Introduction to Machine Learning (CS 771A, IIT Kanpur)

Course Notes and Exercises

Suggested Citation: P. Kar. Introduction to Machine Learning (CS 771A, IIT Kanpur), Course Notes and Exercises, 2019.

Purushottam Kar
IIT Kanpur
purushot@cse.iitk.ac.in

IIT Kanpur

Contents

1 Support Vector Machines **2**
1.1 Derivation of the CSVM Dual 2

2 Probabilistic Learning Methods **6**
2.1 Exercises 7

3 Learning with Latent Variables **8**
3.1 Exercises 9

4 Learning with Kernels **11**
4.1 Kernel PCA Derivation 11
4.2 Kernel kNN Done Two Ways 12
4.3 Exercises 13

Acknowledgements **15**

Appendices **16**

A Calculus Refresher **17**
A.1 Extrema 17
A.2 Derivatives 18
A.3 Second Derivative 19
A.4 Stationary Points 19
A.5 Useful Rules for Calculating Derivatives 20
A.6 Multivariate Functions 21
A.7 Visualizing Multivariate Derivatives 23
A.8 Useful Rules for Calculating Multivariate Derivatives 25
A.9 Subdifferential Calculus 28
A.10 Exercises 30

B Convex Analysis Refresher **32**
B.1 Convex Set 32
B.2 Convex Functions 34
B.3 Operations with Convex Functions 36
B.4 Exercises 39

C Probability Theory Refresher **41**
C.1 Empirical Median 41
C.2 Exercises 46

D Linear Algebra Refresher **47**
D.1 Exercises 49

References **53**

Introduction to Machine Learning (CS 771A, IIT Kanpur)

Purushottam Kar[1]*

[1]*IIT Kanpur; purushot@cse.iitk.ac.in*

ABSTRACT

Machine Learning is the art and science of designing algorithms that can learn patterns and concepts from data to modify their own behavior without being explicitly programmed to do so. This monograph is intended to accompany a course on an introduction to the design of machine learning algorithms with a modern outlook. Some of the topics covered herein are *Preliminaries* (multivariate calculus, linear algebra, probability theory), *Supervised Learning* (local/proximity-based methods, learning by function approximation, learning by probabilistic modeling), *Unsupervised Learning* (discriminative models, generative models), practical aspects of machine learning, and additional topics.

Although the monograph will strive to be self contained and revisit basic tools such as calculus, probability, and linear algebra, the reader is advised to not completely rely on these refresher discussions but also refer to a standard textbook on these topics.

*The contents of this monograph were developed as a part of successive offerings of various machine learning related courses at IIT Kanpur.

1

Support Vector Machines

1.1 Derivation of the CSVM Dual

Let us recall the CSVM primal problem

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\cdot\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y^i(\mathbf{w}^\top\mathbf{x}^i + b) \geq 1-\xi_i, && \text{for i = 1,...,n} \\ & \xi_i \geq 0, && \text{for i = 1,...,n} \end{aligned}$$

We follow the usual steps of deriving the dual problem below

1. Step 1 (Convert the problem into conventional form). The problem is already a minimization problem so nothing to be done there. However, we do need to convert the constraints into ≤ 0 type constraints

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\cdot\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & 1-\xi_i - y^i(\mathbf{w}^\top\mathbf{x}^i + b) \leq 0, && \text{for i = 1,...,n} \\ & -\xi_i \leq 0, && \text{for i = 1,...,n} \end{aligned}$$

2. Step 2 (Introducing dual variables/Lagrange multipliers). Since there are two sets of n constraints above, let us introduce $\alpha_1,\ldots,\alpha_n$ for the constraints of the kind $1-\xi_i - y^i(\mathbf{w}^\top\mathbf{x}^i + b) \leq 0$ and $\beta_1,\ldots,\beta_n$ for the constraints of the kind $-\xi_i \leq 0$.

3. Step 3 (Create the Lagrangian). This is easy to do

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\cdot\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1-\xi_i - y^i(\mathbf{w}^\top\mathbf{x}^i + b)) - \sum_{i=1}^{n}\beta_i\xi_i$$

Note that we have $-\beta_i\xi_i$ occurring with a negative sign in the above because the constraints are $-\xi_i \leq 0$ and not $\xi_i \leq 0$.

4. Step 3 (Create the dual problem). This is easy to do as well. Just keep in mind that there are constraints on the dual variable that they must be non-negative. We use the shorthand $\mathbf{x} \geq 0$ to say that all coordinates of the vector $\mathbf{x}$ must be non-negative i.e. $\mathbf{x}_i \geq 0$ for all i.

$$\max_{\boldsymbol{\alpha} \geq 0, \boldsymbol{\beta} \geq 0} \min_{\substack{\mathbf{w} \in \mathbb{R}^d \\ b \in \mathbb{R} \\ \boldsymbol{\xi} \in \mathbb{R}^n}} \left\{ \frac{1}{2} \|\mathbf{w}\|_2^2 + C \cdot \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i (1 - \xi_i - y^i(\mathbf{w}^\top \mathbf{x}^i + b)) - \sum_{i=1}^{n} \beta_i \xi_i \right\}$$

5. Step 4 (Apply first order optimality with respect to all primal variables). Recall that we do this since in the dual problem, there are no more constraints on the primal variables and the Lagrangian is a differentiable function of the primal variables and so the derivatives of the Lagrangian must vanish with respect to all the primal variables.

   (a) Optimality w.r.t $\mathbf{w}$. Setting $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0}$ gives us $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{x}^i$.

   (b) Optimality w.r.t b. Setting $\frac{\partial \mathcal{L}}{\partial b} = 0$ gives us $\sum_{i=1}^{n} \alpha_i y^i = 0$.

