
Computational functional genomics

(Spring 2002: Lecture 11)

David K. Gifford

(Adapted from a lecture by Tommi S. Jaakkola)

MIT LCS and AI Lab

gifford@mit.edu

Classification approaches

- The two main classification approaches
 1. Generative approach
 - build a statistical model
e.g., mixture model
 2. Discriminative approach
 - specify a decision rule/boundary directly
e.g., logistic regression

Discriminative approach: motivation

- One reason for using discriminative approaches is **robustness**:

Class conditional models	type of decision boundary
Gaussian, equal covariances	linear
Independent exponential	linear
Independent binomial	linear
...	...

If we estimate a linear decision boundary directly we are less dependent on what the true class conditional distributions are

- Examples of discriminative classifiers
 - linear discriminant analysis
 - Logistic regression (generalized linear models generalized additive models)
 - Support vector machines

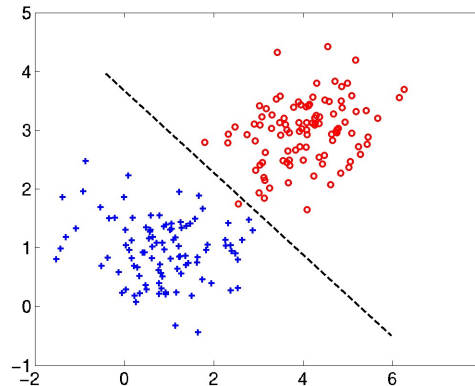
Discriminative approach to classification

- Simple example: **linear decision boundaries**

$$f(x; \theta) = \theta_0 + \theta_1 x_1 + \dots + \theta_m x_m > 0, \text{ class} = 1 \quad (1)$$

$$\leq 0, \text{ class} = 0 \quad (2)$$

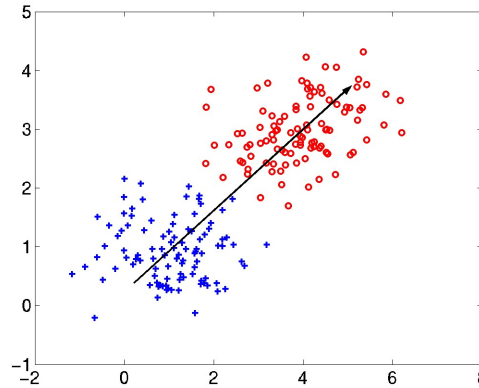
parameters $\theta = \{\theta_0, \theta_1, \dots, \theta_m\}$ and $x = \{x_1, \dots, x_m\}$ is a vector of expression levels.



- Similarly to the generative model case, we have to solve the
 1. estimation problem
 2. variable selection problem

Fisher linear discriminant analysis

- Set the parameters θ so that the two class populations are maximally separated along this direction



- We try to maximize

$$J(\theta) = \frac{(\text{Separation of means along } \theta)^2}{\text{Sum of within population variances along } \theta} \quad (3)$$

In fact, this is exactly the same as assuming two Gaussian distributions with equal covariances

Logistic regression

- In the logistic regression model, the log-odds of decisions is a linear function of the explanatory variables

$$\log \frac{P(\text{class} = 1|x, \theta)}{P(\text{class} = 0|x, \theta)} = \theta_0 + \theta_1 x_1 + \dots + \theta_m x_m \quad (4)$$

Note that the value of the discriminant function (distance away from the boundary) is now interpreted as our confidence in the decisions

- How do we estimate logistic regression models from observed data?

Logistic regression: estimation

- We can estimate the parameters θ by maximizing the likelihood of the training labels $\{c^{(1)}, \dots, c^{(n)}\}$ corresponding to the measurements $\{x^{(1)}, \dots, x^{(n)}\}$

$$\prod_t P(\text{class} = c^{(t)} | x^{(t)}, \theta) \quad (5)$$

- Suppose the labels are binary $c^{(t)} \in \{0, 1\}$. We can derive a simple incremental update rule for the parameters

$$\theta_i \leftarrow \theta_i + \epsilon \cdot (c^{(t)} - P(\text{class} = 1 | x^{(t)}, \theta)) \cdot x_i^{(t)} \quad (6)$$

where ϵ is a **learning rate**.

