Module 5: Classification

≈ perceptron + regularization

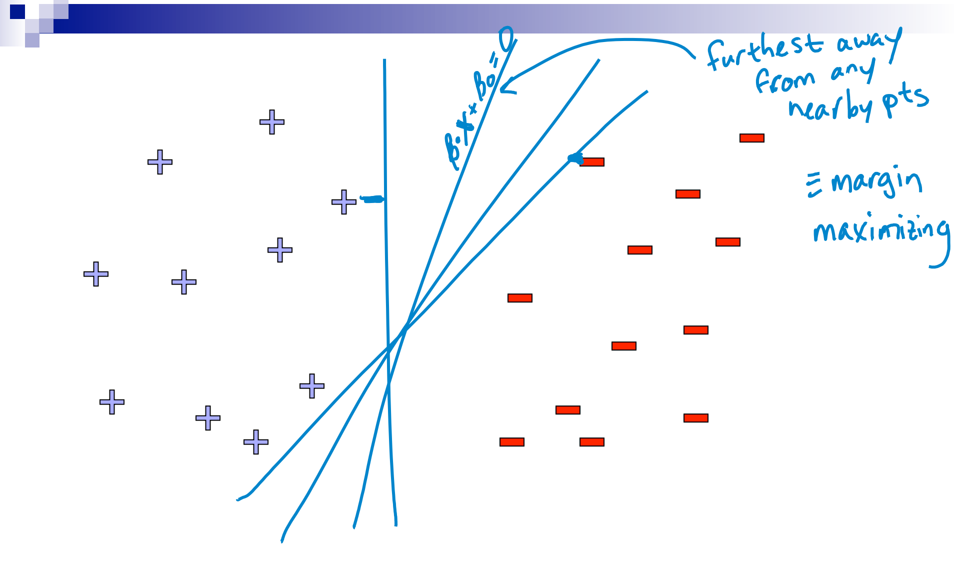
Support Vector Machines

STAT/BIOSTAT 527, University of Washington

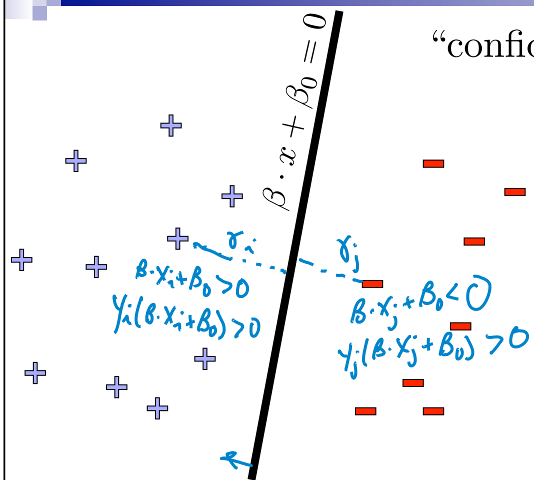
Emily Fox

June 6th, 2013

Linear classifiers – Which line is better?



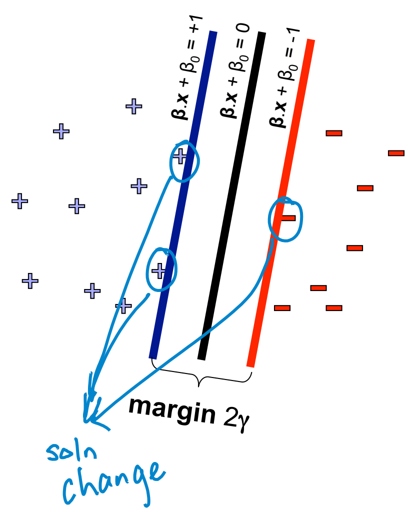
Pick the one with the largest margin!



