A short introduction to supervised learning,

with applications to cancer pathway analysis

Dr. Christina Leslie

Computational Biology Program

Memorial Sloan-Kettering Cancer Center

http://cbio.mskcc.org/leslielab

Outline

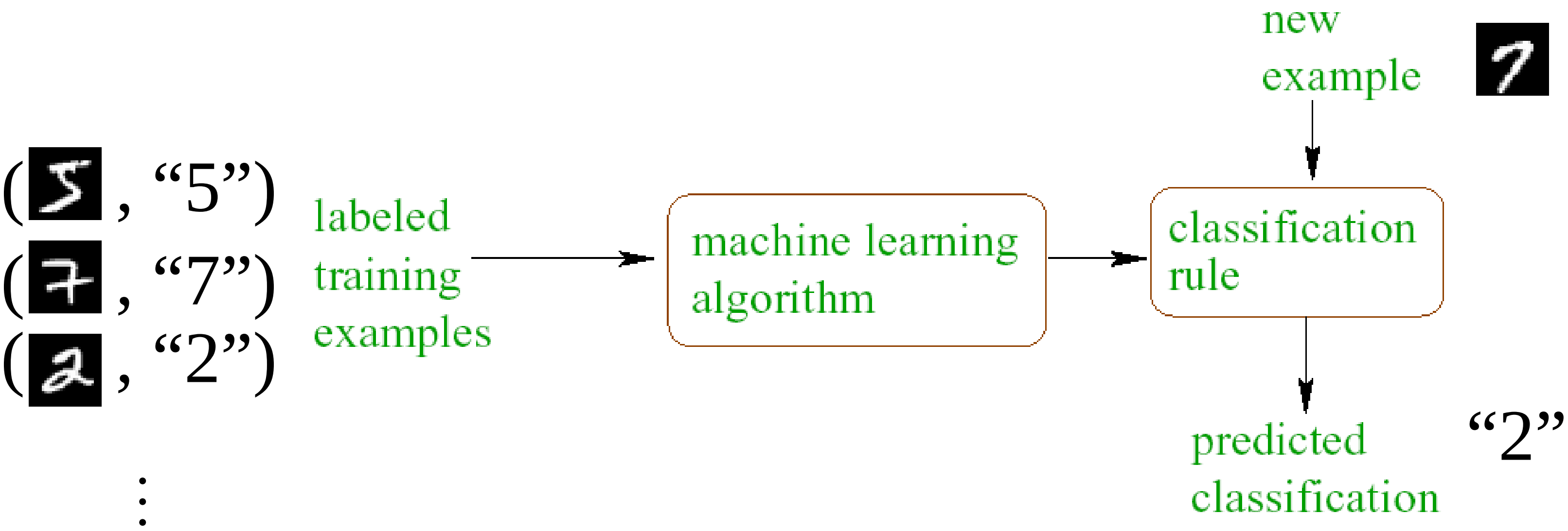
- General ideas about supervised learning
  - (Not specific to biological domain)
  - Training, generalization, overfitting
  - Small bit of theory
- Cancer classification, gene signatures
  - Lab preview: Nevins paper (Bild et al., Nature Medicine 2006, “Oncogenic pathways…”) as a concrete example
- SVMs in some (mathematical) detail

What is machine learning?

- “Statistics with more than 20 variables”
- “Intersection of computer science and statistics”
- Provisional definition: [R. Schapire]
  - Machine learning studies how to *automatically learn* to make *predictions* based on past observations

Classification problems

- Classification:
  - Learn to classify examples into a given set of categories (“classes”)
  - Example of *supervised learning* (“labeled” training examples, i.e. known class labels)



(, “5”)
(, “7”)
(, “2”)
⋮

labeled training examples

machine learning algorithm

classification rule

new example

predicted classification

“2”

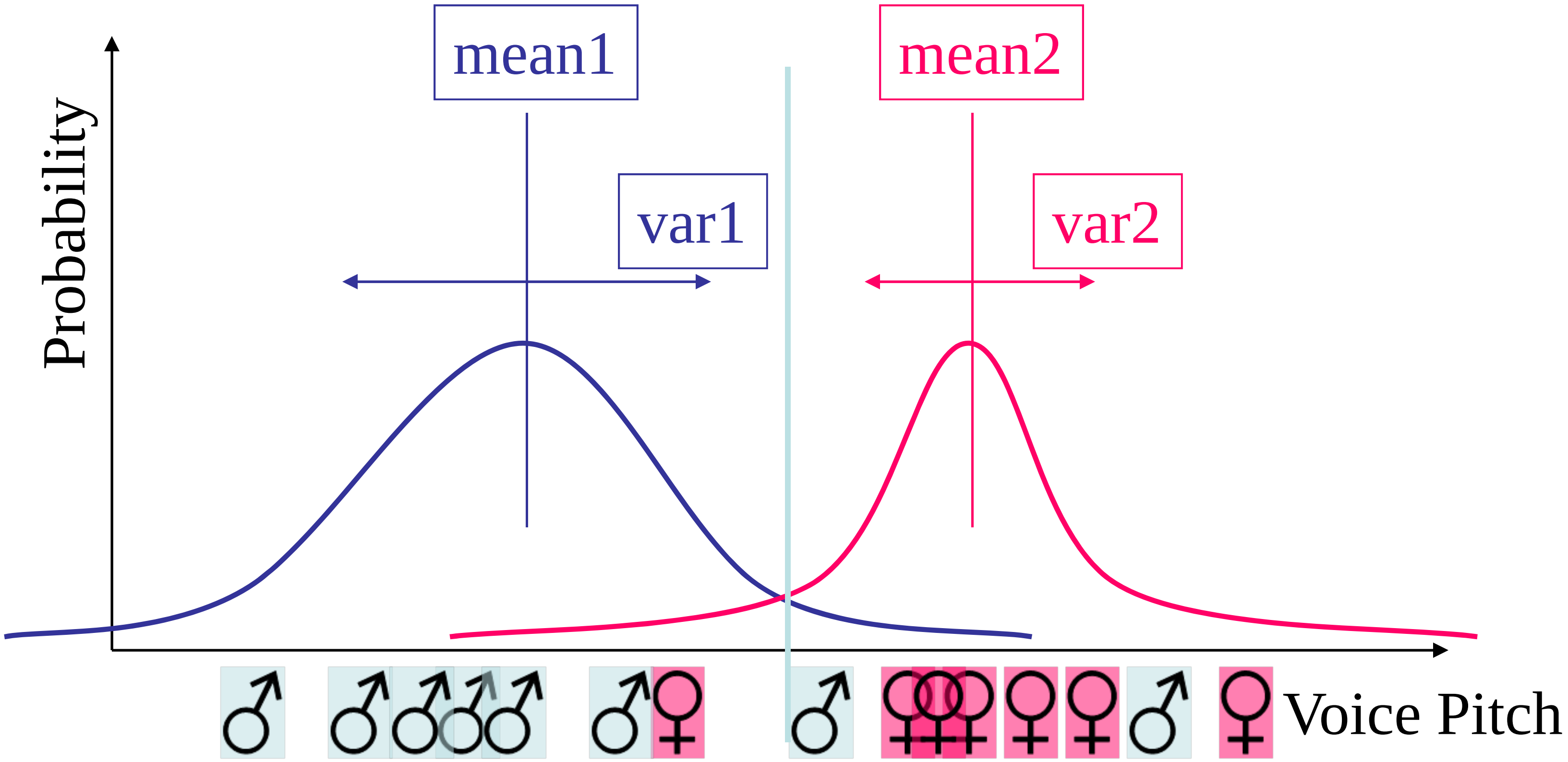
ML vs. “Traditional Statistics”

- L. Breiman: “The two cultures”, Statistical Science, 2001
- “Data modeling culture” (Generative models)
  - Assume probabilistic model of known form, not too many parameters (<50)
  - Fit model to data
  - Interpret model and parameters, make predictions after

ML vs “Traditional Statistics”

