Linear Classifiers and Perceptron (2014)

Note Title 11/12/2006

N samples: $\{(\underline{x}_\mu, y_\mu) : \mu = 1 \text{ to } N\}$

$y_\mu \in \{\pm 1\}$

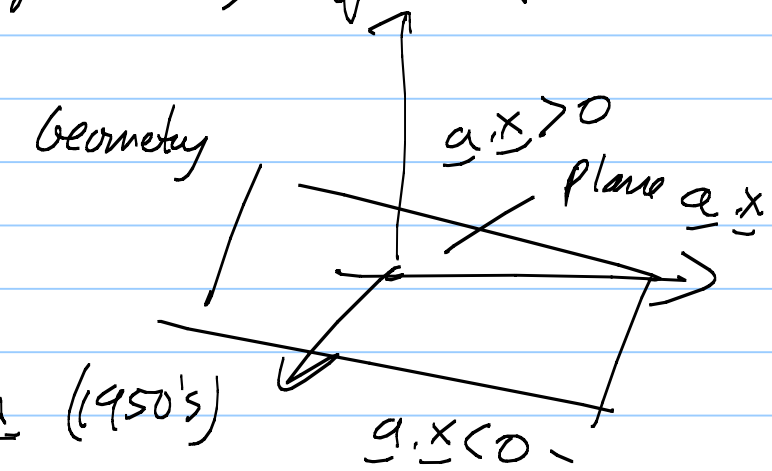
Can we find a linear classifier that separates the positive and negative examples?

Eg. a plane $\underline{a} \cdot \underline{x} = 0$ s.t. $\text{sign}(\underline{a} \cdot \underline{x}) = y$

s.t. $\underline{a} \cdot \underline{x}_\mu \geq 0$, if $y_\mu = +1$

$\underline{a} \cdot \underline{x}_\mu \leq 0$, if $y_\mu = -1$

Plane goes through the origin ($\underline{a} \cdot \underline{0} = 0$)



Perceptron Algorithm (1950's)

First, replace -ve examples by +ve examples

If $y_\mu = -1$, set $\underline{x}_\mu \to -\underline{x}_\mu$, $y_\mu \to -y_\mu$

(Note. require $\text{sign}(\underline{a} \cdot \underline{x}_\mu) = y_\mu$, this is equivalent to $\text{sign}(-\underline{a} \cdot \underline{x}_\mu) = -y_\mu$)

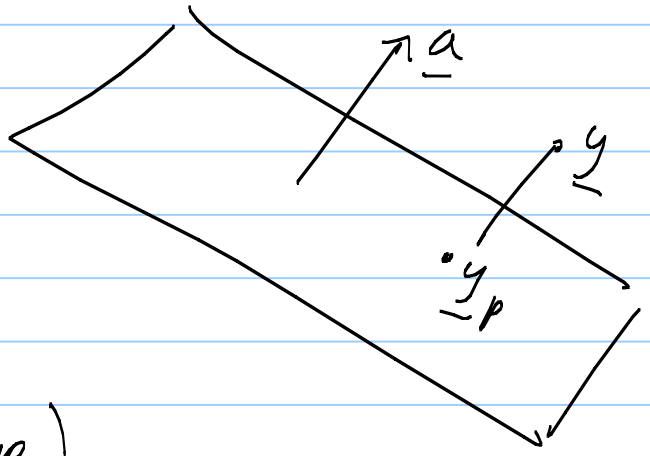
(2) This reduces to finding a plane s.t. $\underline{a} \cdot \underline{x}_\mu \geq 0$, for $\mu = 1 \text{ to } N$

Note: the vector $\underline{a}$ need not be unique. It is better to try to maximize the margin (see next lectures). To find $\underline{a}$ with $|\underline{a}| = 1$, so that $\underline{a} \cdot \underline{x}_\mu \geq m$, $\forall \mu = 1 \text{ to } N$ for the maximin value of m.

More geometry

Claim: If $\underline{a}$ is a unit vector $|\underline{a}| = 1$, then $\underline{a} \cdot \underline{y}$ is the sign(*) distance of $\underline{y}$ to the plane $\underline{a} \cdot \underline{x} = 0$.