"confidence" $= y_i(\beta \cdot x_i + \beta_0)$

pos. if no mistake

Support vector machines (SVMs)



$$\min_{\beta,\beta_0} \quad ||\beta||_2^2$$

$$y_i(\beta \cdot x_i + \beta_0) \geq 1, \forall i \in \{1, \ldots, n\}$$

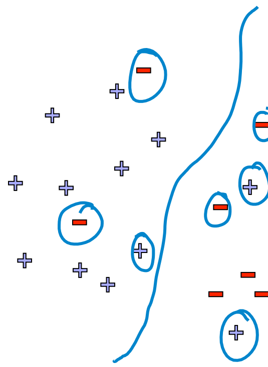
- Solve efficiently by many methods, e.g.,
  - quadratic programming (QP)
    - Well-studied solution algorithms
  - Stochastic gradient descent
- Hyperplane defined by support vectors

* only perturbations of these pts will change the soln (contrast w/ LDA, QDA)

What if the data are still not linearly separable?

$$\min_{\beta,\beta_0} \quad ||\beta||_2^2$$

$$y_i(\beta \cdot x_i + \beta_0) \geq 1, \forall i \in \{1, \dots, n\}$$



- If data are not linearly separable, some points don't satisfy margin constraint:

$\exists i$ s.t. $y_i(\beta \cdot x_i + \beta_0) < 1$ (violates)

or $1 - y_i(\beta \cdot x_i + \beta_0) > 0$

- How bad is the violation? like hinge loss

$$\text{constraint violation} = \begin{cases} 0 & \text{if } 1 - y_i(\beta \cdot x_i + \beta_0) < 0 \\ 1 - y_i(\beta \cdot x_i + \beta_0) & \text{otherwise} \end{cases}$$

- Tradeoff margin violation with $||\beta||$:

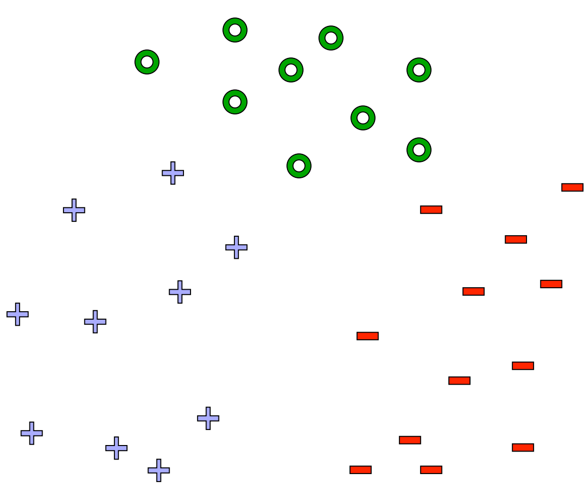
$$\min_{\beta,\beta_0} ||\beta||_2^2 + C \sum_{i=1}^{n} (1 - y_i(\beta \cdot x_i + \beta_0))_+$$

SVM objective

Now support vectors are
- on margin
or
- misclassified pts

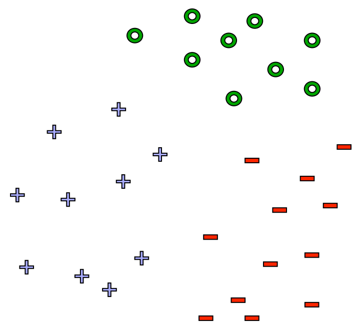
$(1 - y_i(\beta \cdot x_i + \beta_0))_+$

What about multiple classes?



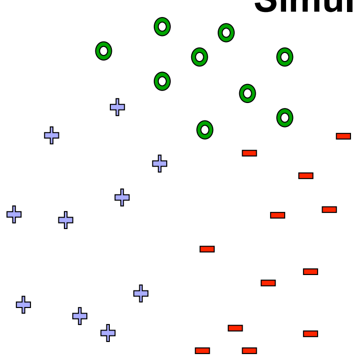
One against All

Learn 3 classifiers:



Learn 1 classifier: Multiclass SVM

Simultaneously learn 3 sets of weights

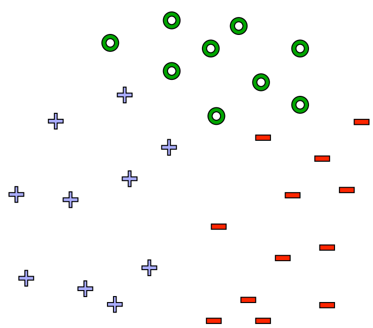


$$\beta^{(y_i)} \cdot x_i + \beta_0^{(y_i)} \geq \beta^{(y')} \cdot x_i + \beta_0^{(y')} + 1, \forall y' \neq y_i, \forall i$$