Logistic regression: regularization

- In the absence of any data, we'd like the parameters to go to zero (equal probability class predictions)

We add a penalty term (prior) to the likelihood criterion that encourages small parameter values

$$\max_{\theta} \left\{ \log\text{-likelihood}(\theta) + \overbrace{\log\text{-prior}(\theta)}^{-C \cdot \|\theta\|^2 / 2} \right\} \quad (7)$$

- With such regularization the learning/update rule changes only slightly

$$\theta_i \leftarrow \theta_i + \epsilon \cdot \left[\left(c^{(t)} - P(\text{class} = 1 | x^{(t)}, \theta) \right) \cdot x_i^{(t)} - C \cdot \theta_i \right] \quad (8)$$

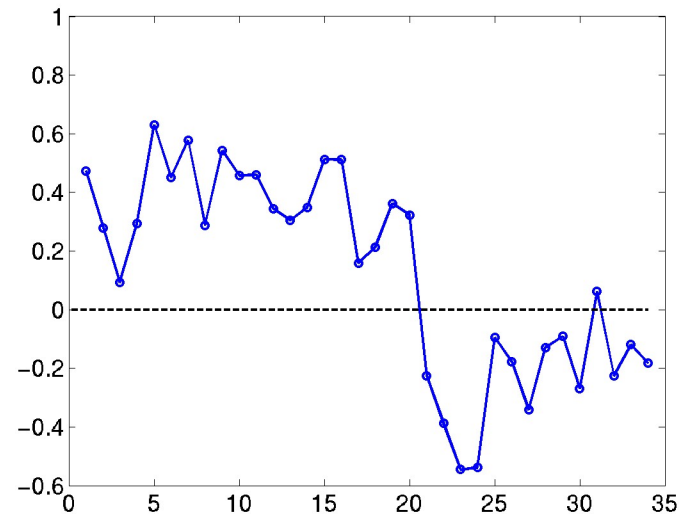
where $C > 0$ is a regularization parameter.

Logistic regression: example

- Golub et al. leukemia classification problem
 - 7130 ORFs
 - 38 labeled training examples,
 - 34 test examples
- If we apply the regularized logistic regression model without variable selection, we get **1 test error (out of 34)**.

I cheated just a bit ...

Logistic regression: example



The figure shows the values of the discriminant function

$$f(x; \theta) = \theta_0 + \theta_1 x_1 + \dots + \theta_m x_m \quad (9)$$

across the test examples

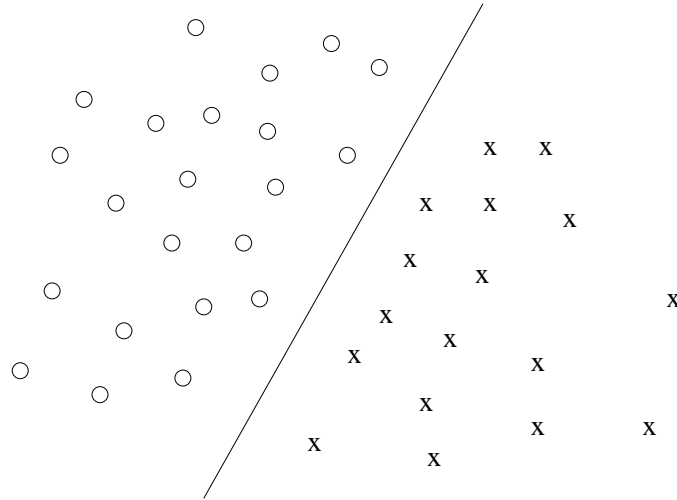
- Variable selection?

Support vector machines

- Basics of support vector machines
 - optimal hyperplane
 - finding the optimal hyperplane
 - kernel function
 - complexity
- Examples

“Optimal” hyperplane

- Let's assume for simplicity that the classification problem is **linearly separable**



- Maximum margin hyperplane** is maximally removed from all the training examples
- This hyperplane can be defined on the basis of only a few training examples called **support vectors**

“Optimal” hyperplane cont’d

- We are estimating a linear classifier:

$$\begin{aligned} f(x; \Theta) &= \theta_0 + x_1\theta_1 + \dots + x_d\theta_d \\ &= \theta_0 + \theta \cdot x \end{aligned} \tag{10}$$

where $\Theta = \{\theta_0, \theta\}$

- We can try to find the “optimal” hyperplane by requiring that the sign of the decision boundary (clearly) agrees with all the training labels

$$y^{(t)} [\theta_0 + \theta \cdot x^{(t)}] \geq 1, \quad t = 1, \dots, n \tag{11}$$

where the labels $y^{(t)}$ are ± 1 .

