MATHVSMACHINE: THE BOOK

LOUIS DE THANHOFFER DE VOLCSEY

Contents

Part I

Mathematical background

Chapter 1

Linear algebra

1.1 the (Moore-Penrose) Pseudo-inverse

In this section, we will let V, W be finite-dimensional vector spaces and $f \in \mathrm{Hom}_{\mathbb{R}}(V, W)$.
It is well-known that f does not have an inverse in general. There is however a natural generalization of the notion of inverse which can be defined for *any* map: a *pseudo-inverse*. More precisely, if f either has a nonzero kernel or if the image of f is not the whole of W, then the inverse of f will not exist. One natural way to remediate this issue is to consider complements for both subspaces and write

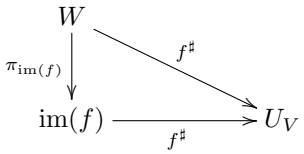
$$V \overset{\text{def}}{=} \ker(f) \oplus U_V \text{ and } W \overset{\text{def}}{=} \mathrm{im}(f) \oplus U_W$$

It's easy to see that restricting f to appropriate subspaces now does produce an invertible map as follows:

Lemma 1.1.0.1. *the map $f : U_V \longrightarrow \mathrm{im}(f)$ is an isomorphism.*

We'll denote the inverse of f on U_V by $f^\sharp : \mathrm{im}(f) \longrightarrow U_V$. A pseudo-inverse is now the natural lift of $f^\sharp$ to the whole of W:

Lemma 1.1.0.2. *There exists a unique map $f^\sharp : W \longrightarrow U_V$ making the following diagram commute:*



Proof. The commutativity of the diagram means that for $u \in U_V$, we have

$$f^\sharp(w) \overset{\text{def}}{=} u \iff f^\sharp(\pi_{\mathrm{im}(f)}(w)) = u \iff \pi_{\mathrm{im}(f)}(w) = f(u)$$

Where the second equivalence follows from the fact that $f^\sharp$ is the inver of f on U_V.
The claim will thus follow if we show that the above assignment is indeed a well-defined linear map. To this end assume that $u, u' \in U_V$ satisfy $f(u') = \pi_{\mathrm{im}(f)}(w) = f(u)$.
Then $u - u' \in \ker(f)$, hence $u - u' \in \ker(f) \cap U_V$ in particular. Now since $\ker(f) \oplus U_V = V$, we have $u - u' = 0$, so that $u = u'$, showing the well-definedness.
We leave the linearity to the reader. ∎

It will be helpful to note that the map $f^\sharp \in \mathrm{Hom}(W, V)$ can also be characterized by $\mathrm{im}(f^\sharp) \subset U_V$ and $f \circ f^\sharp = \pi_{\mathrm{im}(f)}$.
To give the map $f^\sharp$ a name, we first let $\Lambda(f)$ denote the set

$$\Lambda(f) \overset{\text{def}}{=} \{(U_V, U_W) |\ \ker(f) \oplus U_V = V \text{ and } \mathrm{im}(f) \oplus U_W = W\}$$

and conclude from Lemma 1.1.0.2 that there is a assignment:

$$\Phi : \Lambda(f) \longrightarrow \mathrm{Hom}_{\mathbb{R}}(W, V) : (U_V, U_W) \mapsto f^\sharp$$

where $f^\sharp \in \mathrm{Hom}_{\mathbb{R}}(W, V)$ is the unique map satisfying

$$f \circ f^\sharp = \pi_{\mathrm{im}(f)} \text{ and } \mathrm{im}(f^\sharp) \subset U_V$$

Let's denote the image of Φ by $\Pi(f)$. Summarizing the discussion, we make the following:

Definition 1.1.0.3. Let $(U_V, U_W) \in \Lambda(f)$. Then the pseudo-inverse of (U_V, U_W, f) is the map $\Phi(f)$. We say that $g \in \mathrm{Hom}_{\mathbb{R}}(W, V)$ is a pseudo-inverse to f if $g \in \Pi(f)$

We can give a slightly different description of pseudo-inverses by describing them on the 2 components in the decomposition $\mathrm{im}(f) \oplus U_W = W$:

Lemma 1.1.0.4. *Let (U_V, U_W) in $\Lambda(f)$. Then the following are equivalent:*

1. *$f^\sharp$ is the pseudo-inverse to (U_V, U_W, f)*
2. *$f^\sharp|_{\mathrm{im}(f)}$ is the inverse to $f : U_V \longrightarrow \mathrm{im}(f)$ and $f^\sharp|_{U_W} = 0$*

Proof. Since the pseudo=inverse to (U_V, U_W, f) is unique, it suffices to show that the pseudo-inverse indeed satisfies the conditions of (2). The fact that $f^\sharp|_{\mathrm{im}(f)}$ is the inverse of $f|_{U_V}$ follows from

$$(f \circ f^\sharp)|_{\mathrm{im}(f)} = \big(\pi_{\mathrm{im}(f)}\big)|_{\mathrm{im}(f)} = \mathrm{Id}\,|_{\mathrm{im}(f)}$$

Moreover, if $w \in U_W$, then $\pi_{\mathrm{im}(f)}(w) = 0$ since $\mathrm{im}(f) \oplus U_W$. Hence $f^\sharp(w) = f^\sharp(\pi_{\mathrm{im}(f)}(w)) = 0$ by Lemma 1.1.0.2 ∎

Our next order of business is to give an explicit description of the set $\Pi(f)$ of pseudo-inverses to f. We begin by showing that we can describe the complements U_V and U_W solely by using the maps f and $f^\sharp$:

Lemma 1.1.0.5. *Let $f^\sharp$ be the pseudo-inverse to (U_V, U_W, f). Then $U_V = \mathrm{im}(f^\sharp)$ and $U_W = \ker(f^\sharp)$*

Proof. We have $\mathrm{im}(f^\sharp) \subset U_V$ by Definition 1.1.0.3 . Moreover, $f^\sharp$ is a composition of surjections and hence itself surjective, proving the first claim.
To prove the second claim, note that the second condition of Lemma 1.1.0.4 immediately implies that $U_W \subset \ker(f^\sharp)$. We can also show the other inclusion by assuming that $w \in W$ satisfies $f^\sharp(w) = 0$, in which case $\pi_{\mathrm{im}(f)}(w) = f(f^\sharp(w)) = f(0) = 0$, implying that w lies in the component U_W of the decomposition $\mathrm{im}(f) \oplus U_W = W$ as required ∎

Taking the above lemma one step further allows us to describe the set $\Pi(f)$ of pseudo-inverses as promised:

Lemma 1.1.0.6. *Let $f \in \mathrm{Hom}(V, W)$. Then the following are equivalent:*

1. *$g \in \Pi(f)$*
2. *$(f \circ g)|_{\mathrm{im}(f)} = \mathrm{Id}$ and $(g \circ f)|_{\mathrm{im}(g)} = \mathrm{Id}$*

Proof. Let g be a pseudo-inverse to f and define $U_V \overset{\text{def}}{=} \mathrm{im}(g)$ and $U_W \overset{\text{def}}{=} \ker(f)$. Then Lemma 1.1.0.5 shows that g is in fact the pseudo-inverse to the triple $\big(U_V, U_W, f\big)$. Now, since $g|_{\mathrm{im}(\,f)}$ is the inverse to $f|_{U_V}$ by Lemma 1.1.0.4, we have $(f \circ g)|_{\mathrm{im}(f)} = \mathrm{Id}$ and $(g \circ f)|_{\mathrm{im}(g)} = (g \circ f)|_{U_V} = \mathrm{Id}$.
Conversely, assume that g satisfies the conditions in (2).
We begin by showing that $\big(\mathrm{im}(g), \ker(g)\big) \in \Lambda(f)$. Let's show that $\mathrm{im}(f) \oplus \ker(g) = W$ by way of

example. Indeed, first note that $\mathrm{im}(f) \cap \ker(g) = 0$, as any w in this intersection must satisfy $w = (f \circ g)(w) = f(0) = 0$. Moreover, if we write $w = \big(w - f(g(w))\big) + f(g(w))\big)$, we see that trivially $f(g(w)) \in \mathrm{im}(f)$ and

$$g(w - f(g(w))) = g(w) - (g(f(g(w)) = g(w) - g(w) = 0$$

so that $\big(w - f(g(w))\big) \in \ker(g)$. This indeed shows that $\mathrm{im}(f) \oplus \ker(g) = W$. The proof of $\mathrm{im}(g) \oplus \ker(f) = V$ is completely analogous, allowing us to conclude that $(\mathrm{im}(g), \ker(g)) \in \Lambda(f)$.
It now remains to show that g is indeed a pseudo-inverse to the triple $(\mathrm{im}(g), \ker(f), f)$. By Lemma 1.1.0.4, it suffices to show that $g|_{\mathrm{im}(f)}$ is the inverse to $f|_{\mathrm{im}(g)}$ and that $g|_{\ker(g)} = 0$. The first claim follows immediately from the fact that g is a left inverse to $f : \mathrm{im}(g) \longrightarrow W$ and the second claim is trivial. ∎

In order to summarize the previous 2 lemmas, we introduce the following assignment, which is well-defined by Lemma 1.1.0.5

$$\Psi : \Pi(f) \longrightarrow \Lambda(f) : g \mapsto \big(\mathrm{im}(g), \ker(g)\big)$$

We now have:

Lemma 1.1.0.7. *Let $f \in \mathrm{Hom}(V, W)$. Then:*

- $\Pi(f) = \big\{ g \in \mathrm{Hom}(W, V) \;\big|\; (f \circ g)|_{\mathrm{im}(f)} = \mathrm{Id} \;\; \textit{and} \;\; (g \circ f)|_{\mathrm{im}(g)} = \mathrm{Id} \big\}$
- *The assignments Φ and Ψ define 1:1 correspondences between $\Lambda(f)$ and $\Pi(f)$*

Proof. The first claim simply restates Lemma 1.1.0.6. To prove the second, we note that $\Psi \circ \Phi = \mathrm{Id}$ by Lemma 1.1.0.5. Moreover, Φ is surjective by definition, implying that $\Phi \circ \Psi = \mathrm{Id}$ as well ∎

We finish our discussion of pseudo-inverses by discussing a special choice of pseudo-inverse in $\Pi(f)$ that one can make if the vector spaces V and W are equipped with inner products. Indeed, recall the following standard result:

Lemma 1.1.0.8. *Let $U \subset V$ be a subspace of a finite dimensional inner product space. Then $U \oplus U^{\perp} = V$*

This leads us to the following Definition:

Definition 1.1.0.9. Let V, W be finite-dimensional inner product spaces and let $f \in \mathrm{Hom}_{\mathbb{R}}(V, W)$. Then the *Moore-Penrose pseudo-inverse* is the pseudo-inverse to the triple $(\ker(f)^{\perp}, \mathrm{im}(f)^{\perp}, f)$.
We will denote it by f^{+}

It turns out that we can give a very satisfying description of Moore-Penrose pseudo-inverses:

Lemma 1.1.0.10. *Let V, W be finite-dimensional inner product spaces and $f \in \mathrm{Hom}(V, W)$. Then the following are equivalent:*

1. *g is the Moore-Penrose pseudo-inverse f^{+} to f*
2. *g is a pseudo-inverse to f and $g \circ f$ and $f \circ g$ are self-adjoint linear maps*
3. *f and g satisfy $f \circ g \circ f = f$, $g \circ f \circ g = g$, $(g \circ f)^{*} = g \circ f$ and $(f \circ g)^{*} = f \circ g$*

