Statistical learning, high dimension and big data

Stéphane Gaïffas

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

Today

- Again binary classification
- The linear SVM
- Construction of the hinge loss
- Kernels methods

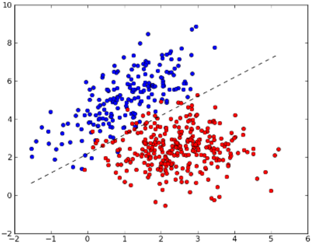
Setting

- Binary classification problem
- We observe a training dataset D of pairs (x_i, y_i) for $i = 1, \ldots, n$
- Features $x_i \in \mathbb{R}^d$ and labels $y_i \in \{-1, 1\}$
- Aim is to learn a classification rule that **generalizes** well
- Given a features vector $x \in \mathbb{R}^d$, we want to predict the label y
- Without **overfitting**

Linear classification. Why?

- Let's start simple!
- On very large datasets (n is large, say $n \geq 10^7$), no other choice (training complexity)
- Big data paradigm: lots of data $\Rightarrow$ simple methods are enough

A linear classifier



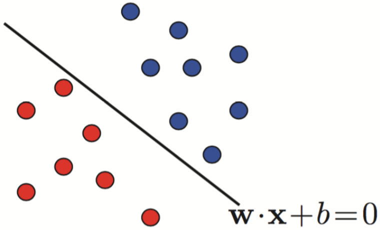
Learn $\hat{w} \in \mathbb{R}^d$ and $\hat{b}$ such that

$$\hat{y} = \mathrm{sign}(\langle x, \hat{w} \rangle + \hat{b})$$

is a good classifier

A dataset is **linearly separable** if we can find an hyperplane H that puts

- Points $x_i \in \mathbb{R}^d$ such that $y_i = 1$ on one side of the hyperplane
- Points $x_i \in \mathbb{R}^d$ such that $y_i = -1$ on the other
- H do not pass through a point x_i



An hyperplane

$$H = \{x \in \mathbb{R}^d : \langle w, x \rangle + b = 0\}$$

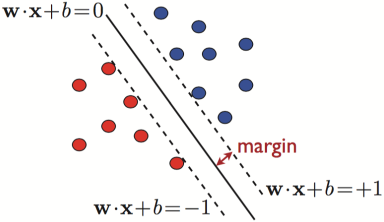
is a translation of a set of vectors orthogonal to w

- $w \in \mathbb{R}^d$ is a non-zero vector normal to the hyperplane
- $b \in \mathbb{R}$ is a scalar

Definition of H is invariant by multiplication of w and b by a non-zero scalar

If H do not pass through any sample point x_i, we can scale w and b so that

$$\min_{(x,y)\in D} |\langle w, x\rangle + b| = 1$$



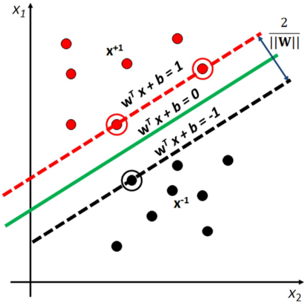
For such w and b, we call H the *canonical* hyperplane

The distance of any point $x' \in \mathbb{R}^d$ to H is given by

$$\frac{|\langle w, x' \rangle + b|}{\|w\|}$$

So, if H is a canonical hyperplane, its **margin** is given by

$$\min_{(x,y)\in D} \frac{|\langle w, x \rangle + b|}{\|w\|} = \frac{1}{\|w\|}.$$



In summary: if D is strictly linearly separable, we can find a canonical separating hyperplane

$$H = \{x \in \mathbb{R}^d : \langle w, x \rangle + b = 0\}.$$

that satisfies

$$|\langle w, x_i \rangle + b| \geq 1 \ \text{ for any } \ i = 1, \ldots, n,$$

which entails that a point x_i is correctly classified if

$$y_i(\langle x_i, w \rangle + b) \geq 1.$$

The margin of H is equal to $1/\|w\|$.

Linear SVM: separable case

From that, we deduce that a way of classifying D with maximum margin is to solve the following problem:

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2}\|w\|_2^2$$
$$\text{subject to} \quad y_i(\langle x_i, w\rangle + b) \geq 1 \; \text{ for all } \; i = 1, \dots, n$$

Note that:

- This problem admits a **unique** solution
- It is a "quadratic programming" problem, which is easy to solve numerically
- Dedicated optimization algorithms can solve this on a large scale very efficiently

Some tools from **constrained optimization**

- Consider a constrained optimization problem

$$\min_{x \in \mathbb{R}^d} \quad f(x)$$
$$\text{subject to} \quad g_i(x) \leq 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

  where $f, g_1, \ldots, g_n : \mathbb{R}^d \to \mathbb{R}$

- We denote $P^* = f(x^*)$ the minimum of this objective (minimum of the **primal**)
- The associated **Lagrangian** is the function given on $\mathbb{R}^d \times \mathbb{R}^n_+$ by

$$L(x, \alpha) = f(x) + \sum_{i=1}^n \alpha_i g_i(x)$$

  for **Lagrange** or **dual** variables $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n_+$

- The **Lagrange dual** function is defined by

$$D(\alpha) = \inf_{x \in \mathbb{R}^d} L(x, \alpha) = \inf_{x \in \mathbb{R}^d} \Big(f(x) + \sum_{i=1}^n \alpha_i g_i(x) \Big)$$

  for $\alpha \in \mathbb{R}_+^n$
- D is always concave, as the infimum of linear functions
- We denote $D^* = D(\alpha^*) = \max_{\alpha \geq 0} D(\alpha)$ the optimal value of the dual. It is a convex problem (maximum of a concave function)
- For any **feasible** x and any $\alpha \geq 0$ we have $D(\alpha) \leq f(x)$, hence

$$D^* \leq P^*$$

  This is called the **weak duality** inequality and always holds
- Something that does not always holds is **strong duality**:

$$D^* = P^*$$

Strong duality holds under **constraint qualifications** (sufficient but not necessary)

