Support Vector Machines (SVMs)

Machine Learning - 10601

Geoff Gordon, MiroslavDudík

[partly based on slides of Ziv-Bar Joseph]

http://www.cs.cmu.edu/~ggordon/10601/

November 11, 2009

Boosting

weak classifiers $\Rightarrow$ strong classifiers

- **weak:** slightly better than random on training data
- **strong:** eventually zero error on training data

AdaBoost

- begin with equal weights on all examples
- in each round **t** call a weak learner, get a classifier h_t
- reweight examples:
  - **increase** weight where h_t makes **mistakes**
  - **decrease** weight where h_t is **correct**
- after **T** rounds, return a weighted ensemble

AdaBoost

- **training error** decreases exponentially
  (if **weak learner** assumption satisfied)
- **training error** decreases even **faster**
  if weak learners allowed **continuous outputs**
- **margins** of training examples **increase**
- **large margins**
  ≈ **simple** final (strong) classifier
  ≈ **good generalization** bounds

Overfitting regimes

- **weak learner too strong: use smaller trees** or **stop early**
- **data noisy: stop early** or **regularize α_t**

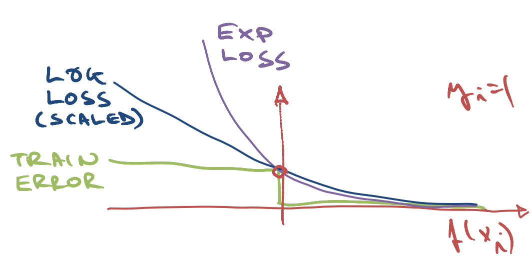
Logistic Regression vs AdaBoost

- Minimize *log loss*

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

- Minimize *exponential loss*

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



Types of classifiers

- We can divide the large variety of classification approaches into roughly three major types

1. Instance based classifiers
   - Use observation directly (no models)
   - e.g. K nearest neighbors

2. Generative:
   - build a generative statistical model
   - e.g., Bayesian networks

3. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., logistic regression, boosting, decision trees

Which classifier is the best?

[Caruana & Niculescu-Mizil 2006]

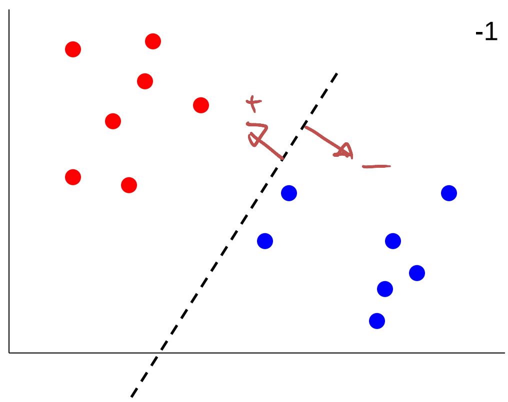
MODEL	1ST	2ND	3RD	4TH	5TH
BST-DT	0.580	0.228	0.160	0.023	0.009
RF	0.390	0.525	0.084	0.001	0.000
BAG-DT	0.030	0.232	0.571	0.150	0.017
SVM	0.000	0.008	0.148	0.574	0.240
ANN	0.000	0.007	0.035	0.230	0.606
KNN	0.000	0.000	0.000	0.009	0.114
BST-STMP	0.000	0.000	0.002	0.013	0.014
DT	0.000	0.000	0.000	0.000	0.000
LOGREG	0.000	0.000	0.000	0.000	0.000
NB	0.000	0.000	0.000	0.000	0.000

Logistic regression

Recall logistic regression classifiers

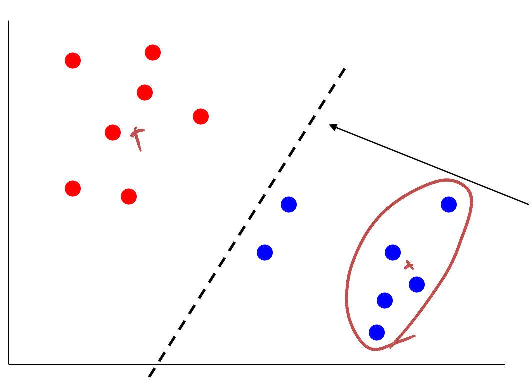
+1 if sign(w^Tx+b)≥0

-1 if sign(w^Tx+b)<0



Logistic regression

Recall logistic regression classifiers



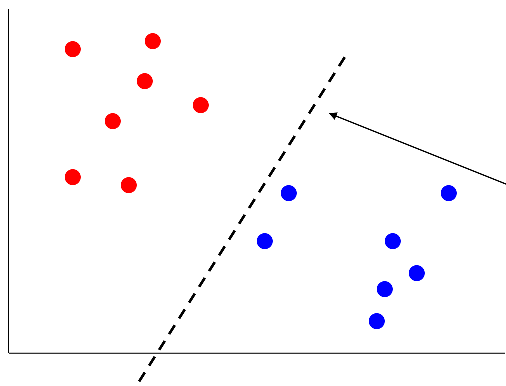
Line closer to the blue nodes since many of them are far away from the boundary

