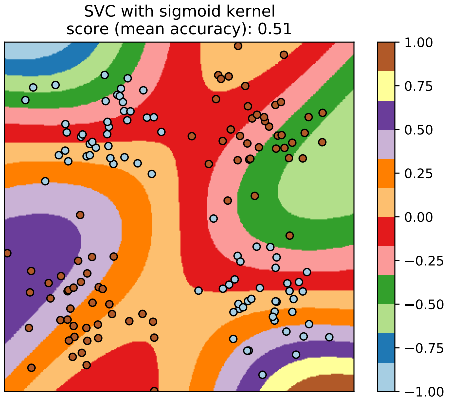
Fondamentaux théoriques du machine learning



Overview of lecture 5

Gradient descent

We want to minimize a function f defined over $\mathbb{R}^d$.

$$\theta \leftarrow \theta - \gamma \nabla_f(\theta) \tag{1}$$

Least-squares problem

In **lecture_notes.pdf**, general results about convex and strongly convex functions are summarized. To develop our intuition, we will study the specific case of ERM for OLS with GD.

- $X \in \mathbb{R}^{n,d}$
- $y \in \mathbb{R}^n$.

$$f(\theta) = \frac{1}{2n} ||X\theta - y||_2^2 \tag{2}$$

$$\theta \leftarrow \theta - \gamma \nabla_\theta f(\theta) \tag{3}$$

$$f(\theta) = \frac{1}{2n}||X\theta - y||_2^2 \tag{4}$$

The gradient and the Hessian write :

$$\nabla_\theta f(\theta) = \frac{1}{n}X^T(X\theta - y) \tag{5}$$

$$H = \frac{1}{n}X^T X \tag{6}$$

Minimizers

We note θ^* the minimizers of f. All minimizers verify that

$$\nabla_\theta f(\theta^*) = 0 \tag{7}$$

or

$$H\theta^* = \frac{1}{n} X^T y \tag{8}$$

If H is not invertible, they might be not unique, but all have the same function value $f(\theta^*)$.

Invertibility of H

$$H = \frac{1}{n} X^T X \tag{9}$$

- H is symmetric, positive semi-definite.
- H is invertible if and only if its smallest eigenvalue μ is > 0, in which case f is strongly convex.

We assume that $\mu > 0$. Let us study the convergence speed of GD towards θ^* (that exsits and is unique).

Minimizers

With a taylor expansion, we have that

$$f(\theta) - f(\theta^*) = \frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*) \tag{10}$$

Gradient update

Exercice 1 : We perform a gradient update with step size γ. t denotes the iteration number. Show that

$$\theta_t = \theta_{t-1} - \gamma H(\theta_{t-1} - \theta^*) \tag{11}$$

Gradient update

Exercice 2 : Deduce that :

$$\theta_t - \theta^* = (I - \gamma H)(\theta_{t-1} - \theta^*) \tag{12}$$

and that

$$\theta_t - \theta^* = (I - \gamma H)^t(\theta_0 - \theta^*) \tag{13}$$

Measure of performance

We can use two measures of performance of the gradient algorithm :

- Distance to minimizer :

$$||\theta_t - \theta^*||_2^2 = (\theta_0 - \theta^*)^T (I - \gamma H)^{2t} (\theta_0 - \theta^*) \qquad (14)$$

- Convergence in function values :

$$f(\theta_t) - f(\theta^*) = \frac{1}{2} (\theta_0 - \theta^*)^T (I - \gamma H)^{2t} H (\theta_0 - \theta^*) \qquad (15)$$

Distance to minimizer

If we can bound the eigenvalues of $(I - \gamma H)^{2t}$, we can bound $||\theta_t - \theta^*||_2^2$.
Exercice 3 : We note λ_i the eigenvalues of H. What are the eigenvalues of $(I - \gamma H)^{2t}$?