Probably the best known one is **strong duality**:

- The primal problem is **convex**: $f, g_1, \ldots, g_n$ are convex
- **Slater**'s condition holds: there is some strictly feasible point $x \in \mathbb{R}^d$ such that

$$g_i(x) < 0 \quad \text{for all } i = 1, \ldots, n$$

- **Slater**'s condition is obvious for **affine** functions: inequality no longer strict, reduces to the original constraint $g_i(x) \leq 0$

Now, a fundamental tool: **KKT theorem** (Karush-Kuhn-Tucker)

- Assume that $f, g_1, \ldots, g_n$ are **differentiable**, assume **strong duality**.
- Then, $x^* \in \mathbb{R}^d$ is a solution of the primal problem if and only if there is $\alpha^* \in \mathbb{R}_+^n$ such that

$$\begin{aligned} \nabla_x L(x^*, \alpha^*) = \nabla f(x^*) + \sum_{i=1}^{n} \alpha_i^* \nabla g_i(x^*) = 0 \\ g_i(x^*) \leq 0 \quad & \text{for any } i = 1, \ldots, n \\ \alpha_i^* g_i(x^*) = 0 \quad & \text{for any } i = 1, \ldots, n \end{aligned}$$

- These are known as the KKT conditions
- The last one is called **complementary slackness**

In summary: if

- primal problem is **convex** and
- constraint functions satisfy the **Slater**'s conditions

then

- **strong duality** holds.

If in addition we have that

- functions $f, g_1, \ldots, g_n$ are **differentiable**

then

- KKT conditions are **necessary and sufficient** for optimality

Back to the Linear SVM. The problem has the form

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \; f(w)$$
$$\text{subject to} \quad g_i(w, b) \le 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

where

- $f(w) = \frac{1}{2}\|w\|_2^2$ is **strongly convex**, since

$$\nabla^2 f(w) = \boldsymbol{I}_d \succ 0$$

- Constraints are $g_i(w, b) \le 0$ with **affine** functions

$$g_i(w, b) = 1 - y_i(\langle x_i, w \rangle + b)$$

  so that the constraints are **qualified**

We can apply the KKT theorem

Use this theorem to obtain a condition at the optimum

- It will lead to crucial properties on the SVM
- Allow to obtain the dual formulation of the problem

Lagragian

- Introduce dual variables $\alpha_i \geq 0$ for $i = 1, \ldots, n$ corresponding to the constraints $g_i(w, b) \leq 0$
- For $w \in \mathbb{R}^d$, $b \in \mathbb{R}$ and $\alpha = (\alpha_1, \ldots \alpha_n) \in \mathbb{R}_+^n$, introduce the Lagrangian

$$L(w, b, \alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{n} \alpha_i \big(1 - y_i(\langle w, x_i \rangle + b)\big)$$

$$L(w, b, \alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{n} \alpha_i \big(1 - y_i(\langle w, x_i \rangle + b)\big)$$

KKT conditions

Set the gradient to zero

$$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \quad \text{namely} \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\nabla_b L(w, b, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{namely} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Write the complementary slackness condition

$$\alpha_i \big(1 - y_i(\langle w, x_i \rangle + b)\big) = 0 \quad \text{namely} \quad \alpha_i = 0 \;\text{ or }\; y_i(\langle w, x_i \rangle + b) = 1$$

for all $i = 1, \ldots, n$

This entails the following properties **at the optimum**

- There are **dual** variables $\alpha_i \geq 0$ such that the **primal** solution (w, b) satisfies
$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$
- We have that
$$\alpha_i \neq 0 \quad \text{iff} \quad y_i(\langle w, x_i \rangle + b) = 1$$

This means that

- w writes as a linear combination of the features vectors x_i that belong to the marginal hyperplanes $\{x \in \mathbb{R}^d : \langle w, x \rangle + b = \pm 1\}$
- These vectors x_i are called **support vectors**

The support vectors fully define the maximum-margin hyperplane, hence the name **Support Vector Machine**

Dual optimization problem

Lagrangian is

$$L(w, b, \alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{n} \alpha_i \big(1 - y_i(\langle w, x_i\rangle + b)\big)$$

Plug $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ in it to obtain

$$L(w, b, \alpha) = \frac{1}{2}\Big\| \sum_{i=1}^{n} \alpha_i y_i x_i \Big\|_2^2 + \sum_{i=1}^{n} \alpha_i - b \sum_{i=1}^{n} \alpha_i y_i \\ - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

Recalling that $\sum_{i=1}^{n} \alpha_i y_i = 0$ and doing some algebra we arrive at the dual formulation

$$\max_{\alpha \in \mathbb{R}^n} \qquad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad \alpha_i \geq 0 \;\; \text{and} \;\; \sum_{i=1}^{n} \alpha_i y_i = 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

Remarks

- As in the primal formulation, it is again a quadratic programming problem
- At optimum, we have (using KKT conditions) that the decision function is expressed using the dual variables as

$$x \mapsto \operatorname{sgn}(\langle w, x\rangle + b) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \langle x, x_i\rangle + b\right)$$

- The intercept b can be expressed for any support vector x_i as

$$b = y_i - \sum_{j=1}^{n} \alpha_j y_j \langle x_i, x_j\rangle$$

This allows to write the margin as a function of the dual variables

- Multiplying the last equality by $\alpha_i y_i$ and summing entails

$$\sum_{i=1}^{n} \alpha_i y_i b = \sum_{i=1}^{n} \alpha_i y_i^2 - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

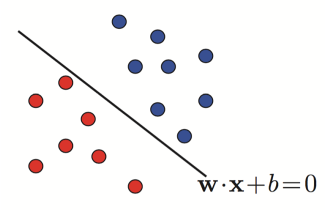
- Namely recalling that at optimum $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ we get

$$0 = \sum_{i=1}^{n} \alpha_i = \|w\|_2^2, \quad \text{namely}$$

$$\text{margin} = \frac{1}{\|w\|_2^2} = \frac{1}{\sum_{i=1}^{n} \alpha_i} = \frac{1}{\|\alpha\|_1}$$

- Okay, this is a nice theory, but...

