Presentation for Stat 526
Jochen Brumm

April 6, 2003

- genomic studies often try to find the function of a gene by comparing its behaviour to other genes with known function
- an example: time-course in yeast
- would often be analyzed with cluster analysis; we'll try to see if we can find a suitable classifier

- choosing the “right” classifier not necessarily an easy problem
- black-box approach: limited usefulness
- have to cut the box open to make some progress
- that’s how this example is tied in to this lecture: use the “scissors” we aquired in this class

- introduce our classification problem in more detail
- derive a classification algorithm that attempts to solve the problem (*support vector machines*)
- derive properties of that algorithm
- the smoothing-related issues are more towards the end

- *Problem 1:* can't expect linear separation
- *Problem 2:* don't have many labels (yeast o.k., human hardly any)
- solution to problem 1: go from linear boundaries to smooth boundaries
- we'll introduce it as an optimization problem (typical) and then show the relation to a smoothing problem
- we'll talk about problem 2 later

- linear rule:

$$\{f(x) = x^T\beta + \beta_0 = 0\}$$

- separation:

$$y_i f(x) > 0$$

  for all i

- find the optimal rule: maximum distance

- restate as an optimization problem:

$$\begin{aligned} \max_{\beta,\beta_0} \quad & C \\ \text{s.t.} y_i(x_i^T\beta + \beta_0) \quad & \geq C \quad i = 1, \ldots, N \end{aligned}$$

- now extend to non-separable case
- modify constraints:

$$\begin{aligned} y_i(x_i^T\beta + \beta_0) &\geq C(1-\xi_i) \quad i = 1, \ldots, N \\ \sum \xi_i &\leq cons; \qquad \xi_i \geq 0 \end{aligned}$$

- ξ are called “slack-variables”; $\xi_i > 1$ means misclassification
- $cons$ bounds the number of misclass. in training set

- now extend to non-linear case
- idea: find basis $h_m(x); m = 1, \ldots, M$ and fit linear solution with input features $h(x_i) = (h_1(x_i), \ldots, h_M(x_i))$ to produce non-linear function $f(x) = h(x)^T\beta + \beta_0$
- note that M could be very big; how to avoid overfitting ?
- typical smoothing issues; in fact this can be stated as such a problem

- can restate the problem from above as:

$$\min_{\beta_0,\beta} \sum_i [1 - y_i(h(x_i)^T\beta + \beta_0)]_+ + \lambda||\beta||^2$$

  with $\lambda = 1/2\gamma$

- this states the problem as a penalization method with λ as the tuning parameter

- the basis arises from the eigen-expansion of a positive definite kernel K; for SVM's we have typically d-th degree polynomial, radial and neural network kernels available

- let's see how this works...

- there is a useful C-library available: *libsvm*
- there is also an R-interface (in package *e1071*; I couldn't get that to run since it requires g++ $>$ 3.00 (and Redhat Linux 7.3 has some weird gcc 2.96 version). The pleasures of portable C-code ...
- turns out that I can compile the *libsvm* library directly without any problems and it is easy to use
- I've created some mixture data and tried different kernels and smoothing parameters on it

- now we turn our attention to the missing-labels problem
- examine the loss-function: 0 for $yf(x) \geq 1$; that is points well within their margin and linear penalty for points on the wrong side
- I'm not convinced that this is a good method for genome-classification: very few labels are known; so downweighting the things we know doesn't seem like a reasonable procedure
- also: no label - no information (similar to logistic regression)
- might be better off with a potentially misspecified mixture-model and Bayesian/full likelihood estimation; often have some reasonable guess at prior class memberships

- but certainly an efficient algorithm for difficult classification problems