Learn 1 classifier: Multiclass SVM

$$\beta^{(y_i)} \cdot x_i + \beta_0^{(y_i)} \geq \beta^{(y')} \cdot x_i + \beta_0^{(y')} + 1, \forall y' \neq y_i, \forall i$$

$$\min_{\beta, \beta_0} \sum_y \beta^{(y)} \cdot \beta^{(y)} + C \sum_i \xi_i$$



What you need to know

- Maximizing margin
- Derivation of SVM formulation
- Non-linearly separable case
  - Hinge loss
  - A.K.A. adding slack variables
- SVMs = Perceptron + L2 regularization
- Can optimize SVMs with SGD
  - Many other approaches possible
- Handling multiple classes

Reading

- Hastie, Tibshirani, Friedman – 12.1-12.3

Module 5: Classification

Boosting

STAT/BIOSTAT 527, University of Washington
Emily Fox
June 6th, 2013

Fighting the Bias-Variance Tradeoff

- **Simple (a.k.a. weak) learners are good**
  - e.g., naïve Bayes, logistic regression, shallow decision trees
  - Low variance, don't usually overfit too badly
- **Simple (a.k.a. weak) learners are bad**
  - High bias, can't solve hard learning problems
- Can we make weak learners always good???
  - **No!!!**
  - **But often yes…**

Voting (Ensemble Methods)

- Instead of learning a single (weak) classifier, learn **many weak classifiers** that are **good at different parts of the input space**
- **Output class:** (Weighted) vote of each classifier
  - Classifiers that are most "sure" will vote with more conviction
  - Classifiers will be most "sure" about a particular part of the space
  - On average, do better than single classifier!
- **But how do you ???**
  - force classifiers to learn about different parts of the input space?
  - weigh the votes of different classifiers?

Boosting [Schapire, 1989]

- Idea: given a weak learner, run it multiple times on (reweighted) training data, then let learned classifiers vote
- On each iteration *t*:
  - weight each training example by how incorrectly it was classified
  - Learn a hypothesis – h_t
  - A strength for this hypothesis – α_t
- Final classifier:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

- **Practically useful**
- **Theoretically interesting**

Learning from Weighted Data

- **Sometimes not all data points are equal**
  - Some data points are more equal than others
- **Consider a weighted dataset**
  - D(i) – weight of *i* th training example ($\mathbf{x}_i$,y_i)
  - Interpretations:
    - *i* th training example counts as D(i) examples
    - If I were to "resample" data, I would get more samples of "heavier" data points
- **Now, in all calculations, whenever used, *i* th training example counts as D(i) "examples"**

AdaBoost

- Initialize weights to uniform dist: $D_1(i) = 1/n$
- For t = 1…T
  - Train weak learner h_t on distribution D_t over the data
  - Choose weight α_t
  - Update weights:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

    - Where Z_t is normalizer:

$$Z_t = \sum_{i=1}^{n} D_t(i) \exp(-\alpha_t y_i h_t(x_i))$$

- Output final classifier:

Picking Weight of Weak Learner

- Weigh h_t higher if it did well on training data (weighted by D_t):

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

  - Where ε_t is the weighted training error:

$$\epsilon_t = \sum_{i=1}^{n} D_t(i) \mathbb{I}[h_t(x_i) \neq y_i]$$

Why choose α_t for hypothesis h_t this way?

[Schapire, 1989]

Training error of final classifier is bounded by:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[H(x_i) \neq y_i] \leq \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i f(x_i))$$

Where $f(x) = \sum_t \alpha_t h_t(x); H(x) = sign(f(x))$

Why choose α_t for hypothesis h_t this way?

[Schapire, 1989]

Training error of final classifier is bounded by:

$Z_t = \sum_{i=1}^{n} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[H(x_i) \neq y_i] \leq \frac{1}{n}\sum_{i=1}^{n}\exp(-y_i f(x_i)) = \prod_{t=1}^{T} Z_t$$

Where $f(x) = \sum_t \alpha_t h_t(x); H(x) = sign(f(x))$

Why choose α_t for hypothesis h_t this way?