BUT...

- this is actually an alternative definition of linear separability
- there are multiple answers
- larger values of θ_0, θ would yield larger separation.

Support vector machine

- We find the smallest parameter values that still satisfy the classification constraints
- We minimize

$$\|\theta\|^2/2 = \sum_{j=1}^d \theta_j^2/2 \quad (12)$$

subject to the classification constraints

$$y^{(t)} [\theta_0 + \theta \cdot x^{(t)}] \geq 1, \quad t = 1, \dots, n \quad (13)$$

- Only a few of the classification constraints are relevant
⇒ support vectors

Solution*

- Solving for $\{\theta_0, \theta\}$ leaves us with the following (dual) optimization problem over Lagrange multipliers associated with the constraints:

We maximize

$$J(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)} \cdot x^{(j)}) \quad (14)$$

subject to the constraints

$$\alpha_i \geq 0, \quad i = 1, \dots, n, \quad \sum_{i=1}^n \alpha_i y^{(i)} = 0 \quad (15)$$

(For non-separable problems we simply limit $\alpha_i \leq C$ for some positive constant C)

- This is a **quadratic programming problem**

Interpretation of support vector machines

- Before:
 - example vectors $x^{(t)}$ of dimension m (the number of genes)
 - parameters $\theta_0, \dots, \theta_m$ which multiply **each component of x** (genes)
- After:
 - **real valued inner products** $(x^{(t)} \cdot x^{(t')})$ measuring how similar the training examples are
 - **weights α_i on the examples** indicating how important each training example is to the classification task

Interpretation of support vector machines cont'd

- To use support vector machines we
 1. specify similarities between the examples (i.e., $(x \cdot x')$)
 2. set the example weights $\{\alpha_i\}$ by enforcing the classification constraints.
- We make decisions by comparing each new sample x with **only** the k support vectors $x^{(t_1)}, \dots, x^{(t_k)}$

$$\hat{y} = \text{sign} \left(\sum_{i=1}^k \underbrace{\hat{\alpha}_i}_{\text{weight}} y^{(t_i)} \underbrace{(x^{(t_i)} \cdot x)}_{\text{similarity}} + \theta_0 \right) \quad (16)$$

Non-linear classifier

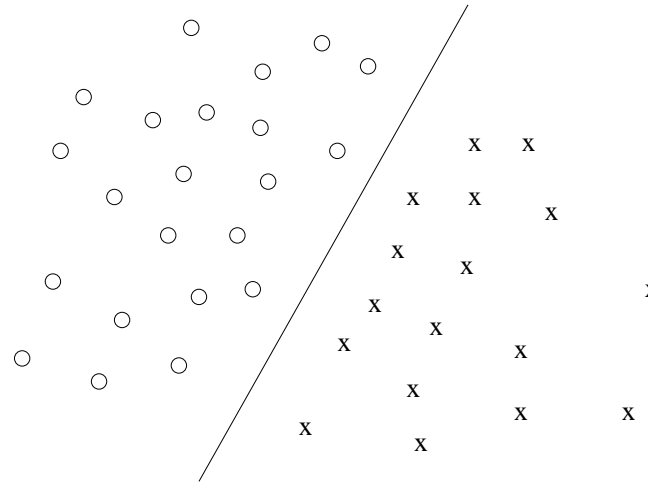
- So far the SVM classifier is able to separate our sample populations only linearly
- We can easily obtain a non-linear classifier by mapping our samples $x = [x_1 \ x_2]$ into longer feature vectors $\phi(x)$

$$\phi(x) = [x_1^2 \ x_2^2 \ \sqrt{2}x_1x_2 \ \sqrt{2}x_1 \ \sqrt{2}x_2 \ 1] \quad (17)$$