Proof. The equivalence (2) $\iff$ (3) is simply a restatement of Lemma 1.1.0.7.
We now prove (2) $\implies$ (1):
Assume that g is a pseudo-inverse to f and that $g \circ f$ and $f \circ g$ are both self-adjoint. then Lemma 1.1.0.5 implies that g is the pseudo-inverse to the triple $\big(\mathrm{im}(g), \ker(f), f\big)$. The claim will thus follow

if we show that $\mathrm{im}(g) = \ker(f)^\perp$ and $\ker(g) = \mathrm{im}(f)^\perp$. By way of example, we will prove the former equality: First note that since $\mathrm{im}(g) \oplus \ker(f) = V$, it suffices to show that $\mathrm{im}(g) \perp \ker(f)$. Indeed, for $w \in W$ and $v \in \ker(f)$, we have:

$$\langle v, g(w)\rangle = \langle v, (g\circ f)(g(w))\rangle = \langle (g\circ f)^*(v), g(w)\rangle = \langle (g\circ f)(v), g(w)\rangle = \langle g(0), g(w)\rangle = 0$$

The proof of $\ker(g) = \mathrm{im}(f)^\perp$ is analogous.
Finally, we show (1) $\implies$ (2):
Assume that g is the Moore Penrose pseudo-inverse to f. Ie g is the pseudo-inverse to the triple $(\ker(f)^\perp, \mathrm{im}(f)^\perp, f)$. We will show that $(f\circ g)$ is self-adjoint and leave the other claim to the reader. To this end, let $v, v' \in V$. Then

$$\begin{aligned}\langle v, g(f(v'))\rangle &= \left\langle v - g(f(v)) + g(f(v)), g(f(v')) - v' + v' \right\rangle \\ &= \left\langle v - g(f(v)), g(f(v'))\right\rangle + \left\langle g(f(v)), g(f(v')) - v'\right\rangle + \left\langle g(f(v)), v'\right\rangle\end{aligned}$$

Now, since $f\circ g\circ f = f$, we conclude that $v - g(f(v))$ and $g(f(v')) - v'$ lie in $\ker(f)$. Moreover, since $\ker(f) = \mathrm{im}(g)^\perp$, we conclude that

$$\langle v - g(f(v)), g(f(v'))\rangle = \langle g(f(v)), g(f(v')) - v'\rangle = 0$$

So that

$$\langle v, g(f(v'))\rangle = \langle g(f(v)), v'\rangle$$

implying that $f\circ g = (f\circ g)^*$. The equality $g\circ f = (g\circ f)^*$ is completely analogous. ✗

As mentioned in the introduction of this section, our main motivation for studying the Moore-Penrose pseudo-inverse, is to provide a description of the projection of a vector onto the image of a linear map. We begin with the following preparatory lemma:

Lemma 1.1.0.11. *Let V, W be finite-dimensional inner product spaces and $f \in \mathrm{Hom}(V, W)$. Let $v \in V$ and $w \in W$. Finally denote the Moore-Penrose inverse of f by f^+. Then the following are equivalent:*

1. *$f(v)$ is the projection of w onto the subspace $\mathrm{im}(f)$*
2. *v satisfies the normal equation $(f^*\circ f)(v) = f^*(w)$*
3. *v lies in the affine subspace $f^+(w) + \ker(f)$*

Proof. The equivalence of (1) $\iff$ (2) is simply a restatement of Lemma **??**.
To show the equivalence of (1) $\iff$ (3), we first note that $f(f^+w)) = \pi_{\mathrm{im}(f)}$, where $\pi_{im(f)}$ is the projection onto the subspace $\mathrm{im}(f) \subset W$ by Lemma **??**. This shows that the vector $f^+(w) \in V$ indeed satisfies the condition (1). Next, assume (1), so that $v \in V$ satisfies $f(v) = \pi_{\mathrm{im}(f)}(v)$ and write $v = f^+(w) + v'$. Then

$$f(v) = \pi_{\mathrm{im}(f)(w)} \iff f(f^+(w) + v') = \pi_{\mathrm{im}(f)}(w) \iff \pi_{\mathrm{im}(f)}(w) + f(v') = \pi_{\mathrm{im}(f)}(w) \iff v' \in \ker(f)$$

This proves the claim ✗

This lemma has an interesting corollary which allows us to write the Moore-Penrose even more explicitly which will play an important role later on:

Corollary 1.1.0.12. *Let V be a finite dimensional vector space and W a finite dimensional inner product space. Let $f \in \mathrm{Hom}(V, W)$ be injective and choose* any *inner product on V. Then*

$$f^+ = (f^*\circ f)^{-1}\circ f^*$$

Proof. Since f is injective (so that $\ker(f) = 0$), f^+ is the pseudo-inverse to the triple $(V, \mathrm{im}(f)^\perp, f)$ by Definition 1.1.0.9. It follows immediately that this condition is independent of the inner product on V. To prove the formula, simply note that $f^* \circ f$ is invertible if f is injective and apply the second criterium of Lemma 1.1.0.11 ∎

We finish this section by giving a more explicit description of this map after introducing coordinates:
To this end, let $\{v_1, \ldots, v_n\}$ be an orthonormal basis for V and $\{w_1, \ldots, w_m\}$ be an orthonormal basis for W. We denote by $M : \mathrm{Hom}(V, W) \longrightarrow \mathrm{Mat}_{n\times m}(\mathbb{R})$ the isomorphism that assigns to any $f \in \mathrm{Hom}_{\mathbb{R}}(V, W)$ its associated matrix M_f.

Lemma 1.1.0.13. *For any f, we have $M_{f^+} = \left(M_f\right)^+$ where $\left(M_f\right)^+$ is the unique matrix M satisfying*

- *$M \cdot M_f \cdot M = M$ and $M_f \cdot M \cdot M_f = M_f$*
- *$M \cdot M_f$ and $M_f \cdot M$ are symmetric*

Proof. The first claim follows from the compatibility between composition of maps and multiplication of matrices. The second follows from the fact that the inner product is the standard one since the bases are orthonormal ∎

Chapter 2

Convex Analysis

2.1 Overview

One major aspect of machine learning is the study of *supervised learners*.
For these algorithms one is given a space of *features* $\mathfrak{X}$ together with a space of *labels* $\mathfrak{y}$ -the underlying idea being that to each feature $x \in \mathfrak{X}$, one can attach its correct label $y \in \mathfrak{y}$.
In order to do this, One first considers a *hypothesis space* $\mathfrak{H}$ of functions $\mathfrak{X} \longrightarrow \mathfrak{y}$ which consist of all the ways one wishes to make a prediction, and collects a *dataset* $\Delta \in \mathfrak{X} \times \mathfrak{y}$.
Using this dataset , one in turn defines a cost function $c : \mathfrak{H} \longrightarrow \mathbb{R}$ which intuitively describes how accurate any chosen hypothesis $h \in \mathfrak{H}$ is. The appropriate h hypothesis will then be the one that minimizes this cost function.

As such the problem of finding minima for functions plays a key role in the field of machine learning.
Arguably, the most popular approach to this problem is to construct a sequence of hypotheses $x_i \in \mathfrak{H}$ which decreases the cost function c at each step and hopefully converges to this global minimum.
One particularly elegant way of constructing such a sequence is the method of gradient descent.
In this chapter, we investigate this method and prove that it is indeed descending and converging to a global minimum under certain circumstances. Along the way, we give a bound for the error at each iteration.

2.2 Background and Notation

In this context, the hypothesis space $\mathfrak{H}$ will be a Hilbert space, i.e. a real vector space V, together with an inner product $\langle -, - \rangle$ such that the associated norm $||x|| \stackrel{\text{def}}{=} \sqrt{\langle x, x \rangle}$ is complete. For the reader not comfortable with the language of Hilbert spaces, we assure him it is safe to assume that V is finite-dimensional, in which case the Gram-Schmidt algorithm exhibits an orthonormal basis, and shows in particular that V is isomorphic to $\mathbb{R}^n$, together with the usual inner product $\langle x_1, \ldots, x_n, y_1, \ldots y_n \rangle \stackrel{\text{def}}{=} \sum_i x_i y_i$.

We recall that given Hilbert spaces V and W, we denote by $\text{Hom}_{\mathbb{R}}(V, W)$ the vector space of continuous and linear functions[1]. In particular if $W = \mathbb{R}$, we define $V^\star = \text{Hom}_{\mathbb{R}}(V, \mathbb{R})$. Moreover, for each element $x \in V$, we can consider $\langle x, - \rangle : V \longrightarrow \mathbb{R}$ which is clearly linear and continuous. This in turn defines a map $V \longrightarrow V^\star : x \mapsto \langle x, - \rangle$ and it is a well known fact from the theory of Hilbert spaces that this map is an isomorphism.

2.3 Subgradients

We begin our analysis of gradient descent sequences by analyzing the role of convexity in minimizing functions. To this end, recall that:

Definition 2.3.0.1. A function $f : V \longrightarrow \mathbb{R}$ is *convex* if for any x, y and $\alpha \in [0, 1]$, we have

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x') + (1 - \alpha) f(y)$$

Our main reason for considering this property is the following characterization of the gradient:

Theorem 2.3.0.2. *Let $f : V \longrightarrow \mathbb{R}$ be differentiable at x and convex.*
Then the gradient is the only vector $\nabla f(x) \in V$ satisfying

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle \tag{2.1}$$

for all $y \in V$

Proof. First, we show that the statement is true for the vector $\nabla f(x)$:
By lemma 2.4.0.1, given $\epsilon > 0$, for any $||y - x|| < \delta$, we have

$$\frac{|f(y) - f(x) - \text{Dir}_x f(v)|}{||y - x||} = \frac{|f(y) - f(x) - \langle \nabla f(x), y - x \rangle|}{||y - x||} < \epsilon$$

Removing the absolute values and reorganizing yields that

$$f(y) - f(x) - \epsilon ||y - x|| \geq \langle \nabla f(x), y - x \rangle$$

for all y in some ball $B(x, \delta)$.
We now use the convexity of f to show this in fact holds for *any* $y \in V$. Since $\epsilon > 0$ was chosen arbitrarily, the inequality (2.3) will follow immediately.
To this end, for any $y \in V$, consider the function

$$\psi(y) \stackrel{\text{def}}{=} f(y) - f(x) - \epsilon ||y - x|| - \langle \nabla f(x), y - x \rangle \tag{2.2}$$

[1] in the case where $\dim V \neq \infty$, any linear map is automatically continuous so that the latter condition becomes superfluous

Equation (2.4) states that $\psi(y) \geq 0$ whenever $y \in B(x, \delta)$. We must now show that $\psi(y) \geq 0$ for arbitrary $y \in V$.
We leave it to the reader to verify that ψ itself is convex and note that it is trivial that $\psi(x) = 0$. Now, pick $\alpha < \frac{\delta}{||y-x||}$ small enough so that $\alpha y + (1-\alpha)x \in B(x, \delta)$. Then by convexity, we have:

$$\alpha\psi(y) = \alpha\psi(y) + (1-\alpha)\psi(x) \geq \psi\big(\alpha y + (1-\alpha)x\big) \geq 0$$

It follows in particular that $\psi(y) \geq 0$, hence the existence.
To show that $\nabla f(x)$ is the unique vector satisfying the inequality (2.3), we assume that $v \in V$ satisfies

$$f(y) - f(x) \geq \langle v, y - x\rangle$$

And conclude that

$$\frac{\langle v - \nabla f(x), y - x\rangle}{||y-x||} \leq \frac{f(y) - f(x) - \langle \nabla f(x), y - x\rangle}{||y-x||} \leq \frac{|f(y) - f(x) - \langle \nabla f(x), (y-x)\rangle\,|}{||y-x||}$$

The differentiability of$f$ at x now implies that for any $\epsilon > 0$

$$\frac{\langle v - \nabla f(x), y - x\rangle}{||y-x||} < \epsilon$$

for all y in some ball $B(x, \delta)$. Using the cosine rule we conclude that

$$\frac{\langle v - \nabla f(x), y - x\rangle}{||y-x||} = ||v - \nabla f(x)|| \cdot \cos(\theta) < \epsilon$$

Where θ denotes the angle between v and $\nabla f(x)$ which immediately implies that $v = \nabla f(x)$, as ϵ was chosen arbitrarily. ✗

This characterization allows us to introduce gradients in the more general setting of convex, non-differentiable functions:

Definition 2.3.0.3. Let $f : V \longrightarrow \mathbb{R}$ be a convex function. The *subgradient* at x is the set $\partial f(x)$ of all vectors $v \in V$ such that

$$f(y) - f(x) \geq \langle v, y - x\rangle$$

for any $y \in V$

The above theorem shows that in the case where f is differentiable, we have

$$\partial f(x) = \{\nabla f(x)\}$$

A crucial theorem in the theory of convex optimization is:

Theorem 2.3.0.4. *Let $f : V \longrightarrow \mathbb{R}$ be convex, then the subgradient is a nonempty set.*

The interpretation of the subgradient is rather powerful, as the next two results are easily proven:

Lemma 2.3.0.5. *Let $f : V \longrightarrow \mathbb{R}$ be convex and differentiable everywhere. Then*

- *any local minimum is a global one*
- *Any stationary point is an either a global minimum or maximum*

Proof. To prove the first point, we use a standard argument already used in 2.5.0.2: If $f(y) \geq f(x)$ for all y in some ball $B(x, \delta)$, then we pick $\alpha < \frac{\delta}{||y-x||}$ small enough so that $\alpha y + (1-\alpha)x \in B(x, \delta)$, then using convexity, we obtain:

$$\alpha f(y) + (1-\alpha)f(x) \geq f(\alpha y + (1-\alpha)x) \geq f(x)$$

Which immediately proves the first claim.
For the second, if $\nabla f(x) = 0$, then using the fact that $\nabla f(x)$ is the subgradient, for any $y \in V$, we have

$$f(y) \geq f(x) - \nabla f(x)(y - x) = f(x)$$

∎

We now go on to prove an important inequality in a certain setting, which in some sense provides a converse to inequality (2.3)

Lemma 2.3.0.6. *Assume f is convex and differentiable everywhere. Moreover, assume the gradient satisfies the Lipschitz condition:*

$$|\nabla f(y) - \nabla f(x)| \leq K \cdot ||y - x||$$

Then for any $x, y \in V$, we have

$$f(x) + \nabla f(x)(y - x) \leq f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + K \cdot ||y - x||^2$$

Proof. The left hand side of the inequality is simply the subgradient inequality (2.3).
To prove the right-hand side we once again use the subradient inequality (2.3) to compute:

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y - x \rangle &\leq f(y) - \Big(f(y) + \langle \nabla f(y), x - y \rangle \Big) - \langle \nabla f(x), y - x \rangle \\ &= \langle \nabla f(y) - \nabla f(x), y - x \rangle \\ &\leq ||\nabla f(y) - \nabla f(x)|| \cdot ||y - x|| \\ &\leq K \cdot ||y - x||^2 \end{aligned}$$

∎

2.4 Gradient descent sequences

Given a function $f : V \longrightarrow W$ and a point $x \in V$, we say that f is *differentiable* at x if for each $v \in V$, the *directional derivative*

$$\mathrm{Dir}_x f(v) \stackrel{\mathrm{def}}{=} \lim_{\lambda \to 0} \frac{f(x + \lambda v) - f(x)}{\lambda}$$

is defined for any $v \in V$ and that $\mathrm{Dir}_x f(-) : V \longrightarrow W$ is linear and continuous. A classical result in analysis is the folllowing:

Lemma 2.4.0.1. *A function is differentiable at x iff there exists a linear map $L_x \in \mathrm{Hom}_{\mathbb{R}}(V, W)$ such that*

$$\lim_{||v|| \longrightarrow 0} \frac{||f(x + v) - f(x) - L_x(v)||}{||v||} = 0$$

In which case $L_x = \mathrm{Dir}_x f$

Furthermore, if $W = \mathbb{R}$, we can make the following observation:

Lemma 2.4.0.2. *Assume that $f : V \longrightarrow \mathbb{R}$ is differentiable. Then there exists a unique element $\nabla f(x) \in V$ such that*

$$\mathrm{Dir}_x f(-) = \langle \nabla f(x), - \rangle$$

Proof. Since $\mathrm{Dir}_x f$ lies in $V^\star$ by the definition of differentiability and since the map

$$V \longrightarrow V^* : x \longrightarrow \langle x, - \rangle$$

is an isomorphism, the result follows. ∎

Definition 2.4.0.3. Let $f : V \longrightarrow \mathbb{R}$ be differentiable. The function

$$\nabla f : V \longrightarrow V : x \mapsto \nabla f(x)$$

is the *gradient* of f.

The gradient has another interesting interpretation. To this end, for any x, we define the direction $\overline{x}$ of x as the class under the equivalence relation $x \sim y \iff \mathbb{R}^+ x = \mathbb{R}^+ y$.

Remark 2.4.0.4. *The direction of $\nabla f(x)$ yields the minimal directional derivative. Indeed, for any other direction $\overline{Y}$ represented by a unit vector y, we compute*

$$\mathrm{Dir}_x f(y) = \langle \nabla f(x), y \rangle = ||\nabla f(x)|| \cdot ||y|| \cos(\theta) = ||\nabla f(x)|| \cdot \cos\theta$$

where θ denotes the angle between x and y. This minimum is attained when $\cos(\theta) = -1$, in other words for any vector of direction $-\nabla f(x)$, by a simple application of the Cauchy-Schwarz theorem

This observation motivates gradient descent.
Indeed, as mentioned in the Overview, our goal is to minimize the function f. One approach would be to begin with any choice of starting point x_0 and subsequently build a sequence in such a way that the direction of any vector $x_{i+1} - x_i$ is $\nabla f(x_i)$, the minimal directional derivative.

Definition 2.4.0.5. A gradient descent sequence is a sequence $(x_i)_{i \in \mathbb{N}} \in V$ such that the direction of two subsequent elements satisfies

$$\overline{x_{i+1} - x_i} = \overline{\nabla f(x_i))}$$

In this case, we necessarily have

$$x_{i+1} = x_i - \lambda_i \nabla f(x_i)$$

We call $(\lambda_i)_i$ the sequence of *learning rates.*

This chapter is concerned with understanding when gradient descent sequences converge. The main result here being that in the case where f is differentiable and convex, δf is a Lipschitz function with constant K and the learning rates are constant such that $\lambda \leq \frac{1}{K}$, we can describe the convergence rather well.

2.5 Subgradients

We begin our analysis of gradient descent sequences by analyzing the role of convexity in minimizing functions. To this end, recall that:

Definition 2.5.0.1. A function $f : V \longrightarrow \mathbb{R}$ is *convex* if for any x, y and $\alpha \in [0, 1]$, we have

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x') + (1 - \alpha) f(y)$$

Our main reason for considering this property is the following characterization of the gradient:

Theorem 2.5.0.2. *Let $f : V \longrightarrow \mathbb{R}$ be differentiable at x and convex.*
Then the gradient is the only vector $\nabla f(x) \in V$ satisfying

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle \tag{2.3}$$

for all $y \in V$

Proof. First, we show that the statement is true for the vector $\nabla f(x)$:
By lemma 2.4.0.1, given $\epsilon > 0$, for any $||y - x|| < \delta$, we have

$$\frac{|f(y) - f(x) - \mathrm{Dir}_x f(v)|}{||y - x||} = \frac{|f(y) - f(x) - \langle \nabla f(x), y - x\rangle|}{||y - x||} < \epsilon$$

Removing the absolute values and reorganizing yields that

$$f(y) - f(x) - \epsilon||y - x|| \geq \langle \nabla f(x), y - x\rangle$$

for all y in some ball $B(x, \delta)$.
We now use the convexity of f to show this in fact holds for *any* $y \in V$. Since $\epsilon > 0$ was chosen arbitrarily, the inequality (2.3) will follow immediately.
To this end, for any $y \in V$, consider the function

$$\psi(y) \stackrel{\text{def}}{=} f(y) - f(x) - \epsilon||y - x|| - \langle \nabla f(x), y - x\rangle \tag{2.4}$$

Equation (2.4) states that $\psi(y) \geq 0$ whenever $y \in B(x, \delta)$. We must now show that $\psi(y) \geq 0$ for arbitrary $y \in V$.
We leave it to the reader to verify that ψ itself is convex and note that it is trivial that $\psi(x) = 0$.
Now, pick $\alpha < \frac{\delta}{||y-x||}$ small enough so that $\alpha y + (1 - \alpha)x \in B(x, \delta)$. Then by convexity, we have:

$$\alpha\psi(y) = \alpha\psi(y) + (1 - \alpha)\psi(x) \geq \psi\big(\alpha y + (1 - \alpha)x\big) \geq 0$$

It follows in particular that $\psi(y) \geq 0$, hence the existence.
To show that $\nabla f(x)$ is the unique vector satisfying the inequality (2.3), we assume that $v \in V$ satisfies

$$f(y) - f(x) \geq \langle v, y - x\rangle$$

And conclude that

$$\frac{\langle v - \nabla f(x), y - x\rangle}{||y - x||} \leq \frac{f(y) - f(x) - \langle \nabla f(x), y - x\rangle}{||y - x||} \leq \frac{|f(y) - f(x) - \langle \nabla f(x), (y - x)\rangle|}{||y - x||}$$

The differentiability of$f$ at x now implies that for any $\epsilon > 0$

$$\frac{\langle v - \nabla f(x), y - x\rangle}{||y - x||} < \epsilon$$

for all y in some ball $B(x, \delta)$. Using the cosine rule we conclude that

$$\frac{\langle v - \nabla f(x), y - x\rangle}{||y - x||} = ||v - \nabla f(x)|| \cdot \cos(\theta) < \epsilon$$

Where θ denotes the angle between v and $\nabla f(x)$ which immediately implies that $v = \nabla f(x)$, as ϵ was chosen arbitrarily. ∎

This characterization allows us to introduce gradients in the more general setting of convex, non-differentiable functions:

Definition 2.5.0.3. Let $f : V \longrightarrow \mathbb{R}$ be a convex function. The *subgradient* at x is the set $\partial f(x)$ of all vectors $v \in V$ such that

$$f(y) - f(x) \geq \langle v, y - x\rangle$$

for any $y \in V$

The above theorem shows that in the case where f is differentiable, we have

$$\partial f(x) = \{\nabla f(x)\}$$

A crucial theorem in the theory of convex optimization is:

Theorem 2.5.0.4. *Let $f : V \longrightarrow \mathbb{R}$ be convex, then the subgradient is a nonempty set.*

The interpretation of the subgradient is rather powerful, as the next two results are easily proven:

Lemma 2.5.0.5. *Let $f : V \longrightarrow \mathbb{R}$ be convex and differentiable everywhere. Then*

- *any local minimum is a global one*
- *Any stationary point is an either a global minimum or maximum*

Proof. To prove the first point, we use a standard argument already used in 2.5.0.2: If $f(y) \geq f(x)$ for all y in some ball $B(x, \delta)$, then we pick $\alpha < \frac{\delta}{||y-x||}$ small enough so that $\alpha y + (1-\alpha)x \in B(x, \delta)$, then using convexity, we obtain:

$$\alpha f(y) + (1-\alpha)f(x) \geq f(\alpha y + (1-\alpha)x) \geq f(x)$$

Which immediately proves the first claim.
For the second, if $\nabla f(x) = 0$, then using the fact that $\nabla f(x)$ is the subgradient, for any $y \in V$, we have

$$f(y) \geq f(x) - \nabla f(x)(y-x) = f(x)$$

∎

We now go on to prove an important inequality in a certain setting, which in some sense provides a converse to inequality (2.3)

Lemma 2.5.0.6. *Assume f is convex and differentiable everywhere. Moreover, assume the gradient satisfies the Lipschitz condition:*

$$|\nabla f(y) - \nabla f(x)| \leq K \cdot ||y - x||$$

Then for any $x, y \in V$, we have

$$f(x) + \nabla f(x)(y-x) \leq f(y) \leq f(x) + \langle \nabla f(x), y-x \rangle + K \cdot ||y-x||^2$$

Proof. The left hand side of the inequality is simply the subgradient inequality (2.3).
To prove the right-hand side we once again use the subradient inequality (2.3) to compute:

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y-x \rangle &\leq f(y) - \Big(f(y) + \langle \nabla f(y), x-y \rangle \Big) - \langle \nabla f(x), y-x \rangle \\ &= \langle \nabla f(y) - \nabla f(x), y-x \rangle \\ &\leq ||\nabla f(y) - \nabla f(x)|| \cdot ||y-x|| \\ &\leq K \cdot ||y-x||^2 \end{aligned}$$

∎

This inequality forms the key to the gradient descent convergence theorem.