Have you ever seen a dataset that looks that this?



Datasets are **not** linearly separable!

Keep cool and **relax** !

Replace the constraints

$$y_i(\langle w, x_i \rangle + b) \geq 1 \quad \text{for all} \quad i = 1, \ldots, n,$$

that are too strong, by the **relaxed** ones

$$y_i(\langle w, x_i \rangle + b) \geq 1 - s_i \quad \text{for all} \quad i = 1, \ldots, n,$$

for **slack variables** $s_1, \ldots, s_n \geq 0$

Slack rope

Linear SVM: non-separable case

Relax, but keep the slacks s_i as small as possible (goodness-of-fit)

Replace the original problem

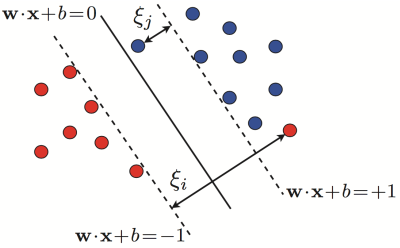
$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2}\|w\|_2^2$$
$$\text{subject to} \quad y_i(\langle x_i, w \rangle + b) \geq 1 \;\; \text{for all} \;\; i = 1, \ldots, n$$

by the relaxed one using slack variables:

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}, s \in \mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C \sum_{i=1}^n s_i$$
$$\text{subject to} \quad y_i(\langle x_i, w \rangle + b) \geq 1 - s_i \;\; \text{and} \;\; s_i \geq 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

where $C > 0$ is the "goodness-of-fit strength"

- The slack $s_i \geq 0$ measures the the distance by which x_i violates the desired inequality $y_i(\langle x_i, w\rangle + b) \geq 1$
- A vector x_i with $0 < y_i(\langle x_i, w\rangle + b) < 1$ is correctly classified but is an outlier, since $s_i > 0$
- If we omit outliers, training data is correctly classified by the hyperplane $\{x \in \mathbb{R}^d : \langle x, w\rangle + b = 0\}$ with a margin $1/\|w\|_2^2$
- The margin $1/\|w\|_2^2$ is called a **soft-margin** (in the non-separable case), while it is a **hard-margin** in the separable case



Linear SVM: non-separable case

So, we arrived at:

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, s\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} s_i$$

$$\text{subject to} \quad y_i(\langle x_i, w\rangle + b) \geq 1 - s_i \;\text{ and }\; s_i \geq 0 \;\text{ for all }\; i = 1, \ldots, n$$

Once again:

- This problem admits a **unique** solution
- It is a quadratic programming problem

The constant $C > 0$ is chosen using V-fold cross-valiation

Lagrangian

$$L(w, b, s, \alpha, \beta) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} s_i$$
$$+ \sum_{i=1}^{n} \alpha_i \big(1 - s_i - y_i(\langle w, x_i \rangle + b)\big) - \sum_{i=1}^{n} \beta_i s_i$$

At optimum, let's again:

- set the gradients ∇_w, ∇_b and ∇_s to zero
- write the complementary conditions

$$\nabla_w L(w,b,s,\alpha,\beta) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \quad \text{i.e.} \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\nabla_b L(w,b,s,\alpha,\beta) = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{i.e.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\nabla_s L(w,b,s,\alpha,\beta) = C - \alpha_i - \beta_i = 0 \quad \text{i.e.} \quad \alpha_i + \beta_i = C$$

and the complementary condition

$$\alpha_i\big(1-s_i-y_i(\langle w,x_i\rangle+b)\big) = 0 \text{ i.e. } \alpha_i = 0 \;\text{ or }\; y_i(\langle w,x_i\rangle+b) = 1-s_i$$

$$\beta_i s_i = 0 \quad \text{i.e.} \quad \beta_i = 0 \;\text{ or }\; s_i = 0$$

for all $i = 1, \ldots, n$

This means that

- $w = \sum_{i=1}^{n} \alpha_i y_i x_i$
- If $\alpha_i \neq 0$ we say that x_i is a support vector and in this case $y_i(\langle w, x_i \rangle + b) = 1 - s_i$
    - If $s_i = 0$ then x_i belongs to a margin hyperplane
    - If $s_i \neq 0$ then x_i is an outlier and $\beta_i = 0$ and then $\alpha_i = C$

Support vectors either belong to a marginal hyperplane, or are outliers with $\alpha_i = C$

Dual problem

- Plugging $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ in $L(w, b, s, \alpha, \beta)$ leads to the same formula as before

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

- with the constraints

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

that can be rewritten for as

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

for all $i = 1, \ldots, n$

Leading to the following **dual problem**

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C \ \text{ and } \ \sum_{i=1}^{n} \alpha_i y_i = 0 \ \text{ for all } \ i = 1, \ldots, n$$

- This is the same problem as before, but with the extra constraint

$$\alpha_i \leq C$$

- It is again a convex quadratic program

As in the linearly separable case, the label prediction is expressed using the dual variables as

$$x \mapsto \operatorname{sgn}\left(\langle w, x\rangle + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \langle x, x_i\rangle + b\right)$$

The intercept b can be expressed for a support vector x_i such that $0 < \alpha_i < C$ as

$$b = y_i - \sum_{j=1}^{n} \alpha_j y_j \langle x_i, x_j\rangle$$

A very important remark

The dual problem

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C \;\text{ and }\; \sum_{i=1}^n \alpha_i y_i = 0 \;\text{ for all }\; i = 1, \ldots, n$$

and the label prediction (using dual variables)

$$x \mapsto \operatorname{sgn}(\langle w, x \rangle + b) = \operatorname{sgn}\left(\sum_{i=1}^n \alpha_i y_i \langle x, x_i \rangle + b \right)$$

depends only on the features x_i via their **inner products** $\langle x_i, x_j \rangle$!