Bounding eigenvalues

We introduce the **condition number** $\kappa = \frac{L}{\mu}$ where L is the largest eigenvalue of H. By convention, if $\mu = 0$, $L = +\infty$.
All eigenvalues of $(I - \gamma H)^{2t}$ have a magnitude that is smaller than

$$\Big(\max_{\lambda \in [\mu, L]} |1 - \gamma\lambda| \Big)^{2t} \tag{16}$$

Hence, we want to find γ such that $\max_{\lambda \in [\mu, L]} |1 - \gamma\lambda|$ is **minimum** (or at least small).

Bounding eigenvalues

Exercice 4 : Find γ such that

$$\max_{\lambda \in [\mu, L]} |1 - \gamma\lambda| \leq (1 - \frac{\mu}{L}) = (1 - \frac{1}{\kappa}) \tag{17}$$

Exponential convergence

With $\gamma = \frac{1}{L}$, $\max_{\lambda \in [\mu, L]} |1 - \gamma \lambda| = 1 - \frac{1}{\kappa}$.
We obtain an **exponential convergence**

$$||\theta_t - \theta^*||_2^2 \leq \left(1 - \frac{1}{\kappa}\right)^{2t} ||\theta_0 - \theta^*||_2^2 \tag{18}$$

We have that

$$(1 - \frac{1}{\kappa})^{2t} \leq \exp(-\frac{1}{\kappa})^{2t} = \exp(-\frac{2t}{\kappa}) \quad (19)$$

Exercice 5 : What number of iterations is suffcient in order to have a relative reduction of $||\theta_t - \theta^*||_2^2$ of ϵ ?

$$
\begin{aligned}
&\exp\left(-\frac{2t}{\kappa}\right) \leq \epsilon \\
\Leftrightarrow\ & -\log(\epsilon) \leq \frac{2t}{\kappa} \\
\Leftrightarrow\ & \frac{\kappa}{2}\log\left(\frac{1}{\epsilon}\right) \leq t
\end{aligned}
\tag{20}
$$

Large condition number

If $\kappa = +\infty$ $(\mu = 0)$, we do not have a convergence guarantee.

Summary

- if H is invertible ($\mu > 0$), we have a convergence rate in $\exp(-\frac{2t}{\kappa})$.
- if H is not invertible ($\mu = 0$), we have a convergence rate in $\mathcal{O}(\frac{1}{t})$ (probably one of the exercises of the project).

Importantly, these results generalize to general convex / strongly convex functions, under some additional hypotheses (not necessary OLS) (see dedicated textbooks for reference).

Convergence for SGD

Due to the stochastic nature of SGD, the convergence results and proofs are more abstract / harder to derive. In the general case :

- Either we have results on expected values, such as that of $||\theta_t - \theta^*||$.
- Or convergence guarantees on **averages of the iterates.**

GD or SGD ?

In the case of a strongly convex least squares problem, let us admit that SGD has a convergence rate of $\mathcal{O}(\frac{\kappa}{t})$.
Exercice 6 : If we want an error of order of magnitude ϵ, should we choose GD or SGD ?

Comparison

The ridge regression problem is smooth and strongly convex.

- GD has a convergence rate of $\mathcal{O}(\exp(-\frac{t}{\kappa}))$. To get an error of ϵ, we must have $t = \mathcal{O}(\kappa \log \frac{1}{\epsilon})$. Since each iteration requires $\mathcal{O}(nd)$ computations, the computation time will be $\mathcal{O}(\kappa nd \log \frac{1}{\epsilon})$.
- SGD has a convergence rate of $\mathcal{O}(\frac{\kappa}{t})$. To get an error of ϵ, we must have $t = \mathcal{O}(\frac{\kappa}{\epsilon})$. Since each iteration is $\mathcal{O}(d)$, we have a computation time of $\mathcal{O}(\frac{\kappa d}{\epsilon})$.