   (c) Optimality w.r.t ξ_i. Setting $\frac{\partial \mathcal{L}}{\partial \xi_i} = 0$ gives us $\alpha_i + \beta_i = C$

   The above identities are necessarily true at the optimum, so we take them as constraints in the dual problem. Note that we already have positivity constraints on the dual variables i.e. $\boldsymbol{\alpha}, \boldsymbol{\beta} \geq 0$.

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n} \min_{\substack{\mathbf{w} \in \mathbb{R}^d \\ b \in \mathbb{R} \\ \boldsymbol{\xi} \in \mathbb{R}^n}} \quad \frac{1}{2} \|\mathbf{w}\|_2^2 + C \cdot \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i (1 - \xi_i - y^i(\mathbf{w}^\top \mathbf{x}^i + b)) - \sum_{i=1}^{n} \beta_i \xi_i$$

$$\begin{aligned} \text{s.t.} \quad & \alpha_i \geq 0, \quad \text{for i} = 1,\ldots,\text{n} \\ & \beta_i \geq 0, \quad \text{for i} = 1,\ldots,\text{n} \\ & \mathbf{w} = \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{x}^i \\ & \sum_{i=1}^{n} \alpha_i y^i = 0 \\ & \alpha_i + \beta_i = C \end{aligned}$$

6. Step 5 (Simplify the objective function by *possibly* eliminating primal variables). We can rearrange terms in the objective function as

$$\sum_{i=1}^{n} \alpha_i + \frac{1}{2} \|\mathbf{w}\|_2^2 - \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{w}^\top \mathbf{x}^i - b \sum_{i=1}^{n} \alpha_i y^i + \sum_{i=1}^{n} \xi_i (C - \alpha_i - \beta_i)$$

   Applying $\sum_{i=1}^{n} \alpha_i y^i = 0$ and $\alpha_i + \beta_i = C$ simplifies the objective to

$$\sum_{i=1}^{n} \alpha_i + \frac{1}{2} \|\mathbf{w}\|_2^2 - \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{w}^\top \mathbf{x}^i$$

   Applying $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{x}^i$ tells us that $\sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{w}^\top \mathbf{x}^i = \|\mathbf{w}\|_2^2$ which further simplifies the objective to

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \|\mathbf{w}\|_2^2$$

Applying $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{x}^i$ once more completely eliminates $\mathbf{w}$ from the objective function

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j \left\langle \mathbf{x}^i, \mathbf{x}^j \right\rangle$$

Thus, we obtain the dual problem with primal variables completely eliminated from the objective.

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\beta} \in \mathbb{R}^n} \min_{\mathbf{w} \in \mathbb{R}^d} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j \left\langle \mathbf{x}^i, \mathbf{x}^j \right\rangle \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad \text{for i} = 1,\ldots,\text{n} \\ & \beta_i \geq 0, \quad \text{for i} = 1,\ldots,\text{n} \\ & \mathbf{w} = \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{x}^i \\ & \sum_{i=1}^{n} \alpha_i y^i = 0 \\ & \alpha_i + \beta_i = C \end{aligned}$$

Note that we have removed the variables $b, \boldsymbol{\xi}$ from the optimization problem since they no longer appear anywhere, either in the constraints or the objective. However, we still have a constraint, namely $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y^i \cdot \mathbf{x}^i$, which contains a primal variable. However, since we have already incorporated that constraint while simplifying the objective and the objective is no more linked to the primal variable $\mathbf{w}$ either directly or indirectly, we may safely remove this constraint – we will reuse this constraint again after solving the dual problem to reconstruct the model vector $\mathbf{w}$ using the dual solution $\boldsymbol{\alpha}$. Thus, we have completely removed primal variables from the objective and the constraints so we may as well remove them from consideration altogether

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\beta} \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j \left\langle \mathbf{x}^i, \mathbf{x}^j \right\rangle \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad \text{for i} = 1,\ldots,\text{n} \\ & \beta_i \geq 0, \quad \text{for i} = 1,\ldots,\text{n} \\ & \sum_{i=1}^{n} \alpha_i y^i = 0 \\ & \alpha_i + \beta_i = C \end{aligned}$$

7. Step 5 (Simplify by *possibly* eliminating some dual variables). It turns out that we can even eliminate the dual variables β_i. Note that the β_i variables do not appear in the objective function at all. Instead, the variable β_i appears in just two constraints, namely $\beta_i \geq 0$, $\alpha_i + \beta_i = C$. The second equation gives us $\beta_i = C - \alpha_i$. Putting this into the first equation gives us $\alpha_i \leq C$. This means that the β_i variables are

not required themselves and they were just an indirect way of the dual problem telling us that we must have $\alpha_i \leq C$.

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j \left\langle \mathbf{x}^i, \mathbf{x}^j \right\rangle \\ \text{s.t.} \quad & \alpha_i \in [0, C], \qquad \text{for i} = 1,\ldots,\text{n} \\ & \sum_{i=1}^{n} \alpha_i y^i = 0 \end{aligned}$$

The above is indeed the final form of the dual of the CSVM problem that is used in professional solvers such as `liblinear` and `sklearn`.

We note that the above derivation is by no means an indication of what must happen while deriving the dual problems for other optimization problems. Specifically, we warn the reader about the following

1. A constraint of the form $\alpha_i \geq 0$ will always appear in a dual problem if α_i is a dual variable corresponding to a $\leq$ type constraint (unless of course the α_i variable gets eliminated altogether like we eliminated the β_i variable above). This is because being non-negative is an inherent property of dual variables that correspond to $\leq$ type constraints.

2. However, constraints on dual variables such as $\alpha_i \leq C$ need not always appear in the dual problem. Such constraints do appears if the primal is the CSVM formulation which effectively uses the hinge loss. However, such a constraint on α_i need not appear (or some other funny-looking constraint may appear instead) if we use some other primal formulation, e.g. replace the hinge loss with logistic or squared loss. The reader would have noticed how the constraint $\alpha_i \leq C$ appeared in the above derivation because some other dual variable got eliminated, which is something that need not happen with all optimization problems.