- This will be particularly important next week: **kernel methods**

The hinge loss

Going back to the primal problem

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, s\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} s_i$$

$$\text{subject to} \quad y_i(\langle x_i, w\rangle + b) \geq 1 - s_i \ \text{ and } \ s_i \geq 0 \ \text{ for all } \ i = 1, \dots, n$$

We remark that it can be rewritten as

$$\operatorname*{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} \max\Big(0, 1 - y_i(\langle x_i, w\rangle + b)\Big).$$

Introducing the **hinge loss**

$$\ell(y, y') = \max(0, 1 - yy') = (1 - yy')_+,$$

the problem can be written as

$$\operatorname*{argmin}_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2}\|w\|_2^2 + C \sum_{i=1}^{n} \ell(y_i, \langle x_i, w \rangle + b).$$

Leads to an alternative understanding of the linear SVM.

Recalling that the 0/1 loss given by

$$\ell_{0/1}(y, z) = \mathbf{1}_{yz \leq 0},$$

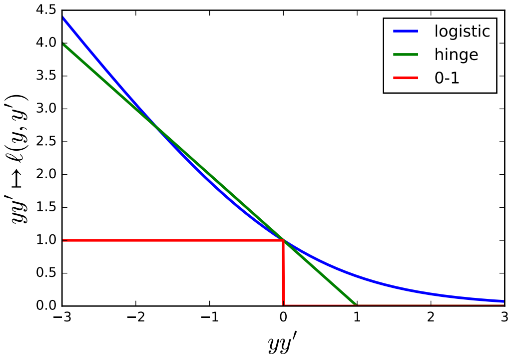
we can understand the linear SVM, as an approximation of the problem

$$\operatorname*{argmin}_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2}\|w\|_2^2 + C \sum_{i=1}^{n} \mathbf{1}_{y_i(\langle x_i, w\rangle + b) \leq 0},$$

which is impossible numerically (NP-hard)

Hinge loss is a **convex surrogate** for the 0/1 loss

The losses we've seen so far for classification



$$\ell_{0-1}(y, y') = \mathbf{1}_{yy' \leq 0} \quad \ell_{\text{hinge}}(y, y') = (1 - yy')_+$$
$$\ell_{\texttt{logistic}}(y, y') = \log(1 + e^{-yy'}).$$

Grandmother's recipe:

Grandmother's recipes for logistic regression vs linear SVM

Logistic regression

- Logistic regression has a nice probabilistic interpretation
- Relies on the choice of the logit link function

SVM

- No model, only aims at separating points

No one is not better than the other in general. Depends on the data.

Once again, what is always important though is the **construction of the features** you'll use for training

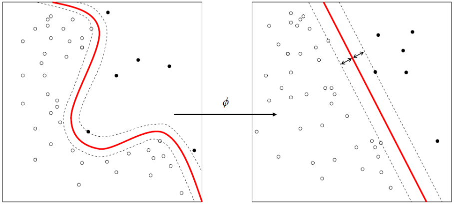
Features engineering and kernel methods

- Given raw features $x_1, \ldots, x_n \in \mathbb{R}^d$, we can construct **new** features
- For instance, we can add second order polynomials of the features
$$x_j^2, x_j x_k \quad \text{for any} \quad 1 \leq j, k \leq d$$
- It increases the number of features, hence the dimension of the model weights w learned from it

A **feature map**

- Consider a feature map $\varphi : \mathbb{R}^d \to \boldsymbol{H}$ that adds all these new features
- $\boldsymbol{H}$ is an Hilbert space (eventually infinite dimensional), endowed with an inner product $\langle \cdot, \cdot \rangle_{\boldsymbol{H}}$
- The decision boundary $x \to \langle w, \varphi(x) \rangle + b = 0$ is **not an hyperplane anymore** (but $\varphi(x) \to \langle w, \varphi(x) \rangle + b = 0$ is)

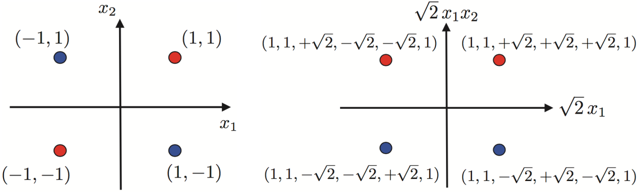
A common belief: **increasing dimension** of features space makes data **almost linearly separable**



The **polynomial** mapping $\varphi : \mathbb{R}^2 \to \mathbb{R}^6$ for $x = (x_1, x_2) \in \mathbb{R}^2$

$$\varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

solves the XOR (Exclusive OR) classification problem



XOR : label y_i is blue iff one of the coordinates of x_i equals 1.

- Blue and red points **cannot be linearly separated** in $\mathbb{R}^2$
- But **they can using the mapping** φ, using the hyperplane $x_1x_2 = 0$

This mapping φ is call **polynomial mapping of order 2**.

Note that for $x, x' \in \mathbb{R}^2$ we have

$$\begin{aligned}
\langle \varphi(x), \varphi(x') \rangle &= \left\langle \begin{bmatrix} x_1^2 \\ x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ 1 \end{bmatrix}, \begin{bmatrix} x_1^2 \\ x_1'^2 \\ x_2'^2 \\ \sqrt{2}x_1' x_2' \\ \sqrt{2}x_1' \\ \sqrt{2}x_2' \\ 1 \end{bmatrix} \right\rangle \\
&= (x_1 x_1' + x_2 x_2' + 1)^2 \\
&= (\langle x, x' \rangle + 1)^2
\end{aligned}$$

This motivates the definition of

$$K(x, x') = \langle \varphi(x), \varphi(x') \rangle = (\langle x, x' \rangle + c)^q$$

where $q \in \mathbb{N} - \{0\}$ and $c > 0$. In this case K is called the polynomial **kernel** of degree q.