Comparison

As a consequence :

- When n is large and ϵ not too small, GD will need more computation time to reach error ϵ. An order of magnitude can be obtained by studying the value ϵ^* such that

  $$\kappa n d \log \frac{1}{\epsilon^*} = \frac{\kappa d}{\epsilon^*}$$

  Which translates to

  $$\epsilon^* \log \epsilon^* = -\frac{1}{n}$$

- When $\epsilon \to 0$, GD becomes faster than SGD to reach this precision.

Conclusion

For low precision and large n, SGD is a preferable.
In machine learning, due to the estimation error that is $\mathcal{O}(\frac{1}{\sqrt{n}})$, a very high precision is often not needed.
Final remark : remember that bounds are just bounds. An algorithm will sometimes converge **faster** than the theoretical upper bound !

Introduction to Kernel methods

Replace inputs $x \in \mathcal{X}$ by a function $\phi(x) \in \mathcal{H}$, with $\mathcal{H}$ a $\mathbb{R}$-Hilbert space. We then perform linear predictions on $\phi(x)$. This means that estimators have the form :

$$f(x) = \langle \theta, \phi(x) \rangle_{\mathcal{H}} \tag{21}$$

Introduction to Kernel methods

Replace inputs $x \in \mathcal{X}$ by a function $\phi(x) \in \mathcal{H}$, with $\mathcal{H}$ a $\mathbb{R}$-Hilbert space. We then perform linear predictions on $\phi(x)$. This means that estimators have the form :

$$f(x) = \langle \theta, \phi(x) \rangle_{\mathcal{H}} \tag{22}$$

- $\langle ., . \rangle_{\mathcal{H}}$: inner product defined on $\mathcal{H}$. When there is no ambiguity, we will note $\langle ., . \rangle = \langle ., . \rangle_{\mathcal{H}}$.
- $\theta \in \mathcal{H}$
- $\phi(x)$: **feature** associated to x, $\mathcal{H}$: **feature space** .

Interest

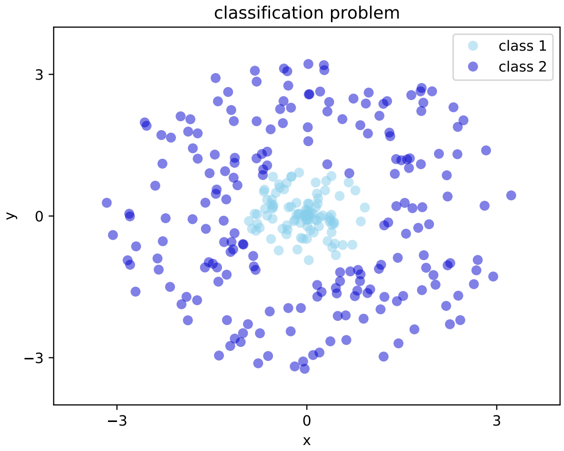
- Kernel methods provide stable algorithms, with theoretical convergence guarantees.
- They can benefit from the smoothness (regularity) of the target function, whereas local averaging methods cannot.
- They can be applied in high dimension.

In some supervised learning problems with many observations, such as computer vision and natural language processing, they are now outperformed by neural networks.

Feature maps

- $\mathcal{X}$ need not be a vector space.
- $\phi(x)$ can provide more useful **representation** of the input for the considered problem (classification, regression).
- The prediction function is then allowed to depend **non-linearly** on x.

Nonlinear data separation



Feature space

Often, $\mathcal{H} = \mathbb{R}^d$, but importantly, we will see that d can even be **infinite**, thanks to a computation trick called the **kernel trick**.

Representer theorem

We consider a framework where we look for a minimizer $\hat{\theta}$ of a loss such as

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \langle\theta, \phi(x_i)\rangle) + \lambda||\theta||^2 \tag{23}$$

The key aspect is that the input observations $x_i \in \mathcal{X}$ are only accessed through inner products with θ.