Theorem 2.5.0.7 (convergence of GD). *Assume f is differentiable and convex, and that ∇f is Lipschitz with constant K. Assume λ is a constant sequence of learning rates such that*

$$\lambda < \frac{1}{K}$$

. Assume that f reaches a minimum at x_μ. Then

- *the sequence $||x_{i+1} - x_i||$ is square summable and converges to x_μ*

- *For $f^\mu \overset{\text{def}}{=} f(x_\mu)$. We have the following bound:*

$$|f(x_i) - f^\mu| \leq \frac{||x_0 - x_\mu||}{2\lambda \cdot K}$$

Proof. Let $x \in V$, $\lambda \in \mathbb{R}$ and define $x^+ \overset{\text{def}}{=} x - \lambda \nabla f(x)$.
then we compute:

$$\begin{aligned} f(x^+) &\leq f(x) + \langle \nabla f(x), x^+ - x \rangle + K||x^+ - x||^2 \\ &= f(x) - \lambda ||\nabla f(x)||^2 + K\lambda^2 ||\nabla f(x)||^2 \\ &= f(x) - \left(\lambda - K\lambda^2\right) \cdot ||\nabla f(x)||^2 \end{aligned}$$

From this last in equality we can draw a few conclusions:

- First, since $\lambda \leq \frac{1}{K}$, we have $\lambda - K\lambda^2 \geq 0$. In particular:

$$f(x^+) \leq f(x)$$

- Next, rewriting, we see that

$$||\nabla f(x)||^2 \leq \frac{1}{K\lambda^2 - \lambda} \Big(f(x) - f(x^+) \Big)$$

  Implying that for a gradient sequence $(x_i)_i$ (using telescopic sums), we have

$$\sum_i^n ||x_{i+1} - x_i||^2 = \lambda^2 \sum_i^n ||\nabla f(x_i)||^2 \leq \frac{\lambda^2}{K\lambda^2 - \lambda} |f(x_0) - f(x_n)| \leq \frac{\lambda^2}{K\lambda - \lambda^2} |f(x_0) - f^\mu|$$

  We conclude that the sequence $||x_{i+1} - x_i||$ is square summable. Moreover, since the gradient is Lipschitz, it is continuous in particular. Hence

$$0 = \lim \nabla f(x_i) = \nabla f(\lim x_i))$$

  meaning that $\nabla f\big(\lim(x_i)\big)$ is a stationary point, hence the global minimum x_μ by Lemma 2.5.0.5

To prove the second bound, we first apply the subgradient inequality (2.3) to x:

$$f(x) \leq f(x_\mu) + \langle \nabla f(x), x - x_\mu \rangle$$

to obtain

$$f(x^+) \leq \Big(f(x_\mu) + \langle \nabla f(x), x - x_\mu \rangle \Big) + (K\lambda^2 - \lambda) \cdot ||\nabla f(x)||^2$$

Now since K and λ are positive, we conclude $K\lambda^2 - \lambda \geq \frac{\lambda}{2}$, so that

$$\begin{aligned} f(x^+) - f(x_\mu) &\leq f(x_\mu) + \langle \nabla f(x), x - x_\mu \rangle - (K\lambda^2 - \lambda) \cdot ||\nabla f(x)||^2 \\ &\leq \frac{1}{2\lambda} \Big(2\lambda \langle \nabla f(x), x - x_\mu \rangle - \lambda^2 ||\nabla f(x)||^2 \Big) \\ &= \frac{1}{2\lambda} \Big(||x - x_\mu||^2 - \big(||x - x_\mu||^2 - 2\lambda \langle \nabla f(x), x - x_\mu \rangle + \lambda^2 ||\nabla f(x)||^2 \big) \Big) \\ &= \frac{1}{2\lambda} \Big(||x - x_\mu||^2 - ||x - \lambda \nabla f(x) - x_\mu||^2 \Big) \\ &= \frac{1}{2\lambda} \Big(||x - x_\mu||^2 - ||x^+ - x_\mu||^2 \Big) \end{aligned}$$

We return to our gradient descent sequence $(x_i)_i$. Then

$$\sum_i f(x_{i+1}) - f^\mu \leq \sum_i \frac{1}{2\lambda}\left(||x_i - x_\mu||^2 - ||x_{i+i} - x_\mu||^2\right) \quad = \frac{1}{2\lambda}\left(||x_0 - x_\mu||^2 - ||x_{i+1} - x_\mu||^2\right) \leq \frac{1}{2\lambda}||x_0 - x||^2$$

Moreover, since f decreases with each iteration by the above, it follows that

$$k \cdot \left(f(x_n) - f^\mu\right) \leq \sum_i^n f(x_i) - f^\mu$$

, implying that

$$f(x_n) - f^\mu \leq \frac{||x_0 - x^*||^2}{2\lambda k}$$

∎

Chapter 3

Probability theory

3.1 Overview

Definition 3.1.0.1.

3.2 Sampling

Convention 3.2.0.1. *We will fix a probability space* $(U, \mathcal{U}, P)$ *and throughout any random variable will always have this as its domain*

Definition 3.2.0.2. Let $(\Omega, \mathcal{F})$ be a measure space.
Assume an underlying probability space with measure P(following conv 3.2.0.1).
The *sample space* of Ω is the set

$$S\Omega \stackrel{\text{def}}{=} \coprod_{i \in \mathbb{N}} \Omega^i$$

Together with the canonical $\sigma-$algebra.
The space of random samples is

$$RS\Omega \stackrel{\text{def}}{=} \coprod_{i \in \mathbb{N}} \left\{ (X_1, \dots X_n) | \, X_i \text{ i.i.d } \sim P \right\}$$

We say that $(x_1 \dots x_n)$ is sampled from P if it lies in the image of the canonical map

$$\text{ev} : \Omega \times RS\Omega \longrightarrow S\Omega : \Big(\omega, \big(X_1, \dots X_n \big) \Big) \mapsto \Big(X_1(\omega), \dots X_n(\omega) \Big)$$

Remark 3.2.0.3. *The above map* $\text{ev} : \Omega \times RS\Omega \longrightarrow S\Omega$ *: will be referred to as the sampling map*

Chapter 4

Statistics

4.1 Overview

Convention 4.1.0.1. *Throughout, we will fix a probability space* $(U, \mathcal{U}, \mu)$

4.2 Statistical models

Definition 4.2.0.1. Let $(\Theta, \mathcal{O})$ and $(\Omega, \mathcal{F})$ be measurable spaces.
A Markov kernel (denoted $\Theta \implies \Omega$) is a map

$$\Sigma : \Omega \times \mathcal{F} \longrightarrow \mathbb{R}$$

Where for each $\theta \in \Theta$ the function $\Sigma(\theta, -)$ is a probability measure on Ω and for each $A \in \mathcal{F}$, the function $\sigma(-, A)$ is $\mathcal{O}$-measurable.
If each probability measure $\Sigma(\theta, -)$ is dominated by the measure λ (conv. 4.1.0.1), then we write $\Sigma(\theta, -) \stackrel{\text{def}}{=} P_\theta$ and call it a *statistical model*

Example 4.2.0.2. *Let* $(\Omega, \mathcal{F})$ *be a measurable space and consider the space of samples (def. 3.2.0.2) given by* $S\Omega \stackrel{\text{def}}{=} \coprod_{in\in\mathbb{N}} \Omega^i$.
Then we can define a statistical model $\Sigma : S\Omega \implies \Omega$ *through*

$$\sigma\big((x_1, \ldots, x_n), A\big) = \frac{1}{n} \sum \text{card}\left\{1_A(x_i)\right\}$$

This is the descriptive statistical model of $(\Omega, \mathcal{F})$

4.2.1. the category Stat(Θ) The collection of statistical models on the parameter space Θ naturally forms a category by defining a morphism as follows:

the pullback functor

Definition 4.2.1.1. Let $p : \Theta' \longrightarrow \Theta$ be any map (we intuitively think of p as focussing on a property of Θ).
Let $\Sigma : \Theta \implies \Omega$ be a statistical model.
Then we can define the push-forward of Σ along p, denoted $p^*\Sigma$ as

$$p^*\Sigma : \Theta \times \mathcal{F} \longrightarrow \mathbb{R} : \big(\omega, A\big) \mapsto \Sigma\big(p(\omega), A\big)$$

4.2.2. Estimators An important aspect of statistical models is that given a sample $x \in S\Omega$ we'd like to find a way to asociate a parameter $\theta \in \Theta$ (a process known as *estimation*).
To this end, we recall definition **??** and introduce an *estimation space* as a set E and define:

Definition 4.2.2.1. An estimator for the statistical model Σ is a sequence of functions

$$S\Omega \xrightarrow[e]{} E \xrightarrow[p]{} \Theta \cup \{\infty\}$$

such that the image of the composition lies in Θ.
Note that for any $\omega \in \Omega$, we immediately obtain a map $RS\Omega \longrightarrow \Theta$ from the random sample space given by

$$\begin{array}{ccc} S\Omega & \xrightarrow{\quad e \quad} & E \\ {\scriptstyle \mathrm{ev}_\omega}\uparrow & & \downarrow{\scriptstyle p} \\ RS\Omega & \xrightarrow[e_R]{} & \Theta \cup \{\infty\} \end{array}$$

4.3 Bayesian models

Definition 4.3.0.1. Let $(\Theta, \mathcal{O})$ and $(\Omega, \mathcal{F})$ be measurable spaces. A Bayesian model is a probability measure Π on the measurable space $(\Theta \times \Omega, \mathcal{O} \times \mathcal{F})$

Part II

statistics

If a statistical model is dominated, one can consider the following map

$$\text{MLE} : \Sigma \longrightarrow \mathcal{P}(\Omega) : x \mapsto \text{argsup}_\theta \, f(\theta, x)$$

We call $L \in \text{Hom}(\Sigma, \Omega$ a maximum likelihood estimator if $L(\theta) \in MLE$.

4.4 Bayesian Models

An important subset of statistical models are so-called Bayesian models in which one considers extra structure which lifts $(\Theta, \mathcal{O})$ to a probability space itself. Assume we are given a probability measure P on $(\Theta, \mathcal{O})$, then together with the statistical model $P_{\mathcal{O}}$, one can form the probability $\Pi \stackrel{\text{def}}{=} P \rtimes P_\theta$ following theorem **??** and since one can recover P and P_θ from Π, it makes sense to define:

Definition 4.4.0.1. A stochastic statistical model is a probability measure Π on $\Theta \times \Sigma, \mathcal{O} \times \mathcal{S})$ which decomposes into $P \rtimes P_{\mathcal{O}}$

One can go one step further and demand this definition be symmetric:

Definition 4.4.0.2. A Bayesian model Π is a probability measure on $\Theta \times \Sigma, \mathcal{O} \times \mathcal{S})$ which decomposes into $\mu \rtimes \nu_{\mathcal{O}}$ and $\nu \rtimes \mu_{\mathcal{S}}$

Before we go on, we wish to introduce a bit of abuse of notation in two steps. First we denote

$$P(E) \stackrel{\text{def}}{=} \mu(E) = \Pi(E \times \Sigma \, , P(F) \stackrel{\text{def}}{=} \nu(F) = \Pi(\Theta \times F)$$

where the second equalities come from lemma **??**. Following this principle we will always denote $\theta \times \Sigma$ simply by θ whenever the context is clear. The second step of abuse of notation is motivated by the case where the parameter space Θ is finite.