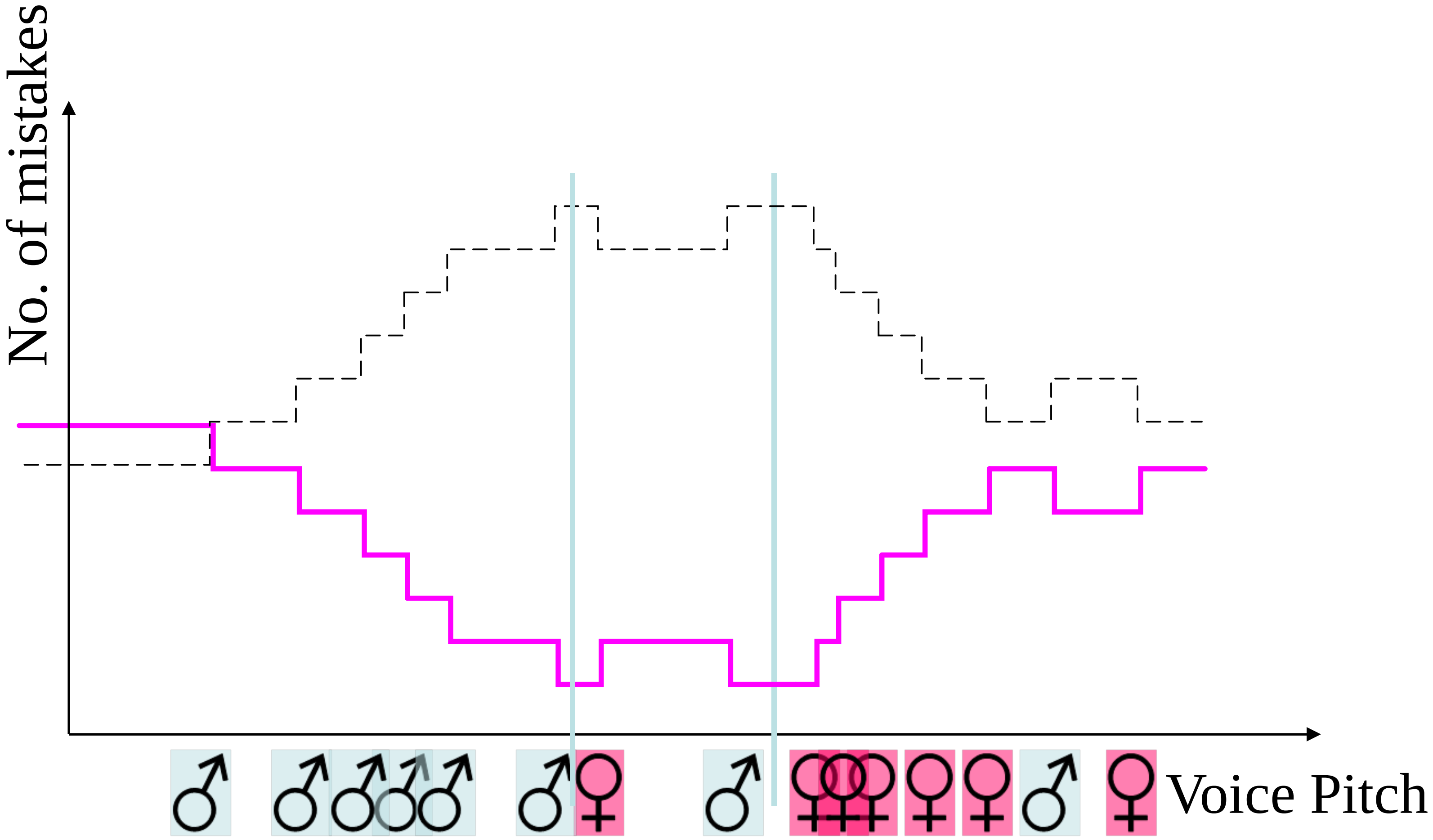
- “Algorithmic modeling culture” (Predictive models)
  - Learn a *prediction function* from inputs to outputs, possibly many parameters (e.g. 10^2 - 10^6)
  - Design algorithm to find good prediction function
  - Primary goal: accurate predictions on new data, i.e. avoid *overfitting*, good *generalization*
  - Interpret after, finding “truth” is not central goal (but some “truth” in accurate prediction rule?)
- “Never solve a more difficult problem than you need to” [V. Vapnik]

Example: Generative model



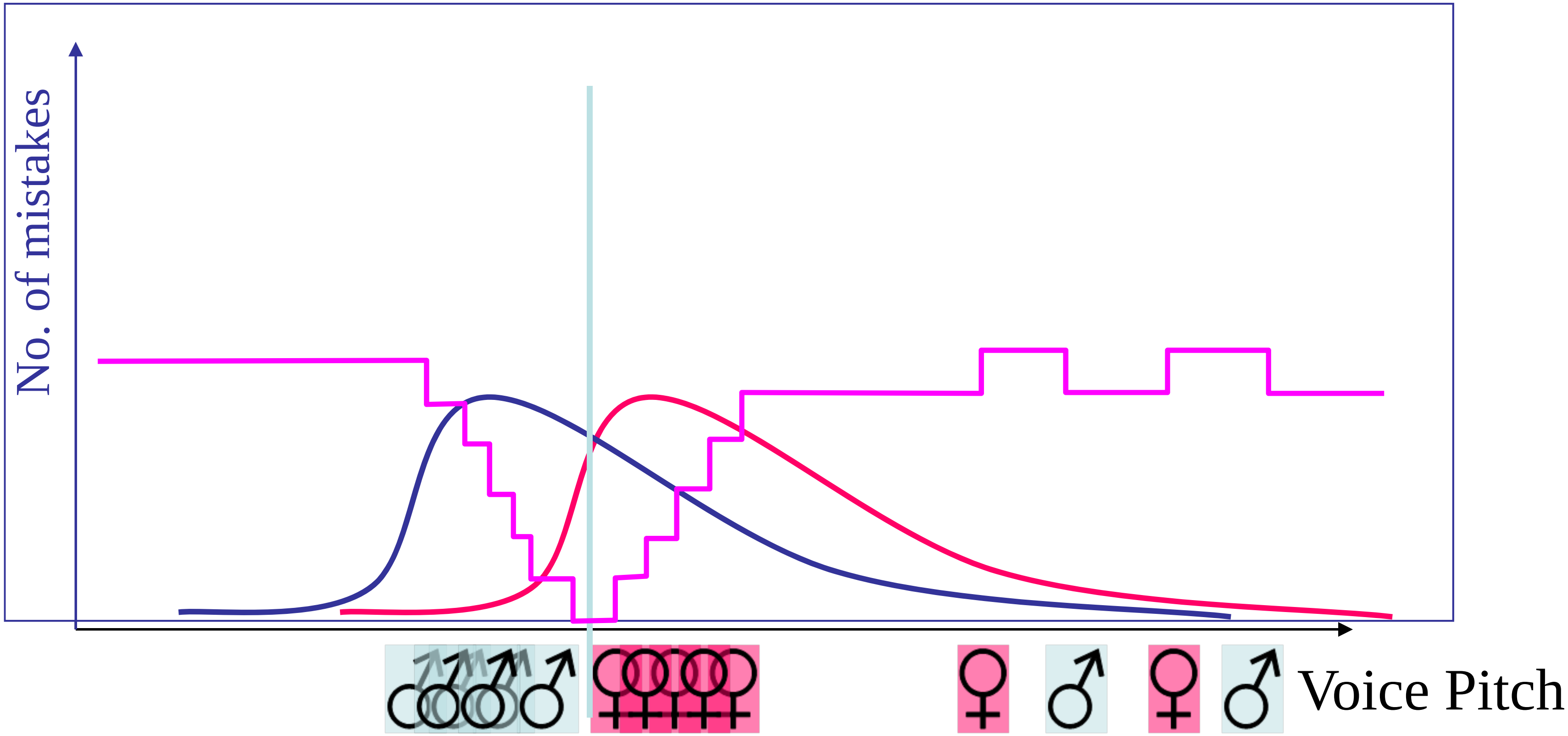
[Figure: Y. Freund]

Example: Prediction function



[Figure: Y. Freund]

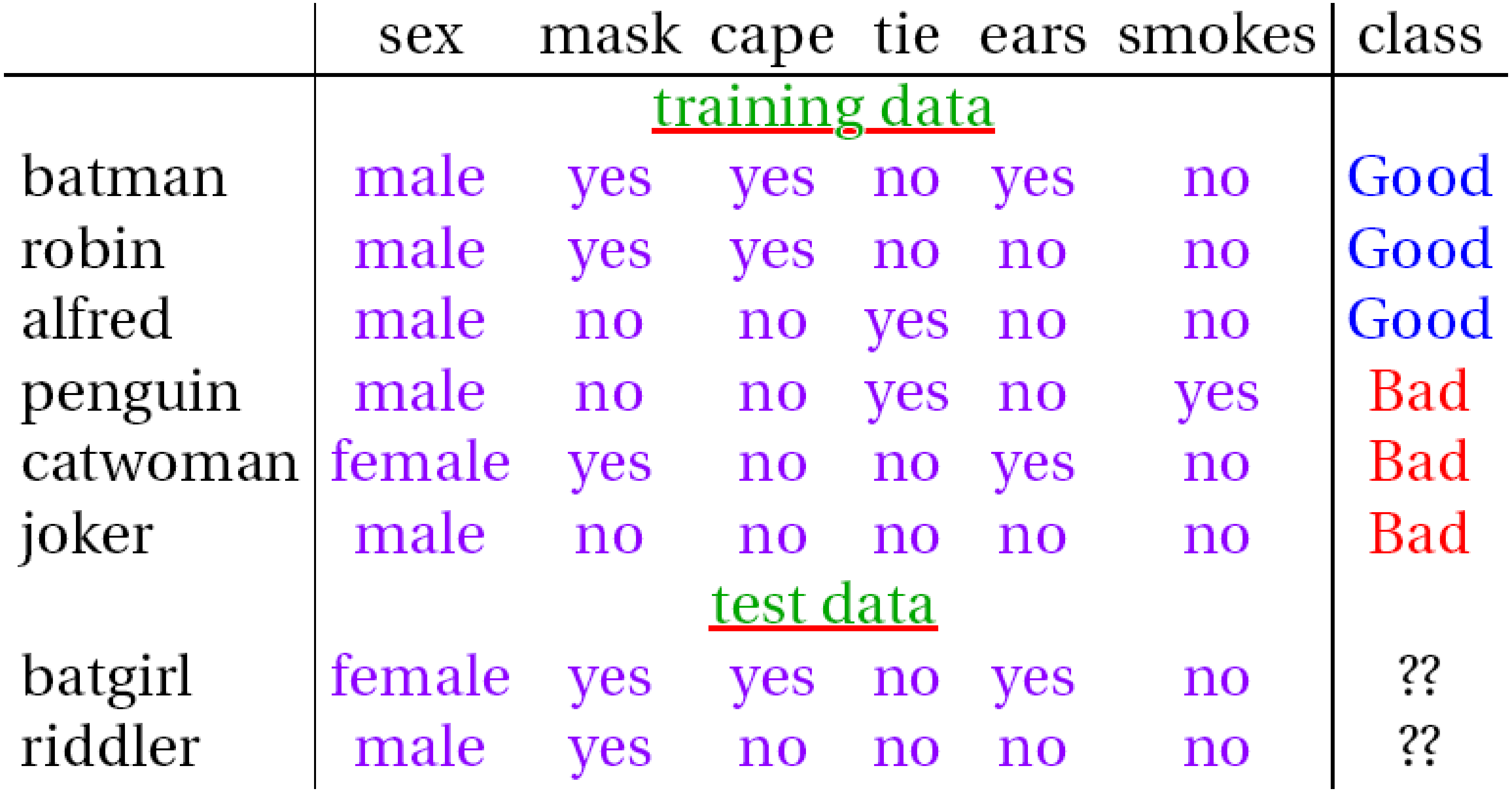
Poorly behaved training data



[Figure: Y. Freund]