Representer theorem

Theorem

Let $\Psi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ be a strictly increasing function with respect to the last variable. Then, the minimum of

$$L(\theta) = \Psi(\langle\theta, \phi(x_i)\rangle, \dots, \langle\theta, \phi(x_n)\rangle, ||\theta||^2) \tag{24}$$

is attained for $\hat{\theta} \in Vect(\{\phi(x_i)\}$. We can write

$$\hat{\theta} = \sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n \tag{25}$$

Proof I

Let $\mathcal{H}_D = \{\sum_{i=1}^n \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n\}$. For all $\theta \in \mathcal{H}$, we have a decomposition

$$\theta = \theta_D + \theta_{D^\perp} \tag{26}$$

with $\theta_D \in \mathcal{H}_D$ and $\theta_{D^\perp} \in \mathcal{H}_D^\perp$.
Then, $\forall i \in \{1, \dots, n\}$,

$$\begin{aligned} \langle \theta, \phi(x_i) \rangle &= \langle \theta_D, \phi(x_i) \rangle + \langle \theta_{D^\perp}, \phi(x_i) \rangle \\ &= \langle \theta_D, \phi(x_i) \rangle \end{aligned} \tag{27}$$

Furthermore,

$$||\theta||^2 = ||\theta_D||^2 + ||\theta_{D^\perp}||^2 \tag{28}$$

Proof II

Hence

$$\begin{aligned} &\Psi(\langle \theta, \phi(x_i)\rangle, \dots, \langle \theta, \phi(x_n)\rangle, ||\theta||^2) \\ &= \Psi(\langle \theta_D, \phi(x_i)\rangle, \dots, \langle \theta_D, \phi(x_n)\rangle, ||\theta_D||^2 + ||\theta_{D^\perp}||^2) \\ &\geq \Psi(\langle \theta_D, \phi(x_i)\rangle, \dots, \langle \theta_D, \phi(x_n)\rangle, ||\theta_D||^2) \end{aligned} \tag{29}$$

This means that

$$\begin{aligned} &\inf_{\theta \in \mathcal{H}} \Psi(\langle \theta, \phi(x_i)\rangle, \dots, \langle \theta, \phi(x_n)\rangle, ||\theta||^2) \\ &= \inf_{\theta \in \mathcal{H}_D} \Psi(\langle \theta, \phi(x_i)\rangle, \dots, \langle \theta, \phi(x_n)\rangle, ||\theta||^2) \end{aligned} \tag{30}$$

Application to supervised learning

The loss

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \langle\theta, \phi(x_i)\rangle + \lambda||\theta||^2 \tag{31}$$

is of the form Ψ and is increasing in the last variable.
As a direct consequence, the minimum of 31 is attained at

$$\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n \tag{32}$$

We note that no convexity hypothesis on l is required.

Consequence

We note

- α the vector such that $\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i)$.
- $K \in \mathbb{R}^{n,n}$ the matrix defined by

$$K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$$

Consequence

We remark that $\forall i \in [1, n]$,

$$\begin{aligned}
\langle \theta, \phi(x_i) \rangle &= \sum_{j=1}^{n} \alpha_j \langle \phi(x_j), \phi(x_i) \rangle \\
&= \sum_{j=1}^{n} \alpha_j K_{ij} \\
&= (K\alpha)_i
\end{aligned}$$

And we also remark that

$$\begin{aligned}
||\theta||^2 &= \langle\theta, \theta\rangle \\
&= \langle\sum_{i=1}^{n} \alpha_i \phi(x_i), \sum_{i=j}^{n} \alpha_j \phi(x_j)\rangle \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \langle\phi(x_i), \phi(x_j)\rangle \\
&= \sum_{i=1}^{n} \alpha_i (\sum_{j=1}^{n} K_{ij}\alpha_j) \\
&= \sum_{i=1}^{n} \alpha_i (K\alpha)_i \\
&= \alpha^T K \alpha
\end{aligned}$$

Finally

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, (K\alpha)_i) + \lambda \alpha^T K \alpha$$

$L(\theta)$ can be written **only** with K and α, instead of $\phi(x_i)$.
Natural question : But this does not make sense, as $\phi(x_i)$ and $\phi(x_j)$ are required to compute $K_{ij} = k(x_i, x_j)$?
Yes, **but**, in some situations, it is possible to compute $k(x_i, x_j)$ **without explicit knowledge** of ϕ. This is known as the **kernel trick**.