Lemma 4.4.0.3. *Assume Θ and Σ are finite spaces and let Π be as in 4.4.0.2. Then*

- $\mu_{\mathcal{S}}(\theta, F) = \Pi(\theta \times \Sigma | \Theta \times F) \stackrel{\text{def}}{=} P(\theta|F)$
- $P_{\mathcal{O}}(E, x) = \Pi(E|x)$

Proof. By definition, we have

$$\Pi(\theta \times F) = P(\theta \cup F) = \int_{\theta \times \Sigma} \mu_{\mathcal{S}}(-, F) d\Pi = \mu_{\mathcal{S}}(-, F) P(\theta)$$

and vice versa. ∎

Hence we will now denote the kernel $\mu_S(\theta, F)$ by $P(\theta|F)$ and $P_{\mathcal{O}}(E, x)$ by $P(E|x)$
We call $P(E)$ on Θ and $P(F)$ on Σ the prior and predictive probabilities whereas $P(E|x)$ and $P(\theta|F)$ are the sampling and posterior kernels respectively.

Example 4.4.0.4. *A good example to keep in mind is the following: assume we have a bunch of emails from people. We let Θ be the set of people, Σ the set of words in the emails and Π(Allen, work) be the number of times Allen has written the word 'work' (normalized). In this case the prior P is the number of words a person has written, the predictive P is the number of people having written a given word. The sampling kernel P(work | allen) is the probability among the words written by allen, we pick 'work' and the posterior probability P(Allen | work) is the probability that the word work was in fact written by Allen*

We now assume that the Bayesian model is dominated (meaning that both the posterior and sampling kernel are dominated). In this case it defines two estimators. The MLE given by considering the distribution of kernel $P(-|\theta)$, written as $f_{x|\theta}$ as well as the maximum a posteriori estimate, given by considering

$$MAP : \Sigma \longrightarrow \mathbb{R} : x \mapsto \text{argsup}_\theta \, f_{\theta|x}(\theta|x)$$

4.5 Bayesian ML Learning

Based off the Bayesian statistical model we described, one defines a Bayesian Machine learning model in which one parametrizes *data* by hypothesis. Let $H \subset \text{Hom}((\Theta, \mathcal{O}), (\Sigma, \mathcal{S}))$ be a finite set of *hypothesis* endowed with the discrete σ-algebra. We consider $D \subset \Theta \times \Sigma, \mathcal{O} \times \mathcal{S}$ as a space of data.

Definition 4.5.0.1. A Bayesian ML model is a statistical model on $H \times D$

If we assume given a probability P on H. Then there is a canonical way to define a Bayesian ML model by letting

$$\Pi(h \times F) \stackrel{\text{def}}{=} \mu(h)\delta_{h,F}$$

where $\delta_{h,F}$ if the graph of h lies in F and zero else. We can this the noiseless model with prior P.

Lemma 4.5.0.2. *Consider the above model. and let $VS_{H,F}$, the version space of F be the set of all $h \in H$ whose graph lies in F. Then*

- *the prior is given by $P(h)$*
- *the sampling kernel is given by $P(E|h) = \Pi(F|h) = \delta_{h,F}$* [1]
- *The predictive probability is given by $P(F) = \mu(VS_H, F)$*
- *the posterior kernel is given by $\mu_S(h, D) = \Pi(h|D) = \frac{\delta_{h,F} P(h)}{\mu(VS_{H,F})}$*

We will also be interested in Bayesian ML models wich incorporate noise. Here one is not just interested in wanting $\Pi(h \times F)$ to only be nonzero if the graph of f describes the data. Rather, one defines noise on space as a probability and keeps track of that. We are not sure how to formalize this in the the above context forn now.

4.6 Bayesian Networks

Definition 4.6.0.1. Let $(\Omega, \mathcal{A}, P)$ be a probability space and $(X_i)_{v \in V} : \Omega \longrightarrow \Sigma$ a set of random variables. A Bayesian network is an acyclic quiver $\mathcal{Q}$ with vertices v.

Given a bayesian network and a node i, the parents of i are all node with an arrow coming in to i.

Definition 4.6.0.2. We say that the random variables X_v follow the Bayesian network $\mathcal{Q}$ if

$$f_{X_1 \wedge \ldots \wedge X_n} = \prod_i f_{X_i | \wedge X_{j, \text{j parent of i}}}$$

In the case where the quiver consists of a single node 1 and a single arrow to $2 \ldots n$. We say the network is naive, so that

$$P(A_1, \ldots A_n) = P(A_1) \prod_i P(A_i|A_1)$$

Where we used the definition of the density function of conditioning one random variable wrt to another.

[1] recall that we identify $h with h \times D$ and F with $H \times F$ implicitely

4.7 Naive Bayes

Definition 4.7.0.1. We say that a Bayesian model $\Theta \times \Sigma$ is naive if $P(-|\theta)$ is given by a naieve bayes network.

4.8 Types of Bayesian Models

Let P_θ denote a statistical model. Let $\mathcal{Q}$ be a Bayesian quiver.

Definition 4.8.0.1. We say that a Bayesian model is of P_θ-$\mathcal{Q}$ type if there exist random variables $X_i : \Omega \longrightarrow \Sigma$ whose distributions lie in the family P_theta who follow the Bayesian quiver $\mathcal{Q}$

4.8.1. A Naive Bernouilli Model We consider the problem of classifiying text messages according to people. We let Θ be the space of people, let V, the vocabulary of words. We order these words to obtain V^n and let $\Sigma = \{0,1\}^n$ (so that a an element represents a text message containing the words corresponding to places with a 1). We assume the sampling probabilities are bernouilli distributed so that for $\theta \in \Theta$ there exist $\pi_theta \in [0,1]^n$ such that for a vector $x \in \Sigma$:

$$P(x|\theta) \stackrel{\text{def}}{=} \prod_i (\pi_\theta)_i^{x_i} (1 - (\pi_\theta)_i^{(1-x_i)}$$

We assume Θ has a prior distribution given by $P(\theta)$

Lemma 4.8.1.1. *Let Π be a finite Bayesian model with Bernouilli sampling distribution*

$$MAP(x) = \text{argmax}_\theta \prod_i P(x_i|\theta)^{x_1} (1 - P(x_i|\theta)^{1-x_i}) P(\theta)$$

Proof. Let $x \in \Sigma$. Since the model is naive we can assume

$$P(x \cap \theta) = p(\theta) p(x|\theta)$$

so that it suffices to maximize $p(\theta)p(x|\theta)$ Writing the binomial distribution out and taking ln yields

$$\delta_{x,\sigma} \ln(\pi_{x,\theta}) + (1 - \delta_{x,\sigma}) \ln(1 - \pi_{x,\theta}) + p(\theta)$$

In order to maximize this, we view this as a function of the variable $\pi_{x,\theta}$ and compute the singular points to get

$$\delta_{x,\sigma}(1 - \pi_{x,\theta}) - (1 - \delta_{x,\sigma})\pi_{x,\theta} = 0 \iff \delta_{x,\sigma} = \pi_{x,\theta}$$

∎

4.9 Estimators and Parameters

Arguably statistics is about the following problem: We are given a measurable function

$$X : (\Omega, \mathcal{F}, P) \longrightarrow (T, \mathcal{T}, \tau)$$

and assume that τ dominates P_X so that P_X has a density function $f_X : (T, \mathcal{T}, \tau) \longrightarrow (\mathbb{R}, \mathcal{L})$.
We assume given a parameter space

$$(\Theta, \mathcal{G}) \times T \longrightarrow \mathbb{R}$$

such that for each $\theta \in \Theta$, f_θ is a density function. and $f_X = f_\theta$ for some $\theta \in \Theta$ (called the parameter).
An *estimator of the random variable X* is a sequence of random variables

$$\hat{\theta}_n : (\Omega, \mathcal{F}, P) \longrightarrow (\Theta, \mathcal{G})$$

which has *desirable properties.*

The exact meaning of desirable properties explains the ambiguity of statistics. Since statistics is concerned with making conclusions based on a sample size $x_1, \dots x_n$ of outcomes as opposed to X itself, we shall only concern ourselves with definitions of this type

4.9.1. Consistent Estimators

Definition 4.9.1.1. An estimator $\hat{\theta}_n$ is consistent if it converges in probability to θ $\hat{\theta_n} \xrightarrow{P} \theta$. I.e. for any choice ϵ

$$P(|\theta_n - \theta| < \epsilon) \overset{n\to\infty}{\longrightarrow} 1$$

we have the following alternate characterization of constitency:

Lemma 4.9.1.2. *The following are equivalent:*

- *the estimator $\hat{\theta_n}$ is consistent*
- $E(\theta_n) = \theta$ *and* $\mathrm{Var}[\theta_n] = 0$

Proof. ∎

4.10 Maximum Likelihood Estimation

Let $(\Theta, \mathcal{O}) \Longrightarrow (\Omega, \mathcal{F})$ be a statistical model. Taking associated density functions yields

$$\Theta \times \Omega \longrightarrow \mathbb{R} : (\theta, x) \mapsto f_\theta(x)$$

Where F_θ is the density of the distribution θ. We now assume given a set of points $(x_1, \dots x_n)$. The purpose of maximum likelihood estimation is to pick the parameter θ such that the *'likelihood'* of the outcomes x_i is maximized. More precisely, we consider the associated statistical model $(\Theta, \mathcal{O}) \Longrightarrow (\Omega^n, \mathcal{F}^n)$, with associated density functions

$$\Theta \times \Omega^n \longrightarrow \mathbb{R} : (\theta, x_1 \dots x_n) \mapsto \prod_i f_\theta(x_i)$$

Then the maximum likelihood is the function $\hat{\theta} : \Omega^n \longrightarrow \Theta$ given by

$$\hat{\theta}(x_1 \dots x_n) = \max_{\theta \in \Theta} \prod_i f_\theta(x_i)$$

whenever this maximum exists. For technical reasons, we sometimes compute the maximum log-likelihood instead

$$\max_\theta \sum_i \ln f_\theta(x_i)$$

Example 4.10.0.1. *(a biased coin) Let $\Omega \overset{\text{def}}{=} \{H, T\} \longrightarrow \{0, 1\}$ be the random variable associated to throwing a coin in the air with P(heads)= p. Assume we repeat the experiment n times to obtain the random variable. And let X be the number of heads. Then X is binomially distributed with success p. assume that after performing the experiment X, we obtain the number i of heads. which p yields the maximum likelihood?*
Well, X is binomially distributed so that

$$\ln f_p(k) = \ln(\binom{n}{k} p^k (1-p)^{n-k}) = \ln \binom{n}{k} + k \ln(p) + (n-k) \ln(1-p)$$

setting the derivative with respect to p to 0 yields $p = \frac{n}{k}$ so that given an outcome of i heads, $p = \frac{i}{n}$ yields the highest likelihood

Example 4.10.0.2. *Another example comes from logistical regression*

4.10.1. Gauss' Principle the techniaue of MLE can be used to motivate the normal (or Gaussian) distribution.

Definition 4.10.1.1. we say that a probability on $\mathbb{R}$ is normally distributed if it is dominated by the Lebuesgue measure with density function

$$\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}$$

Or more generally after the change of variable $x \mapsto \frac{x-\mu}{\sigma}$

$$\frac{1}{\sqrt{2\pi}}e^{\frac{-\frac{(x-\mu)^2}{\sigma}}{2}}$$

Lemma 4.10.1.2. *if X is normally distributed, then*

- $X[X] = \mu$
- $S[X] = \sigma$

One interesting characterization of the normal distribution is that it is

the sole distribution whose MLE always coincides with the sample mean

To give a more detailed explanation of this sentence, we start with the following lemma:

Lemma 4.10.1.3. *Let*

$$f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R} : (\mu, \sigma, x) \mapsto \frac{1}{\sqrt{2\pi}}e^{\frac{-\frac{(x-\mu)^2}{\sigma}}{2}}$$

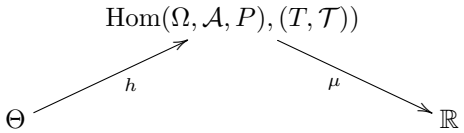
Then the MLE associated to the sample $x_1 \dots x_n$ is

$$\hat{\mu} = \sum \frac{x_i}{n} \text{ and } \hat{\sigma} = (\frac{1}{n}\sum (x_i - \hat{\mu})^2)^{\frac{1}{2}}$$

Proof. This is a straightforward exercise. ∎

4.11 Estimators

Definition 4.11.0.1. Let $X : (\Omega, \mathcal{A}, P) \longrightarrow (T, \mathcal{T})$ be a random variable. and $(x_1, \dots, x_n)$ a sample (set of outcomes) An estimator consists of a daigram



such that $\Theta \longrightarrow \mathbb{R}$ attains a maximum for a certain function h_θ, the estimation

There are two important classes of estimation:

4.11.1. Maximum likelihood estimation In this case, we suppose given $x_1, \dots x_n$ outcomes of the random variable X. Given the map $\Theta \longrightarrow \text{Hom}(\Omega, \mathcal{A}, P), (T, \mathcal{T}))$, for $h_\theta \in \Theta$, we let

$$\mu(h_\theta) = f_{h_\theta^n}(x_1, \dots x_n)$$

This number is the likelihood of $(x_1, \dots x_n)$ assuming θ

4.11.2. Maximum A Posteriori Estimate We assume that $\Theta, \mathcal{A}, P)$ now has the structure of a probability space and let d given outcomes $x_1, \dots x_n$ as well as a random variable $\Theta \longrightarrow \mathbb{R}$

4.12 on Classifiers

We start by defininining a predictor as follows:

Definition 4.12.0.1. For a probability space Ω and a measurable space $\mathcal{O}$. A cost is a function

$$C : \mathrm{Hom}(\Omega, \mathcal{O}) \longrightarrow \mathrm{Hom}(\Omega, \mathbb{R})$$

Let $f \leq g \iff (f) \leq (g)$. This defines a pseudo-order. We sat that f is a predictors if its equivalence class $\bar{f}$ is maximal.