Conditions for accurate learning

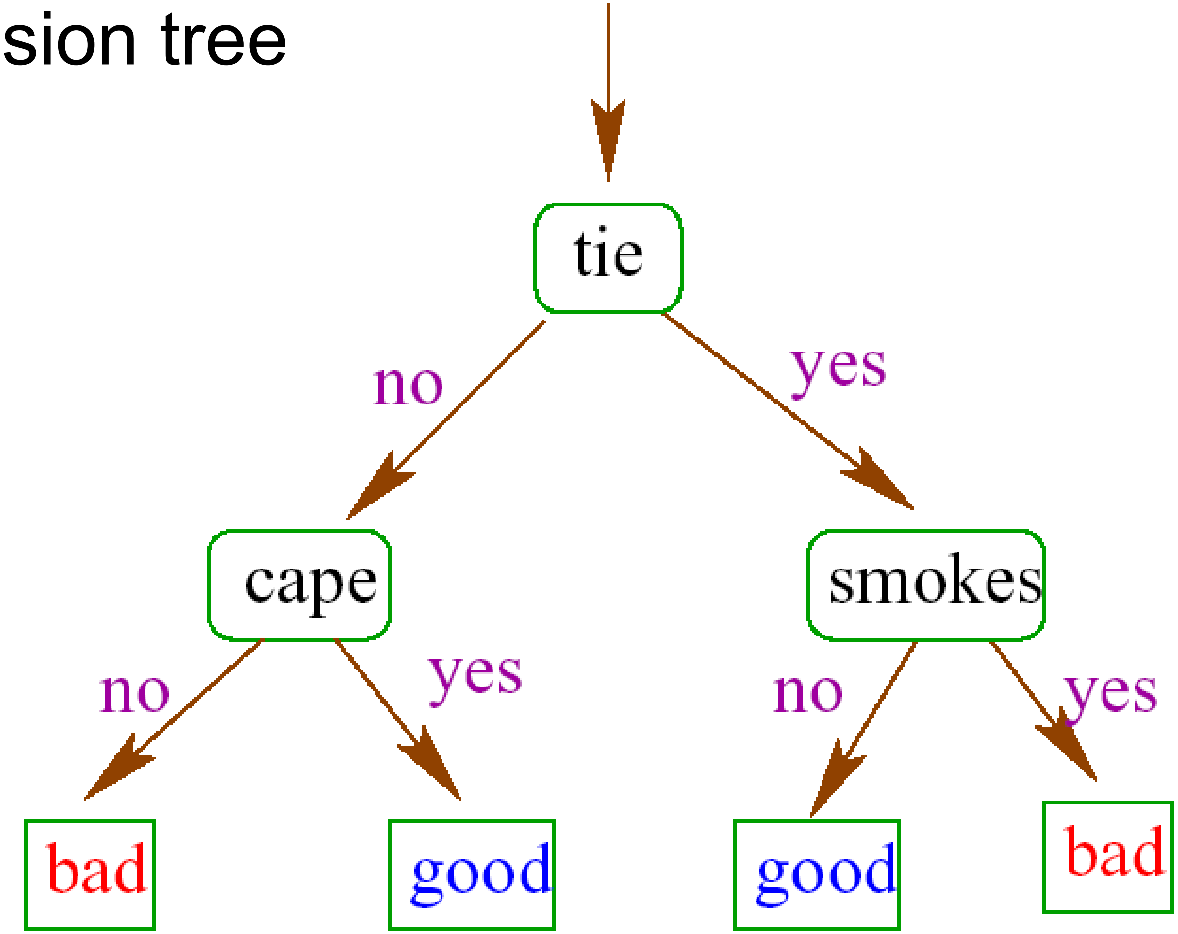
- Example: predict “good” vs. “bad”   [R. Schapire]



	sex	mask	cape	tie	ears	smokes	class
training data							
batman	male	yes	yes	no	yes	no	Good
robin	male	yes	yes	no	no	no	Good
alfred	male	no	no	yes	no	no	Good
penguin	male	no	no	yes	no	yes	Bad
catwoman	female	yes	no	no	yes	no	Bad
joker	male	no	no	no	no	no	Bad
test data							
batgirl	female	yes	yes	no	yes	no	??
riddler	male	yes	no	no	no	no	??