Given a "raw feature" space $\mathcal{X}$ (often $\mathcal{X} = \mathbb{R}^d$), a function

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

is called a **kernel** over $\mathcal{X}$.

Definition. We say that a kernel K is **symmetric** iff

$$K(x, x') = K(x', x)$$

for any $x, x' \in \mathcal{X}$

Definition. We say that a kernel is PDS (positive definite symetric) iff

- it is symmetric
- for any $N \in \mathbb{N}$ and any $\{x_1, \dots x_N\} \subset \mathcal{X}$ we have

  $$\boldsymbol{K} = [K(x_i, x_j)]_{1 \leq i,j \leq N} \succeq 0$$

  meaning that $\boldsymbol{K}$ is positive semi-definite (symmetric), or equivalently that

  $$u^\top \boldsymbol{K} u = \sum_{1 \leq i,j \leq N} u_i u_j K(x_i, x_j) \geq 0$$

  for any $u \in \mathbb{R}^N$, or equivalently that all eigenvalues of $\boldsymbol{K}$ are non-negative.

For a sample $x_1, \dots, x_n$ we call $\boldsymbol{K} = [K(x_i, x_j)]_{1 \leq i,j \leq n}$ the **Gram matrix** of this sample.

Definition. Hadamard product $\boldsymbol{A} \odot \boldsymbol{B}$ between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ (or vectors) with the same dimensions is given by

$$(\boldsymbol{A} \odot \boldsymbol{B})_{i,j} = \boldsymbol{A}_{i,j} \odot \boldsymbol{B}_{i,j}$$

Theorem. The sum, product, pointwise limit and composition with a power series $\sum_{n \geq 0} a_n x^n$ with $a_n \geq 0$ for all $n \geq 0$ preserves the PDS property.

Proof. Consider two $N \times N$ Gram matrices $\boldsymbol{K}, \boldsymbol{K}'$ of PDS kernels K, K' and take $u \in \mathbb{R}^N$. Observe that

$$u^\top (\boldsymbol{K} + \boldsymbol{K}')u = u^\top \boldsymbol{K} u + u^\top \boldsymbol{K}' u \geq 0$$

So PDS is preserved by the sum and finite sums by reccurence.

Now, to prove that the product $\boldsymbol{K} \odot \boldsymbol{K}'$ is PDS, write $\boldsymbol{K} = \boldsymbol{M}\boldsymbol{M}^\top$, where $\boldsymbol{M}$ is the square-root of $\boldsymbol{K}$ (which is SDP) and note that

$$
\begin{aligned}
u^\top(\boldsymbol{K} \odot \boldsymbol{K}')u &= \sum_{1 \le i,j \le N} u_i u_j \boldsymbol{K}_{i,j} \boldsymbol{K}'_{i,j} = \sum_{1 \le i,j \le N} \sum_{k=1}^{N} u_i u_j \boldsymbol{M}_{i,k} \boldsymbol{M}_{k,j} \boldsymbol{K}'_{i,j} \\
&= \sum_{k=1}^{N} z_k^\top \boldsymbol{K}' z_k \ge 0
\end{aligned}
$$

with $z_k = u \odot \boldsymbol{M}_{\bullet,k}$.

This proves that finite products of PDS kernels is PDS.

Assume that $K_n \to K$ as $n \to +\infty$ pointwise, where K_n is a sequence of PDS kernels.
It means that any associated sequence of Gram matrices $\boldsymbol{K}_n$ and the its limit $\boldsymbol{K}$ satisfies $\boldsymbol{K}_n \to \boldsymbol{K}$ entrywise, so that for any $u \in \mathbb{R}^N$ we have

$$u^\top \boldsymbol{K}_n u \to u^\top \boldsymbol{K} u$$

so $u^\top \boldsymbol{K} u \geq 0$ since $u^\top \boldsymbol{K}_n u \to u$ for all n.

This proves stability of PDS property under pointwise limit.

Now, let K be a kernel such that $|K(x,x')| < r$ for all $x, x' \in \mathcal{X}$ and $\sum_{n\geq 0} a_n x^n$ a power series with radius of convergence r.

By stability under sum and product, we have that

$$\sum_{k=0}^{N} a_n K^n$$

is PDS, and

$$\lim_{N\to+\infty} \sum_{n=0}^{N} a_n K^n = \sum_{n\geq 0} a_n K^n$$

remains PDS since PDS is kept under pointwise limit.

This concludes the proof of the theorem.

Theorem. The following inequality holds for K, K' two PDS kernels

$$K(x, x')^2 \leq K(x, x)K(x', x')$$

for any $x, x' \in \mathcal{X}$. It is called the **Cauchy-Schwartz inequality** for PSD kernels.

Proof. Take $x, x' \in \mathcal{X}$ and consider the Gram matrix

$$\boldsymbol{K} = \begin{bmatrix} K(x, x) & K(x, x') \\ K(x', x) & K(x', x') \end{bmatrix}.$$

Since K is PDS, then $\boldsymbol{K} \succeq 0$, which entails that

$$0 \leq \det \boldsymbol{K} = K(x, x)K(x', x') - K(x, x')^2$$

Theorem [Reproducing kernel Hilbert space]. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PDS kernel. Then, there is a Hilbert space $\boldsymbol{H} \subset \mathbb{R}^{\mathcal{X}}$ endowed with an inner product $\langle \cdot, \cdot \rangle$ and a mapping $\varphi : \mathcal{X} \to \boldsymbol{H}$ such that

$$K(x, x') = \langle \varphi(x), \varphi(x') \rangle \tag{1}$$

and such that the **reproducing property** holds:

$$h(x) = \langle h, K(x, \cdot) \rangle$$

for any $h \in \boldsymbol{H}$ and $x \in \mathcal{X}$.

