LECTURE NOTE #8

PROF. ALAN YUILLE

1. Linear Classifiers and Perceptrons

A dataset contains N samples: $\{ (x_\mu, y_\mu) : \mu = 1 \text{ to } N \}$, $y_\mu \in \{\pm 1\}$

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

E.G. a plane $\underline{a} \cdot \underline{x} = 0$, see figure (1), which separates the data with a decision rule $\hat{y}(\vec{x}) = sign(\underline{a} \cdot \underline{x})$
s.t. $\underline{a} \cdot \underline{x} \geq 0$, if $y_\mu = +1$
s.t. $\underline{a} \cdot \underline{x} \leq 0$, if $y_\mu = -1$

Plane goes through the origin ($\underline{a} \cdot \underline{0} = 0$) (special case, can be relaxed later).

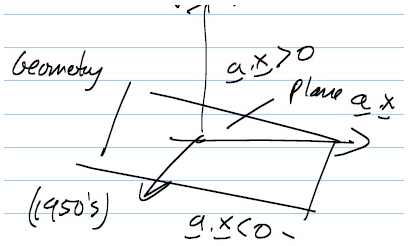

FIGURE 1. A plane $\vec{a} \cdot \vec{x} =$ separates the space into two regions $\vec{a} \cdot \vec{x} > 0$ and $\vec{a} \cdot \vec{x} < 0$.

Perceptron Algorithm (1950's)

First, replace negative examples by positive examples
if $y_\mu = -1$, set $\underline{x}_\mu \rightarrow -\underline{x}_\mu$, $y_\mu \rightarrow -y_\mu$

(note, $\text{sign}(\underline{a} \cdot \underline{x}_\mu) = y_\mu$) 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 $\mu = 1, ..., N$.

Note: the vector $\underline{a}$ need not be unique. It is better to try to maximize the margin (see later this lecture), which requires finding $\underline{a}$ with $|\underline{a}| = 1$, so that $(\underline{a} \cdot \underline{x}_\mu) \geq m$, for $\mu = 1, ..., N$ for the maximum 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 $\underline{a} \cdot \underline{x} = 0$, see figure (2).
(i.e. $\underline{a} \cdot \underline{y} > 0$, is $\underline{y}$ is above plane)
(i.e. $\underline{a} \cdot \underline{y} < 0$, is $\underline{y}$ is below plane)

Proof write $\underline{y} = \lambda \underline{a} + \underline{y}_p$, 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 $|\underline{a}| = 1$

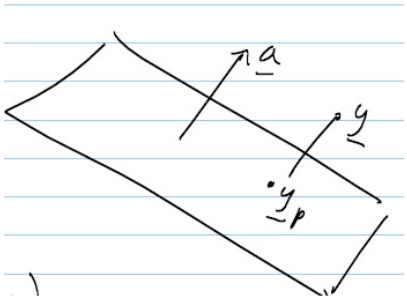

FIGURE 2. $\vec{y}_p$ is the projection of point $\vec{y}$ onto the plane $\vec{a} \cdot \vec{x} = 0$. This means that $\vec{y} - \vec{y}_p \propto \vec{a}$, i.e. the vector joining $\vec{y}$ and $\vec{y}_p$ is parallel to the normal $\vec{a}$ to the plane. If $|\vec{a}| = 1$, then the sign projection is $\vec{a} \cdot (\vec{y} - \vec{y}_p) = \vec{a} \cdot \vec{y}$ (since $\vec{y}_p$ lies on the plane and so $\vec{a} \cdot \vec{y}_p = 0$). The sign projection is positive $\vec{a} \cdot \vec{y} > 0$ if $\vec{y}$ lies above the plane and is negative $\vec{a} \cdot \vec{y} < 0$ if $\vec{y}$ lies below the plane.

3. Perceptron Algorithm

Initialize: $\underline{a}(0) = 0$
loop over $\mu = 1$ to N
If $\underline{x}_\mu$ is misclassified (i.e. $sign(\vec{a} \cdot \vec{x}_\mu) < 0$), set $\underline{a} \to \underline{a} + \underline{x}_\mu$,
Repeat until all samples are classified correctly.

Novikov's Thm
The perception algorithm will converge to a solution weight $\vec{a}$ that classifies all the samples correctly (provided this is possible)

Proof.
Let $\hat{\underline{a}}$ be a separating weight ($m > 0$)
Let $m = \min_\mu \quad \hat{\underline{a}} \cdot \underline{x}_\mu$
Let $\beta^2 = \max_\mu |\underline{x}_\mu|^2$

Suppose
$\underline{x}_t$ is misclassified at time t
so $\underline{a} \cdot \underline{x}_t < 0$
then $\underline{a}_{t+1} - (\beta^2/m)\hat{\underline{a}} = \underline{a}_t - (\beta^2/m)\hat{\underline{a}} + \underline{x}_t$

4.