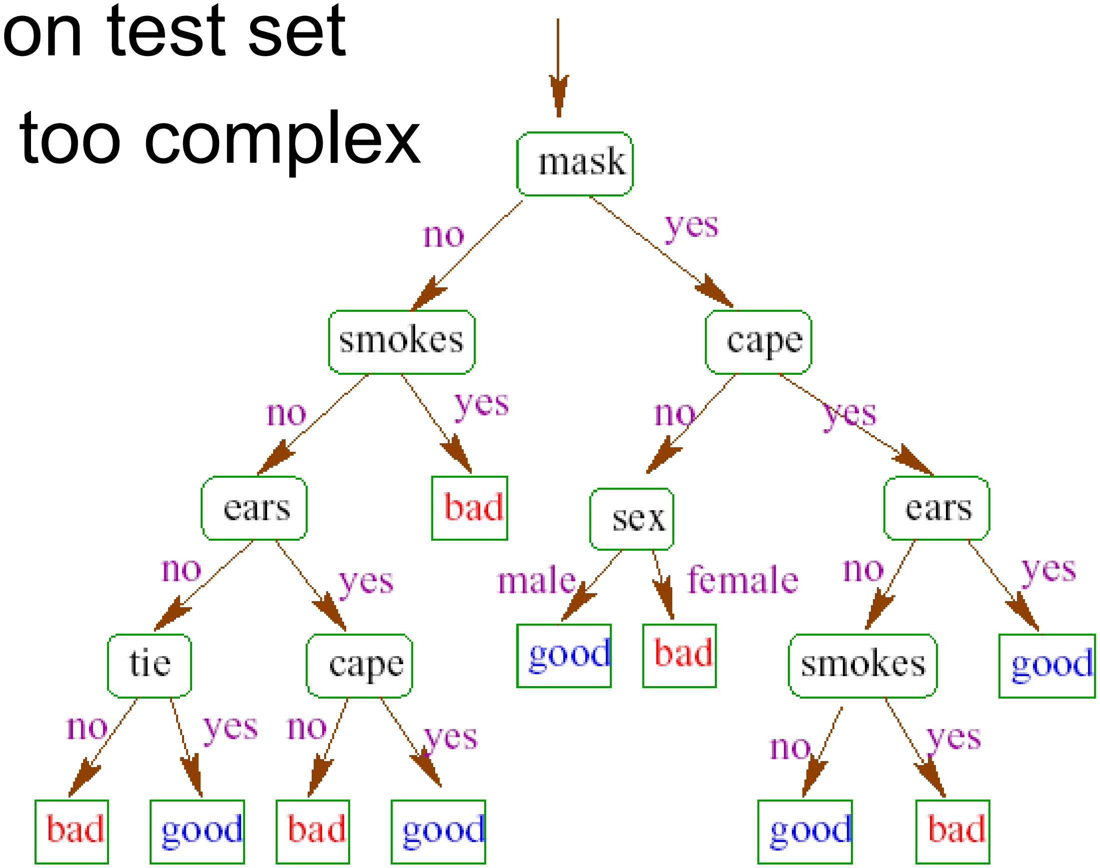
An example classifier

- Decision tree



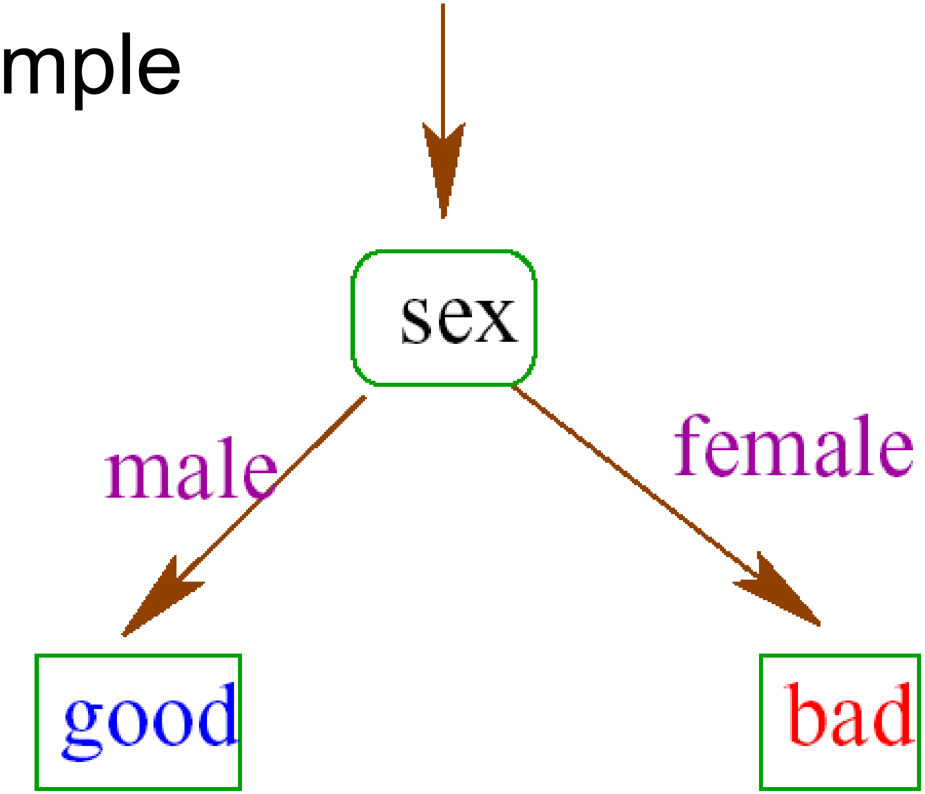
Another possible classifier

- Perfectly classifies training data, makes mistakes on test set
- Intuitively too complex



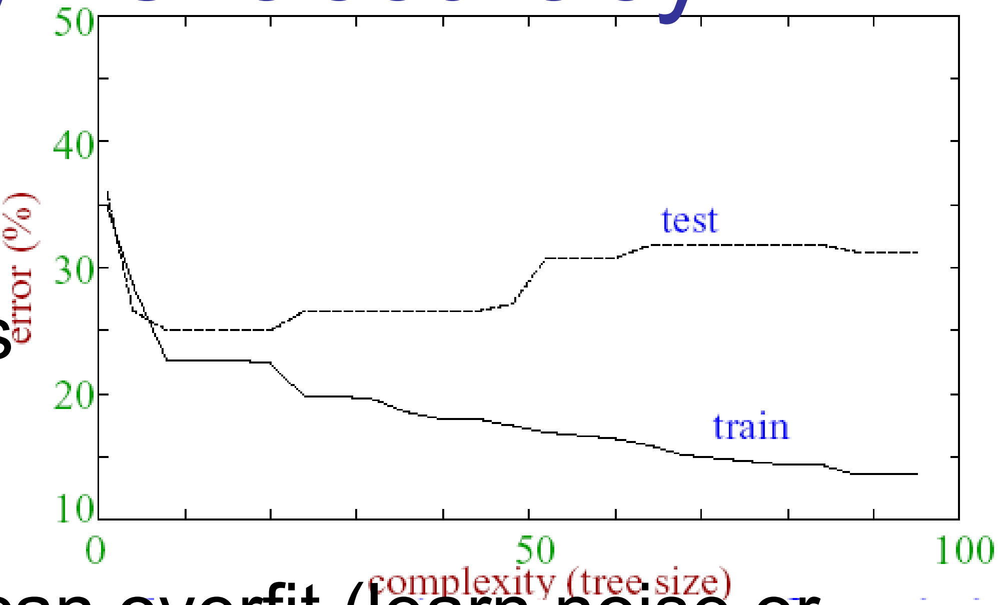
Yet another classifier

- Fails to fit from training data
- Overly simple



Complexity vs. accuracy

- Classifiers must be expressive enough to capture “true” patterns in training data…



- …but if too complex, can overfit (learn noise or spurious patterns)
- Problem: Can’t tell best classifier from training error
- *Controlling overfitting* is central problem of ML

Conditions for accurate learning

- To learn an accurate classifier, need
  - Enough training examples
  - Good performance on training set
  - Control over “complexity” (Occam’s razor)
- Measure complexity by:
  - Minimum description length (number of bits needed to encode rule)
  - Number of parameters
  - VC dimension

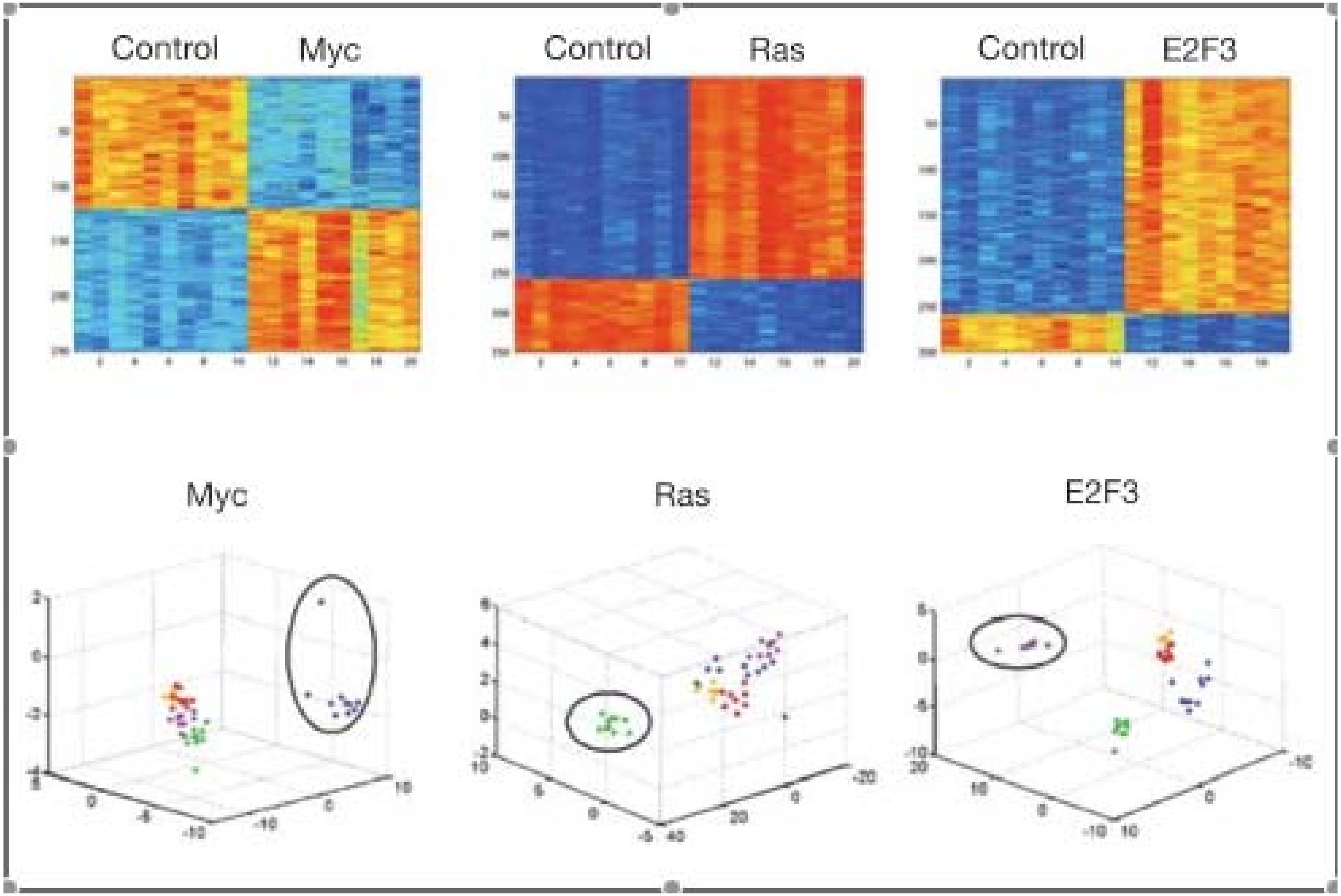
Cancer classification

- Training data: expression data from different tumor types; few examples, high dimensional feature space
- Goals:
  - (Accurately predict tumor type)
  - Learn *gene signature* = smaller set of whose expression pattern discriminates between classes
- “Feature selection” problem