Logistic regression

Recall logistic regression classifiers

$$\min_w \sum_i \ln(1+\exp(y^i w^T x^i))$$

Goes over all training points x



Line closer to the blue nodes since many of them are far away from the boundary

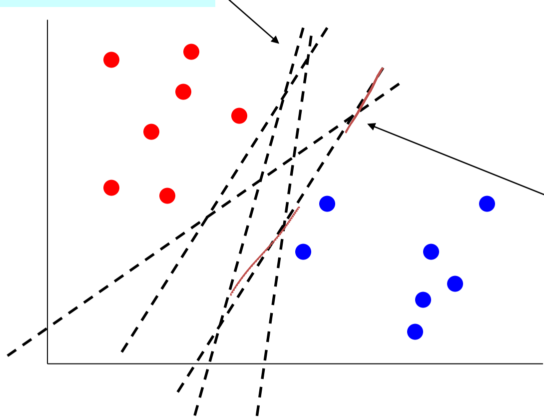
Logistic regression

Recall logistic regression classifiers

$$\min_w \sum_i \ln(1+\exp(y^i w^T x^i))$$

Many more possible classifiers

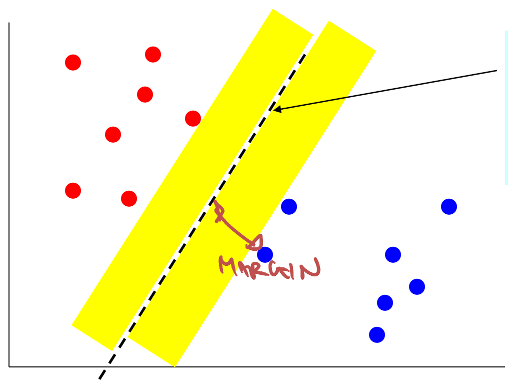
Goes over all training points x



Line closer to the blue nodes since many of them are far away from the boundary

Max margin classifiers

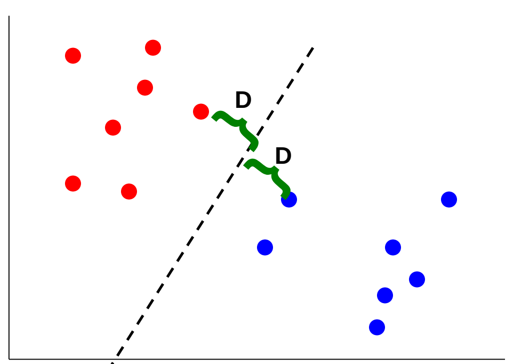
- Instead of fitting all points, focus on boundary points
- Learn a boundary that leads to the largest margin from points on both sides



From all the possible boundary lines, this leads to the largest margin on both sides

Max margin classifiers

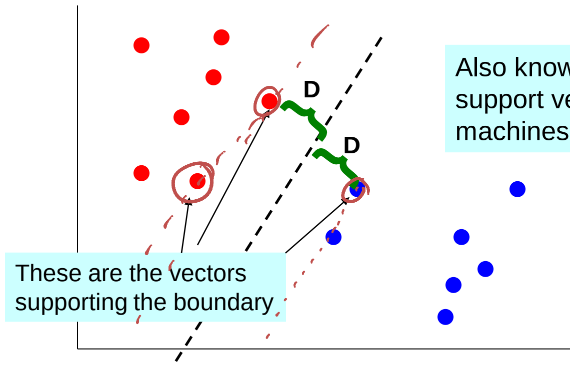
- Instead of fitting all points, focus on boundary points
- Learn a boundary that leads to the largest margin from points on both sides



Why?
- Intuitive, 'makes sense'
- Theoretical support
- Works well in practice

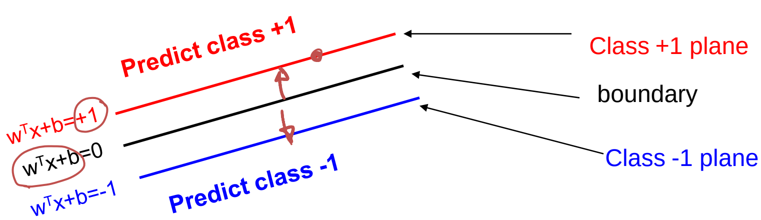
Max margin classifiers

- Instead of fitting all points, focus on boundary points
- Learn a boundary that leads to the largest margin from points on both sides



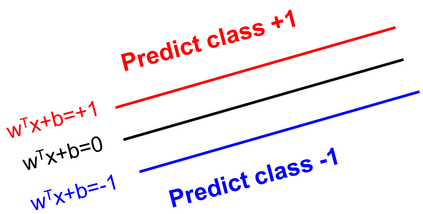
Also known as linear support vector machines (SVMs)

Specifying a max margin classifier



Classify as +1	if	$w^Tx+b \geq 1$
Classify as -1	if	$w^Tx+b \leq -1$
Undefined	if	$-1 < w^Tx+b < 1$

Specifying a max margin classifier

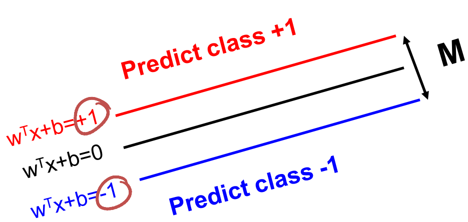


Is the linear separation assumption realistic?