[Schapire, 1989]

Training error of final classifier is bounded by:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[H(x_i) \neq y_i] \leq \frac{1}{n}\sum_{i=1}^{n} \exp(-y_i f(x_i)) = \prod_{t=1}^{T} Z_t$$

Where $f(x) = \sum_t \alpha_t h_t(x); H(x) = sign(f(x))$

If we minimize $\prod_t Z_t$, we minimize our training error

We can tighten this bound greedily, by choosing α_t and h_t on each iteration to minimize Z_t.

$$Z_t = \sum_{i=1}^{n} D_t(i) \exp(-\alpha_t y_i h_t(x_i))$$

Why choose α_t for hypothesis h_t this way?

[Schapire, 1989]

We can minimize this bound by choosing α_t on each iteration to minimize Z_t.

$$Z_t = \sum_{i=1}^{n} D_t(i) \exp(-\alpha_t y_i h_t(x_i))$$

For boolean target function, this is accomplished by [Freund & Schapire '97]:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

Strong, weak classifiers

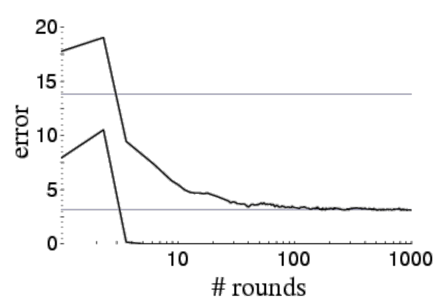
- If each classifier is (at least slightly) better than random
  - $\varepsilon_t < 0.5$
- AdaBoost will achieve zero *training error* (exponentially fast):

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[H(x_i) \neq y_i] \leq \prod_{t=1}^{T} Z_t \leq \exp\left(-2\sum_{t=1}^{T}(1/2 - \epsilon_t)^2\right)$$

- Is it hard to achieve better than random training error?

Boosting results – Digit recognition

[Schapire, 1989]

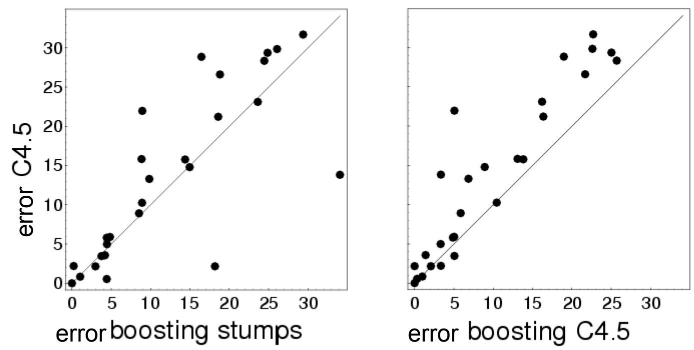


- Boosting often
  - Robust to overfitting
  - Test set error decreases even after training error is zero