Usually one builds from the data of a training set, which is a map $\tau : \mathcal{T} \longrightarrow \mathcal{O}$ where $\mathcal{T} \subset \Omega$

4.13 Naive Bayes

Definition 4.13.0.1. We say that a Bayesian model $\Theta \times \Sigma$ is naive if $P(-|\theta)$ is given by a naieve bayes network.

4.14 the Naive Bayes Classifier

In this section, we consider the following classification problem: assume we are given a probability space partitioned into a finite set of outcomes (which we call authors here) $O_k \subset \mathcal{O}$. We are given a set of words D and let $\Omega = D^n$ assume given a feature set $\mathcal{F} \subset \Omega \longrightarrow \mathcal{O}$ which maps each text to its author. By abuse of notation for an author, we let $P(O_k) = P(\tau^{-1}(O_k))$ and for a word $\omega \in \Omega$, we let $P(\omega) \overset{\text{def}}{=} P(\cup_i \pi_i^{-1}(\omega))$ (where π_k projects a text to its k-th word) Then we define the cost as

$$\mathrm{Hom}(\Omega, \mathcal{O}) \longrightarrow \mathrm{Hom}(\Omega, \mathbb{R}) : f \longrightarrow P(f(\omega) \cap \cap_i \pi_i(\omega))$$

We say that the problem is naive if the following condition is met for a text $omega \in \Omega$ consisting of words $(\omega_1, \ldots \omega_n)$, we have

$$P(\omega_1 | \omega_2 \ldots \omega_n) = P(\omega_1)$$

Then the predictor associated to this problem is simply given by

$$NB(\omega) = \max_k P(O_k \cap \pi_1(\omega) \cap \ldots \pi_n(\omega))$$

It can be easily checked assuming naivity that

$$NB(\omega) = \max_k \left(P(\pi_1(\omega)|O_1) \cdot \ldots \cdot P(\pi_n(\omega)|O_k) P(O_k) \right)$$

Example 4.14.0.1. *Assume we have a set of emails from John and Sarah (split 50/50). Those emails contain the words brown fox jumps in the following way: for john we have* $(0.1, 0.3, 0.1)$ *and Sarah* $(0.2, 0.5, 0.1)$ *We wish to know who wrote the word Brown fox. Assume naivity (wish is realistic in this scenario), we have* $NB_{John} = 0, 1 \cdot 0.3 \cdots 0.5$ *whereas* $NB_{Sarah} = 0.2 \cdot 0.5 \ldots 0.5$. *So Sarah wrote it.*

Part III

Supervised Learners

Chapter 5

the General Theory

5.1 Defining supervised learners

We begin our study of machine learning by suggesting a definition for supervised learners.
Recall that supervised learning (roughly) corresponds to the following paradigm: one is given *data* which consists of *features* and their corresponding *labels*. A supervised learner now assigns to any new feature a new label in a way that matches the given data as well as possible.

Let's look at this idea in a little more detail:
To begin, we denote the sets of features and labels by $\mathfrak{X}$ and $\mathfrak{y}$ respectively. We'll also define the *dataspace* $\mathfrak{D}$, which consists of a set of possible datasets, each of which is a finite subset of $\mathfrak{X} \times \mathfrak{y}$ (ie the given features with their assigned labels). Third, we introduce a *hypothesis space* $\mathfrak{H} \subset \mathfrak{y}^{\mathfrak{X}}$ which contains all the possible ways we'll want to assign a label to a new feature.
A supervised learner now assigns a choice a hypothesis $h_\Delta \in \mathfrak{H}$ given a dataset $\Delta \in \mathfrak{D}$ in an *optimal* way. In other words, we wish to construct an assignment from the dataspace to the hypothesis space

$$h : \mathfrak{D} \longrightarrow \mathfrak{H} : \Delta \mapsto h_\Delta$$

such that h_Δ *fits* the data Δ optimally.
To formalize this optimality condition, we introduce a cost function which assigns a real number given any choice of data and hypothesis:

$$c : \mathfrak{D} \times \mathfrak{H} \longrightarrow \mathbb{R} : (\Delta, h) \mapsto c(\Delta, h)$$

The idea that the choice h_Δ of hypothesis fits the data best is now translated into h_Δ minimizing the cost:

$$c(\Delta, h_\Delta) = \min_{h \in \mathfrak{H}} c(\Delta, h) \tag{5.1}$$

Leading us to the following definition:

Definition 5.1.0.1. A supervised learner $\mathfrak{L}$ (or simply learner) is a tuple $(\mathfrak{X}, \mathfrak{y}, \mathfrak{D}, \mathfrak{H}, c, h)$ where $\mathfrak{D}$ consists of finite subsets of $\mathfrak{X} \times \mathfrak{y}$, $\mathfrak{H} \subset \mathfrak{y}^{\mathfrak{X}}$ and $h : \mathfrak{H} \longrightarrow \mathfrak{D}$, $c : \mathfrak{D} \times \mathfrak{H} \longrightarrow \mathbb{R}$ are functions satisfying the learning condition

$$c(\Delta, h_\Delta) = \min_{h \in \mathfrak{H}} c(\Delta, h)$$

We say that $\mathfrak{L}$ has *learned* the hypothesis h_Δ from the data Δ.
The functions $c_\Delta \stackrel{\text{def}}{=} c(\Delta, -) : \mathfrak{H} \longrightarrow \mathbb{R}$ are the *cost functions* of $\mathfrak{L}$

To make our exposition a little more transparent we'll introduce a bit of notation:

Notation 1. For any dataset $\Delta \subset \mathfrak{X} \times \mathfrak{y}$, we define $\Delta_{\mathfrak{X}} \stackrel{\text{def}}{=} \left\{(x)_{(x,y)\in\Delta}\right\}$ and define $\Delta_{\mathfrak{y}}$ similarly.
For a hypothesis $h \in \mathfrak{H}$, we let $h(\Delta_{\mathfrak{X}}) = \left\{h(x)_{(x,y)\in\Delta}\right\}$

In the rest of our discussion, we will encounter a few interesting properties that a learner can possess:
For one, it will be convenient to study learners where the cost function really only depends on the values of the hypothesis on the features of the dataset as follows:

Definition 5.1.0.2. A learner $\mathfrak{L}$ is *regular* if for any hypotheses $h, k \in \mathfrak{H}$ and any dataset $\Delta \in \mathfrak{D}$

$$h(\Delta_{\mathfrak{X}}) = k(\Delta_{\mathfrak{X}}) \implies c(\Delta, h) = c(\Delta, k)$$

Additionally, it will be convenient to give a specific name to learners where the function h_Δ is unique:

Definition 5.1.0.3. We say that a learner is *sharp* if $h_\Delta = \operatorname{argmin}_{h\in\mathfrak{H}} c(\Delta, h)$ for any dataset $\Delta \in \mathfrak{D}$.
In particular h_Δ is fully determined by the cost function c

The first few chapters in this book are dedicated to proving how some of the main learning algorithms that are being used today can indeed be interpreted in the context of Definition ??. Along the way we shall give a clean interpretation of some of these algorithms and build on the theory of learners just described...

5.2 Trainers

5.2.1. Generalities Throughout, we consider a supervised learner $\mathfrak{L}$ whose hypothesis space (see 5.1.0.1) $\mathfrak{H}$ is a first countable topological space and denote by $\operatorname{Conv}(\mathfrak{H})$, the set of convergent sequences in $\mathfrak{H}$.

Definition 5.2.1.1. Let $\mathfrak{L}$ be a learner. A *trainer* consists of a relation

$$\tau \in \mathfrak{D} \times \operatorname{Conv}(\mathfrak{H})$$

such that $\lim(h_i)_i = h_\Delta$ if $(\Delta, (h_i)_i) \in \tau$

5.2.2. An example: Gradient Descent

5.2.3. Boosting We begin with a Euclidean learner (ie where $\mathfrak{H}$ is a Euclidean space.)
The idea of boosting is to consider a subset $\mathfrak{W} \subset \mathfrak{H}$ of the hypothesis space and train them to compute the learned hypothesis..

Definition 5.2.3.1. Let $\mathfrak{L}$ be a learner and let $\mathfrak{W} \subset \mathfrak{H}$ be a set of so-called *weak hypotheses*.
A boost for the dataset Δ consists of a sequence of learners h_i satisfying

$$h_{i+1} \stackrel{\text{def}}{=} h_i + \operatorname{argmin}_{w\in\mathfrak{W}} \Big(c(\Delta, h_i + w)\Big)$$

We hope that the following theorem is true

Theorem 5.2.3.2. *Let $\mathfrak{L}$ be a Euclidean learner and consider $\mathfrak{W} \subset \mathfrak{H}$.*
Let $\Delta \in \mathfrak{D}$ and let $(h_i)_{i\in\mathbb{N}}$ be a boost for Δ. Then

- $(h_i)_{i\in\mathbb{N}}$ *converges*
- $\lim h_i = h_\Delta$

5.3 Accuracy

Let $\mathfrak{y}$ be a set. Recall that $\mathrm{FRel}(\mathfrak{y})$ is the set of all finite subsests of $\mathfrak{y} \times \mathfrak{y}$.