3. In more complicated optimization problems, it may not be possible to eliminate dual variables like we eliminated β_i above. However, if we are lucky, we may be able to eliminate some dual variables and obtain a simpler dual problem, which would then be faster to solve since there are less dual variables about which to worry.

4. In still more complicated optimization problems, it may not be possible to even eliminate all the primal variables. In such cases, we still must apply first order optimality with respect to all primal variables to obtain constraints but we may find ourselves unable to exploit those constraints to remove all the primal variables. If this happens, we will be left with the dual problem looking like a *max-min* problem which we must solve as is. Such max-min problems are called *saddle point optimization* problems. So we would have reduced our dual problem to a saddle point problem. However, in nice cases where we are able to completely remove the primal variables, our dual problem gets reduced to a *maximization* problem.

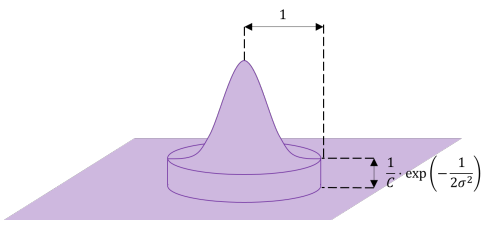
2

Probabilistic Learning Methods

Example 2.1. Consider the following optimization problem that offers a novel way to solve the linear regression problem. In the following $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \mathbb{R}$.

$$\begin{aligned} \hat{\mathbf{w}}_{\text{MAP}} = \underset{\mathbf{w}\in\mathbb{R}^d}{\arg\min} \quad & \|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} (y^i - \left\langle \mathbf{w}, \mathbf{x}^i \right\rangle)^2 \\ \text{s.t.} \quad & \|\mathbf{w}\|_2 \leq 1 \end{aligned}$$

Can we find a likelihood distribution for $\mathbb{P}\left[y^i \,|\, \mathbf{x}^i, \mathbf{w}\right]$ and prior $\mathbb{P}\left[\mathbf{w}\right]$ such that $\hat{\mathbf{w}}_{\text{MAP}}$ becomes the MAP estimate for these choices? This example is interesting since the optimization problem involves an L_2 regularizer, as well as a constraint. In the following, fix any $\sigma > 0$.



The solution for the likelihood function turns out to be straightforward: choose the likelihood density function as $\mathbb{P}\left[y^i \,|\, \mathbf{x}^i, \mathbf{w}\right] = \mathcal{N}(\left\langle \mathbf{w}, \mathbf{x}^i \right\rangle, \sigma^2)$. The prior distribution needs more thought: since the MAP solution is necessarily constrained to the region $\|\mathbf{w}\|_2 \leq 1$, the prior has the responsibility of ensuring that it places zero density outside this region. Moreover, within this region, the prior must enable a regularization that looks like the L_2 regularizer. All this motivate the following choice for the prior distribution:

$$\mathbb{P}\left[\mathbf{w}\right] = \begin{cases} C \cdot \mathcal{N}(\mathbf{0}, \sigma^2 \cdot I) & \text{if } \|\mathbf{w}\|_2 \leq 1 \\ 0 & \text{if } \|\mathbf{w}\|_2 > 1 \end{cases}$$

where $C^{-1} = \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \int_{\|\mathbf{x}\|_2 \leq 1} \exp\left(-\frac{1}{2\sigma^2} \cdot \|\mathbf{x}\|_2^2\right)$ is a normalization constant chosen so that $\mathbb{P}\left[\mathbf{w}\right]$ becomes a proper probability distribution.

It can be verified that for the above choices for the prior and likelihood distributions, $\hat{\mathbf{w}}_{\text{MAP}}$ is indeed the MAP solution.

2.1 Exercises

Exercise 2.1. Recall Vapnik's ϵ-insensitive loss defined as $\ell_\epsilon(y, \hat{y}) = 0$ if $|y - \hat{y}| \leq \epsilon$ and otherwise $\ell_\epsilon(y, \hat{y}) = (|y - \hat{y}| - \epsilon)^2$ where $\hat{y}, y \in \mathbb{R}$. Consider the following optimization problem with $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \mathbb{R}$ and write down a likelihood distribution for $\mathbb{P}\left[y^i \,|\, \mathbf{x}^i, \mathbf{w}\right]$ and prior $\mathbb{P}\left[\mathbf{w}\right]$ such that $\hat{\mathbf{w}}$ is the MAP estimate for your model. Give explicit forms for the density functions but you may omit calculating explicit normalization constants.

$$\hat{\mathbf{w}} = \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^d} \ \|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \ell_\epsilon(y^i, \left\langle \mathbf{w}, \mathbf{x}^i \right\rangle)$$

3

Learning with Latent Variables

Example 3.1. The website flopcart.com has a customer who uses their account to make purchases for their entire family. There are k members in the family, each indexed by a vector $\mathbf{u}^1, \ldots, \mathbf{u}^k \in \mathbb{R}^d$. Each product on flopcart.com is also indexed by a vector $\mathbf{v} \in \mathbb{R}^d$. It is known that the i^{th} member will give the product $\mathbf{v}$, a rating $r = \langle \mathbf{u}^i, \mathbf{v} \rangle + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$. This random noise is generated afresh independently each time the user gives a rating.

The customer has made n purchases with flopcart. At the t-th purchase, the item $\mathbf{v}^t$ was purchased and a rating r^t was given to that product. If a product was purchased multiple times, it was rated those many times as well (with, as mentioned before, the random noise being generated afresh independently each time a rating is given).

However, it is not known which member gave which rating. We have a dataset $\{(\mathbf{v}^t, r^t)\}_{t=1}^n$ with us. Can we design an algorithm to estimate the user vectors corresponding to the k members of the family (and possibly also estimate which user purchased as well as rated which item)?