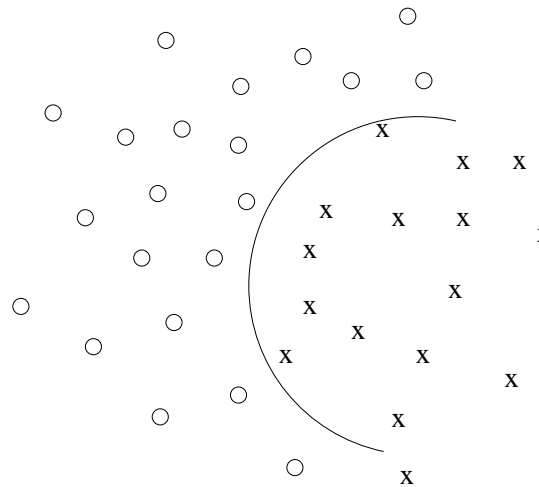
and applying the linear classifier to $\phi(x)$ instead

- This way we can for example take into account dependencies among the genes to better classify tissue samples

Non-linear classifier



Linear separator in the **feature space**



Non-linear separator in the **original space**

Kernel function and feature mapping

- Let's look at the previous example in a bit more detail

$$x \rightarrow \phi(x) = [x_1^2 \ x_2^2 \ \sqrt{2}x_1x_2 \ \sqrt{2}x_1 \ \sqrt{2}x_2 \ 1] \quad (18)$$

- If we try to find the “optimal” hyperplane in the feature space, i.e., using $\phi(x)$ as the observed examples, we have to deal with (only) the inner products (kernels) between such feature vectors

$$\begin{aligned} \phi(x) \cdot \phi(x') &= x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x_2 x_1' x_2' + 2x_1 x_1' + 2x_2 x_2' + 1 \\ &= (1 + x_1 x_1' + x_2 x_2')^2 \\ &= (1 + (x \cdot x'))^2 \end{aligned} \quad (19)$$

But these can be evaluated without ever explicitly constructing the feature vectors $\phi(x)$!

Other examples of kernel functions

- There are a number of feature mappings that behave in nicely in this way

- **Linear kernel**

$$K(x, x') = (x \cdot x') \quad (20)$$

- **Polynomial kernel**

$$K(x, x') = \left(1 + (x \cdot x')\right)^p \quad (21)$$

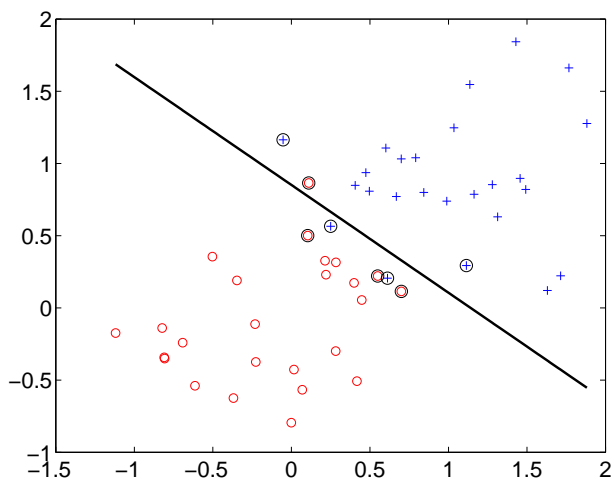
where $p = 2, 3, \dots$. To get the feature vectors we concatenate all p^{th} order polynomial terms of the components of \mathbf{x} (weighted appropriately)

- **Radial basis kernel**

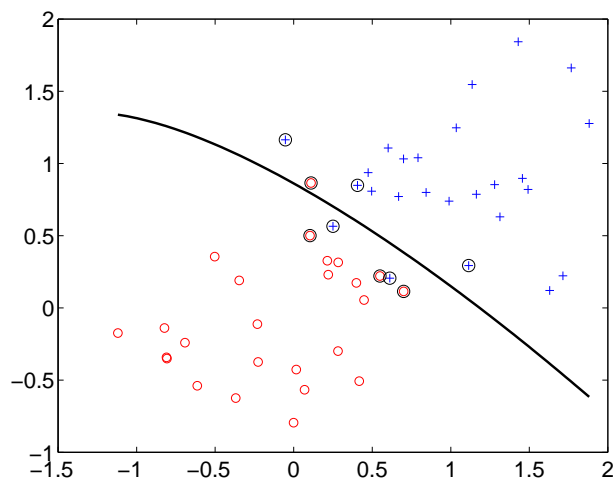
$$K(x, x') = \exp\left(-\frac{1}{2}\|x - x'\|^2\right) \quad (22)$$

In this case the feature space consists of functions and results in a *non-parametric* classifier.