Definition 5.3.0.1. An accuracy on a set $\mathfrak{y}$ is a map of the form

$$\alpha : \mathrm{FRel}(\mathfrak{y}) \longrightarrow \mathbb{R}.$$

Definition 5.3.0.2. Given a supervised learner $\mathcal{L}$ and dataset $\Delta \in \mathfrak{D}$. We choose a partition $\Delta \stackrel{\text{def}}{=} \Delta_{\text{train}} \coprod \Delta_{\text{test}}$ (*a train-test-split*).
For an accuracy α on the target set $\mathfrak{y}$, we call the accuracy of $\mathcal{L}$ given the dataset Δ the number

$$\alpha_{\mathcal{L}}(\Delta) \stackrel{\text{def}}{=} \alpha\left(\left\{ \left(h_{\Delta_{\text{train}}}(x), y\right) \right) \,\middle|\, (x,y) \in \Delta_{\text{test}} \right\} \right).$$

More generally, we can define k-fold accuracy, by choosing an averaging function:

$$\mathrm{avg} : \mathbb{R}^k \longrightarrow \mathbb{R}$$

and performs the same operation k times:

Definition 5.3.0.3. Given a supervised learner $\mathcal{L}$ and a dataset $\Delta \in \mathfrak{D}$, we choose a partition $\Delta = \Delta_1 \coprod \ldots \coprod \Delta_k$.
For an accuracy α on the target set $\mathfrak{y}$, we call the accuracy of $\mathcal{L}$ given the dataset Δ the number

$$\alpha_{\mathcal{L}}(\Delta) \stackrel{\text{def}}{=} \mathrm{avg}\left(\alpha\big(\left\{ \left(h_{\Delta \setminus \Delta_i}(x), y\right) \right) \,\middle|\, (x,y) \in \Delta_i \right\} \big) \right)$$

5.3.1. Accuracy of binary classifiers

Recall that a binary classifier is a supervised learner where $\mathfrak{y} \stackrel{\text{def}}{=} \{0,1\}$.
In this setting, there are a few natural choices of accuracies. To efine them, we'll fix a relation $\mathcal{R} \in \mathrm{FRel}(\mathfrak{y})$ and define

Definition 5.3.1.1. We define the following accuracies:

- $\mathrm{TP} = \mathrm{card}\left(\left\{ (y_{\text{true}}, y_{\text{pred}}) \in \mathcal{R} \,|\, y_{\text{true}} = y_{\text{pred}} = 1 \right\} \right)$
- $\mathrm{TN} = \mathrm{card}\left(\left\{ (y_{\text{true}}, y_{\text{pred}}) \in \mathcal{R} \,|\, y_{\text{true}} = y_{\text{pred}} = 0 \right\} \right)$
- $\mathrm{FP} = \mathrm{card}\left(\left\{ (y_{\text{true}}, y_{\text{pred}}) \in \mathcal{R} \,|\, y_{\text{true}} = 0,\, y_{\text{pred}} = 1 \right\} \right)$
- $\mathrm{TN} = \mathrm{card}\left(\left\{ (y_{\text{true}}, y_{\text{pred}}) \in \mathcal{R} \,|\, y_{\text{true}} = 1,\, y_{\text{pred}} = 0 \right\} \right)$

A great way to summarize this information is through precision, recall and the F_β-score:

Definition 5.3.1.2. Let $\mathfrak{L}$ be a binary classifier.
The recall is given by

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

(intuitively, how good is the classifier at does detecting positives?).
The precision is given by

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

(intuitively, how reliable is a positive in the classifier?) We combine both by taking the harmonic mean of r and $\beta^2 \cdot p$

$$F_\beta = \frac{(1+\beta^2)pr}{\beta^2 p + r}$$

5.3.2. Accuracy of trainers

Chapter 6

Linear learners

6.1 Generalities

We begin our study of learners with one of the most ubiquitous learning algorithms: *linear regression.*
in this context, we endow the label space $\mathfrak{y}$ with the structure of a finite-dimensional inner product space. $\mathfrak{y}$ is thus equipped with a norm in particular. Before we continue our study of linear regression we recall the following standard consructions of inner product spaces:

Lemma 6.1.0.1. *Let V and W be inner product spaces with orthonormal bases $\left(v_1, \dots v_m\right) \in V$ and $\left(w_1, \dots, w_n\right) \subset W$. Then*

1. *$V^* \stackrel{\text{def}}{=} \text{Hom}_{\mathbb{R}}(V, \mathbb{R})$ is an inner product space with $\langle f, g\rangle \stackrel{\text{def}}{=} \langle v, v'\rangle$ if $\langle v, -\rangle = f$ and $\langle -, v'\rangle = g$ and orthonormal basis given by the maps $\left(\langle v_1, -\rangle, \dots, \langle v_m, -\rangle \right)$*
2. *$V \times W$ is an inner product space with $\langle (v, w), (v', w')\rangle \stackrel{\text{def}}{=} \langle v, w\rangle + \langle v', w'\rangle$ and orthonormal basis $\left((v_i, w_j) \right)_{i,j}$*
3. *$V \otimes W$ is an inner product space with $\langle v \otimes w, v' \otimes w\rangle \stackrel{\text{def}}{=} \langle v, w\rangle \cdot \langle v', w'\rangle$ and orthonormal basis $\left((v_i \otimes w_j) \right)_{i,j}$*
4. *$\text{Hom}(V, W)$ is an inner product space with an orthonormal basis given by maps $e_{i,j}$ defined as*

$$e_{i,j}(v_k) = \begin{cases} 0, & \text{if } k \neq i \\ w_j, & \text{otherwise} \end{cases} \tag{6.1}$$

Proof. Items (1) through (3) consist of routine calculations. To show item (4), recall that the function

$$\iota : V^* \otimes W \longrightarrow \text{Hom}_{\mathbb{R}}(V, W)$$

which assigns to $f \otimes w$ the linear map $\iota_{f\otimes w}(v) \stackrel{\text{def}}{=} f(v) \cdot w$ is an isomorphism. Now, items (1) and (3) together imply that $V^* \otimes W$ is indeed an inner product space with orthonormal basis $\left\{ \langle v_i, -\rangle \otimes w_j \right\}_{i,j}$. Now simply note that the map associated to $\langle v_i, -\rangle \otimes w_j$ under the function ι is exactly $e_{i,j}$ ∎

Returning to our discussion above, we consider a finite-dimensional inner product space $\mathfrak{y}$ of labels and $\mathfrak{X}$ any set of features. Now, for any finite dataset $\Delta \subset \mathfrak{X} \times \mathfrak{y}$, the space $\mathfrak{y}^{\Delta}$ in turn carries an inner product by Lemma 6.1.0.1, so that for any map $f \in \mathfrak{y}^{\mathfrak{X}}$, we can define a cost as follows:

$$c(\Delta, f) = ||y - f(x)||_{\mathfrak{y}^{\Delta}} \stackrel{\text{def}}{=} \sqrt{\sum_{\Delta} ||y - f(x)||^2}$$

The main result of this chapter will be to show that this cost indeed describes a learner under the right conditions. To this end, we will make the following definition:

Definition 6.1.0.2. Let $\mathfrak{H} \subset \mathfrak{y}^{\mathfrak{X}}$ and $\Delta \subset \mathfrak{X} \times \mathfrak{y}$. Then we say that Δ separates $\mathfrak{H}$ if for any $f, g \in \mathfrak{H}$:

$$f|_{\Delta_{\mathfrak{X}}} = g|_{\Delta_{\mathfrak{X}}} \implies f = g$$

The main result of this chapter is:

Theorem 6.1.0.3. *Let $\mathfrak{y}$ be a finite-dimensional inner product space, $\mathfrak{X}$ any set and let $\mathfrak{H} \subset \mathfrak{y}^{\mathfrak{X}}$ be a finite-dimensional subspace of $\mathfrak{y}^{\mathfrak{X}}$. Assume that any $\Delta \in \mathfrak{D}$ separates $\mathfrak{H}$ and let $c(\Delta, f) = ||y - f(x)||_{\mathfrak{y}^{\Delta}}$. Then $(\mathfrak{X}, \mathfrak{y}, \mathfrak{D}, \mathfrak{H}, c)$ defines a sharp learner (as in Definition 5.1.0.3)*

Definition 6.1.0.4. We say that a learner is linear if it is of the above form

One particular type of linear learner will be of particular interest, as it allows us to be a little more explicit with certain constructions: if we assume that $\mathfrak{X}$ itself carries the structure of a finite-dimensional inner product space, and put $\mathfrak{H} \stackrel{\text{def}}{=} \text{Hom}_{\mathbb{R}}(\mathfrak{X}, \mathfrak{y})$ as well as consider finite datasets $\Delta \subset \mathfrak{X} \times \mathfrak{y}$ such that $\Delta_{\mathfrak{X}}$ spans the whole of $\mathfrak{X}$ (where we recall our use of the notation 1), then it's easy to see that the conditions of Theorem 6.1.0.3 are satisfied so that we indeed obtain an example of a linear learner:

Definition 6.1.0.5. A *Euclidean learner* is a sharp (linear) learner where $\mathfrak{X}, \mathfrak{y}$ are finite-dimensional inner product spaces,

$$\mathfrak{D} = \left\{ \Delta \subset \mathfrak{X} \times \mathfrak{y} \;\middle|\; |\Delta| \neq \infty, \text{ and } \text{span}(\Delta_{\mathfrak{X}}) = \mathfrak{X} \right\}, \; \mathfrak{H} = \text{Hom}_{\mathbb{R}}(\mathfrak{X}, \mathfrak{y})$$

and

$$c : \mathfrak{D} \times \mathfrak{H} \longrightarrow \mathbb{R} : (\Delta, f) \mapsto ||\big(y - f(x)\big)_{(x,y) \in \Delta}||_{\mathfrak{y}^{\Delta}}$$

Chapter 7

Statistical leaners

Chapter 8

Bayesian learners

Chapter 9

k-Nearest Neighbors

We begin by picking a parameter $k \in \mathbb{N}$.
Let $\mathfrak{X}$ be a metric space and $\mathfrak{y}$ a finite set. Let $\mathfrak{D}$ be the dataspace consisting of finite subsets $\Delta \subset \mathfrak{X} \times \mathfrak{y}$ such that each Δ is a function.
For $x \in \Delta$ Let us define kNN(x) (the set of *k-nearest neighbors*) as the set of k points in $\Delta \in \mathfrak{D}$ who are closest to x in the metric space $\mathfrak{X}$.
We now consider the hypothesis space $\mathfrak{H} = \mathfrak{y}^{\mathfrak{X}}$ and look at the following cost function

$$c : \mathfrak{D} \times \mathfrak{H} \longrightarrow \mathbb{R} : (\Delta, f) \longrightarrow \sum_{(x,y)\in\Delta} |k - \text{nearest neighbors of } x \text{ in class } y|$$

We now define for $\Delta \in \mathfrak{D}$ the learned hypothesis $h_\Delta \in \mathfrak{H}$ to be

$$h_\Delta(x) = \text{argmax}_{y\in\mathfrak{y}} |k - \text{nearest neighbors of } x \text{ in class } y|$$

Then we trivially have:

Theorem 9.0.0.1. *The above defines a supervised learner, called k-nearest neighbors.*

Chapter 10

Support Vector Machines

10.1 the Linearly Separable Case

Let V be a normed space and $f : V \longrightarrow \{-1, 1\}$ a function.