One way to model this problem is as a mixed regression problem with k-components. Denote $U = [\mathbf{u}^1, \ldots, \mathbf{u}^k] \in \mathbb{R}^{d \times k}, V = [\mathbf{v}^1, \ldots, \mathbf{v}^n] \in \mathbb{R}^{d \times n}, \mathbf{r} = [r^1, \ldots, r^n]^\top \in \mathbb{R}^n$. For each of the data points, $z^t \in [k]$ is a latent variable that denotes the identity of the member who rated the item $\mathbf{v}^t$ and we denote $\mathbf{z} = [z^1, \ldots, z^n] \in [k]^n$.

We choose to use (for the sake of simplicity) a Gaussian likelihood model for the ratings $\mathbb{P}\left[r^t \mid \mathbf{v}^t, z^t, U\right] = \frac{1}{\sqrt{2\pi}} \exp\left(-(r^t - \left\langle \mathbf{u}^{z^t}, \mathbf{v}^t \right\rangle)^2/2\right)$ and assume that members are equally likely to rate items by assuming a uniform prior on the latent variables $\mathbb{P}\left[z^t = i\right] = \frac{1}{k}$ for all $i \in [k]$. We also assume a standard Gaussian prior on the user vectors $\mathbb{P}\left[\mathbf{u}^i\right] = \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2} \left\|\mathbf{u}^i\right\|_2^2\right)$.

Given a setting of the latent variables z^t, we can find the MAP solution for the user vectors as follows

$$U = \arg\max \ \mathbb{P}\left[U \mid \mathbf{r}, V, \mathbf{z}\right] = \arg\max \ \mathbb{P}\left[\mathbf{r} \mid V, U, \mathbf{z}\right] \cdot \mathbb{P}\left[U\right]$$

The above reduces to a sequence of k ridge regression problems

$$\mathbf{u}^i = \arg\min \ \frac{1}{2}\|\mathbf{u}\|_2^2 + \frac{1}{2}\sum_{t:z^t=i}(r^t - \left\langle \mathbf{u}, \mathbf{v}^t \right\rangle)^2,$$

all of which have closed form solutions. Given an estimate of the user vectors, we can find the MAP assignments to the latent variables as follows

$$\mathbf{z} = \arg\max \ \mathbb{P}\left[\mathbf{z} \,|\, \mathbf{r}, V, U\right] = \arg\max \ \mathbb{P}\left[\mathbf{r} \,|\, \mathbf{z}, V, U\right] \cdot \mathbb{P}\left[\mathbf{z}\right]$$

Since we have assumed uniform priors, the above just becomes

$$z^t = \arg\max \ \mathbb{P}\left[\mathbf{r} \,|\, \mathbf{z}, V, U\right] = \arg\min_{i\in[k]} \ (r^t - \left\langle \mathbf{u}^i, \mathbf{v}^t \right\rangle)^2$$

The above readily gives us an alternating optimization algorithm for this problem. At every time step, first we use the existing user models to assign every rating to the user that is most likely to have made that rating. Next, having updated the assignments, we update the user models for each user.

Note that we may even perform the EM algorithm to solve this problem by attributing a rating to every member of the family with different weights (that are positive and add up to one).

Algorithm 1: User Profiling in RecSys

Input: Ratings $\left\{(r^t, \mathbf{v}^t)\right\}_{t\in[n]}$

Initialize $\mathbf{u}^1, \ldots, \mathbf{u}^k$
for $s = 1, 2, \ldots,$ **do**
  for $t = 1, \ldots, n$ **do**
    $z^t = \arg\min_{i\in[k]} \ |r^t - \langle \mathbf{u}^i, \mathbf{v}^t \rangle|$ `//Reassign ratings`
  end for
  for $1 = 1, \ldots, k$ **do**
    Let $X^i = \left[\mathbf{v}^t\right]_{t:z^t=i}$ and $\mathbf{y}^i = \left[r^t\right]_{t:z^t=i}$ `//Handy shorthand`
    $\mathbf{u}^i = \left(X^i (X^i)^\top + I_d\right)^{-1} X^i \mathbf{y}^i$ `//Update user models`
  end for
end for
return User models $\left\{\mathbf{u}^i\right\}_{i=1}^k$ and assignments $\left\{z^t\right\}_{t=1}^n$

3.1 Exercises

Exercise 3.1. Consider the problem of heteroscedastic regression, a curious variant of linear regression where the noise added to each data point comes from a Gaussian distribution with potentially different variance. Let $\mathbf{x}^i \in \mathbb{R}^d, i = 1, \ldots, n$ denote the covariates/feature vectors. The responses are generated as $y^i = \langle \mathbf{w}, \mathbf{x}^i \rangle + \epsilon^i$, where the noise $\epsilon^i \sim \mathcal{N}(0, \sigma_i^2)$ for the i-th data point has variance σ_i^2 . We are shown $\left\{(\mathbf{x}^i, y^i)\right\}_{i\in[n]}$ but model $\left\{\sigma_i\right\}_{i\in[n]}$ as latent variables. Note that this is a discriminative model and $\mathbf{x}^i$ are not probabilistically modelled. You may find the shorthands $X = [\mathbf{x}^1, \ldots, \mathbf{x}^n], \mathbf{y} = [y^1, \ldots, y^n], \Sigma =$

$\text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$ to be helpful. In all the problems below, your expressions may have unspecified normalization constants.

1. Derive an expression for $\mathbb{P}\left[\sigma_i \,|\, y^i, \mathbf{x}^i, \mathbf{w}\right]$ using the (uniform) prior $\mathbb{P}\left[\sigma_i\right] = 1$ if $\sigma_i \in [0, 1]$ and $\mathbb{P}\left[\sigma_i\right] = 0$ otherwise. Then derive the MAP estimate for σ_i i.e. $\arg\max\ \mathbb{P}\left[\sigma_i \,|\, y^i, \mathbf{x}^i, \mathbf{w}\right]$ assuming the model $\mathbf{w}$ is known. For simplicity, assume $\mathbb{P}\left[\sigma_i \,|\, \mathbf{x}^i, \mathbf{w}\right] = \mathbb{P}\left[\sigma_i\right]$ i.e. $\mathbf{w}$ and $\mathbf{x}^i$ had nothing to do with the selection of σ_i.

