Statistical Data Mining and Machine Learning
Hilary Term 2016

Dino Sejdinovic
Department of Statistics
Oxford

Slides and other materials available at:
http://www.stats.ox.ac.uk/~sejdinov/sdmml

Support Vector Machines

These slides are based on Arthur Gretton's UCL course on Advanced Topics in Machine Learning

Optimization and the Lagrangian

Optimization problem on $x \in \mathbb{R}^d$ / primal,

$$\begin{aligned} \text{minimize} \quad & f_0(x) & \\ \text{subject to} \quad & f_i(x) \leq 0 & i = 1, \ldots, m \\ & h_j(x) = 0 & j = 1, \ldots r. \end{aligned}$$

- domain $\mathcal{D} := \bigcap_{i=0}^{m} \mathrm{dom} f_i \;\cap\; \bigcap_{j=1}^{r} \mathrm{dom} h_j$ (nonempty).
- p^*: the (primal) optimal value

Idealy we would want an unconstrained problem

$$\text{minimize}\, f_0(x) + \sum_{i=1}^{m} I_-\left(f_i(x)\right) + \sum_{j=1}^{r} I_0\left(h_j(x)\right),$$

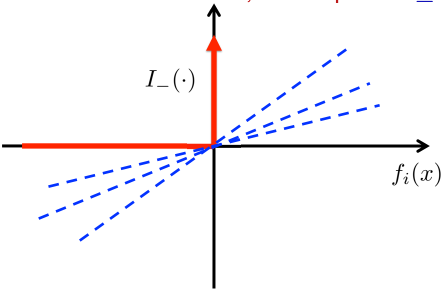
where $I_-(u) = \begin{cases} 0, & u \leq 0 \\ \infty, & u > 0 \end{cases}$ and $I_0(u) = \begin{cases} 0, & u = 0 \\ \infty, & u \neq 0 \end{cases}$.

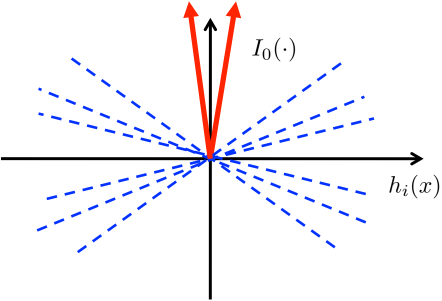
Lower bound interpretation of Lagrangian

The **Lagrangian** $L : \mathbb{R}^d \times \mathbb{R}^m \times \mathbb{R}^r \to \mathbb{R}$ is an (easier to optimize) lower bound on the original problem:

$$L(x, \lambda, \nu) := f_0(x) + \sum_{i=1}^{m} \underbrace{\lambda_i f_i(x)}_{\leq I_-(f_i(x))} + \sum_{j=1}^{r} \underbrace{\nu_j h_j(x)}_{\leq I_0(h_j(x))},$$

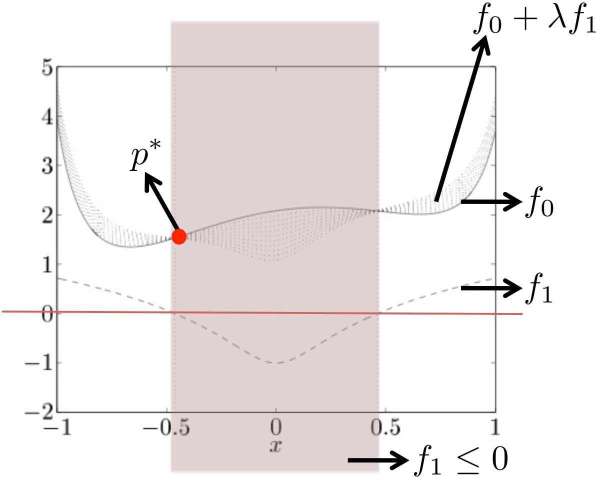
The vectors λ and ν are called **Lagrange multipliers** or **dual variables**. To ensure a lower bound, we require $\lambda \succeq 0$





Lower bound interpretation of Lagrangian

Simplest example: minimize over x the function $L(x, \lambda) = f_0(x) + \lambda f_1(x)$