Oncogenic pathways

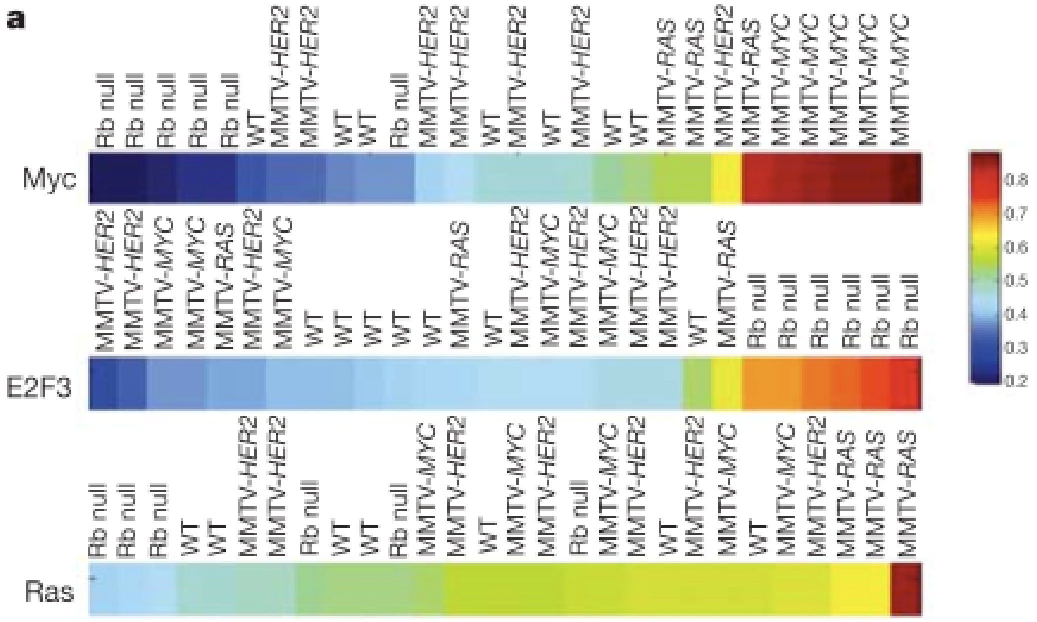
- [Nevins lab, Nature 2006]
- Training data:
  - Human cell cultures where specific oncogenic pathway has been activated vs. control cells (Myc, Ras, E2F3, etc)
- Prediction scores ↔ probability/confidence that pathway is activated in sample
- Test data:
  - Mouse models for pathways
  - Human cancer cell lines

Pathway signatures



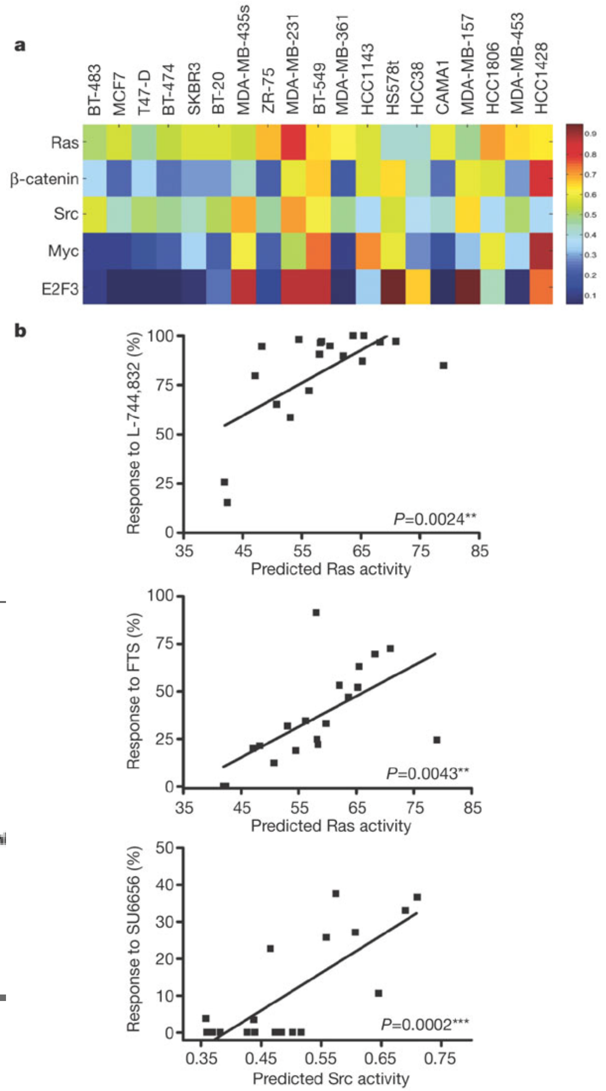
Prediction in mouse models

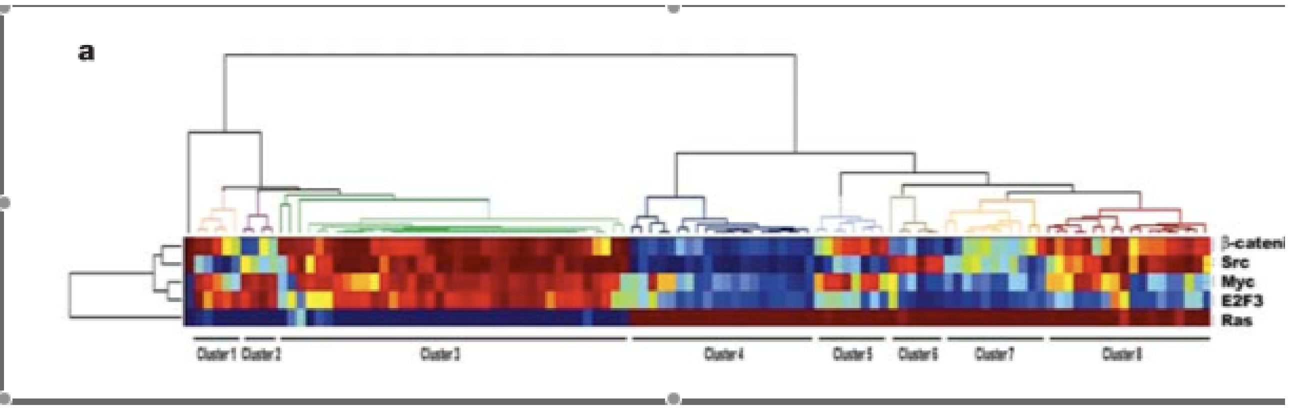
- Rank tumors from mouse models using trained pathway vs control classifiers



Prediction scores as features

- Oncogenic pathway prediction scores used to represent tumors for clustering
- Pathway scores on cell lines correlate with response to inhibitors





Support vector machines

- SVMs are a family of algorithms for learning a *linear classification* rule from labeled training data

  $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, $y_i = 1$ or -1

- Well-motivated by learning theory
- Various properties of the SVM solution help *avoid overfitting*, even in very *high dimensional* feature spaces

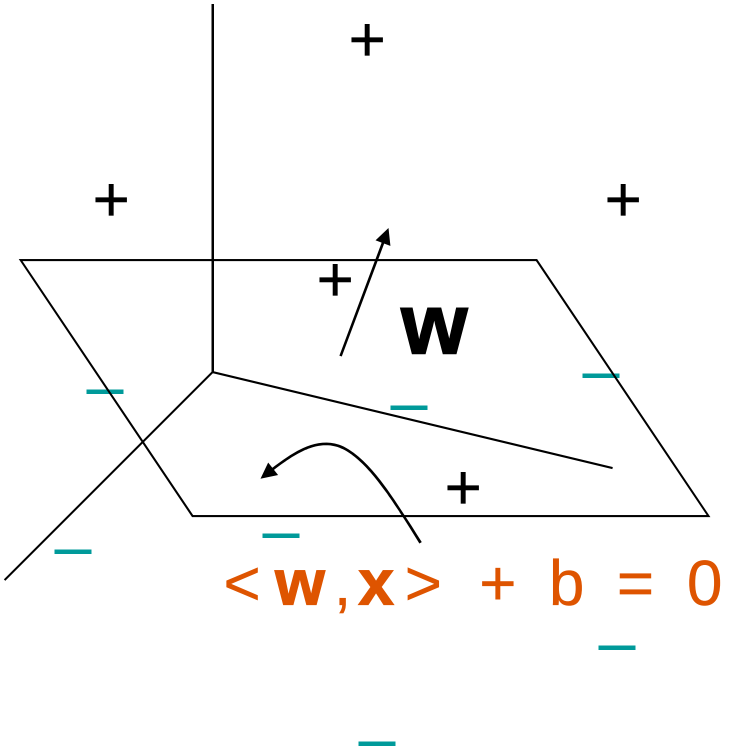
Vector space preliminaries

- Inner product of two vectors:

$<\mathbf{w},\mathbf{x}> = \Sigma_g w_g x_g$

- Hyperplane with normal vector **w** and bias b:

$<\mathbf{w},\mathbf{x}> + b = 0$



Linear classification rules

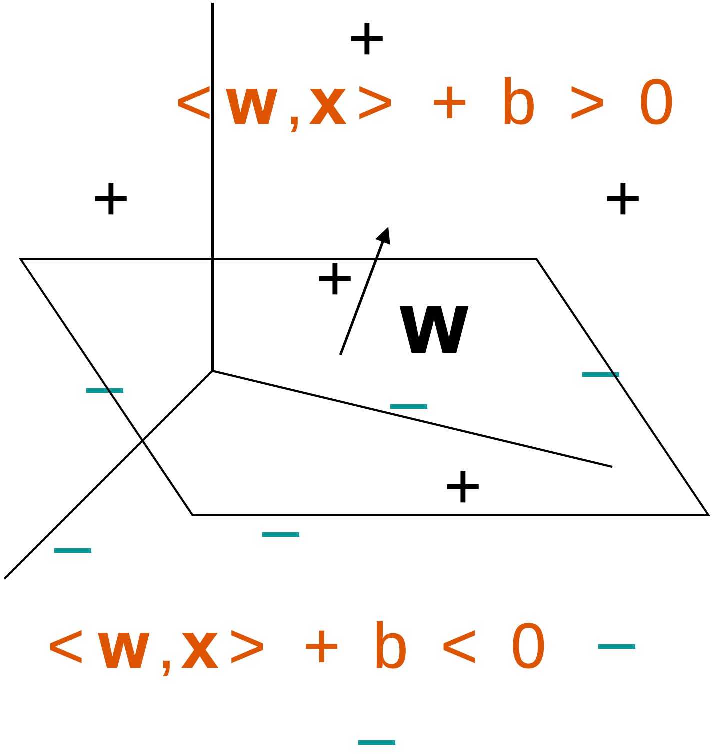
- SVMs consider only linear classifiers:

  $$f_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

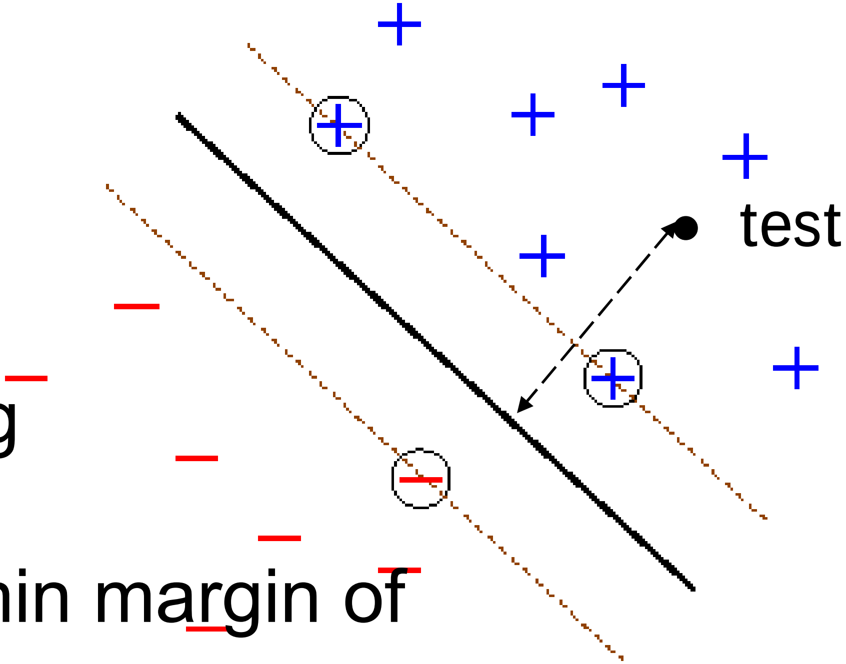
- Leads to linear prediction rules:

  $$h_{\mathbf{w},b}(\mathbf{x}) = \text{sign}(f_{\mathbf{w},b}(\mathbf{x}))$$

- Decision boundary is a *hyperplane*
- Prediction score $f_{\mathbf{w},b}(\mathbf{x})$ interpreted as "confidence" in prediction



Support vector machines

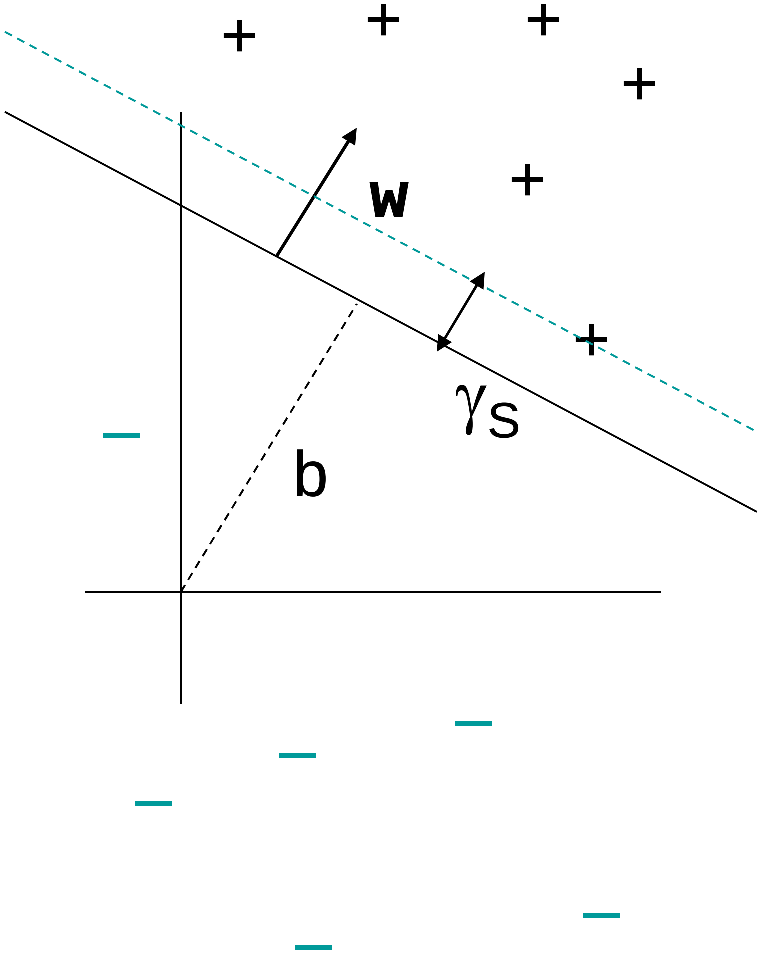


- Assume linearly separable training data
- Margin of example = distance to separating hyperplane
- Margin of training set = min margin of examples
- Choose (unique) hyperplane that *maximizes the margin*
- Prediction score for test example f(x) ~ signed distance of x to hyperplane

Geometric margin

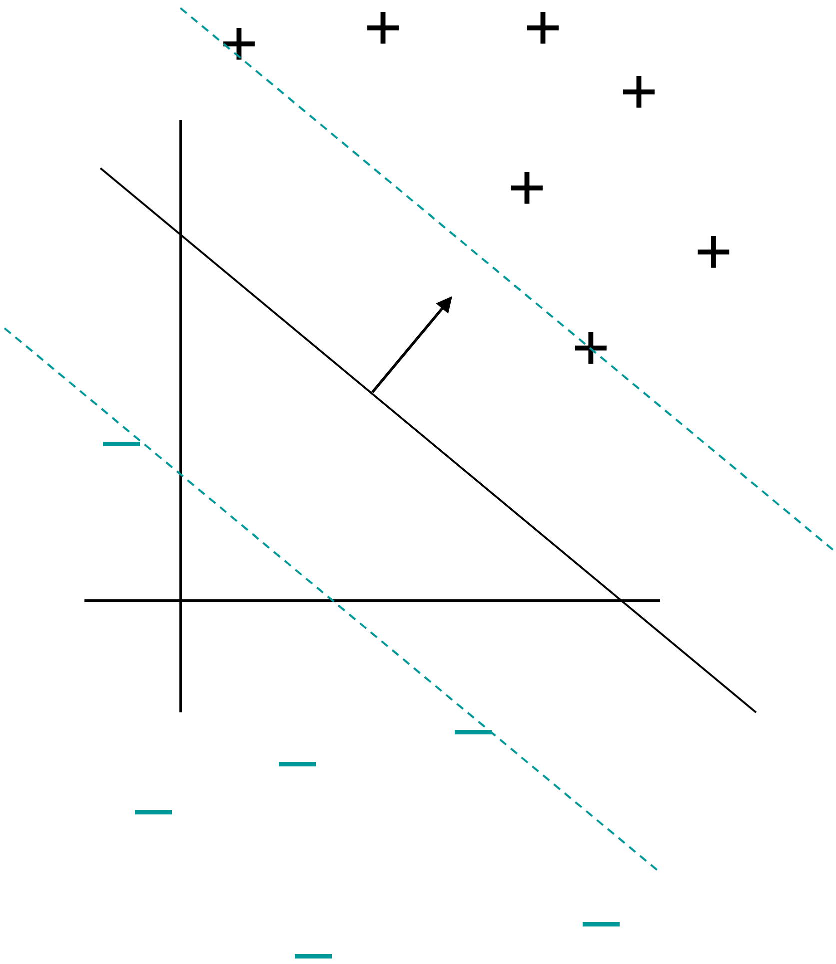
- Consider training data S and a particular linear classifier $f_{\mathbf{w},b}$
- If $||\mathbf{w}|| = 1$, then the *geometric margin* of training data for $f_{\mathbf{w},b}$ is

$\gamma_S = \mathrm{Min}_S\ y_i\ (\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$



Maximal margin classifier

- Hard margin SVM: given training data S, find linear classifier $f_{\mathbf{w},b}$ with *maximal geometric margin* γ_S
- Solve optimization problem to find **w** and b that give maximal margin solution



Hard margin SVMs

- Equivalently, enforce a *functional margin* ≥ 1 for every training vector, and minimize ||**w**||
- Primal problem:

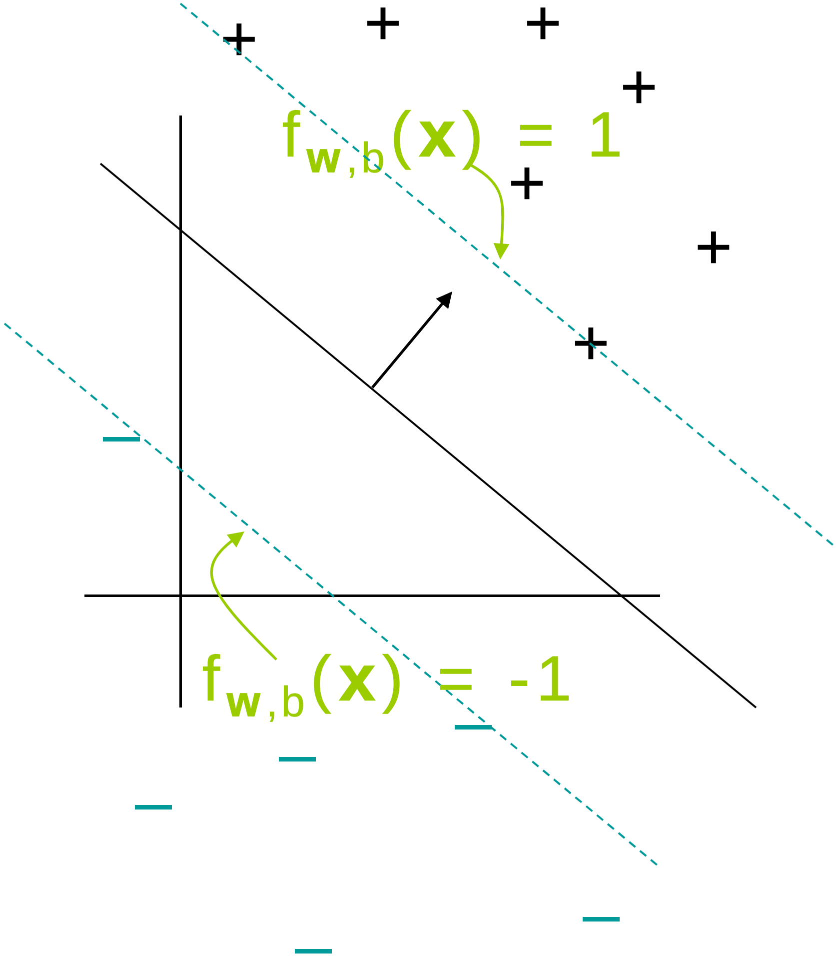
Minimize

$$\frac{1}{2} \langle \mathbf{w},\mathbf{w} \rangle$$

subject to

$$y_i (\langle \mathbf{w},\mathbf{x}_i \rangle + b) \geq 1$$

for all training vectors $\mathbf{x}_i$



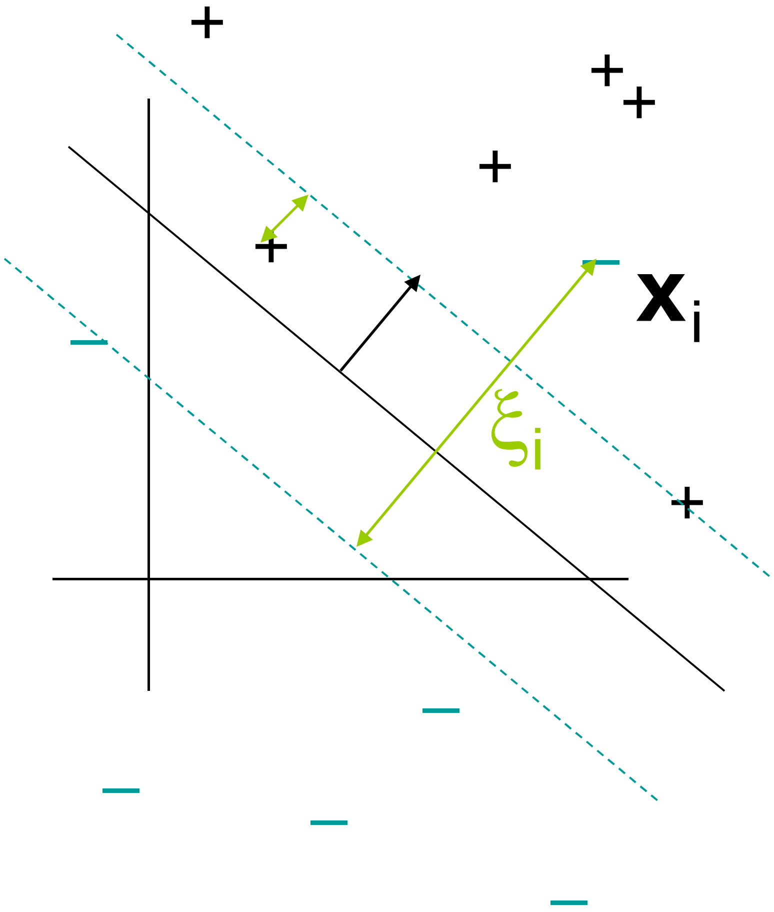
Non-separable case

- If training data is not linearly separable, can:
  - Penalize each example by the amount it violates the margin (“soft margin SVM”)
  - Map examples to a higher dimensional space where data is separable
  - Combination of above 2 solutions

Soft margin SVMs

- Introduce slack variable ξ_i to represent margin violation for training vector $\mathbf{x}_i$
- Now constraint becomes:

  $y_i(<\mathbf{w},\mathbf{x}_i>+b) \geq 1- \xi_i$



Soft margin SVMs

- Primal optimization problem becomes:

  Minimize

  $\frac{1}{2} \langle \mathbf{w},\mathbf{w} \rangle + C \sum_i \xi_i$ *("1-norm")* ← LIBSVM

  or

  $\frac{1}{2} \langle \mathbf{w},\mathbf{w} \rangle + C \sum_i \xi_i^2$ *("2-norm")* ← SVM-light

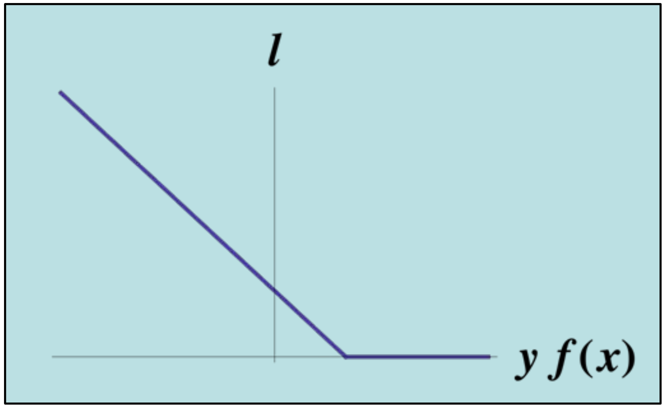
  subject to

  $y_i(\langle \mathbf{w},\mathbf{x}_i \rangle + b) \geq 1 - \xi_i \,,\ \xi_i \geq 0$

- C: "trade-off" parameter

Regularization viewpoint

- Trade-off optimization problem (1-norm soft margin): minimize



$$||\mathbf{w}||^2 + C\ \Sigma_i\ (1 - y_i\, f_{\mathbf{w},b}(x_i))_+$$

  - $(1 - y\ f(x))_+$: "hinge loss", penalty for margin violation
  - $||w||^2$: "regularization term"; intuitively, prevents overfitting by constraining **w**

Properties of SVM solution

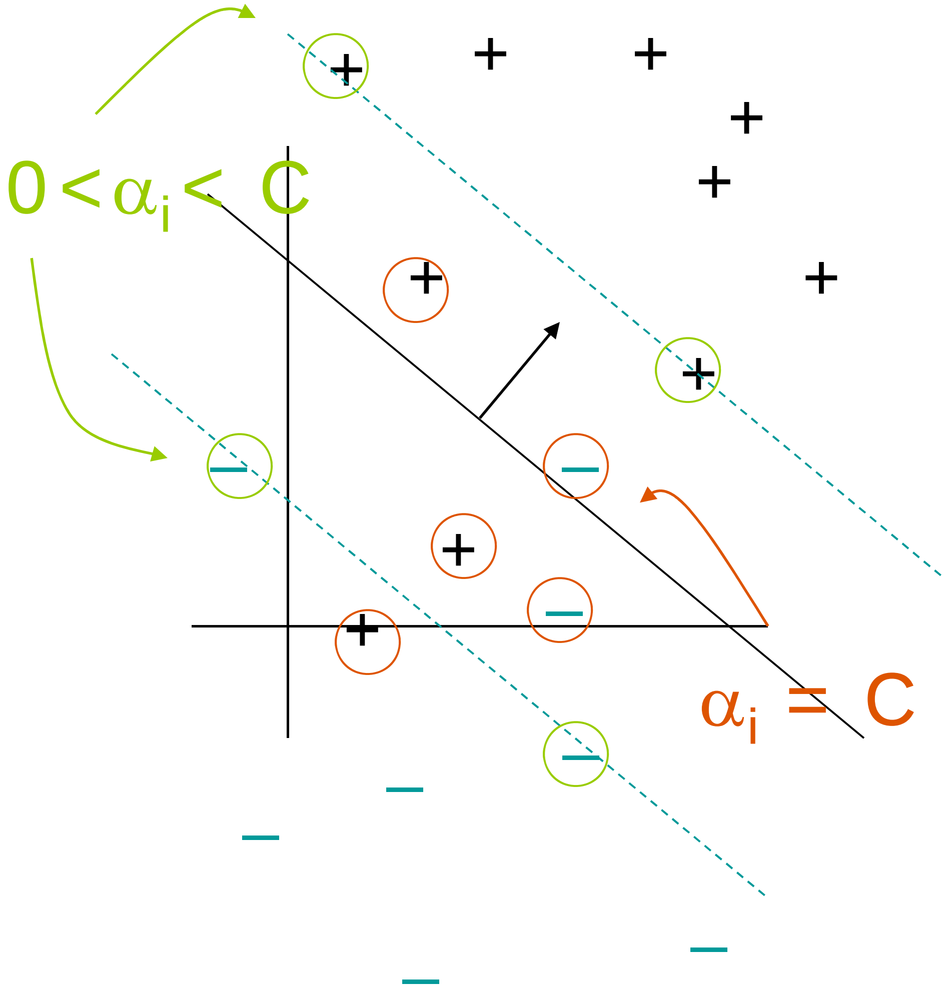
- Introduce dual variable (“weight”) α_i for each constraint, i.e. for each training example
- Solve dual optimization problem to find α_i
  - Convex quadratic problem → unique solution, good algorithms
- $\mathbf{w} = \Sigma_i \alpha_i y_i \mathbf{x}_i$
  - Normal vector is linear combination of support vectors, i.e. training vectors with $\alpha_i > 0$