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

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

It follows that $\|\underline{a}_{t+1} - \frac{\beta^2}{m}\hat{\underline{a}}\|^2 \leq \|\underline{a}_t - \frac{\beta^2}{m}\hat{\underline{a}}\|^2 + \beta^2 - 2\frac{\beta^2 m}{m}$

Hence $\|\underline{a}_{t+1} - \frac{\beta^2}{m}\hat{\underline{a}}\|^2 \leq \|\underline{a}_t - \frac{\beta^2}{m}\hat{\underline{a}}\|^2 - \beta^2$

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

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

This guarantees convergence.

5.

The Perceptron was very influential (i.e. hyped) - and unrealistic claim were made about its effectiveness. But it is very important as a starting point for one style of machine learning.

The Perceptron was criticized (Minksy and Papert) because it was not all to represent all decision rules – i.e. you cannot always separate data into positive and negative by using a plane. But, from the point of view of generalization, it is good that perceptrons cannot learn everything! In technical language, this means that perceptrons have limited capacity and this enables good generalization for some types of data.

6.

Perceptrons can only represent a restricted set of decision rules (e.g. separation by hyperplane). This is a limitation and a virtue. If we can find a separating hyperplane, then it is probably not due to chance alignment of the data (provided $n > (d+1)$), and so it is likely to generalize. In Learning Theory (Vapnik) the quantity $(d+1)$ is the VC dimension of perceptrons and is a measure of the capacity of perceptrons. There is a hypothesis space of classifiers – the set of all perceptrons in this case – and this hypothesis space has a capacity which is $d+1$ for perceptrons. To ensure generalization, you need much more training data than the capacity of the hypothesis space that you are using. We will return to this is later lectures.

Alternative : Multilevel Perceptron. See previous lecture.

7. Linear Separation: Margins & Duality

Modern approach to linear separation

Data: $\{(\underline{x}_\mu, y_\mu) : \mu = 1 \text{ to } N\}$, $\quad y_\mu \in < -1, 1 >$
Hyperplane: $< \underline{x} : \underline{x} \cdot \underline{a} + b = 0 >$ $\quad |\underline{a}| = 1$

The signed distance of a point $\underline{x}$ to the plane is $\underline{a} \cdot \underline{x} + b$
Line $\underline{x}(\lambda) = \underline{x} + \lambda \underline{a})$ $\quad \leftarrow$ to project onto the plane.
Hits plane when $\underline{a} \cdot (\underline{x} + \lambda \underline{a}) = -b$
$\lambda = -(\underline{a} \cdot \underline{x} + b)/|\underline{a}|^2 = -(\underline{a} \cdot \underline{x} + b), \quad$ if $|\underline{a}| = 1$

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

i.e. the positive examples are at least distance C above the plane, and negative examples are at least C below the plane, see figure (3)

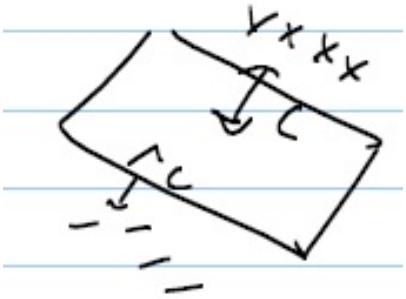

FIGURE 3. Want positive examples to be above the plane by more than C, and negative examples below the plane by more than C.

Large margin is good for generalization (less chance of an accidental alignment).

8.

Now, allow for some data points to be misclassified. Slack variables $< z_1, ..., z_n >$ allow data points to move in direction $\underline{a}$, so that they are on the right side of the margin, see figure (4).

Criteria:
$\max_{a,b,|\underline{a}|=1} C$ 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, $y_\mu\{(\underline{x}_\mu + Cz_\mu\underline{a}) \cdot \vec{a} + b\} \geq C$, like moving $\underline{x}_\mu$ to $\underline{x}_\mu + z_\mu\underline{a}$.

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 data point is correctly classified and is past the margin. If $z_\mu > 0$, then the data point is on the wrong side of the margin, and so had to be moved.

