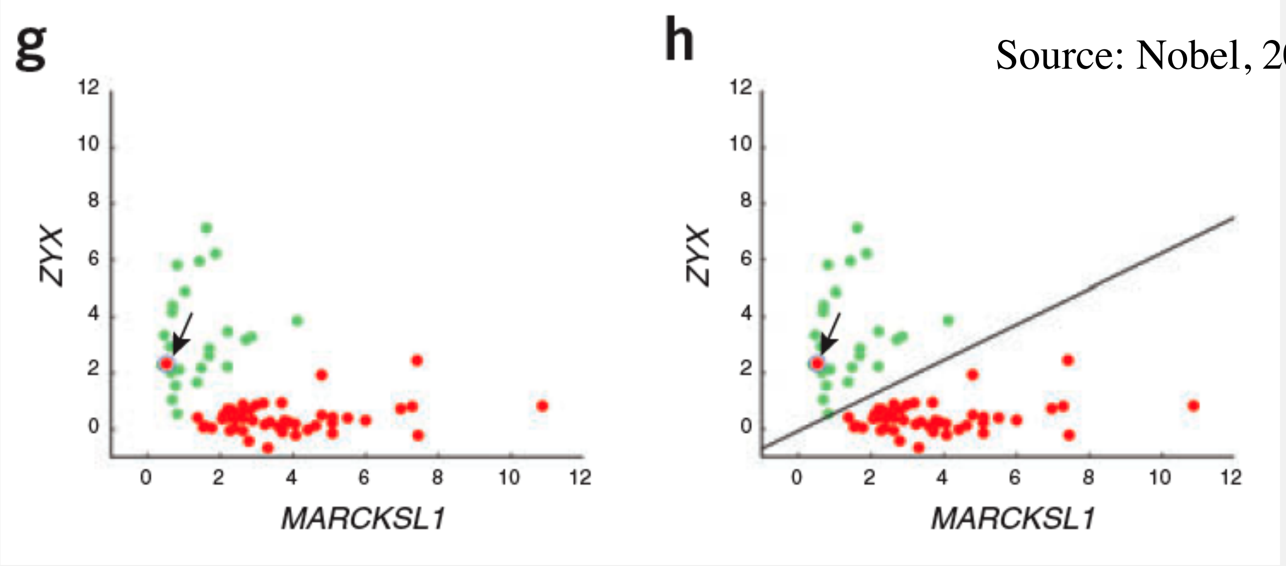
- ***Linearly separable*** case: A line (hyperplane) exists that separates the data into two distinct classes.
- An SVM finds the separating plane that ***maximizes*** the distance between distinct classes.



