Linear Classifiers and Perceptron (2014)

**Sample:** \( \{ (x_\mu, y_\mu) : \mu \in \{1 \ldots N\} \} \)

\[ y_\mu \in \{\pm 1\} \]

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

**Example:** a plane \( a \cdot x = 0 \) s.t. \( \text{sign}(a \cdot x) = y \)

\[ a \cdot x_\mu > 0, \quad y_\mu = +1 \]

\[ a \cdot x_\mu \leq 0, \quad y_\mu = -1 \]

Plane goes through the origin \((a \cdot 0 = 0)\)

**Perceptron Algorithm (1950’s)**

First, replace -ve examples by the example:

If \( y_\mu = -1 \), set \( x_\mu \rightarrow -x_\mu, \quad y_\mu \rightarrow -y_\mu \)

(Note: require \( \text{sign}(a \cdot x_\mu) = y_\mu \), this is equivalent to \( \text{sign}(-a \cdot x_\mu) = -y_\mu \)
2) This reduces to finding a plane

\[ \text{s.t. } a \cdot x \mu \geq 0, \text{ for } \mu = \text{lin N} \]

Note: the vector \( a \) need not be unique.

It is better to try to maximize the margin (see next lecture). To find a

with \( |a| = 1 \), so that \( a \cdot x \mu \geq m \), \( \forall \mu \geq 0 \)

for the maximum value of \( m \).

More geometry

Claim:

\( a \) is a unit vector

\( |a| = 1 \), then \( a \cdot y \) is

the sign distance of \( y \)

to the plane \( a \cdot x = 0 \).

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

Proof: write \( y = y_p + \lambda a \), where \( y_p \) is the projection

of \( y \) into the plane. By definition \( a \cdot y_p = 0 \),

hence \( \lambda = (a \cdot y) / (a \cdot y) = (a \cdot y), \quad \forall \lambda \)
(3) **Perceptron Algorithm.**

Initialze: \( a(0) = 0 \).

Loop over \( \mu = 1 \to N \)

if \( x_\mu \) is misclassified, set \( a \to a + x_\mu \)

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 \( m = \min_{\mu=1}^{N} \hat{a} \cdot x_\mu \quad (m > 0) \)

Let \( \beta^2 = \max_{\mu=1}^{N} |x_\mu|^2 \)

Suppose \( x_t \) is misclassified at time \( t \)

\[ a_\mu \cdot x_t < 0 \]

\[ a_{t+1} = (\beta^2/m) \hat{a} = a_t - (\beta^2/m) \hat{a} + x_t \]
\[ (4) \quad \| \mathbf{a}_{t+1} - \beta^2 \mathbf{m} \mathbf{a}_{t} \|_2^2 = \| \mathbf{a}_{t} - (\mathbf{b}^2 \mathbf{m}) \mathbf{a} \|_2^2 + \| \mathbf{x}_t \|_2^2 - 2 \mathbf{a}_t \cdot (\mathbf{b}^2 \mathbf{m}) \cdot \mathbf{x}_t. \]

Using \( \| \mathbf{x}_t \|_2 \leq \beta \), \( \mathbf{a}_t \cdot \mathbf{x}_t < 0 \), \( -\mathbf{a}_t \cdot \mathbf{x}_t < -1 \)

It follows that
\[ \| \mathbf{a}_{t+1} - \beta^2 \mathbf{m} \mathbf{a}_{t} \|_2^2 \leq \| \mathbf{a}_{t} - \beta^2 \mathbf{m} \mathbf{a} \|_2^2 + \beta^2 - 2 \beta^2 \mathbf{m} \mathbf{m} \]

Hence
\[ \| \mathbf{a}_{t+1} - \beta^2 \mathbf{m} \mathbf{a}_{t} \|_2^2 \leq \| \mathbf{a}_{t} - \beta^2 \mathbf{m} \mathbf{a} \|_2^2 - \beta^2. \]

So, each time we update a weight, we reduce the quantity \( \| \mathbf{a}_{t} - \beta^2 \mathbf{m} \mathbf{a} \|_2^2 \) by a fixed amount \( \beta^2 \). \( \| \mathbf{a}_{t} - \beta^2 \mathbf{m} \mathbf{a} \|_2^2 \) is bounded by \( \frac{\beta^4 \| \mathbf{a} \|_2^2}{\mathbf{m}^2} \).

So we can update the weights at most \( \frac{\beta^2 \| \mathbf{a} \|_2^2}{\mathbf{m}^2} \) times.