Definition 10.1.0.1. A support vector machine is a hyperplane $H \subset V$ which maximizes the tuple

$$(d(H, f^1(-1), d(H, f^{-1}(1)))$$

Convention 10.1.0.2. *We will always denote $f^{-1}(1)$ (resp. $f^{-1}(-1)$) by V_1 (resp. V_{-1}).*

Theorem 10.1.0.3. *Assume that V_{-1}, V_1 are linearly separable. Then there exists a unique support vector machine*

Lemma 10.1.0.4. *Let $H_1 = a + V, H_2 = b + V$ parallell hyperplanes. and $o \in V^\perp$*

$$d(H_1, H_2) = \frac{|\langle o, b - a\rangle|}{||o||}$$

Proof. Since $-b$ is an isometry, we can assume $b = 0$. Now, for a vector $x \in a + V$, we have that $d(a + v, V)$ is the distance to the orthogonal projection of x on V. Since by definition, we have $x = \pi_V(x) + \pi_{V^\perp}(x)$, $d(x, V) = ||\pi_{V^\perp}(x)|| = \pi_o(x)$.
Now the projection of x onto the line with direction o is

$$\frac{|\langle o, x\rangle|}{||o||} = \frac{|\langle o, a\rangle|}{||o||}$$

hence, the claim ∎

Definition 10.1.0.5. A linear separator is an affine function $V \longrightarrow \mathbb{R}$ such that $f \cdot \sigma \geq 1$

Since σ is affine, the fibers $\sigma^{-1}(1)$ and $\sigma^1(-1)$ are parallel hyperplanes. This leads to the following definition

Definition 10.1.0.6. A support vector machine is a linear separator σ which maximizes

$$d(\sigma^{-1}(1), \sigma^{-1}(-1))$$

Theorem 10.1.0.7. *A support vector machine is equivalent to the data of a couple of vectors (a, w)such that $\langle w, x_i + a\rangle \geq 1$ and $||w||$ is minimal for this condition.*

Proof. Since σ is affine, the kernel is a hyperplane $a + V$. let $win V^\perp$. Then ∎

Chapter 11

Neural Networks

We let X be a topological space

Definition 11.0.0.1. A perceptron is a quiver with a marked node, n ingoing vertices and 1 outgoing vertex, together with a continous function $Hom^n(X, X)$

Chapter 12

Errors in Classifiers

12.1 Bias

Intuitively, bias occurs when the model isn't complex enough to fit the data

12.2 Learning Curves

Recall that a learning scheme consists of spaces X and Y together with subspaces $H \subset Hom(X, Y)$, $D \subset X \times Y$ and a decomposition $D = D_{\text{Train}} \cup D_{\text{Test}}$.
Assume we have an error function $E : \mathcal{H} \times \mathcal{P}(X) \longrightarrow \mathbb{R})$. Assume that X_{Train} and X_{Test} are parametrized by some functions θ_{Train} and θ_{Test}. Then we define the learning curves as

$$LC_{\text{Train}} : \mathbb{R} \longrightarrow \mathbb{R} : x \mapsto E(h, \theta^{-1}_{\text{Train}}(]-\infty, x])))$$
$$LC_{\text{Train}} : \mathbb{R} \longrightarrow \mathbb{R} : x \mapsto E(h, \theta^{-1}_{\text{Test}}(]-\infty, x])))$$

Chapter 13

Machine Learning

13.1 Basic Definition

Definition 13.1.0.1. We consider $\mathfrak{X}$ and $\mathfrak{y}$, the set of *features* and *labels*. Additionally, we define $\mathfrak{D} \subset \mathrm{Rel}(\mathfrak{X}, \mathfrak{y})$, the *dataspace*. and $\mathfrak{H} \subset \mathrm{Hom}(\mathfrak{X}, \mathfrak{y})$ the hypothesis space.
A regressor is a function

$$\rho : \mathfrak{D} \longrightarrow \mathfrak{H}$$

called the *regressor* (or *classifier*) if $\mathfrak{y}$ is finite) together with a *cost function*

$$\eta : \mathrm{Rel}(\mathfrak{y}) \longrightarrow \mathbb{R}^+$$

such that for each $\Delta \in \mathfrak{D}$:

$$\min_{f \in \mathfrak{H}} \eta(\Delta_f) = \rho_\Delta$$

where $\Delta_f \stackrel{\text{def}}{=} \{(f(x), y) | (x, y) \in \Delta\}$

Example 13.1.0.2. *Assume that $\mathfrak{X} = V$ is a real inner product space and $\mathfrak{y} = \mathbb{R}$. and let $\mathfrak{D} = \mathfrak{H} = V^*$ For a relation $\mathcal{R} \in \mathrm{Rel}(\mathfrak{y})$), we define*

$$\eta(\mathcal{R}) = \sqrt{\sum_{(a,b) \in rR} ||a - b||^2}$$

The associated ρ which exists and is unique is the linear regressor.

13.1.1. MLE Regressors Assume that $\mathfrak{y} = \mathbb{R}$ and that $\mathfrak{H} \subset \mathrm{Hom}(\mathfrak{X}, \mathbb{R})$ has the structure of a Markov kernel

$$\mathfrak{H} \stackrel{P}{\Longrightarrow} \mathbb{R}$$

which satisfies the MLE hypothesis.

Definition 13.1.1.1. We will fix a set $\mathfrak{D} \stackrel{\text{def}}{=} \mathfrak{X} \times \mathfrak{y}$ called the dataspace consisting of *the example space* and *feature space*. We let $F(\mathcal{D})$ denote the set of functions $X \longrightarrow \mathfrak{y}$ where $X \subset \mathcal{X}$ is a finite subset. A learning scheme is a function

$$F(\mathcal{D}) \times \mathfrak{X} \longrightarrow \mathfrak{y}$$

Called the *regressor*, If $\mathfrak{y}$ is finite, we call it a classifier, otherwise we call it a regressor.

Example 13.1.1.2. *Assume $\mathfrak{X}$ is a (real) inner product space. Then the linear regressor is defined by sending (D, x) to $f(x)$ where*

$$\mathrm{argmin}_{f \in \mathfrak{X}^*} ||y - x||$$

Here the norm is taken in $\mathfrak{X}^{\oplus |D|}$ and $(x, y) \in D$. We will call this a linear regressor.

There is another interpretation of the above regressor:

Example 13.1.1.3. *WE let $\mathfrak{X}$ be a real inner space and $\mathfrak{y} = \mathbb{R}$. For each n, We define a statistical model*

$$(\mathfrak{X}^*, \mathcal{B}^*) \implies (\mathfrak{D}, \mathcal{B})$$

by defining for $\alpha \in \mathfrak{X}^*$ and

$$f_\alpha((x,y)) = e^{\left(\frac{y-\alpha(x)}{\sigma}\right)^2}$$

This induces a canonical statistical model $(\mathfrak{X}^*, \mathcal{B}^*) \implies (\mathfrak{D}^n, \mathcal{B})$ For a fixed $D \subset \mathfrak{D}^n$, we let α be the associated maximum likelihood estimator. Then α coincides with the linear regressor.

Example 13.1.1.4. *Let $\mathfrak{D} = (\mathfrak{X}, \mathfrak{y})$. A decision tree for $\mathcal{D}$ is a graph such that*

- *The graph is dichotomic*
- *the nodes are function $\mathfrak{X} \longrightarrow \mathbb{Z}_2$ (called attributes*
- *There is a bijection between the final nodes and $\mathfrak{y}$*

If $\mathbb{T}$ is a decision tree, we associate a function $\lambda_{\mathbb{T}} : \mathfrak{X} \longrightarrow \mathfrak{y}$ by sending x to the final node. The scheme ID_3 associates a decision tree to any $F(\mathfrak{D}$ yielding a learning scheme:

$$F(\mathfrak{D}) \times \mathfrak{X} \longrightarrow \mathfrak{y} : (D, x) \mapsto \lambda_{\mathrm{ID}_3(D)}(x)$$

Example 13.1.1.5. *We again assume that $\mathfrak{X}$ is an inner product space and $\mathfrak{Y} = \{-1, 1\}$. We say that $f \in F(fD)$ can be separated if there exists an affine function $f \in \mathrm{aff}(\mathfrak{X}, \mathbb{R})$ $f \cdot \sigma \geq 1$. If f cannot be separated, we let $\mathrm{SVM}(f, x) = 0$, If f can be separated, we consider $SVM_f \stackrel{\mathrm{def}}{=} \mathrm{argmin}_{\sigma \in \mathrm{sep}(f)} d(\sigma^{-1}(1), \sigma^{-1}(1))$ and let*

$$\mathrm{SVM} : F(\mathfrak{D}) \times \mathcal{X} \longrightarrow \mathfrak{y} : (f, x) \longrightarrow \mathrm{SVM}_f(x)$$

This is the hard margin Support vector machine.

Example 13.1.1.6. *Assume we have a Bayesian model on Π on $\mathfrak{y} \times \mathfrak{X}$*

13.2 Classification

In machine learning, start with a space of *instances* and try and find function called the *candidate concept* which maps to a finite space. The image space being the target concepts. The concept comes from a class of functions, called the hypothesis class. Part of the data includes 2 subset of the instances $\times$ targets, the sample (or training set) and testing set.

13.3 Decision Trees

Recall:

Definition 13.3.0.1. A tree is a graph where two nodes are connected by exactly one path. Equivalently, a tree is an acyclic connected graph.

Definition 13.3.0.2. A decision tree is a tree with a start node wich has two edges. A subset of end nodes wich have one edge whereas all other nodes have 3.

We can construct a partition on decision trees as follows:

- the start node is equivalent to itself call this level 0
- two nodes are in level i if there is a common node in level $i-1$ connecting them.

Lemma 13.3.0.3. *Being in the same level is an equivalence relation*

Definition 13.3.0.4. A truth coloring is a binary coloring on a decision tree such that for each node n at level i, the two edges out of n to level $i+1$ are colored true and false

Lemma 13.3.0.5. *A truth coloring exists*

There's another way of thinking about decision trees. Recall that

Definition 13.3.0.6. A logical statement in n variables is simply a map $\mathbb{Z}_2^n \longrightarrow \mathbb{Z}_2$

Theorem 13.3.0.7. *There is a 1 to 1 correspondence between logical statements in n variables and decision trees on $n-1$ levels with truth function*

Remark 13.3.0.8. *note that the amount of nodes coming from a logcial expression can be rather large as it equals* 2^{2^n} $(|\operatorname{Hom}(\mathbb{Z}_2^n, \mathbb{Z}_2)|)$

This leads to the ID3 algorithm for machine learning:

- pick a "best attribute" A
- assign A as a decision attribute (logical atom) for a node
- for each value of A creat a descendant
- sort the training axamples to the leaves
- if examples perfectly classified, then $STOP$
- else, iterate over each leaf.

What do we mean by best attribute?

Definition 13.3.0.9. The information gain over S and A is given by

$$\operatorname{Entropy}(S) - \sum_v \frac{|S_v|}{|S|} \operatorname{Entropy}(S_v)$$

It's not clear whether this algorithm will indeed stop, in fact it doesnt necessary. So we assume that some data points are 'noisy' and look for the best hypothesis. by avoiding overfitting (which intuitively leads to a tree which is too big, i.e. there are smaller trees which yield the same classification). This is done by 'pruning' back the tree.

13.3.1. Bias

Bias has two aspects:

- restriction bias, i.e. the hypothesis set (in this cas the collection of decision trees)
- the preference bias: the hypothesis form the hypothesis set we prefer.

ID3 prefers:

- good splits at the top of tree
- correct over incorrect trees
- shorter trees

Part IV

Unsupervised Learners

Chapter 14

Generalities

14.1 the Definition

Definition 14.1.0.1. A cluster set consists of a surjective map $\pi : \mathfrak{U} \longrightarrow \mathcal{B}$ of sets, (sometimes together with a basepoint $p \in \mathcal{B}$). The fibers $\pi^{-1}(b)$ will be denoted $\mathfrak{U}^b$.

Definition 14.1.0.2. An unsupervised learner consists of a set of features $\mathfrak{X}$ and labels $\mathfrak{y}$, a clusterset $\mathfrak{U} \longrightarrow \mathcal{B}$, dataspace $\mathfrak{D} \subset \coprod \mathfrak{X}^i$ and hypothesis space $\mathfrak{H} \subset \mathfrak{y}^{\mathfrak{X}}$ together with maps

$$l : \mathfrak{X} \times \mathfrak{U} \longrightarrow \mathfrak{y},\ m : \mathcal{B} \times \mathfrak{D} \longrightarrow \mathfrak{U},\ \text{and}\ c : \mathfrak{D} \times \mathfrak{U} \longrightarrow \mathbb{R}$$

called the labeling -, cost -, and method maps. [1] We further assume the method m is minimal in the sense that for each dataset $\Delta \in \mathfrak{D}$ and $b \in \mathcal{B}$, we have

$$\min_{\mathfrak{U}^b} c\big(\Delta, -\big) = c\big(\Delta, m(b, \Delta)\big)$$

We can then define the *learned hypothesis* (which intuitively labels each feature optimally) as

$$\mathcal{B} \times \mathfrak{D} \longrightarrow \mathfrak{H} : (b, \Delta) \longrightarrow h_{b,\Delta} \text{ where } h_{b,\Delta}\big(x\big) \stackrel{\text{def}}{=} l\big(x, m(b, \Delta)\big)$$

We finally require that $h_\Delta \in \mathfrak{H}$

[1] we say that the feature obtains a label through the cluster by l, each dataset associates a b-cluster through the method m and this comes with a cost c

Chapter 15

k-Means

Chapter 16

Tree Clustering

Part V

Preprocessing

Chapter 17

Change of Basis

17.1 PCA

17.1.1. Tuning the Parameter