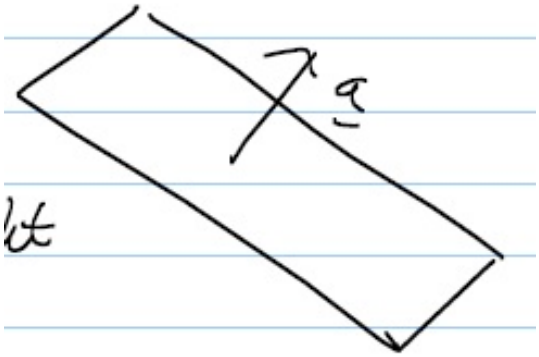

FIGURE 4. Datapoints can use slack variables to move in the direction of the normal $\vec{a}$ to the plane so that they lie on the right side of the margin. But the datapoints must pay a penalty of C times the size of the slack variable.

9. Optimization Criterion

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 criteria that maximizes the margin and minimizes the amount of slack variables used. We remove the constraint $|a| = 1$, set $C = \frac{1}{|\underline{a}|}$.

Criteria: Min $(1/2) \sum \underline{a} \cdot \underline{a} + \gamma \sum_\mu z_\mu$
s.t. $y_\mu(\underline{x}_\mu \cdot \underline{a} + b) \geq 1 - z_\mu, \quad \forall\ \mu z_\mu \geq 0$

Quadratic Primal Problem using lagrange multipliers

$L_p(\vec{a}, b, z; \alpha, \tau) = \frac{1}{2}\underline{a} \cdot \underline{a} + \gamma \sum_\mu z_\mu - \sum_\mu \alpha_\mu < y_\mu(\underline{x}_\mu \cdot \underline{a} + b) - (1 - z_\mu) >$
$- \sum_\mu \tau_\mu z_\mu$

The $< \alpha_\mu >$ & $< \tau_\mu >$ are Lagrange parameters needed to enforce the inequality constraints. We require that $\alpha_\mu \geq 0, \tau_\mu \geq 0, \forall \mu$.

The function $L_p(\vec{a}, b, z; \alpha, \tau)$ should be minimized with respect to the primal variables $\vec{a}, z$ and maximized with respect to the dual variables α, τ. Note this means that if the constraints are satisfied then we need to set the corresponding lagrange parameter to be zero (to maximize). For example, if $y_\mu(\underline{x}_\mu \cdot \underline{a} + b) - (1 - z_\mu) > 0$ for some μ then we set $\alpha_\mu = 0$ because the term $-\alpha_\mu\{y_\mu(\underline{x}_\mu \cdot \underline{a} + b) - (1 - z_\mu)\}$ is non-positive, and so the maximum value occurs when $\alpha_\mu = 0$. But if the constraint is not satisfied – e.g., $y_\mu(\underline{x}_\mu \cdot \underline{a} + b) - (1 - z_\mu) < 0$ – then the lagrange parameter will be positive. So there is a relationship between the lagrange parameters which are positive (non-zero) and the constraints which are satisfied. This will have important consequences.

10. Primal and Dual

L_p is a function of the primal variable $\underline{a}, b, < z_\mu >$ 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.

$\frac{\partial L_p}{\partial \underline{a}} = 0 \Rightarrow \hat{\underline{a}} = \sum_\mu \alpha_\mu y_\mu \underline{x}_\mu$
$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_\mu \alpha_\mu y_\mu = 0$
$\frac{\partial L_p}{\partial z_\mu} = 0 \Rightarrow \alpha_\mu = \gamma - \hat{\tau}_\mu, \forall \mu$

The classifier is $sign < \hat{\underline{a}} \cdot \underline{x} + \hat{b} >= sign < \sum_\mu \alpha_\mu y_\mu \underline{x}_\mu \cdot \underline{x} + b >$, by using the equation $\frac{\partial L_p}{\partial \underline{a}} = 0$.

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

11.

The constraints are $y_\mu(\underline{x}_\mu \cdot \tilde{\vec{a}} + \tilde{b}) \geq 1 - \hat{z_\mu} \qquad \hat{z_\mu} \geq 0, \hat{\tau_\mu} \geq 0$

By theory of Quadratic Programming - $\alpha_\mu > 0$, only if either:
(i) $z_\mu > 0$ slack variable is used.
(ii) z_μ, but$y_\mu(\underline{x}_\mu \cdot \tilde{\vec{a}} + \tilde{b}) = 1$ – i.e. data point is on the margin.