(* i.e. $\underline{a} \cdot \underline{y} > 0$, if $\underline{y}$ is above plane
$\underline{a} \cdot \underline{y} < 0$, if $\underline{y}$ is below plane)



Proof: write $\underline{y} = \underline{y}_p + \lambda \underline{a}$, where $\underline{y}_p$ is the projection of $\underline{y}$ into the plane. By definition $\underline{a} \cdot \underline{y}_p = 0$, hence $\lambda = (\underline{a} \cdot \underline{y}) / (\underline{a} \cdot \underline{a}) = (\underline{a} \cdot \underline{y})$, if $|\underline{a}| = 1$.

(3) Perceptron Algorithm.

Initialize: $\underline{a}(0) = 0$.

Loop over $\mu = 1$ to N

if $\underline{x}_\mu$ is missclassified, set $\underline{a} \to \underline{a} + \underline{x}_\mu$

Repeat until all samples are classified correctly.

Novikov's Thm. The Perceptron algorithm will converge to a solution weight that classifies all the samples correctly (provided this is possible).

Proof. Let $\hat{a}$ be a separating weight

Let $m = \min_{\mu=1}^{n} \hat{a} \cdot \underline{x}_\mu$ $(m > 0)$

Let $\beta^2 = \max_{\mu=1}^{N} |\underline{x}_\mu|^2$

Suppose $\underline{x}_t$ is missclassified at time t

so $\underline{a} \cdot \underline{x}_t < 0$

$$\underline{a}_{t+1} - (\beta^2/m)\,\hat{a} = \underline{a}_t - (\beta^2/m)\,\hat{a} + \underline{x}_t$$

(4) $\|\underline{a}_{t+1} - \beta^2/m\,\hat{\underline{a}}\|^2 = \|\underline{a}_t - (\beta^2/m)\hat{\underline{a}}\|^2 + \|\underline{x}_t\|^2 - 2(\underline{a}_t - (\beta^2/m)\hat{\underline{a}})\cdot\underline{x}_t.$

Using $\|\underline{x}_t\|^2 \le \beta^2$, $\underline{a}_t \cdot \underline{x}_t < 0$, $-\hat{\underline{a}}\cdot\underline{x}_t < -m$

It follows that

$\|\underline{a}_{t+1} - \beta^2/m\,\hat{\underline{a}}\|^2 \le \|\underline{a}_t - \beta^2/m\,\hat{\underline{a}}\|^2 + \beta^2 - 2\beta^2/m \cdot m$

Hence $\|\underline{a}_{t+1} - \beta^2/m\,\hat{\underline{a}}\|^2 \le \|\underline{a}_t - \beta^2/m\,\hat{\underline{a}}\|^2 - \beta^2.$

So, each time we update a weight, we reduce the quantity $\|\underline{a}_t - \beta^2/m\,\hat{\underline{a}}\|^2$ by a fixed amount β^2. $\|\underline{a}_0 - \beta^2/m\,\hat{\underline{a}}\|^2$ is bounded by $\frac{\beta^4}{m^2}|\hat{\underline{a}}|^2$.

So we can update the weights at most $\frac{\beta^2}{m^2}|\hat{\underline{a}}|^2$ times

Guarantees Convergence

(5)

Support Vector Machines

Linear Separation: Margins & Duality

Note Title 11/12/2006

Modern approach to linear separation.

Data $\{(\underline{x}_\mu, y_\mu) : \mu = 1 \text{ to } N\}$, $y_\mu \in \{-1, 1\}$

Hyperplane $\{\underline{x} : \underline{x} \cdot \underline{a} + b = 0\}$, $|\underline{a}| = 1$.

The signed distance of a point $\underline{x}$ to the plane is $\underline{a} \cdot \underline{x} + b$.

$\in$ to project on plane

Line $\underline{x}(\lambda) = \underline{x} + \lambda \underline{a}$

hits plane when $\underline{a} \cdot (\underline{x} + \lambda \underline{a}) = -b$

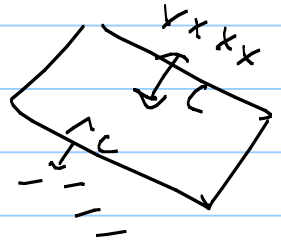
$$\lambda = -(\underline{a} \cdot \underline{x} + b) / |\underline{a}|^2 = -(\underline{a} \cdot \underline{x} + b)$$

if $|\underline{a}| = 1$.

Seek classifier with biggest margin

$$\max_{\underline{a}, b, |\underline{a}|=1} C \quad \text{s.t.} \quad y_\mu (\underline{x}_\mu \cdot \underline{a} + b) \geq C, \quad \forall \mu = 1 \text{ to } N.$$

i.e. the positive examples are at least distance C above the plane, and negative examples are at least C below the plane.