Support vectors

- If $\mathbf{x}_i$ has margin > 1, $\alpha_i = 0$

1-norm SVM: two kinds of support vectors

- If $\mathbf{x}_i$ has margin = 1, $0 < \alpha_i < C$
- If $\mathbf{x}_i$ has margin < 1, $\alpha_i = C$



Feature selection

- How to extract a “cancer signature”?
- Simplest feature selection: filter on training data
  - E.g. Apply t-test or Fisher’s criterion to find genes that discriminate between classes
  - Train SVM on reduced feature set
- Usually better to use results of training to select features

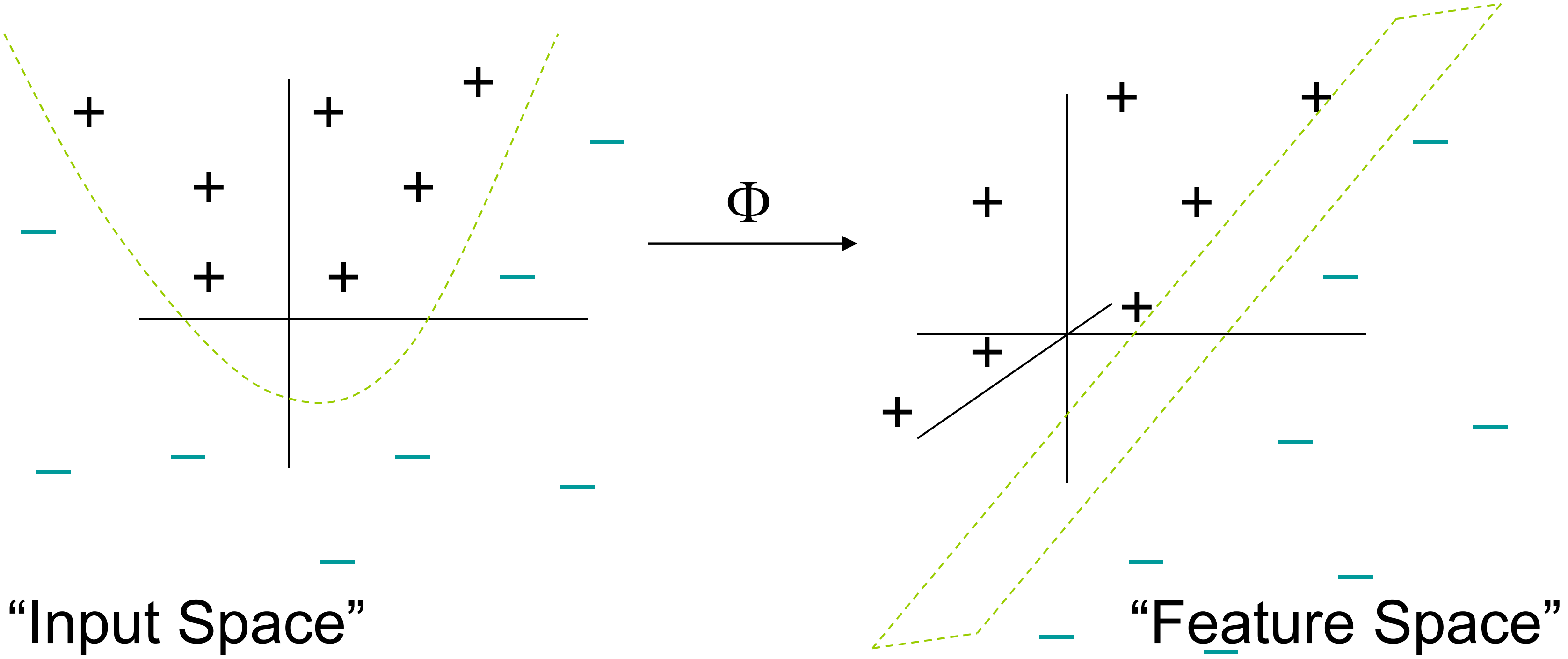
Ranking features

- Normal vector $\mathbf{w} = \Sigma_i \alpha_i y_i \mathbf{x}_i$ gives direction in which prediction scores change
- Rank features by $|w_g|$ to get most significant components
- Recursive feature elimination (RFE): iteratively
  - Throw out bottom half of genes ranked by $|w_g|$
  - Retrain SVM on remaining genes

  Induces ranking on all genes

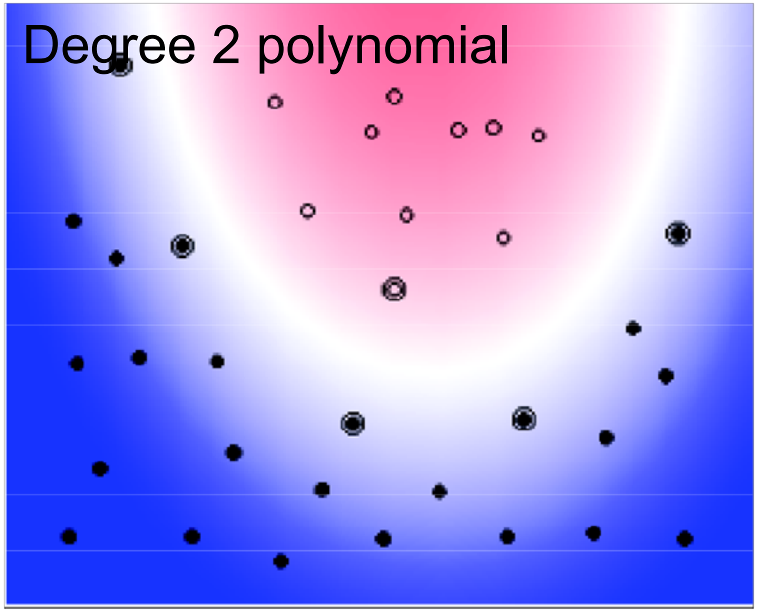
Kernel trick

- Idea: map to higher dimensional feature space
- Only need *kernel* values: $K(\mathbf{x}_1,\mathbf{x}_2) = \Phi(\mathbf{x}_1) \bullet \Phi(\mathbf{x}_2)$ to solve dual optimization problem

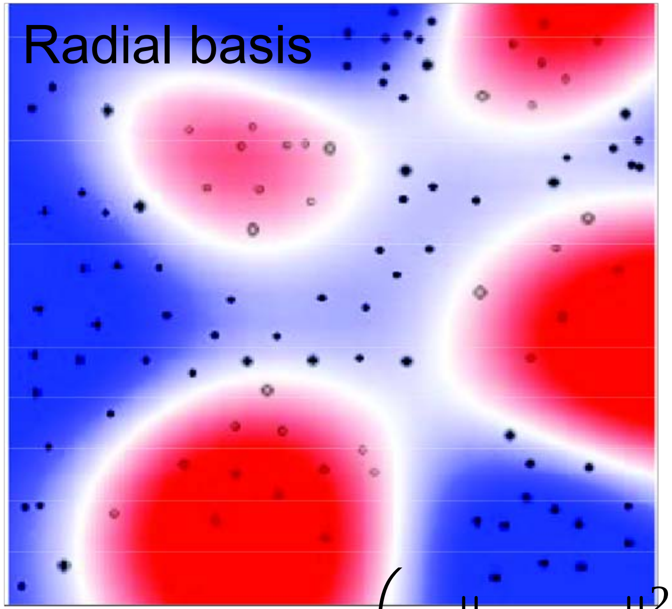


Examples of kernels

- Large margin non-linear decision boundaries
- Not needed with expression data



$$K(x,z) = (x \bullet z + C)^2$$



$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{\sigma^2}\right)$$

Discussion issues for paper

- How well-defined is a cancer signature?
  - How stable is feature selection on small data set?
  - Empirical validation of gene set, number of genes?
- Which analyses are purely training data results, which show prediction performance?
- Significance of prediction performance?
  - Traditional ML does not assert significance via a *p*-value but comparison against other methods
  - Can compare to a baseline method, e.g. single oncogene expression level

Be careful!

- Potti et al., Nature Medicine 2006: Similar analysis to predict response to chemotherapy, based on NCI 60 cell line data
- Coombed et al., Nature Medicine 2007: “Bioinformatics forensics”, unable to reproduce results
  - Mislabeling of samples (+ vs -)
  - Off-by-one indexing error, wrong genes in signature
  - No separation of training and test for feature reduction (“metagene”), not strictly inductive learning
- Summary: poor computational practices and (probably) overfitting lead to erroneous results