Guarantees convergence.
Support Vector Machines

Linear Separation: Margins & Quality

Modern approach to linear separation.

Data \( \{(x_\mu, y_\mu) : \mu \in \{1, \ldots, N\}\} \), \( y_\mu \in \{-1, 1\} \)

Hyperplane \( \langle x : x \cdot a + b = 0 \rangle \), \( ||a|| = 1 \).

The signed distance of a point \( x \) to the plane is \( a \cdot x + b \).

To project on line \( x \cdot (x + 2a) \)

Hit plane when \( a \cdot (x + 2a) = -b \)

\[ \gamma = -\frac{(a \cdot x + b)}{||a||^2} = -\frac{(a \cdot x + b)}{1} \]

Seek classifier with biggest margin

\[ \max C \text{ s.t. } y_\mu (x_\mu \cdot a + b) \geq C \]

\[ a \parallel b, ||a|| = 1 \]

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.
Now, allow for some datapoints to be misclassified. Slack variables allow datapoints to move in direction $\alpha$, so that they are on the right side of the margin. Slack variables $\{z_i, \ldots, z_n\}$

**Criterion:** $\max \ a \ s.t. \ y_\mu (\frac{x_\mu \cdot a + b}{\sqrt{a^T a}}) > c (1 - z_\mu)$

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

**Additionally:** $y_\mu \{(\frac{x_\mu + cz_\mu a}{\sqrt{a^T a}}), a + b\} > c$

like moving $x_\mu$ to $x_\mu + cz_\mu a$.

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

- 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.
Task: We need to estimate several quantities simultaneously:

1. The plane \(a, b\)
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 \(a\) by \(a \rightarrow \frac{1}{c}a\), hence \(C = \frac{1}{c}\alpha\).

The Max Margin Criterion:

\[
\min \quad \frac{1}{2} a^T a + \sum_{\mu} z_{\mu}
\]

s.t.
\[
y_{\mu} (x_{\mu} \cdot a + b) \geq 1 - z_{\mu}, \quad z_{\mu} \geq 0, \quad \forall \mu
\]

Quadratic Primal Problem requires Lagrange multipliers.

\[
L_p = \frac{1}{2} a^T a + \sum_{\mu} z_{\mu} - \sum_{\mu} \alpha_{\mu} \left( y_{\mu} (x_{\mu} a + b) - (1 - z_{\mu}) \right) - \sum_{\mu} \mu z_{\mu}
\]

The \(\alpha_{\mu}\) and \(z_{\mu}\) are Lagrange parameters needed to enforce the inequality constraints: \(\alpha_{\mu} \geq 0, z_{\mu} \geq 0, \forall \mu\).
Lp is a function of the primal variables α, b, and the lagrange parameters \( \{\lambda_\mu \} \).

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

\[
\frac{\partial L_p}{\partial \alpha} = 0 \quad \Rightarrow \quad \hat{\alpha} = \sum_\mu \lambda_\mu y_\mu x_\mu.
\]

\[
\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_\mu \lambda_\mu y_\mu = 0.
\]

\[
\frac{\partial L_p}{\partial \lambda_\mu} = 0 \quad \Rightarrow \quad \hat{\lambda}_\mu = 0 - \hat{\alpha}_\mu, \quad \forall \mu.
\]

The classifier is

\[
\text{sign} \left( \hat{\alpha} \cdot x + \hat{b} \right) = \text{sign} \left( \sum_\mu \lambda_\mu y_\mu x_\mu \cdot x - \hat{b} \right)
\]

Support vectors, the solution depends only on the vectors \( x_\mu \) for which \( \lambda_\mu \neq 0 \).
The constraints are
\[ y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - z_i \]
\[ z_i \geq 0, \quad \xi_i \geq 0. \]

By theory of Quadratic Programming,
\( \xi_i > 0 \), only if either:

(i) \( z_i > 0 \) slack variable is used

(ii) \( z_i = 0 \), but \( y_i (\mathbf{x}_i \cdot \mathbf{w} + b) = 1 \)

data point is on the margin

The classifier depends only on the support vectors; the other data points do not matter.

This is actually reasonable – the classifier plane 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.
(10)

**Dual formulation**

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

\[
\lambda_d = \sum_{\mu} \alpha_\mu - \frac{1}{2} \sum_{\mu,\nu} \alpha_\mu \alpha_\nu y_\mu y_\nu x_\mu x_\nu,
\]

with constraints:

- \( 0 \leq \alpha_\mu \leq 2 \),
- \( \sum_\mu \alpha_\mu y_\mu = 0 \).