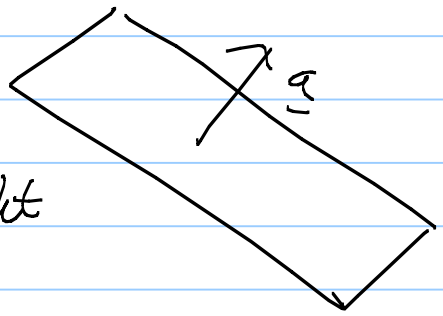


Large margin is good for generalization.

16)

Now, allow for some datapoints to be misclassified.

Slack variables → allow datapoints to move in direction $\underline{a}$, so that they are on the right side of the margin.



Slack variables $\{z_1, \ldots z_n\}$,

Criterion: $\max_{a,\, b/|\underline{a}|=1} C$ s.t. $y_\mu(\underline{x}_\mu \cdot \underline{a} + b) \geq C(1 - z_\mu)$ $\forall \mu \in (1, N)$,

with constraint $z_\mu \geq 0$, $\forall \mu$.

Alternately: $y_\mu\{(\underline{x}_\mu + C z_\mu \underline{a}) \cdot \underline{a} + b\} \geq C$

like moving $\underline{x}_\mu$ to $\underline{x}_\mu + C z_\mu \underline{a}$.

But, you must pay a penalty for using slack variables. A penalty like $\sum_{\mu=1}^{n} z_\mu$.

If $z_\mu = 0$, then the datapoint is correctly classified and is past the margin.

If $z_\mu > 0$, then the datapoint is on the wrong side of the margin

(7)

Task: We need to estimate several quantities simultaneously:

(1) The plane $\underline{a}, b$

(2) The margin C

(3) The slack variables $\{z_\mu\}$.

We need a criterion that maximizes the margin and minimizes the amount of slack variables used.

Absorb C into $\underline{a}$ by $\underline{a} \to \frac{1}{C}\underline{a}$, hence $C = 1/|\underline{a}|$

The Max Margin Criterion.

$$\underline{\text{Min}} \quad \tfrac{1}{2}\,\underline{a}\cdot\underline{a} + \gamma \sum_\mu z_\mu$$

$$\text{s.t.} \quad y_\mu(\underline{x}_\mu \cdot \underline{a} + b) \geq 1 - z_\mu, \quad \forall \mu$$

$$z_\mu \geq 0, \quad \forall \mu.$$

Quadratic Primal Problem requires Lagrange multipliers.

$$L_p = \tfrac{1}{2}\,\underline{a}\cdot\underline{a} + \gamma \sum_\mu z_\mu - \sum_\mu \alpha_\mu \{ y_\mu(\underline{x}_\mu \cdot \underline{a} + b) - (1 - z_\mu) \} - \sum_\mu \tau_\mu z_\mu.$$

The $\{\alpha_\mu\}$ & $\{\tau_\mu\}$ are Lagrange parameters needed to enforce the inequality constraints. $\alpha_\mu \geq 0, \tau_\mu \geq 0, \forall \mu$.

(8)

L_p is a function of the primal variables $\underline{a}, b, \{z_\mu\}$ and the lagrange parameters $\{\alpha_\mu, \tau_\mu\}$

There is no analytic solution for these variables, but we can use analytic techniques to get some understanding of their properties.

$$\frac{\partial L_p}{\partial \underline{a}} = 0 \quad \Rightarrow \quad \hat{\underline{a}} = \sum_\mu \hat{\alpha}_\mu y_\mu \underline{x}_\mu .$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_\mu \hat{\alpha}_\mu y_\mu = 0 .$$

$$\frac{\partial L_p}{\partial z_\mu} = 0 \quad \Rightarrow \quad \hat{\alpha}_\mu = \gamma - \hat{\tau}_\mu , \quad \forall \mu .$$

The classifier is

$$\text{sign}\{\hat{\underline{a}} \cdot \underline{x} + \hat{b}\} = \text{sign}\{\sum_\mu \alpha_\mu y_\mu \underline{x}_\mu \cdot \underline{x} + b\}$$

Support vectors, the solution depends only on the vectors $\underline{x}_\mu$ for which $\alpha_\mu \neq 0$. //

(9)