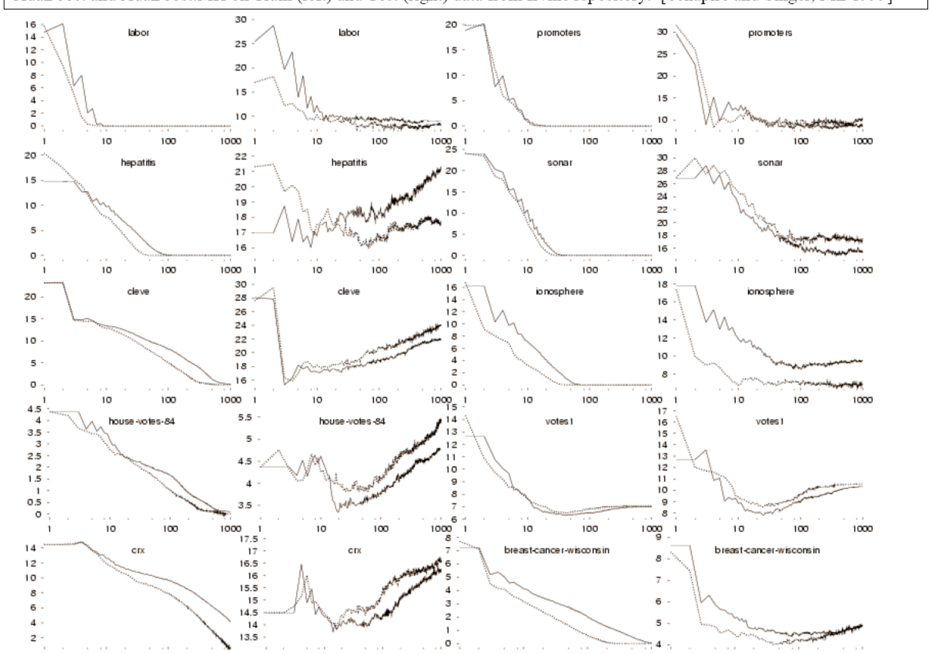
Boosting: Experimental Results

[Freund & Schapire, 1996]

Comparison of C4.5, Boosting C4.5, Boosting decision stumps (depth 1 trees), 27 benchmark datasets



AdaBoost and AdaBoost.MH on Train (left) and Test (right) data from Irvine repository. [Schapire and Singer, ML 1999]



Demo!

http://cseweb.ucsd.edu/~yfreund/adaboost/

Boosting and Logistic Regression

Logistic regression assumes:

$$P(Y = 1|X) = \frac{1}{1 + \exp(f(x))}$$

And tries to maximize data likelihood:

$$P(\mathcal{D}|H) = \prod_{i=1}^{n} \frac{1}{1 + \exp(-y_i f(x_i))}$$

Equivalent to minimizing log loss

$$\sum_{i=1}^{n} \ln(1 + \exp(-y_i f(x_i)))$$

Boosting and Logistic Regression

Logistic regression equivalent to minimizing log loss

$$\sum_{i=1}^{n} \ln(1 + \exp(-y_i f(x_i)))$$

Boosting minimizes similar loss function!!

$$\frac{1}{n}\sum_{i=1}^{n} \exp(-y_i f(x_i)) = \prod_{t=1}^{T} Z_t$$

Both smooth approximations of 0/1 loss!

Logistic regression and Boosting

Logistic regression:

- Minimize loss fn

$$\sum_{i=1}^{n} \ln(1 + \exp(-y_i f(x_i)))$$

- Define

$$f(x) = \beta_0 + \sum_j \beta_j h(x_j)$$

where features x_j are predefined

- Weights β_j are learned in joint optimization

Boosting:

- Minimize loss fn

$$\sum_{i=1}^{n} \exp(-y_i f(x_i))$$

- Define

$$f(x) = \sum_t \alpha_t h_t(x)$$

where $h_t(x)$ defined dynamically to fit data

(not a linear classifier)

- Weights α_t learned incrementally

What you need to know about Boosting

- Combine weak classifiers to obtain very strong classifier
  - Weak classifier – slightly better than random on training data
  - Resulting very strong classifier – can eventually provide zero training error
- AdaBoost algorithm
- Boosting v. Logistic Regression
  - Similar loss functions
  - Single optimization (LR) v. Incrementally improving classification (B)
- Most popular application of Boosting:
  - Boosted decision stumps!
  - Very simple to implement, very effective classifier

Reading

- Hastie, Tibshirani, Friedman – 10.1-10.6

What you need to know from 527

- Module 1: Preliminaries of Nonparametrics
  - Loss functions and optimal predictions
  - Linear smoothers
  - Ridge regression, LASSO
  - Cross validation, etc.
- Module 2: Splines and Kernel Methods
  - Smoothing splines, penalized regression splines
  - Local polynomial regression, KDE
- Module 3: Bayesian Nonparametrics
  - Gaussian processes
  - Dirichlet process mixture of Gaussians

What you need to know from 527

- Module 4: Nonparametrics with Multiple Predictors
  - Thin plate splines, tensor product splines
  - GAMs
  - Projection pursuit
  - Multivariate kernels and KDE
  - Regression trees
- Module 5: Classification
  - Classification trees
  - Logistic regression (also looked at nonparametrics for GLMs)
  - LDA, QDA, KDE, naïve Bayes, mixture models
  - Perceptron, SVM and with kernels for both
  - Boosting

THANK YOU!!!

- You have been a great, interactive class! …especially for a 9am lecture =)
- We're looking forward to the poster session

- Thanks to Shirley, too!