2. Derive an expression for $\mathbb{P}\left[\mathbf{w} \,|\, y^i, \mathbf{x}^i, \sigma_i\right]$ using a standard Gaussian prior $\mathbb{P}\left[\mathbf{w}\right] = \frac{1}{\sqrt{(2\pi)^d}} \exp(-\frac{1}{2} \left\|\mathbf{w}\right\|_2^2)$. Then derive the MAP estimate for $\mathbf{w}$ i.e. $\arg\max\ \mathbb{P}\left[\mathbf{w} \,|\, \mathbf{y}, X, \Sigma\right]$ assuming that $\{\sigma_i\}$ are known.

3. Using the above estimates, give the pseudocode for an alternating optimization algorithm for estimating $\mathbf{w}$ that performs MAP-based hard assignments to the latent variables σ_i to solve the problem.

4

Learning with Kernels

4.1 Kernel PCA Derivation

We will perform the derivations assuming that the kernel K in question corresponds to a feature map ϕ_K that is finite dimensional i.e. $\phi_K : \mathcal{X} \to \mathcal{H} = \mathbb{R}^D$ for some $D < \infty$ (we will often write ϕ instead of the more verbose ϕ_K when the kernel in question is obvious). This will be done merely for the sake of providing intuition without getting harassed by infinite dimensional spaces. We will continue to assume that D, although finite, is so large that it is all but impossible to actually perform computations with the mapped feature vectors $\phi(\mathbf{x}^i)$.

Given a set of raw feature vectors $\mathbf{x}^i \in \mathcal{X} \subseteq \mathbb{R}^d, i \in [n]$, collected in a matrix $X \in \mathbb{R}^{n \times d}$, (linear) PCA involves computing the top k eigenvalues of $X^\top X \in \mathbb{R}^{d \times d}$ for which the power+peeling methods are quite successful. Let $\Phi = [\phi(\mathbf{x}^1), \ldots, \phi(\mathbf{x}^n)]^\top \in \mathbb{R}^{n \times D}$ be the impossibly huge (but still finite) matrix of the mapped features. Naturally, for kernel PCA, we wish to find the top k eigenfunctions of the linear operator $M = \Phi^\top \Phi = \sum_{i=1}^n \phi(\mathbf{x}^i)\phi(\mathbf{x}^i)^\top$.

Suppose $\mathbf{v} \in \mathcal{H}$ is an eigenfunction of the operator M. Then it must be the case that $M\mathbf{v} = \lambda \cdot \mathbf{v}$ for some $\lambda \in \mathbb{R}$ (since M is a symmetric operator, it has only real eigenvalues, just like a symmetric matrix only has real eigenvalues). This means that $\mathbf{v} = \frac{1}{\lambda} \cdot M\mathbf{v} = \frac{1}{\lambda} \sum_{i=1}^n \phi(\mathbf{x}^i)\phi(\mathbf{x}^i)^\top \mathbf{v} = \Phi^\top \boldsymbol{\alpha}$ where $\boldsymbol{\alpha} \in \mathbb{R}^n$ with $\boldsymbol{\alpha}_i = \frac{\phi(\mathbf{x}^i)^\top \mathbf{v}}{\lambda}$.

This means that we may implicitly represent eigenfunctions in terms of the mapped feature vectors and thus, instead of searching for eigenfunctions $\mathbf{v} \in \mathcal{H}$ which are large/infinite dimensional, we may instead search for coefficients $\boldsymbol{\alpha} \in \mathbb{R}^n$ which are always finite and (relatively) small dimensional. To do so we rewrite the equation $M\mathbf{v} = \lambda \cdot \mathbf{v}$ as $\Phi^\top \Phi \Phi^\top \boldsymbol{\alpha} = \lambda \cdot \Phi^\top \boldsymbol{\alpha}$ (using $M = \Phi^\top \Phi$ and $\mathbf{v} = \Phi^\top \boldsymbol{\alpha}$).

The above equation is still infinite dimensional and cannot be solved as is for α. However, notice that the matrix $\Phi\Phi^\top$ is nothing but $G \in \mathbb{R}^{n \times n}$ the *Gram matrix* of pairwise kernel values of the training data points. Multiplying

both sides of the equation with Φ gives us $\Phi\Phi^\top\Phi\Phi^\top\boldsymbol{\alpha} = \lambda \cdot \Phi\Phi^\top\boldsymbol{\alpha}$ which is the same as $G^2\boldsymbol{\alpha} = \lambda \cdot G\boldsymbol{\alpha}$.

Now, if G is invertible (i.e. full rank), then the above can be reduced to $G\boldsymbol{\alpha} = \lambda \cdot \boldsymbol{\alpha}$ which means that the coefficient vectors $\boldsymbol{\alpha}$ that we are searching are nothing but the eigenvectors of G. However, this result is true even if G is not full rank. Below we derive this result.

Lemma 4.1. Every eigenfunction $\mathbf{v}$ of the operator $M = \Phi^\top\Phi$ that corresponds to a non-zero eigenvalue can be expressed as $\frac{1}{\left\|\Phi^\top\boldsymbol{\alpha}\right\|_{\mathcal{H}}} \cdot \Phi^\top\boldsymbol{\alpha}$ where $\boldsymbol{\alpha}$ is an eigenvector of the Gram matrix $G = \Phi\Phi^\top$ corresponding to a non-zero eigenvalue.