The constraints are

$$y_\mu (\underline{x}_\mu \cdot \hat{\underline{a}} + \hat{b}) \geq 1 - \hat{z}_\mu$$

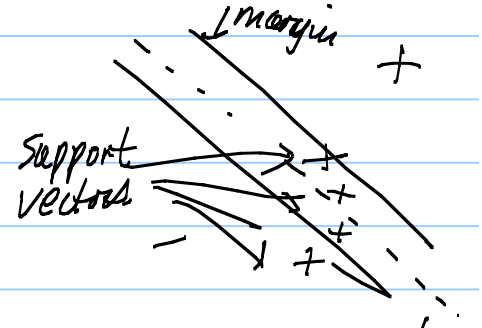
$$\hat{z}_\mu \geq 0, \quad \hat{\tau}_\mu \geq 0.$$

By theory of Quadratic Programming —
$\alpha_\mu > 0$, only if either:

(i) $z_\mu > 0$ slack variable is used

(ii) $z_\mu = 0$, but $y_\mu (\underline{x}_\mu \cdot \hat{\underline{a}} + \hat{b}) = 1$
datapoint is on the margin

The classifier depends only on the support vectors, the other datapoints do not matter.



This is intuitively reasonable — the classifier must pay close attention to the data that is difficult to classify — the data near the boundary.

This differs from the probabilistic approach where we learn probability models for each class and then use the Bayes classifier.

Dual formulation

We can solve the problem more easily in the dual formulation. - function of Lagrange multipliers only.

$$L_d = \sum_\mu \alpha_\mu - \frac{1}{2} \sum_{\mu,\nu} \alpha_\mu \alpha_\nu y_\mu y_\nu \underline{x}_\mu \cdot \underline{x}_\nu .$$

with constraint $0 \le \alpha_\mu \le \tau$, $\sum_\mu \alpha_\mu y_\mu = 0$.

There are standard packages to solve this.

Knowing $\{\hat{\alpha}_\mu\}$, will give us the solution

$\hat{\underline{a}} = \sum_\mu \hat{\alpha}_\mu y_\mu \underline{x}_\mu$, (only a little more work to get $\hat{b}$)

(11)

Relationship between Primal & Dual.

Start with dual formulation. L_P

Rewrite it as

$$L_P = \left(-\tfrac{1}{2}\right) \underline{a}\cdot\underline{a} + \sum_\mu \alpha_\mu + \underline{a}\cdot\left\{\underline{a} - \sum_\mu \alpha_\mu y_\mu \underline{x}_\mu\right\}$$
$$+ \sum_\mu z_\mu (\gamma - \tau_\mu - \alpha_\mu) - b \sum_\mu \alpha_\mu y_\mu .$$

Extremize w.r.t. $\underline{a}, b, \{z_\mu\}$ gives:

$$\hat{\underline{a}} = \sum_\mu \alpha_\mu y_\mu \underline{x}_\mu , \quad \sum_\mu \alpha_\mu y_\mu = 0 , \quad \gamma - \tau_\mu - \alpha_\mu = 0$$

Substituting back into L_P gives:

$$L_D = -\frac{1}{2} \sum_{\mu,\nu} \alpha_\mu \alpha_\nu y_\mu y_\nu \underline{x}_\mu \cdot \underline{x}_\nu + \sum_\mu \alpha_\mu$$

maximize w.r.t. $\{\alpha_\mu\}$.

Note: The Perceptron can be reformulated in this way.

By the theory, the weight hypothesis will always be of form:

$$\underline{a} = \sum_{\mu} \alpha_\mu y_\mu \underline{x}_\mu.$$

Perceptron Update Rule:

If data $\underline{x}_\mu$ is misclassified

i.e. $y_\mu (\underline{a} \cdot \underline{x}_\mu + b) \leq 0$

Set $\alpha_\mu \to \alpha_\mu + 1$

$b \to b + y_\mu R^2$.

R is radius of smallest ball containing the data.

New Topics:

(1) How does the max-margin criterion relate to the empirical risk?

(2) What if we ignore the dual formulation and instead try to learn in the primal space?

Both topics start by re-expressing the max-margin criterion.