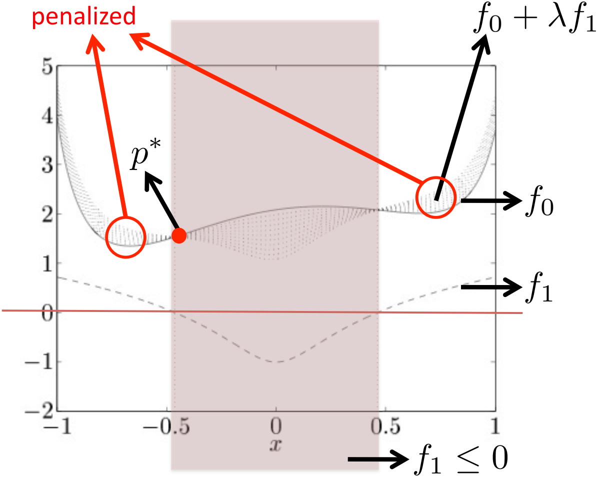
Reminders:

- f_0 is function to be minimized.
- $f_1 \leq 0$ is inequality constraint
- $\lambda \geq 0$ is Lagrange multiplier
- p^* is minimum f_0 **in constraint set**

Figure from Boyd and Vandenberghe

Lower bound interpretation of Lagrangian

Simplest example: minimize over x the function $L(x,\lambda) = f_0(x) + \lambda f_1(x)$



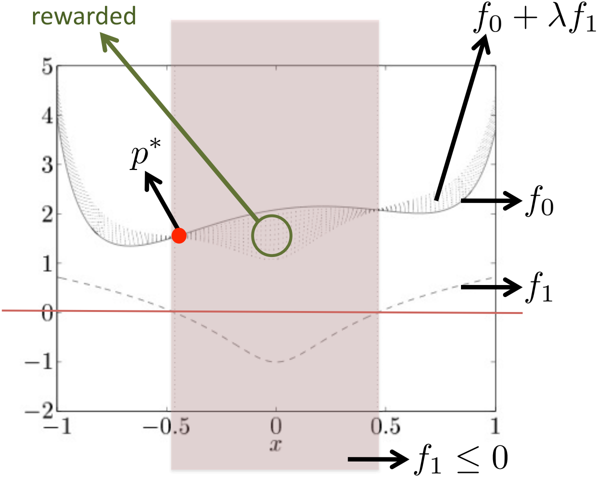
Reminders:

- f_0 is function to be minimized.
- $f_1 \leq 0$ is inequality constraint
- $\lambda \geq 0$ is Lagrange multiplier
- p^* is minimum f_0 **in constraint set**

Figure from Boyd and Vandenberghe

Lower bound interpretation of Lagrangian

Simplest example: minimize over x the function $L(x, \lambda) = f_0(x) + \lambda f_1(x)$



Reminders:

- f_0 is function to be minimized.
- $f_1 \leq 0$ is inequality constraint
- $\lambda \geq 0$ is Lagrange multiplier
- p^* is minimum f_0 **in constraint set**

Figure from Boyd and Vandenberghe

Lagrange dual: lower bound on optimum p^*

The **Lagrange dual function:** minimize Lagrangian When $\lambda \succeq 0$ and $f_i(x) \leq 0$, Lagrange dual function is

$$g(\lambda, \nu) := \min_{x \in \mathcal{D}} L(x, \lambda, \nu).$$

A **dual feasible** pair (λ, ν) is a pair for which $\lambda \succeq 0$ and $(\lambda, \nu) \in \mathrm{dom}(g)$.
We will show: for any $\lambda \succeq 0$ and ν,

$$g(\lambda, \nu) \leq f_0(x)$$

wherever

$$\begin{aligned} f_i(x) &\leq 0 \\ h_j(x) &= 0 \end{aligned}$$

(including at optimal point $f_0(x^*) = p^*$).

Lagrange dual is a lower bound on p^*

Assume $\tilde{x}$ is feasible, i.e. $f_i(\tilde{x}) \leq 0$, $h_i(\tilde{x}) = 0$, $\tilde{x} \in \mathcal{D}$, $\lambda \succeq 0$. Then

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{r} \nu_i h_i(\tilde{x}) \leq 0$$

Thus

$$\begin{aligned} g(\lambda, \nu) \quad &:= \quad \min_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{r} \nu_i h_i(x) \right) \\ &\leq \quad f_0(\tilde{x}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{r} \nu_i h_i(\tilde{x}) \\ &\leq \quad f_0(\tilde{x}). \end{aligned}$$

This holds for every feasible $\tilde{x}$, hence lower bound holds.