Proof. Let $\Phi = U\Sigma V^\top$ be the (thin) SVD of Φ with $U = [\mathbf{u}^1, \ldots, \mathbf{u}^r] \in \mathbb{R}^{n\times r}, \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}, V = [\mathbf{v}^1, \ldots, \mathbf{v}^r] \in \mathbb{R}^{D\times r}$ where $r = \operatorname{rank}(\Phi) \leq \min\{n, D\}$ i.e. $\sigma_i \neq 0$ for all $i \in [r]$. Note that this makes sense only because we assumed ϕ to be a finite dimensional map[1]. Thus, $M = V\Sigma^2V^\top$ and $G = U\Sigma^2U^\top$. Consider any eigenvector $\mathbf{u}^i \in \mathbb{R}^n$ of the matrix G. Since the columns of U are orthonormal, we have

$$\Phi^\top\mathbf{u}^i = V\Sigma U^\top\mathbf{u}^i = \sigma_i \cdot \mathbf{v}^i$$

Thus, if we enumerate over the eigenvectors $\mathbf{u}^i, i = 1, 2, \ldots, r$ of G, the expression $\Phi^\top\mathbf{u}^i$ does give us scaled versions of $\mathbf{v}^i, i = 1, 2, \ldots, r$, the eigenfunctions of M. All we need to do is scale them back to unit norm (as the statement of the lemma indicates) and we are done. This scaling is also very simple to perform. If our power + peeling method has returned $(\lambda_i, \mathbf{u}^i)$ as an eigenpair of G, then we have

$$\left\|\Phi^\top\mathbf{u}^i\right\|_{\mathcal{H}}^2 = (\mathbf{u}^i)^\top\Phi\Phi^\top\mathbf{u}^i = (\mathbf{u}^i)^\top G\mathbf{u}^i = \lambda_i$$

which implies that the normalization constant is simply $\sqrt{\lambda_i}$. □

4.2 Kernel kNN Done Two Ways

Given a kernel K with a corresponding feature map $\phi : \mathcal{X} \to \mathcal{H}$ and training points $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$, we can execute the kernel k-nearest neighbors (KkNN) algorithm in two distinct ways. Let us set $k = 1$ for sake of simplicity.

1. kNN in RKHS: for a test point $\mathbf{x}^t \in \mathcal{X}$, we find its closest neighbor in the RKHS $\mathcal{H}$ i.e. $i^t = \arg\min_{i\in[n]} \left\|\phi(\mathbf{x}^t) - \phi(\mathbf{x}^i)\right\|_{\mathcal{H}}^2 = \arg\min_{i\in[n]} K(\mathbf{x}^i, \mathbf{x}^i) - 2K(\mathbf{x}^t, \mathbf{x}^i)$ and predict $\hat{y}^t = y^{i^t}$ as the label for the test point.

2. Similarity kNN: since kernels are meant to be measures of similarity, we may directly use them to find "neighbors". For a test point $\mathbf{x}^t \in \mathcal{X}$, find its most similar neighbor in terms of kernel value $\tilde{i}^t = \arg\max_{i\in[n]} K(\mathbf{x}^t, \mathbf{x}^i)$ and predict $\hat{y}^t = y^{\tilde{i}^t}$ as the label for the test point.

[1] This lemma also holds if ϕ is a "well-behaved" infinite dimensional map but requires a more elaborate and careful proof.

4.3 Exercises

Exercise 4.1. Suppose we have a binary classification problem $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$ with $\mathbf{x}^i \in \mathbb{R}^d, y^i \in \{-1, 1\}$. Using the technique of introducing dummy constraints that we used to kernelize ridge regression, derive a dual problem for the L_2 regularized logistic regression problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \log(1 + \exp(-y^i \cdot \mathbf{w}^\top \mathbf{x}^i))$$

and conclude that logistic regression may be kernelized as well.

Exercise 4.2. Can the k-means++ initialization algorithm be kernelized? If so how much time does it take to run when working with n data points with initial raw features that are d-dimensional, C clusters, and a kernel such that it takes only $\mathcal{O}(d)$ time to perform a single kernel computation?

Exercise 4.3. Suppose K_1, K_2 are Mercer kernels with corresponding maps ϕ_1, ϕ_2. For sake of simplicity, assume that both maps are finite dimensional i.e. $\phi_1, \phi_2 : \mathcal{X} \to \mathbb{R}^D$ for some very large, but still finite D. Define three new kernels as $K_3(\mathbf{x}^1, \mathbf{x}^2) = K_1(\mathbf{x}^1, \mathbf{x}^2) + K_2(\mathbf{x}^1, \mathbf{x}^2)$ as well as $K_4(\mathbf{x}^1, \mathbf{x}^2) = K_1(\mathbf{x}^1, \mathbf{x}^2) \cdot K_2(\mathbf{x}^1, \mathbf{x}^2)$ as well as $K_5(\mathbf{x}^1, \mathbf{x}^2) = \frac{K_1(\mathbf{x}^1, \mathbf{x}^2)}{\sqrt{K_1(\mathbf{x}^1, \mathbf{x}^1) \cdot K_1(\mathbf{x}^2, \mathbf{x}^2)}}$. Show that K_3, K_4, K_5 are all Mercer kernels too by constructing feature maps for all these kernels.

Exercise 4.4. Can you given an example of two Mercer kernels K_1, K_2 with corresponding (finite dimensional) maps $\phi_1, \phi_2 : \mathcal{X} \to \mathbb{R}^D$ such that the kernel $K_6(\mathbf{x}^1, \mathbf{x}^2) = K_1(\mathbf{x}^1, \mathbf{x}^2) - K_2(\mathbf{x}^1, \mathbf{x}^2)$ is not a Mercer kernel, i.e. for no real valued feature map ϕ_6 can we have $K_6(\mathbf{x}^1, \mathbf{x}^2) = \langle \phi_6(\mathbf{x}^1), \phi_6(\mathbf{x}^2) \rangle$?