(13) The Max-Margin Criterion

$$L_p = \frac{1}{2}\underline{a}\cdot\underline{a} + \gamma \sum_\mu z_\mu$$ with constraints

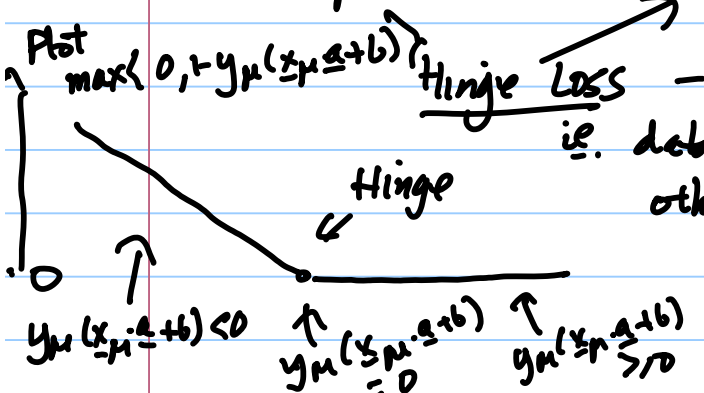
$$y_\mu(\underline{x}_\mu\cdot\underline{a}+b) - (1-z_\mu) \geq 0$$

$$z_\mu \geq 0$$

$$z_\mu \geq 1 - y_\mu(\underline{x}_\mu\cdot\underline{a}+b)$$

Topic 1.

Hence $L_p = \frac{1}{2}\underline{a}\cdot\underline{a} + \gamma \sum_\mu \max\{0, 1-y_\mu(\underline{x}_\mu\cdot\underline{a}+b)\}$



Hinge Loss - loss = 0 if $y_\mu(\underline{x}_\mu\cdot\underline{a}+b) \geq 1$
i.e. data point is on the correct side of margin
otherwise, loss is $1 - y_\mu(\underline{x}_\mu\cdot\underline{a}+b)$

Note: Hinge loss is a convex loss function.

So we can re-express the max-margin criterion as the sum of the empirical risk (with hinge loss function) plus a term $\frac{1}{2}|\underline{a}|^2$ (multiplied by a constant $1/\gamma N$).

$$\frac{L_p}{\gamma N} = \frac{1}{2\gamma N}|\underline{a}|^2 + \frac{1}{N}\sum_{\mu=1}^{N} \max\{0, 1 - y_\mu(\underline{x}_\mu\cdot\underline{a}+b)\}$$

↑ empirical risk

The first term is a "regularizer".
It penalizes decision rules. $\hat{y}(\underline{x}) = \text{sign}(\underline{x}\cdot\underline{a}+b)$ which have large $|\underline{a}|$.

Why? This helps generalization.

If we only tried to minimize the loss function, we may overfit the data, because the space of possible decison rules is very big (all values of $\underline{a}$ and b). If we penalize those rules with big $|\underline{a}|$, then we restrict our set of rules and are more likely to generalize to new data.

(14) Topic (2). Minimizing in the Primal Space.

We can do online learning using the cost function

$$L_P = \tfrac{1}{2}|\underline{a}|^2 + \gamma \sum_{\mu=1}^{N} \max\{0, 1 - y_\mu(\underline{x}_\mu \cdot \underline{a} + b)\}$$

consider the second term for one datapoint $\underline{x}_\mu, y_\mu$

$$\frac{\partial}{\partial \underline{a}} \max\{0, 1 - y_\mu(\underline{x}_\mu \cdot \underline{a} + b)\} = \begin{cases} 0, & \text{if } y_\mu(\underline{x}_\mu \cdot \underline{a} + b) > 1 \\ -y_\mu \underline{x}_\mu, & \text{if } y_\mu(\underline{x}_\mu \cdot \underline{a} + b) < 1 \end{cases}$$

Online learning:

- select data $(\underline{x}_\mu, y_\mu)$ at random
- compute $\frac{\partial}{\partial \underline{a}} \max\{0, 1 - y_\mu(\underline{x}_\mu \cdot \underline{a} + b)\}$

$$\underline{a}^t \to \underline{a}^t - \frac{1}{2N}\underline{a}^t - \gamma \begin{cases} 0, & \text{if } y_\mu(\underline{x}_\mu \cdot \underline{a} + b) > 1 \\ -y_\mu \underline{x}_\mu, & \text{if } y_\mu(\underline{x}_\mu \cdot \underline{a} + b) < 1 \end{cases}$$

This is almost exactly the 1950's perceptron algorithm!

(The $-y_\mu$ term is like converting -ve examples to +ve's)

The $\frac{1}{2N}\underline{a}^t$ is from the regularizer.