Source: Nobel, 2006

Handling Outliers

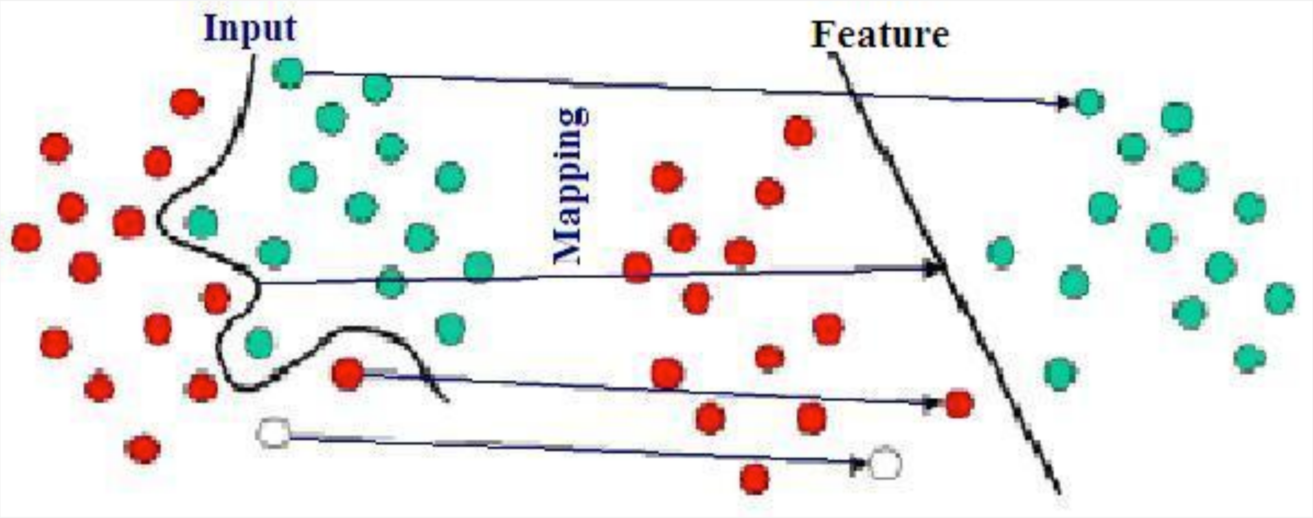
- SVM finds a perfect boundary (sometimes **over fitting**).
- A ***soft margin*** parameter can allow a small number of points on the wrong side of the boundary, diminishing training accuracy.
- **Tradeoff**: Training accuracy vs. predictive power.



Source: Nobel, 2006

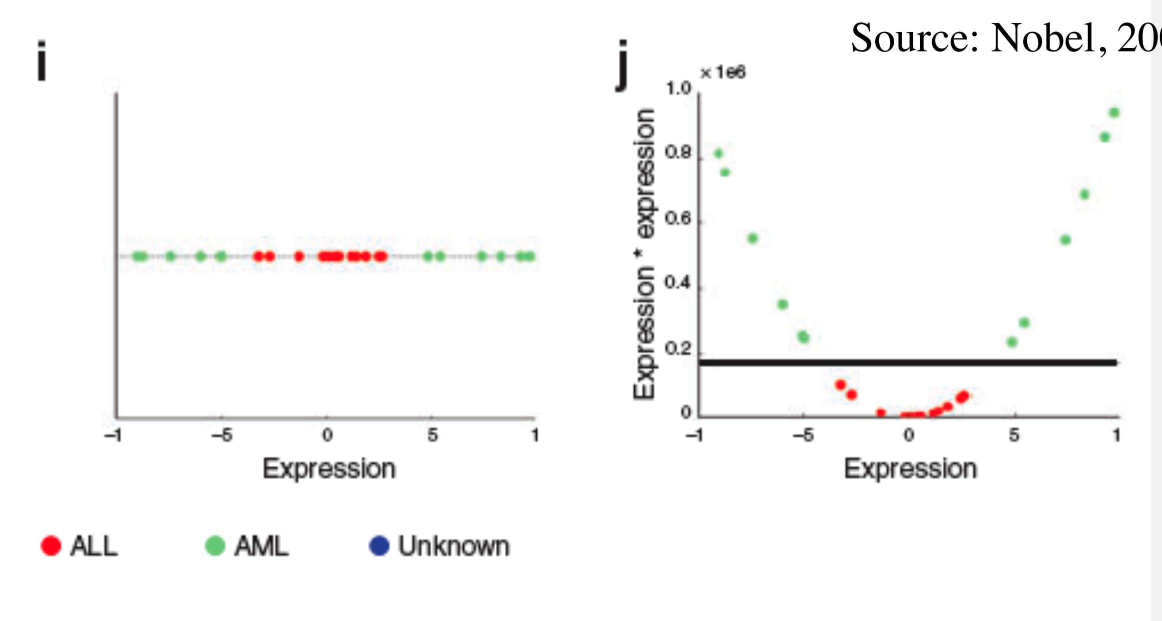
Nonlinear Classification

- Nonseparable data: A SVM will map the data into a higher dimensional space where it is separable by a hyperplane.
- The ***kernel function***: For any consistently labeled data set, there exists a kernel function that maps the data to a linearly separable set.