Remarks.

- $K(x, \cdot)$ means $x' \mapsto K(x, x')$
- $\boldsymbol{H}$ is a space containing functions $\mathcal{X} \to \mathbb{R}$
- (1) stresses the fact that a PDS kernel is some kind of similarity measure, since it is actually an inner product

- We say that $\boldsymbol{H}$ is a **reproducing kernel Hilbert space** associated to the kernel K.
- The Hilbert space $\boldsymbol{H}$ is called the **features space** associated to K
- The corresponding mapping $\varphi : \mathcal{X} \to \boldsymbol{H}$ is called the **features mapping**
- $\boldsymbol{H}$ is endowed with an inner product $\langle h, h' \rangle$ for $h, h' \in \boldsymbol{H}$ and a norm $\|h\| = \sqrt{\langle h, h \rangle}$
- The feature space might is not unique in general

The space $\boldsymbol{H}$ is constructed as the completion of the subspace containing elements of the form

$$h(x) = \sum_{i=1}^{q} a_i K(x_i, x)$$

for $x_1, \ldots, x_N \in \mathcal{X}$ and $a_1, \ldots, a_N \in \mathbb{R}$

In summary

- Choose a kernel K you think relevant, if it's PDS, then there is a mapping φ and a RKHS $\boldsymbol{H}$ for it
- Feature engineernig becomes kernel engineering with kernel methods

Definition. The **normalized kernel** K' associated to a kernel K is given by

$$K'(x, x') = \frac{K(x, x')}{\sqrt{K(x, x)K(x', x')}}$$

if $K(x, x)K(x', x') > 0$ and $K(x, x') = 0$ otherwise.

Theorem. If K is a PDS kernel, its normalized kernel K' is PDS.

Remark. We have that $K(x, x')$ is the cosine of the angle between $\varphi(x)$ and $\varphi(x')$ if K is a normalized kernel (if none is zero). Once again, $K(x, x')$ is a similarity measure between x and x'

Proof. Let $x_1, \dots, x_N \in \mathcal{X}$ and $c \in \mathbb{R}^N$. If $K(x_i, x_i) = 0$ or $K(x_j, x_j) = 0$ then $K(x_i, x_j) = 0$ using Cauchy-Schwartz, so $K'(x_i, x_j) = 0$.

So, we can assume $K(x_i, x_i) > 0$ for all $i = 1, \dots, N$ and write the following:

$$\sum_{1 \leq i,j \leq N} \frac{c_i c_j K(x_i, x_j)}{\sqrt{K(x_i, x_i) K(x_j, x_j)}} = \sum_{1 \leq i,j \leq N} \frac{c_i c_j \langle \varphi(x_i), \varphi(x_j) \rangle}{\|\varphi(x_i)\| \|\varphi(x_j)\|}$$
$$= \left\| \sum_{i=1}^{N} \frac{c_i \varphi(x_i)}{\|\varphi(x_i)\|} \right\| \geq 0$$

which proves the theorem.

Remark. If K is a normalized kernel, then

$$\|\varphi(x)\| = \langle \varphi(x), \varphi(x) \rangle = K(x, x) = 1$$

for any $x \in \mathcal{X}$

The polynomial kernel. For $c > 0$ and $q \in \mathbb{N} - \{0\}$ we define the polynomial kernel

$$K(x, x') = (\langle x, x' \rangle + c)^q.$$

It is a PDS kernel

Proof. It is the power of the PDS kernel $(x, x') \mapsto \langle x, x' \rangle + b$.

We already computed its mapping $\varphi(x)$: it contains all the monomials of degree less than q of the coordinates of x

The RBF kernel (Radial Basis Function). For $\gamma > 0$ it is given by

$$K(x, x') = \exp(-\gamma \|x - x'\|_2^2)$$

Theorem. The RBF kernel is a PDS and normalized kernel.

Proof. First remark that

$$\exp(-\gamma \|x - x'\|_2^2) = \frac{\exp(2\gamma \langle x, x' \rangle)}{\exp(\gamma \|x\|^2) \exp(\gamma \|x'\|^2)} = \frac{K'(x, x')}{\sqrt{K'(x, x) K'(x', x')}}$$

with $K'(x, x') = \exp(2\gamma \langle x, x' \rangle)$ and that K' is PDS since

$$K'(x, x') = \sum_{n \geq 0} \frac{(2\gamma \langle x, x' \rangle)^n}{n!}$$

namely a series of the PDS kernel $(x, x') \mapsto 2\gamma \langle x, x' \rangle$.

The tanh kernel. Also called the sigmoid kernel

$$K'(x,x') = \tanh(a\langle x,x'\rangle + c) = \frac{e^{a\langle x,x'\rangle+c} - e^{a\langle x,x'\rangle+c}}{e^{a\langle x,x'\rangle+c} + e^{a\langle x,x'\rangle+c}}$$

for $a, c > 0$. It is again a PDS kernel (same argument as for the RBF kernel).

Remark. By far, the RBF kernel is the most widely used: uses as a similarity measure the Euclidean norm

Kernel based algorithms how to use kernels for classification and regression?