Exercise 4.5. Suppose K is a Mercer kernel with corresponding map ϕ that is assumed to be finite dimensional for sake of simplicity, i.e. $\phi : \mathcal{X} \to \mathbb{R}^D$ for some very large, but still finite D. Then show that for any $n > 0$ and any set of n points $\mathbf{x}^1, \ldots, \mathbf{x}^n \in \mathcal{X}$, the Gram matrix of kernel K on this data i.e. $G_{ij} = K(\mathbf{x}^i, \mathbf{x}^j)$ is always a positive-semi definite matrix.

Exercise 4.6. Suppose we are given a Gram matrix $G \in \mathbb{R}^{n\times n}$ for n data points but are not told which kernel generated this Gram matrix, nor do we have access to the original features for these data points. Can we still come up with feature representations for these data points i.e. vectors $\mathbf{x}^1, \ldots, \mathbf{x}^n \in \mathbb{R}^D$ for some $D < \infty$ such that $G_{ij} = \langle \mathbf{x}^i, \mathbf{x}^j \rangle$ for all $i, j \in [n]$?

Exercise 4.7. Suppose someone has given us a Gram matrix G and, as in Exercise 4.6, forgot to give us the original raw features or the kernel used to compute this Gram matrix. However, we are told that the kernel used was definitely Mercer. We notice that G is not symmetric i.e. $G^\top \neq G$. Since Mercer kernels always yield symmetric Gram matrices (since dot/inner products are symmetric), we decide to fix this by replacing G by $H = \frac{G+G^\top}{2}$. Show that G is PSD iff H is PSD. More importantly, show that it does not matter whether we use G or H in algorithms such as kernel SVM or kernel ridge regression – both will yeild exactly the same result.

Exercise 4.8. In Section 4.2 we saw how to execute the kNN algorithm using kernels in two ways. Can you give examples of kernels where the two ways would give exactly the same results i.e. we would always have $i^t = \tilde{i}^t$? Can you also come up with an example of a kernel where the two ways would give different results? You may do so by constructing a kernel K with map $\phi : \mathcal{X} \to \mathcal{H}$ alongwith three points $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$ such that $K(\mathbf{x}, \mathbf{y}) > K(\mathbf{x}, \mathbf{z})$ as well as $\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}} > \|\phi(\mathbf{x}) - \phi(\mathbf{z})\|_{\mathcal{H}}$ i.e. $\mathbf{x}$ is more similar to $\mathbf{y}$ according to the kernel similarity value but $\mathbf{x}$ is actually closer to $\mathbf{z}$ in the RKHS. Can you find an example of a kernel where the usual kNN algorithm using Euclidean distance would give the same result as kernel kNN in the RKHS?

Acknowledgements

The author is thankful to the students of successive offerings of the course for their inputs and pointing out various errata in the lecture material. This monograph was typeset using the beautiful style of the Foundations and Trends® series published by now publishers.

Appendices

A

Calculus Refresher

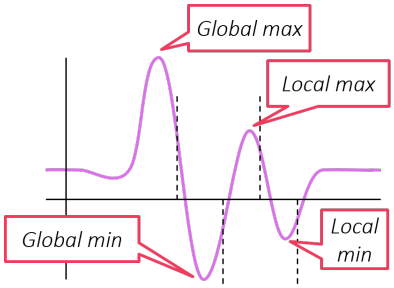
In this chapter we will take a look at basic tools from calculus that would be required to design and execute machine learning algorithms. Before we proceed, we caution the reader that the treatment in this chapter will *not* be mathematically rigorous and frequently, we will appeal to concepts and results based on informal arguments and demonstration, rather than proper proofs. This was done in order to provide the reader with a working knowledge of the topic without getting into excessive formalism. We direct the reader to texts in mathematics, of which several excellent ones are available, for a more rigorous treatment of this subject.

A.1 Extrema

The vast majority of machine learning algorithms learn models by trying to obtain the best possible performance on training data. What changes from algorithm to algorithm is how "performance" is defined and what constitutes "best". Frequently, performance can be defined in terms of an objective function f that takes in a model (say, a linear model $\mathbf{w}$) and outputs a real number $f(\mathbf{w}) \in \mathbb{R}$ called the objective value. Depending on the algorithm designer a large objective value may be better or a small score may be better (e.g. if f encodes margin then we want a large objective value, on the other hand if f encodes the classification error then we want a small objective value).

objective function

objective value



Given a function $f : \mathbb{R}^d \to \mathbb{R}$, a point $\mathbf{x}^* \in \mathbb{R}^d$ is said to be the global maximum of this function if $f(\mathbf{x}^*) \geq f(\mathbf{x})$ for all other $\mathbf{x} \in \mathbb{R}^d$. We similarly define a global minimum of this function as a point $\tilde{\mathbf{x}}$ such that $f(\tilde{\mathbf{x}}) \leq f(\mathbf{x})$ for all other $\mathbf{x} \in \mathbb{R}^d$. Note that a function may have multiple global maxima and global minima. For example the function $f(x) = \sin(x)$ has global maxima at all values of x that are of the form $2k\pi + \frac{\pi}{2}$

global maximum

global minimum

and global minima at all values of x that are of the form $2k\pi - \frac{\pi}{2}$.