There are standard packages to solve this.

Knowing \( \langle \hat{\alpha}_\mu \rangle \), will give us the solution:

\[
\hat{x}_\mu = \sum_\mu \hat{\alpha}_\mu y_\mu x_\mu,
\]

(only a little more work to get \( \hat{b} \).)
(11) Relationship between Primal & Dual

Start with dual formulation $L_d$

Rewrite $L_p$ as:

$$L_p = \left(-\frac{1}{2}\right) a,a + \sum \alpha \mu + a, \left( a - \sum \alpha \mu y \right)x$$

$$+ \sum \frac{\alpha}{\mu} (\delta - \zeta \mu - \alpha \mu) - b \sum \alpha \mu y.$$ 

Extremise w.r.t. $a, b, \{\delta, \zeta\}$ gives:

$$\hat{a} = \frac{\sum \alpha \mu y}{\mu}, \frac{\sum \alpha \mu y}{\mu} = 0, \delta - \zeta \mu - \alpha \mu = 0$$

Substituting back into $L_p$ gives:

$$L_d = -\frac{1}{2} \sum \frac{\alpha}{\mu} a, a y y \mu y \mu y \mu \mu \mu + \sum \alpha \mu$$

Extremise 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:

\[ \alpha = \sum_{\mu} \alpha_{\mu} y_{\mu} x_{\mu}. \]

Perceptron Update Rule:

If data \( x_{\mu} \) is misclassified,

i.e. \( y_{\mu} (\alpha \cdot x_{\mu} + b) \leq 0 \)

Set \( \alpha_{\mu} \rightarrow \alpha_{\mu} + 1 \)

\( b \rightarrow b + y_{\mu} R^2 \)

\( R \) 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.
The Max-Margin Criterion

\[ L_P = \frac{1}{2} \alpha^T \alpha + \sum_{\mu} z_{\mu} \]

with constraints

\[ y_{\mu} (x_{\mu} \cdot a + b) - (1 - z_{\mu}) \geq 0 \]

\[ z_{\mu} \geq 0 \]

\[ z_{\mu} \geq 1 - y_{\mu} (x_{\mu} \cdot a + b) \]

Hence

\[ L_P = \frac{1}{2} \alpha^T \alpha + \sum_{\mu} z_{\mu} \max \{ 0, 1 - y_{\mu} (x_{\mu} \cdot a + b) \} \]

Hinge Loss - Loss = 0 if \( y_{\mu} (x_{\mu} \cdot a + b) \geq 1 \)

is, data point is on the correct side of margin

otherwise, loss is \( 1 - y_{\mu} (x_{\mu} \cdot a + b) \)

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 \[ \frac{1}{2} \alpha^T \alpha \]

\[ \frac{1}{c} \sum_{\mu=1}^{n} \max \{ 0, 1 - y_{\mu} (x_{\mu} \cdot a + b) \} \]

Empirical Risk

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

which have large \( \alpha \).

Why? This helps generalization.

If we only tried to minimize the loss function, we may overfit the data, because the space of possible decision rules is very big (all values of \( a \) and \( b \)). If we penalize these rules with big \( \alpha \) 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

$$L_p = \frac{1}{2} \| a \|^2 + \frac{\lambda}{2} \sum_{\mu=1}^{\ell} \max (0, 1 - y_\mu (x_\mu a + b))$$

Consider the second term for one datapoint $x_\mu , y_\mu$

$$\frac{\partial}{\partial a} \max (0, 1 - y_\mu (x_\mu a + b)) = 0 \quad \text{if} \quad y_\mu (x_\mu a + b) > 1$$

$$= -y_\mu x_\mu \quad \text{if} \quad y_\mu (x_\mu a + b) < 1$$

**Online learning:**

- Select data $(x_\mu , y_\mu)$ at random.
- Compute

$$\alpha^t \to \alpha^t - \frac{1}{2\eta} \alpha^t - T \left\{ 0, \quad \text{if} \quad y_\mu (x_\mu a + b) > 1 \right\}$$

$$= -y_\mu x_\mu \quad \text{if} \quad y_\mu (x_\mu a + b) < 1$$

This is almost exactly the 1950's perception algorithm!  
(The $-y_\mu$ term is like converting -ve examples to +ve.)

The $\frac{1}{2\eta} \alpha^t$ is from the regularizer.