Best lower bound: maximize the dual

Best lower bound $g(\lambda, \nu)$ on the optimal solution p^* of original problem: **Lagrange dual problem**

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0. \end{array}$$

Dual feasible: (λ, ν) with $\lambda \succeq 0$ and $g(\lambda, \nu) > -\infty$.
Dual optimal: solutions (λ^*, ν^*) to the dual problem, d^* is optimal value.
Weak duality always holds:

$$\max_{\lambda \succeq 0, \nu} \underbrace{\min_{x \in \mathcal{D}} L(x, \lambda, \nu)}_{= g(\lambda, \nu)} = d^* \leq p^* = \min_{x \in \mathcal{D}} \underbrace{\max_{\lambda \succeq 0, \nu} L(x, \lambda, \nu)}_{= \begin{cases} f_0(x) & \text{if constraints satisfied,} \\ \infty & \text{otherwise.} \end{cases}}$$

Strong duality: (does **not** always hold, conditions given later):

$$d^* = p^*.$$

If strong duality holds: can solve the **dual problem** to find p^*.

How do we know if strong duality holds?

Conditions under which strong duality holds are called **constraint qualifications** (they are sufficient, but not necessary)

(Probably) best known sufficient condition: Strong duality holds if

- Primal problem is **convex**, i.e. of the form

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, n \\ & Ax = b & \end{array}$$

  for convex $f_0, \ldots, f_m$, **and**

- **Slater's condition:** there exists a strictly feasible point $\tilde{x}$, such that $f_i(\tilde{x}) < 0$, $i = 1, \ldots, n$ (reduces to the existence of any feasible point when inequality constraints are affine, i.e., $Cx \preceq d$).

A consequence of strong duality...

Assume primal is equal to the dual. What are the consequences?

- x^* solution of original problem (minimum of f_0 under constraints),
- (λ^*, ν^*) solutions to dual

$$\begin{aligned}
f_0(x^*) \quad & \underset{\text{(assumed)}}{=} \quad g(\lambda^*, \nu^*) \\
& \underset{\text{(g definition)}}{=} \quad \min_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^* f_i(x) + \sum_{i=1}^{p} \nu_i^* h_i(x) \right) \\
& \underset{\text{(inf definition)}}{\leq} \quad f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{p} \nu_i^* h_i(x^*) \\
& \underset{(4)}{\leq} \quad f_0(x^*),
\end{aligned}$$

(4): (x^*, λ^*, ν^*) satisfies $\lambda^* \succeq 0$, $f_i(x^*) \leq 0$, and $h_i(x^*) = 0$.

...is complementary slackness

From previous slide,

$$\sum_{i=1}^{m} \lambda_i^* f_i(x^*) = 0, \tag{1}$$

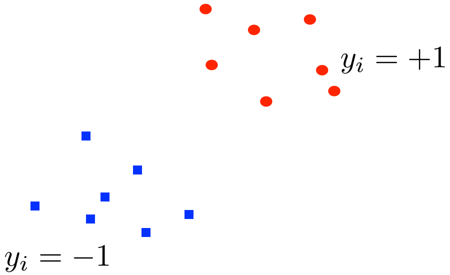
which is the condition of **complementary slackness**. This means

$$\begin{aligned} \lambda_i^* > 0 \quad &\Longrightarrow \quad f_i(x^*) = 0, \\ f_i(x^*) < 0 \quad &\Longrightarrow \quad \lambda_i^* = 0. \end{aligned}$$

From λ_i, read off which inequality constraints are strict.

Linearly separable points

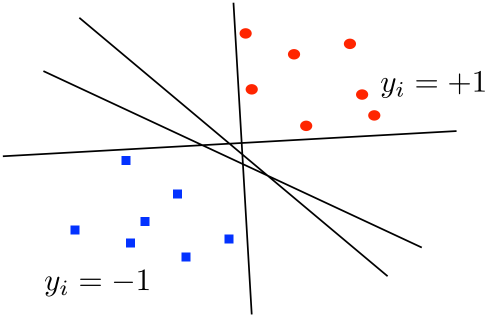
Classify two clouds of points, where there exists a hyperplane which linearly separates one cloud from the other without error.



Data given by $\{x_i, y_i\}_{i=1}^{n}$, $x_i \in \mathbb{R}^p$, $y_i \in \{-1, +1\}$

Linearly separable points

Classify two clouds of points, where there exists a hyperplane which linearly separates one cloud from the other without error.