Alternate minimization problem

$$
\begin{aligned}
&\inf_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \langle \theta, \phi(x_i) \rangle) + \lambda ||\theta||^2 \\
&= \inf_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} l(y_i, (K\alpha)_i) + \lambda \alpha^T K \alpha
\end{aligned}
\tag{33}
$$

Alternate minimization problem

$$\begin{aligned}
&\inf_{\theta\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} l(y_i, \langle\theta, \phi(x_i)\rangle) + \lambda||\theta||^2 \\
=& \inf_{\alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} l(y_i, (K\alpha)_i) + \lambda\alpha^T K\alpha
\end{aligned} \tag{34}$$

It might be easier to optimize in $\mathbb{R}^n$ than in $\mathcal{H}$, especially if $\mathcal{H}$ is infinite dimensional.

Evaluation fonction

Also, we can rewrite the evaluation function as :

$$f(x) = \theta^T \phi(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

Gram matrix

The kernel matrix is a matrix of inner products. It is often called a Gram matrix. If we note the design matrix

$$\phi = \begin{pmatrix} \phi(x_1)^T \\ \dots \\ \phi(x_i)^T \\ \dots \\ \phi(x_n)^T \end{pmatrix}$$

Then

$$K = \phi\phi^T \in \mathbb{R}^{n,n} \tag{35}$$

Gram matrix

K is symmetric positive semi-definite. $\forall \alpha \in \mathbb{R}^n$,

$$\begin{aligned} \alpha K \alpha &= \alpha \phi \phi^T \alpha \\ &= (\phi^T \alpha)^T (\phi^T \alpha) \\ &= ||\phi^T \alpha||^2 \end{aligned} \tag{36}$$

Then, if λ is an eigenvalue of K, with eigenvector α_λ,

$$\begin{aligned} \alpha_\lambda K \alpha_\lambda &= \alpha_\lambda \lambda \alpha_\lambda \\ &= \lambda ||\alpha_\lambda||^2 \end{aligned} \tag{37}$$

Approximations

- when n is large, it can become too costly to compute and store K ($\mathcal{O}(n^2)$) and to solve the optimization problem ($\mathcal{O}(n^3)$).
- to avoid explicitely computing and storing K, **low-rank approximations** may be used (such as Nyström method)
- to solve the optimization problem, **low-rank decomposition** may be used.

See also

- Kernels on structured objects (graphs, texts, etc)
- Reproducing kernel Hilbert space (RKHS) (the space $\mathcal{H}$ that corresponds to k and ϕ)
- Adaptivity of kernel methods to the smoothness of the target function. If the optimization if performed in the right way, the convergence is faster for functions that are more than simply Lipshitz-continuous (in a future lecture).

Duality

An consequence of the lagrangian resolution of the optimization problem is that there exists α such that

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{38}$$

Hence

$$h_{w,b}(x) = \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + b \tag{39}$$

Only the inner products $\langle x_i, x_j \rangle$ are involved in the estimator. Similarly, the dual problem formulation also only uses the inner products. This motivates the use of kernels.

Separation function

$$h(x) = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b \tag{40}$$

Gaussian kernel (RBF)

$$k(x, x') = \exp\Big(- \gamma ||x - x'||^2 \Big) \tag{41}$$

With $k(x, x') = \langle \phi(x), \phi(x') \rangle$, $\phi(x) \in \mathcal{H}$

- $\mathcal{H}$ is of infinite dimension.
- γ is a parameter that should be carefully tuned.

Linear kernel

$$k(x, x') = \langle x, x' \rangle \tag{42}$$