- Let's recall the primal and dual formulation of the SVM

Linear SVM. Primal problem is

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, s\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n s_i$$

$$\text{subject to} \quad y_i(\langle x_i, w\rangle + b) \geq 1 - s_i \;\text{ and }\; s_i \geq 0 \;\text{ for all }\; i = 1, \ldots, n$$

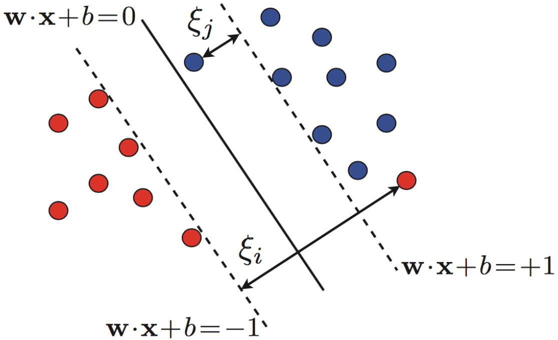
or equivalently

$$\operatorname*{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \ell(y_i, \langle x_i, w\rangle + b)$$

where $\ell(y, y') = \max(0, 1 - yy') = (1 - yy')_+$ is the hinge loss

Label prediction given by

$$y = \operatorname{sgn}\left(\langle x, w\rangle + b\right)$$


$\mathbf{w} \cdot \mathbf{x} + b = 0$
ξ_j
ξ_i
$\mathbf{w} \cdot \mathbf{x} + b = +1$
$\mathbf{w} \cdot \mathbf{x} + b = -1$

Kernel SVM: replace x_i by $\varphi(x_i)$. In the primal this leads to

$$\operatorname*{argmin}_{w \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{2}\|w\|_2^2 + C \sum_{i=1}^{n} \ell(y_i, \langle \varphi(x_i), w \rangle + b)$$

Label prediction is given by

$$y = \operatorname{sgn}\left(\langle \varphi(x), w \rangle + b\right)$$

In the primal, you need to compute $\varphi(x)$!

Dual problem is

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C \;\; \text{and} \;\; \sum_{i=1}^{n} \alpha_i y_i = 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

and the label prediction using dual variables

$$x \mapsto \operatorname{sgn}(\langle w, x \rangle + b) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \langle x, x_i \rangle + b \right)$$

depends only on the features x_i via their inner products $\langle x_i, x_j \rangle$

Fundamental remark. The dual problem depends only on the features via their inner products

Given some kernel K, let's replace the "raw" inner products $\langle x_i, x_j \rangle$ by the "new" inner products $K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$

The kernel trick. Once again, to train the SVM with a kernel, you don't need to know or compute the $\varphi(x_i)$

The kernel SVM

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C \;\; \text{and} \;\; \sum_{i=1}^{n} \alpha_i y_i = 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

and the label prediction using dual variables

$$x \mapsto \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b \right)$$

with the intercept given by

$$b = y_i - \sum_{j=1}^{n} \alpha_j y_j K(x_j, x_i)$$

for any i such that $0 < \alpha_i < C$ (cf previous lecture)

This proves that the hypothesis solution writes

$$h(x) = \operatorname{sgn}\Big(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b\Big),$$

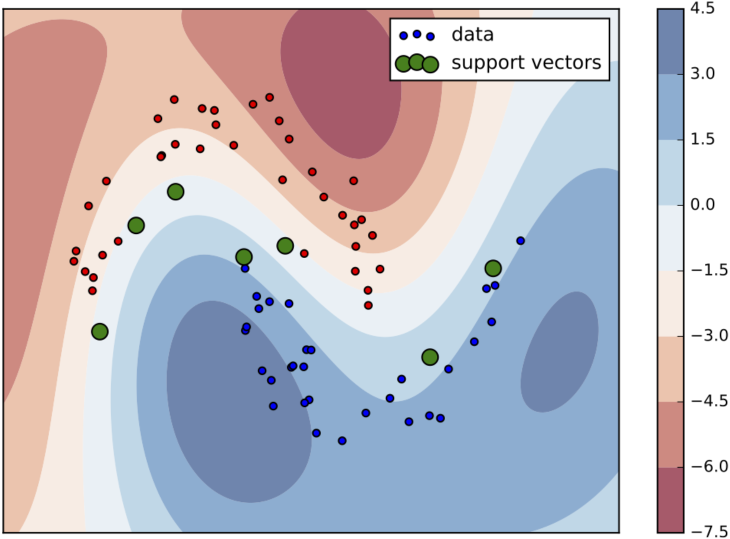
namely a combination of functions $K(x_i, \cdot)$ where x_i are the support vectors.

For the RBF kernel, the decision function is

$$x \mapsto \sum_{i:\alpha_i \neq 0} \alpha_i y_i \exp\big(-\gamma \|x - x_i\|_2^2\big) + b$$

It is a mixture of Gaussian "densities". Let's recall that the x_i with $\alpha_i \neq 0$ are the support vectors

$$x \mapsto \sum_{i:\alpha_i \neq 0} \alpha_i y_i \exp\left(-\gamma \|x - x_i\|_2^2\right) + b$$



The kernel trick is not only for the SVM

Representer theorem. If K is a PDS kernel and $\boldsymbol{H}$ its corresponding RKHS, we have that for any increasing function g and any function $L : \mathbb{R}^n \to \mathbb{R}$ that the optimization problem

$$\operatorname*{argmin}_{h \in \boldsymbol{H}} g(\|h\|) + L(h(x_1), \ldots, h(x_n))$$

admits only solutions of the form

$$h = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot).$$

Kernel ridge regression.