The classifier depends only on the support vectors, the other datapoint do not matter, see figure (5)

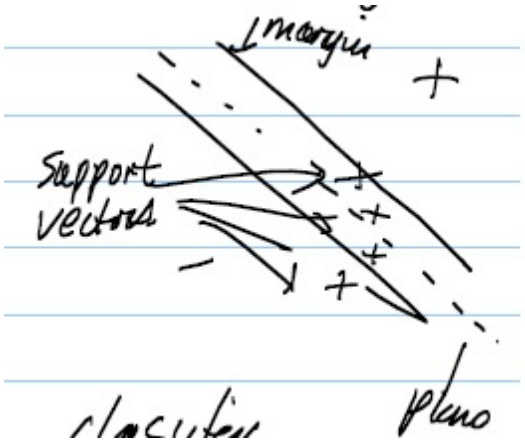

FIGURE 5. The classifier depends only on the support vectors.

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 & when we learn probability models for each class and then use the Bayes classifier (but we can derive it as an approximation to probabilistic methods – see end of this lecture). Note that the support vectors include all the datapoints which have positive slack variables – i.e. which are on the wrong sides of the margin.

12. Dual Formulation

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

$L_p = \sum_\mu \alpha_\mu - \frac{1}{2}\sum_{\mu,\nu} \alpha_\mu \alpha_\nu y_\mu y_\nu \underline{x}_\mu \underline{x}_\nu$
with constraint $0 \leq \alpha_\mu \leq \tau, \sum_\mu \alpha_\mu y_\mu = 0$

There are standard packages to solve this. (Although they get slow if you have a very large amount of data).

Knowing $< \hat{\alpha}_\mu >$, will give us the solution $\underline{\hat{\alpha}} = \sum_\mu \hat{\alpha}_\mu y_\mu \underline{x}_\mu$,
(only a little more work needed to get $\hat{b}$)

13. Relation between Primal & Dual

Now we show how to obtain the dual formulation from the primal. The method we use is only correct if the primal is a convex function (but it is a positive quadratic function with linear constraints, which is convex).

Start with dual formulation. L_p

Rewrite it as
$L_p = -\frac{1}{2}\underline{a} \cdot \underline{a} + \sum_\mu \alpha_\mu + \underline{a} < \underline{a} - \sum_\mu \alpha_\mu y_\mu \underline{x}_\mu >$
$+ \sum_\mu z_\mu < \gamma - \tau_\mu - \alpha_\mu > -b \sum_\mu \alpha_\mu y_\mu$

Extremize w.r.t. $\underline{a}, b, < z_\mu > gives$:
$\underline{\hat{a}} = \sum_\mu \alpha_\mu y_\mu \underline{x}_\mu, \sum_\mu \alpha_\mu y_\mu = 0, \gamma - \tau_\mu - \alpha_\mu = 0$

substituting back into L_p gives:
$L_p = -\frac{1}{2}\sum_{\mu,\nu}\alpha_\mu\alpha_\nu y_\mu y_\nu \underline{x}_\mu \underline{x}_\nu + \sum_\mu \alpha_\mu$

maximize w.r.t. $<\alpha_\mu>$.

14. 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 $\underline{\alpha}_\mu \to \underline{\alpha}_\mu + 1$
$b \to b + y_\mu K^2$

K is radius of smallest ball containing the data.

15. Equivalent Formulation – and alternative criteria.

Suppose we look at the primal function L_p. Consider the constraint $y_\mu(\vec{x}_\mu \cdot \vec{a} + b) - 1 > 0$. If this constraint is satisfied, then it is best to set the slack variable $z_\mu = 0$ (because otherwise we pay a penalty γ for it). If the constraint is not satisfied, then we set the slack variable to be $z_\mu = 1 - y_\mu(\vec{x}_\mu \cdot \vec{a})$ because this is the smallest value of the slack variable which satisfies the constraint. We can summarize this by paying a penalty $\max\{0, 1 - y_\mu(\vec{x}_\mu \cdot \vec{a} + b)\}$ – if the constraint is satisfied, then the maximum is 0 but, if not,

the maximum is $1 - y_\mu(\vec{x}_\mu \cdot \vec{a})$ which is minimum value of the slack variable (to make the constraint satisfied).

This gives an energy function:

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

This can be derived as an approximation to several criteria. We can start with a loss function $l(y_\mu, y)$ and penalize $\sum_\mu l(y_\mu, \hat{y}(x_\mu, \vec{a}, b)$ with an additional term $1/2|\vec{a}|^2$ to maximize the margin. Then we can obtain this criteria as an approximation.

Alternatively, we can define a probabilistic model $P(y|x, \vec{a}, b) = \exp\{y(\vec{a} \cdot \vec{x} + b)/Z[\vec{a}, b, x]$ and use the expected log-risk, with the loss function above. We have a Gaussian prior on $\vec{a}$. Then we obtain this criteria as an approximation. This will be discussed later in the course.