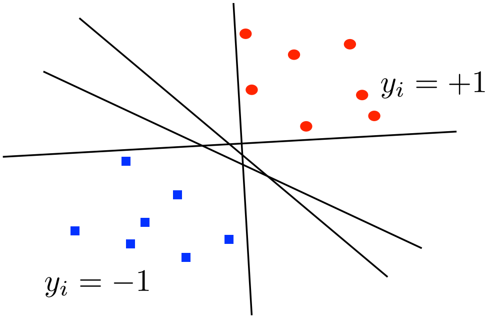


Hyperplane equation $w^\top x + b = 0$. Linear discriminant given by

$$\hat{y}(x) = \text{sign}(w^\top x + b)$$

Linearly separable points

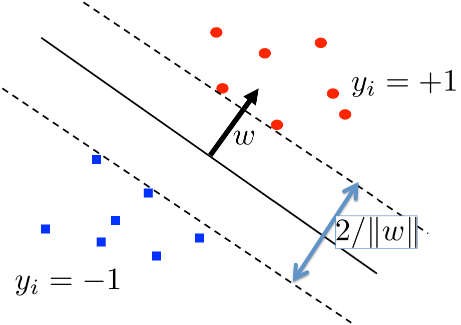
Classify two clouds of points, where there exists a hyperplane which linearly separates one cloud from the other without error.



For a datapoint close to the decision boundary, a small change leads to a change in classification. Can we make the classifier more robust?

Linearly separable points

Classify two clouds of points, where there exists a hyperplane which linearly separates one cloud from the other without error.



Smallest distance from each class to the separating hyperplane $w^\top x + b$ is called the **margin**.

Maximum margin classifier, linearly separable case

This problem can be expressed as follows:

$$\max_{w,b}\,(\text{margin}) = \max_{w,b}\left(\frac{1}{\|w\|}\right)$$

subject to

$$\begin{cases} w^\top x_i + b \geq 1 & i\ :\ y_i = +1, \\ w^\top x_i + b \leq -1 & i\ :\ y_i = -1. \end{cases}$$

The resulting classifier is

$$\hat{y}(x) = \operatorname{sign}(w^\top x + b),$$

We can rewrite to obtain a **quadratic program**:

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

subject to

$$y_i(w^\top x_i + b) \geq 1.$$

Maximum margin classifier: with errors allowed

Allow “errors”: points within the margin, or even on the wrong side of the decision boundary. Ideally:

$$\min_{w,b} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \mathbb{I}[y_i \left(w^\top x_i + b \right) < 0] \right),$$

where C controls the tradeoff between maximum margin and loss. Replace with **convex upper bound**:

$$\min_{w,b} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} h \left(y_i \left(w^\top x_i + b \right) \right) \right).$$

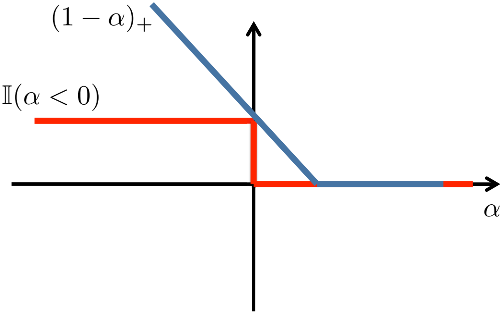
with hinge loss,

$$h(\alpha) = (1-\alpha)_+ = \begin{cases} 1-\alpha, & 1-\alpha > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Hinge loss

Hinge loss:

$$h(\alpha) = (1-\alpha)_+ = \begin{cases} 1-\alpha, & 1-\alpha > 0 \\ 0, & \text{otherwise.} \end{cases}$$



Support vector classification

Substituting in the hinge loss, we get a standard regularised empirical risk minimisation problem - where regularisation naturally arises from the margin penalty.

$$\min_{w,b}\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} h\left(y_i\left(w^\top x_i + b\right)\right)\right).$$

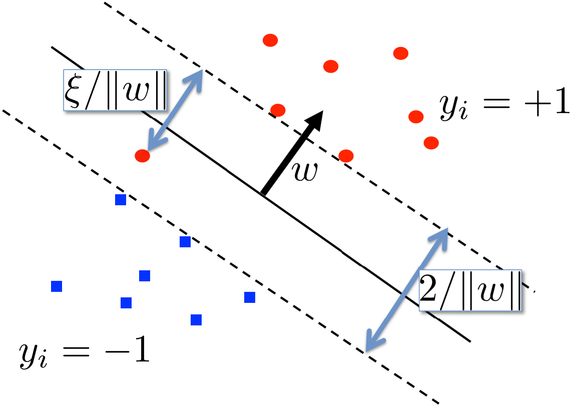
Using substitution $\xi_i = h\left(y_i\left(w^\top x_i + b\right)\right)$, we obtain an equivalent formulation (standard C-SVM):

$$\min_{w,b,\xi}\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i\right)$$

subject to

$$\xi_i \geq 0 \qquad y_i\left(w^\top x_i + b\right) \geq 1 - \xi_i$$

Support vector classification



Duality

As a convex constrained optimization problem with affine constraints in w, b, ξ, strong duality holds.