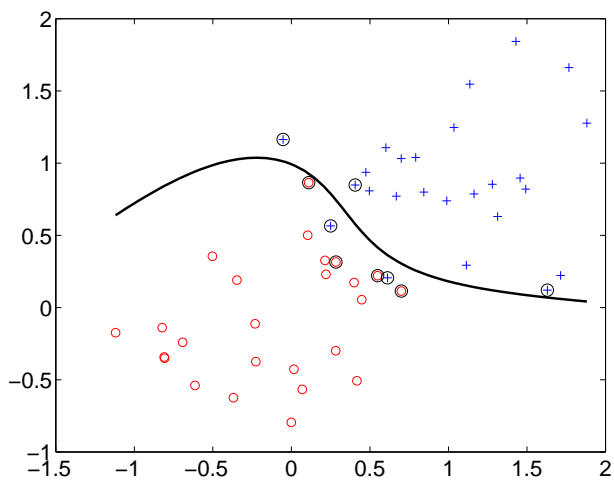
SVM examples



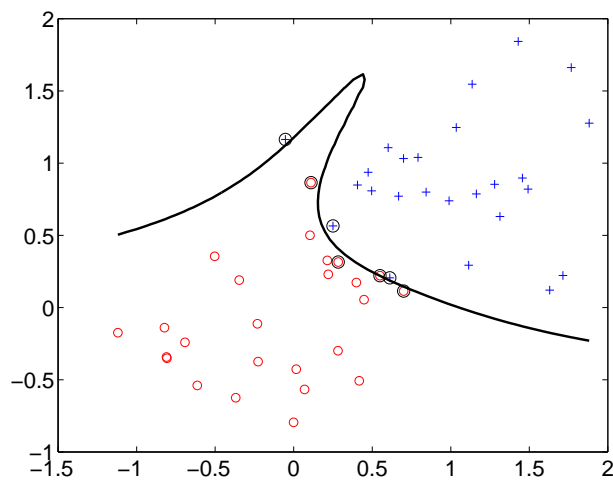
linear



2nd order polynomial



4th order polynomial



8th order polynomial

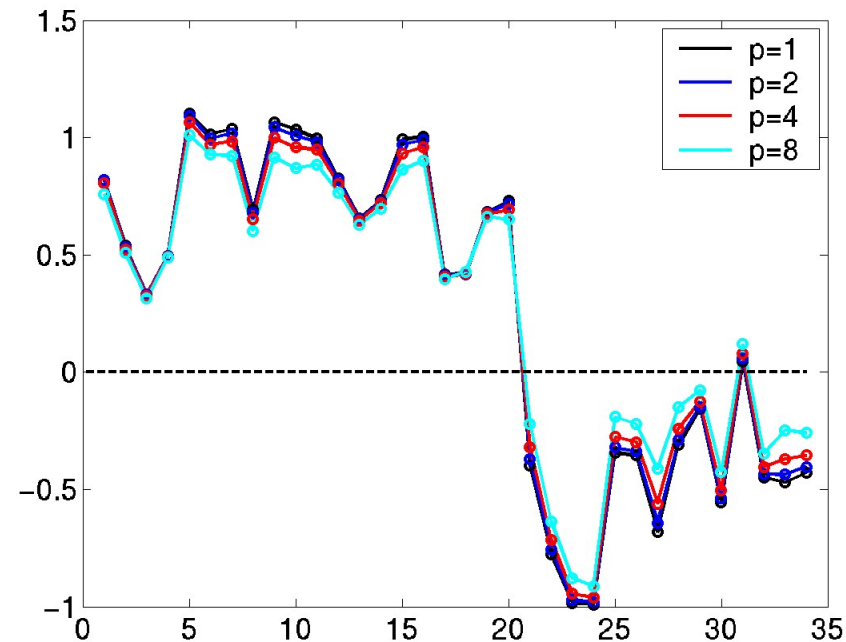
Support vector machine results

- Golub et al. leukemia classification problem
 - 7130 ORFs
 - 38 labeled training examples,
 - 34 test examples
- Let's blindly apply SVMs to this problem using polynomial kernels of degree $p = 1, 2, 4, 8$.

We get **1 test error** for all classifiers regardless of their complexity

There doesn't seem to be much overfitting...

Support vector machine results cont'd



- The figure shows the discriminant function values for the test samples resulting from polynomial kernels of degree $p = 1, 2, 4, 8$