- Consider this time a continuous label $y_i \in \mathbb{R}$, features $x_i \in \mathcal{X}$ for $i = 1, \ldots, n$ and a features mapping $\varphi : \mathcal{X} \to \boldsymbol{H}$ with PDS kernel K
- Kernel ridge regression considers the problem

$$\underset{w}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{n} \ell(y_i, \langle w, \varphi(x_i) \rangle) + \frac{\lambda}{2} \|w\|_2^2 \Big\}$$

where λ is a penalization parameter, and $\ell(y, y') = \frac{1}{2}(y - y')^2$ is the least-squares loss

- Can be written as

$$\underset{w}{\operatorname{argmin}}\, F(x) \quad \text{with} \quad F(w) = \|y - \boldsymbol{X} w\|_2^2 + \lambda \|w\|_2^2$$

with $\boldsymbol{X}$ the matrix with rows containing the $\varphi(x_i)$ and $y = [y_1 \cdots y_n] \in \mathbb{R}^n$

- This problem is strongly convex, and admits a global minimum iff

$$\nabla F(w) = 0 \quad \text{namely} \quad (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})w = \boldsymbol{X}^\top y$$

- Note that $\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}$ is always invertible. Thus kernel ridge allows admits a closed-form solution
- Requires to solve a $D \times D$ linear system, where D is the dimension of $\boldsymbol{H}$
- What if D is large ?
- Let's us the kernel trick, as we did for SVM

- Representer theorem says that we can find α such that

$$h(x) = \langle w, \varphi(x) \rangle = \sum_{i=1}^{n} \alpha_i K(x_i, x) = \sum_{i=1}^{n} \alpha_i \langle \varphi(x_i), \varphi(x) \rangle$$

  for any $x \in \mathcal{X}$

- This means that

$$w = \boldsymbol{X}^\top \alpha$$

Now, use the following trick: for any matrix $\boldsymbol{X}$, we have

$$(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top = \boldsymbol{X}^\top (\boldsymbol{X} \boldsymbol{X}^\top + \lambda \boldsymbol{I})^{-1}$$

This entails

$$w = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top y = \boldsymbol{X}^\top (\boldsymbol{X} \boldsymbol{X}^\top + \lambda \boldsymbol{I})^{-1} y$$

which gives (note that $(\boldsymbol{X} \boldsymbol{X}^\top)_{i,j} = \langle \varphi(x_i), \varphi(x_j) \rangle = K(x_i, x_j)$)

$$\alpha = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} y$$

Proof of the trick. Note that

$$(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})\boldsymbol{X}^\top = \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}).$$

Multiplying on the left by $(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}$ leads to

$$\boldsymbol{X}^\top = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I}).$$

and then on the right by $(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I})^{-1}$ concludes with

$$(\boldsymbol{X}\boldsymbol{X}^\top + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^\top = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^\top$$

A cute trick. But let's do it like we did for the SVMs (just to be sure...)

An alternative formulation of

$$\min_{w} \sum_{i=1}^{n} (y_i - \langle w, \varphi(x_i) \rangle)^2 + \lambda \|w\|_2^2$$

is given by

$$\min_{w} \sum_{i=1}^{n} (y_i - \langle w, \varphi(x_i) \rangle)^2 \ \text{ subject to } \ \|w\|_2^2 \le r^2$$

and also

$$\min_{w} \sum_{i=1}^{n} s_i^2 \ \text{ subject to } \ \|w\|_2^2 \le r^2 \ \text{ and } \ s_i = y_i - \langle w, \varphi(x_i) \rangle$$

Which leads to the following Lagrangian

$$L(w, s, \alpha, \lambda) = \min_w \sum_{i=1}^n s_i^2 + \min_w \sum_{i=1}^n \alpha_i (y_i - s_i - \langle w, \varphi(x_i) \rangle) \\ + \lambda(\|w\|_2^2 - r^2)$$

so that the KKT conditions leads to the following properties:

$$\nabla_w L = -\sum_{i=1}^n \alpha_i \varphi(x_i) + 2\lambda w \Rightarrow w = \frac{1}{2\lambda} \sum_{i=1}^n \alpha_i \varphi(x_i)$$

$$\nabla_{s_i} L = 2s_i - \alpha_i \Rightarrow s_i = \alpha_i / 2$$

and the slackness complementary conditions:

$$\alpha_i (y_i - s_i - \langle w, \varphi(x_i) \rangle) = 0 \quad \text{and} \quad \lambda(\|w\|_2^2 - r^2) = 0$$

Plugging the expressions of w and s_i in functions of α in L gives after some algebra the dual objective

$$
\begin{aligned}
D(\alpha) = -\lambda \sum_{i=1}^{n} \alpha_i^2 + 2 \sum_{i=1}^{n} \alpha_i y_i \\
- \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j) \rangle - \lambda r^2
\end{aligned}
$$

(where we replaced $2\lambda\alpha_i$ by α_i) which can be written matricially as

$$
\begin{aligned}
D(\alpha) &= -\lambda \|\alpha\|_2^2 + 2\langle \alpha, y \rangle - \alpha^\top \boldsymbol{X} \boldsymbol{X}^\top \alpha \\
&= 2\langle \alpha, y \rangle - \alpha^\top (\boldsymbol{K} + \lambda \boldsymbol{I}) \alpha
\end{aligned}
$$

with optimum achieved for

$$
\alpha = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} y
$$

(same as before, of course...)

In summary

- Solving a problem in the dual benefits from the kernel trick
- Allows to construct complex non-linear decision functions
- OK if n is not too large... (if the $n \times n$ Gram matrix $\boldsymbol{K}$ fits in memory)
- Otherwise, stick to the primal!
- But don't forget about feature engineering (yes, again !)

Next week. We have seen a lot of problem of the form

$$\operatorname*{argmin}_w f(w) + g(w)$$

with f a goodness-of-fit function and g is a penalization

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \langle w, x_i \rangle) \quad g(w) = \frac{1}{C}\,\mathrm{pen}(w)$$

where ℓ is some loss and where pen is some penalization function, examples being $\mathrm{pen}(w) = \frac{1}{2}\|w\|_2^2$ (ridge) and $\mathrm{pen}(w) = \|w\|_1$ (Lasso)

Next week we'll learn how to solve this kind of problems using **optimization algorithms** (deterministic and stochastic)

Thank you!