$$\begin{aligned} \text{minimize} \quad & f_0(w, b, \xi) := \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & f_i(w, b, \xi) := 1 - \xi_i - y_i \left(w^\top x_i + b\right) \leq 0,\ i = 1, \ldots, n \\ & f_{n+i}(w, b, \xi) := -\xi_i \leq 0,\ i = 1, \ldots, n. \end{aligned}$$

Support vector classification: Lagrangian

The Lagrangian: $L(w, b, \xi, \alpha, \lambda) =$

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1 - \xi_i - y_i\left(w^\top x_i + b\right)\right) + \sum_{i=1}^{n}\lambda_i(-\xi_i)$$

with dual variable constraints

$$\alpha_i \geq 0, \qquad \lambda_i \geq 0.$$

Minimize wrt the primal variables w, b, and ξ.
Derivative wrt w:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \qquad w = \sum_{i=1}^{n}\alpha_i y_i x_i.$$

Derivative wrt b:

$$\frac{\partial L}{\partial b} = \sum_i y_i\alpha_i = 0.$$

Support vector classification: Lagrangian

Derivative wrt ξ_i:

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \lambda_i = 0 \qquad \alpha_i = C - \lambda_i.$$

Since $\lambda_i \geq 0$,

$$\alpha_i \leq C.$$

Now use complementary slackness:

Non-margin SVs (margin errors): $\alpha_i = C > 0$:

1. We immediately have $y_i \left(w^\top x_i + b \right) = 1 - \xi_i$.
2. Also, from condition $\alpha_i = C - \lambda_i$, we have $\lambda_i = 0$, so $\xi_i \geq 0$

Margin SVs: $0 < \alpha_i < C$:

1. We again have $y_i \left(w^\top x_i + b \right) = 1 - \xi_i$.
2. This time, from $\alpha_i = C - \lambda_i$, we have $\lambda_i > 0$, hence $\xi_i = 0$.

Non-SVs (on the correct side of the margin): $\alpha_i = 0$:

1. From $\alpha_i = C - \lambda_i$, we have $\lambda_i > 0$, hence $\xi_i = 0$.
2. Thus, $y_i \left(w^\top x_i + b \right) \geq 1$

The support vectors

We observe:

1. The solution is sparse: points which are neither on the margin nor “margin errors” have $\alpha_i = 0$
2. The support vectors: only those points on the decision boundary, or which are margin errors, contribute.
3. Influence of the non-margin SVs is bounded, since their weight cannot exceed C.

Support vector classification: dual function

Thus, our goal is to maximize the dual,

$$\begin{aligned}
g(\alpha, \lambda) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1 - y_i\left(w^\top x_i + b\right) - \xi_i\right) \\
&\quad + \sum_{i=1}^{n}\lambda_i(-\xi_i) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\top x_j + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\top x_j \\
&\quad -b\underbrace{\sum_{i=1}^{n}\alpha_i y_i}_{0} + \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n}(C-\alpha_i)\xi_i \\
&= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^\top x_j.
\end{aligned}$$

Dual C-SVM

$$\text{maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^\top x_j,$$

subject to the constraints

$$0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} y_i \alpha_i = 0$$

This is a quadratic program. From α, obtain the hyperplane with

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

(follows from complementary slackness in the derivation of the dual). Offset b can be obtained from any of the margin SVs (for which $\alpha_i \in (0, C)$): $1 = y_i \left(w^\top x_i + b \right)$.