We will deal with this shortly, but lets assume it for now

Classify as +1	if	$w^Tx+b \geq 1$
Classify as -1	if	$w^Tx+b \leq -1$
Undefined	if	$-1 < w^Tx+b < 1$

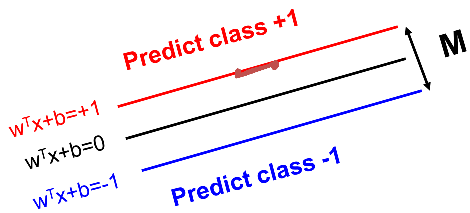
Maximizing the margin



Classify as +1	if	$w^Tx+b \geq 1$
Classify as -1	if	$w^Tx+b \leq -1$
Undefined	if	$-1 < w^Tx+b < 1$

- Lets define the width of the margin by M
- How can we encode our goal of maximizing M in terms of our parameters w and b?
- Lets start with a few observations

Maximizing the margin



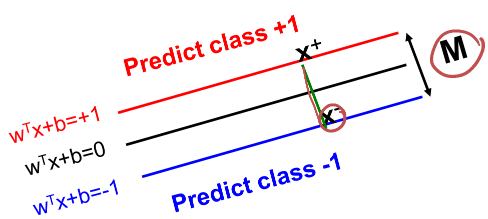
Classify as +1 if $w^Tx+b \geq 1$
Classify as -1 if $w^Tx+b \leq -1$
Undefined if $-1 < w^Tx+b < 1$

- Observation 1: the vector w is orthogonal to the +1 plane
- Why?

Let u and v be two points on the +1 plane, then for the vector defined by u and v we have $w^T(u-v) = 0$

Corollary: the vector w is orthogonal to the -1 plane

Maximizing the margin



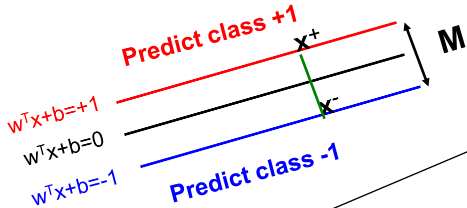
Classify as +1 if $w^Tx+b \geq 1$
Classify as -1 if $w^Tx+b \leq -1$
Undefined if $-1 < w^Tx+b < 1$

- Observation 1: the vector w is orthogonal to the +1 and -1 planes
- Observation 2: if x^+ is a point on the +1 plane and x^- is the closest point to x^+ on the -1 plane then

$$x^+ = \lambda w + x^-$$

Since w is orthogonal to both planes we need to 'travel' some distance along w to get from x^+ to x^-

Putting it together



- $w^T x^+ + b = +1$
- $w^T x^- + b = -1$
- $x^+ = \lambda w + x^-$
- $| x^+ - x^- | = M$

We can now define M in terms of w and b

$w^T x^+ + b = +1$

$\Rightarrow$

$w^T (\lambda w + x^-) + b = +1$

$\Rightarrow$

$\underline{w^T x^- + b} + \lambda w^T w = +1$

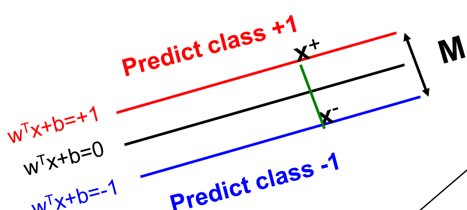
$\Rightarrow$

$-1 + \lambda w^T w = +1$

$\Rightarrow$

$\lambda = 2/w^T w$

Putting it together



- $w^T x^+ + b = +1$
- $w^T x^- + b = -1$
- $\underline{x^+ = \lambda w + x^-}$
- $\underline{| x^+ - x^- | = M}$
- $\underline{\lambda = 2/w^T w}$

We can now define M in terms of w and b

$M = |x^+ - x^-|$

$\Rightarrow$

$$M = |\lambda w| = \lambda |w| = \lambda \sqrt{w^T w}$$

$\Rightarrow$

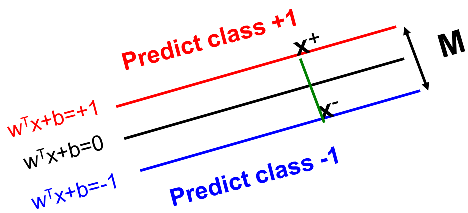
$$M = 2\frac{\sqrt{w^T w}}{w^T w} = \frac{2}{\sqrt{w^T w}}$$

max M

$\Updownarrow$

min $\|w\|^2$

Finding the optimal parameters



$$M = \frac{2}{\sqrt{w^T w}}$$

We can now search for the optimal parameters by finding a solution that:

w, b

1. Correctly classifies all points
2. Maximizes the margin (or equivalently minimizes $w^T w$)

Several optimization methods can be used:
Gradient descent, simulated annealing, EM etc.