Kernel Function Example

- In figure *i* the data are not separable in a 1-dimensional space, so we map them into a 2-dimensional space where they are separable.
- Kernel Function, $\boldsymbol{K}(x_i) \rightarrow (x_i, 10^5 \cdot x_i^2)$

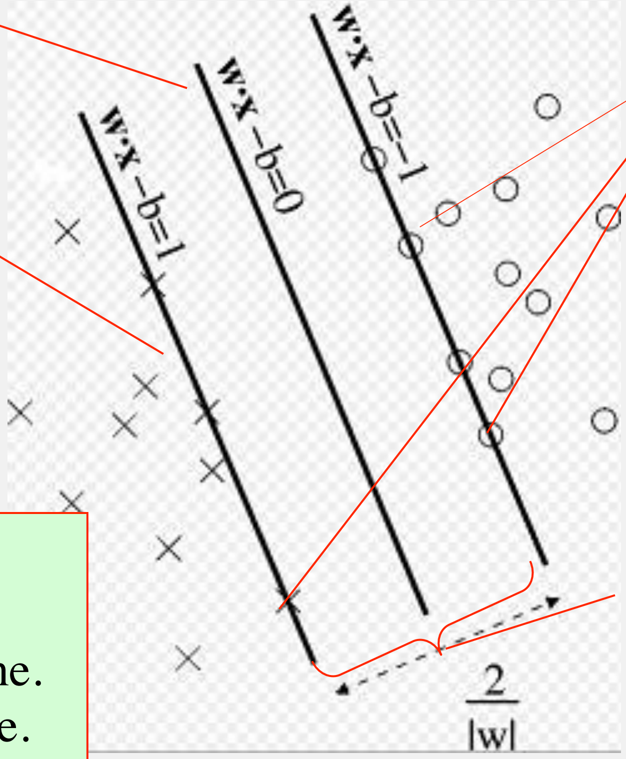


Source: Nobel, 2006

SVM Math

Maximum Margin Hyperplane

Boundary plane.



Support vectors are points on the boundary planes.

We ***maximize*** this margin by minimizing |w|.

<u>Notation:</u>

- ***w*** is a vector perpendicular to the plane.
- ***x*** is a point on the plane.
- *b* is the offset (from the origin) parameter

Source: Burges, 1998

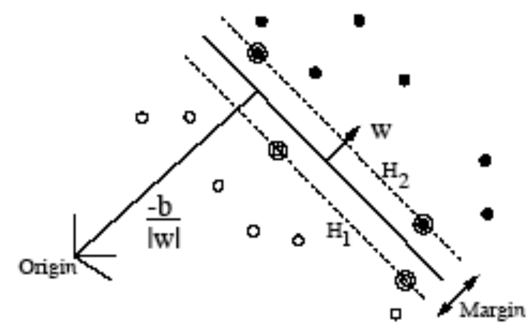
SVM Math (cont)

- Let $S = \{(\boldsymbol{x}_i, y_i)\}, i=1,\ldots, p$ be a set of labeled data points where $\boldsymbol{x}_i \in R^d$ is a ***feature vector*** $y_i \in \{1,-1\}$ is a ***label.***
- We want to exclude points in S from the ***margin*** between the two boundary hyperplanes, which can be expressed by the following ***constraint***:

$$\boxed{y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1, \quad 1 \leq i \leq p.}$$

- To maximize the distance $2/|\mathbf{w}|$ between the two boundary planes, we minimize $|\mathbf{w}|$, the vector perpendicular to the hyperplane.

Source: Burges, 1998



A two-dimensional example.