Solution depends on data through inner products only

Dual program

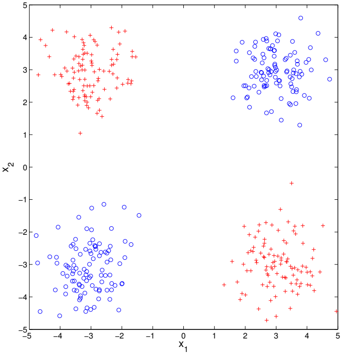
$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^\top x_j \qquad \text{subject to} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \preceq \alpha \preceq C \end{cases}$$

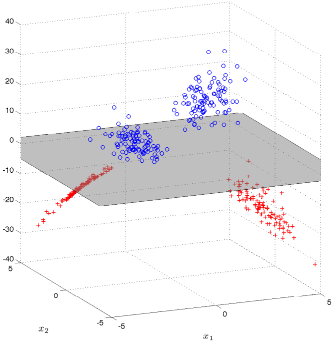
only depends on inputs x_i through their inner products (similarities) with other inputs.
Decision function

$$\hat{y}(x) = \operatorname{sign}(w^\top x + b) = \operatorname{sign}(\sum_{i=1}^{n} \alpha_i y_i x_i^\top x + b)$$

also depends only on the similarity of a test point x to the training points x_i.
Thus, we do not need explicit inputs - just their pairwise similarities.
Key property: even if $p > n$, it is still the case that $w \in \text{span}\{x_i : i = 1, \dots, n\}$ (normal vector of the hyperplane lives in the subspace spanned by the datapoints).

Beyond Linear Classifiers





- No linear classifier separates red from blue.
- Linear separation after mapping to a **higher dimensional feature space**:

$$\mathbb{R}^2 \ni \begin{pmatrix} x^{(1)} & x^{(2)} \end{pmatrix}^\top = x \;\mapsto\; \varphi(x) = \begin{pmatrix} x^{(1)} & x^{(2)} & x^{(1)}x^{(2)} \end{pmatrix}^\top \in \mathbb{R}^3$$

Non-Linear SVM

- Consider the dual C-SVM with explicit non-linear transformation $x \mapsto \varphi(x)$:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \varphi(x_i)^\top \varphi(x_j) \qquad \text{subject to} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \preceq \alpha \preceq C \end{cases}$$

- Suppose $p = 2$, and we would like to introduce quadratic non-linearities,

$$\varphi(x) = \left(1, \sqrt{2}x^{(1)}, \sqrt{2}x^{(2)}, \sqrt{2}x^{(1)}x^{(2)}, \left(x^{(1)}\right)^2, \left(x^{(2)}\right)^2\right)^\top .$$

 Then

$$\begin{aligned} \varphi(x_i)^\top \varphi(x_j) = 1 &+ 2x_i^{(1)}x_j^{(1)} + 2x_i^{(2)}x_j^{(2)} + 2x_i^{(1)}x_i^{(2)}x_j^{(1)}x_j^{(2)} \\ &+ \left(x_i^{(1)}\right)^2 \left(x_j^{(1)}\right)^2 + \left(x_i^{(2)}\right)^2 \left(x_j^{(2)}\right)^2 = (1 + x_i^\top x_j)^2 \end{aligned}$$

- Since only inner products are needed, non-linear transform need not be computed explicitly - inner product between features can be a simple function (**kernel**) of x_i and x_j: $k(x_i, x_j) = \varphi(x_i)^\top \varphi(x_j) = (1 + x_i^\top x_j)^2$
- d-order interactions can be implemented by $k(x_i, x_j) = (1 + x_i^\top x_j)^d$ (**polynomial kernel**). Never need to compute explicit feature expansion of dimension $\binom{p+d}{d}$ where this inner product happens!

Kernel SVM: Kernel trick

- Kernel SVM with $k(x_i, x_j)$. Non-linear transformation $x \mapsto \varphi(x)$ still present, but **implicit** (coordinates of the vector $\varphi(x)$ are never computed).

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad \text{subject to} \quad \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \preceq \alpha \preceq C \end{cases}$$

- Prediction? $\hat{y}(x) = \operatorname{sign}\left(w^\top \varphi(x) + b\right)$, where $w = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i)$ and offset b obtained from a margin support vector x_j with $\alpha_j \in (0, C)$.
  - No need to compute w either! Just need

$$w^\top \varphi(x) = \sum_{i=1}^{n} \alpha_i y_i \varphi(x_i)^\top \varphi(x) = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x).$$

  - Get offset from

$$b = y_j - w^\top \varphi(x_j) = y_j - \sum_{i=1}^{n} \alpha_i y_i k(x_i, x_j)$$

  for any margin support-vector x_j ($\alpha_j \in (0, C)$).

- Fitted a separating hyperplane in a high-dimensional feature space without ever mapping explicitly to that space.