- A ***Lagrangian*** formulation allows us to represent the training data simply as the ***dot product*** between vectors and allows us to simplify the constraint. Given $\boldsymbol{\alpha}_i$ as the Langrange multiplier for each constraint (each point), we maximize:

$$\boxed{L = \sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j}$$

SVM Math Summary

- To summarize:
  - For the separable linear case, training amounts to maximizing ***L*** with respect to α_i. The ***support vectors***--i.e. those points on the boundary planes for which $\alpha_i > 0$ -- are the only points that play a role in training.
  - This maximization problem is solved by ***quadratic programming***, a form of mathematical optimization.
  - For the ***non-separable case*** the above algorithm would fail to find a hyperplane, but solutions are available by:
    - Introducing ***slack variables*** to allow certain points to violate the constraint.
    - Introducing ***kernel functions***, $K(\mathbf{x}_i \cdot \mathbf{x}_j)$ which map the dot product into a higher-dimensional space.
    - Example kernels: *linear, polynomial, radial basis function,* and others.

LIBSVM Example

- Software Tool: LIBSVM
- Data: Astroparticle experiment with 4 features, 3089 training cases and 4000 labeled test cases.
- Command-line experiments:

```
$svmscale train.data > train.scaled
$svmscale test.data > test.scaled
$svmtrain train.scaled > train.model
  Output: Optimisation finished, #iter = 496
$svmpredict test.scaled train.model test.results
  Output: Accuracy = 95.6% (3824/4000) (classification)
```

- Repeat with different parameters, kernels.

Analyzing Baseball Data

- Problem: Predict winner/loser of division or league.
- Major league baseball statistics, 1920-2000.
- Vectors: 30 Features, including (most important)

G (games)	W (wins)	L (losses)
PCT (winning)	**GB (games behind)**	**R (runs)**
OR (opponent runs)	AB (at bats)	H (hits)
2B (doubles)	3B (triples)	HR (home runs)
BB (walks)	SO (strike outs)	**AVG (batting)**
OBP (on base pct)	**SLG (slugging pct)**	SB (steals)
ERA (earn run avg)	CG (complete games)	SHO (shutouts)
SV (saves)	IP (innings)	

Baseball Results
(All numbers are % of predictive accuracy)

Model	Training CV Data	Test Data	Test 50/50	Random Data	Random 50/50	All Zeroes	All Ones
Random Control	85.3	86.7	50	86.7	50	100	0
Trivial Control 1 GB Only	99.8	99.8	100	77.2	48.3	86.8	13.2
Trivial Control 2 PCT Only	99.3	99.3	97.7	85.3	50	84.6	15.4
Trivial Control 3 All features	98.6	98.8	96.5	74.1	49.8	85.0	15.0
Test Model 1 All Minus GB & PCT	91.2	92.4	72.2	79.6	48.0	89.5	10.5
Test Model 2 AVG+OBP +SLG+ERA+SV	89.5	90.4	63.0	76.9	49.7	87.2	12.8
Test Model 3 All Minus GB	92	89.4	69.4	77.5	49.8	91.0	9.0
Test Model 4 R & OR Only	90	89.4	75.9	79.9	47.6	92.6	7.4

Software Tools

- Many open source SVM packages.
  - LIBSVM (C. J. Lin, National Taiwan University)
  - SVM-light (Thorsten Joachims, Cornell)
  - SVM-struct (Thorsten Joachims, Cornell)
  - mySVM (Stefan Ruping, Dortmund U)
- Proprietary Systems
  - Matlab Machine Learning Toolbox

References

- Our WIKI (http://www.cs.trincoll.edu/bioinfo)
- C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*, 121-167, 1998.
- T. S. Furey *et al*. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics 16(10)*, 2000.
- T. R. Golub, *et al*. Molecular classification of cancer: Class discovery and class prediction by gene expression. *Science 286, 531*, 1999.
- I. Guyun, *et al*. Gene selection for cancer classification using support vector machines. *Machine Learning 46*, 389-422, 2002.
- W. S. Noble. What is a support vector machine. *Nature Biotechnology 24(12)*, Dec. 2006.