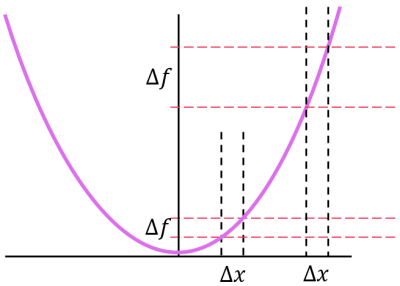
However, apart from global extrema which achieve the largest or the smallest value of a function among all possible input points, we can also have local extrema, i.e. local minimum and local maximum. These are points which achieve the best value of the function (min for local minima and max for local maxima) only in a certain (possibly small) region surrounding the point. local extrema

A practical example to understand the distinction between local and global extrema can be that of population: the city of Kanpur has a large population (that of 3.2 million) which is the highest among cities within the state of Uttar Pradesh. Thus, Kanpur is at least a local maximum. However, it is not a global maximum since if we go outside the state of Uttar Pradesh, we find cities like Mumbai with a population of 12.4 million. However, even Mumbai is a local maximum (among cities within India) since the global maximum (of largest population among all cities on Earth) is achieved at Chongqing, China which has a population of 30.1 million (source: Wikipedia).

It is be clear from the above definitions that all global extrema are necessarily local extrema. For example, Chongqing clearly has the largest population within China itself and thus a local maximum. However, not all local extrema need be global extrema.

A.2 Derivatives

Derivatives are an integral part of calculus (pun intended) and are the most direct way of finding how function values change (increase/decrease) if we move from one point to another. Given a univariate function i.e. a function $f : \mathbb{R} \to \mathbb{R}$ that takes a single real number as input and outputs a real number (we will take care of multivariate functions later), the derivative of f at a point x^0 tells us two things. Firstly, if the sign of the derivative is positive i.e. $f'(x^0) > 0$, then the function value will increase if we move a little bit to the right on the number line (i.e. go from x^0 to $x^0 + \Delta x$ for some $\Delta x > 0$) and it will decrease if we move a little bit to the left on the number line. Similarly if $f'(x^0) < 0$, then moving right decreases the function value whereas moving left increases the function value. univariate function



Secondly, the magnitude of the derivative i.e. $|f'(x^0)|$ tells us by how much would the function value increase or decrease if we move a little bit left or right from the point x^0. For example, consider the function $f(x) = x^2$. Its derivative is $f'(x) = 2x$. This tells us that if $x^0 < 0$ (where the derivative is negative), the function value would decrease if we moved right and increase if we moved left. Similarly, if $x^0 > 0$, the derivative is positive and thus, the function value would increase if we moved to the right and decrease if we moved to the left. However, since the magnitude of the derivative is $2|x|$ which increases as we go away from the origin, it can be seen that the increase in function value, for the same change in the value of x^0 s

much steeper if x^0 is far from the origin.

It is important to note that the above observations (e.g. function value goes up if $f(x^0) > 0$ and we move to the right) hold true only if the movement Δx is "small". For example, $f(x) = x^2$ has a negative derivative at $x^0 = -2$ and so the function value should decrease if we moved right little bit. However, if we move right too much (say we move to $x^0 = 3$) then the above promise does not hold since $f(3) = 9 > 4 = f(-2)$. In fact a corollary of the Taylor's theorem states

Taylor's theorem (first order)

$$f(x^0 + \Delta x) \approx f(x^0) + f'(x) \cdot \Delta x, \text{if } \Delta x \text{ is "small".}$$

How small is small enough for the above result to hold may depend on both the function f as well as the point x^0 where we are applying the result.

A.3 Second Derivative

Just as the derivative of a function tells us how does the function value changes (i.e. goes up/down) and by how much, the second derivative tells us how does the derivative change (i.e. go up/down) and by how much. Intuitively, the second derivative can be thought of as similar to acceleration if we consider the derivative as similar to velocity and the function value as being similar to displacement. If at a point x^0 we have $f''(x^0) > 0$, then this means that the derivative will go up if we move to the right and decrease if we move to the left (similarly if $f''(x^0) < 0$ at a point).

The Taylor's theorem does extend to second order derivatives as well

$$f'(x^0 + \Delta x) \approx f'(x^0) + f''(x^0) \cdot \Delta x, \text{if } \Delta x \text{ is "small".}$$

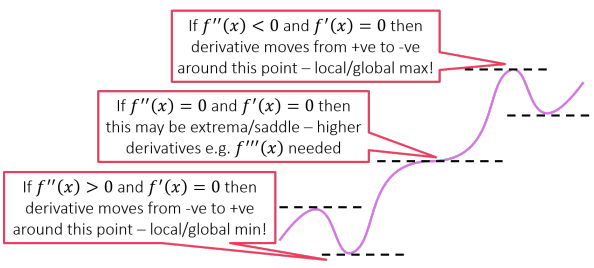
Integrating both sides and applying the fundamental theorem of algebra

$$f(x^0 + \Delta x) \approx f(x^0) + f'(x^0) \cdot \Delta x + \frac{1}{2} f''(x^0) \cdot (\Delta x)^2, \text{if } \Delta x \text{ is "small".}$$

Taylor's theorem (second order)

Although the above derivation is not strictly rigorous, the result is nevertheless true. Thus, knowing the second derivative can help us get a better approximation of the change in function value if we move a bit. The second derivative is most commonly used in machine learning in designing very efficient optimization algorithms (known as *Newton methods* which we will study later). In fact there exist 3rd and higher order derivatives as well (the third derivative telling us how does the second derivative change from point to point etc) but since they are not used all that much, we will not study them here.

A.4 Stationary Points



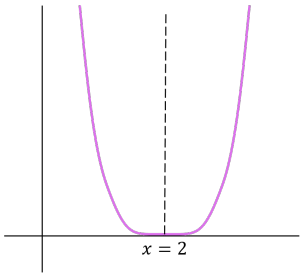
The stationary points of a function $f : \mathbb{R} \to \mathbb{R}$ are defined as the points where the derivative of the function vanishes i.e. $f'(x) = 0$. The

stationary points of a function correspond to either the local/global maxima or minima or else saddle points. Given a stationary point, the second derivative test is used to distinguish extrema from saddle points. second derivative test

If the second derivative of the function is positive at a stationary point x^0 i.e. $f'(x^0) = 0$ and $f''(x^0) > 0$ then x^0 is definitely a local minimum. This result follows directly from the second order Taylor's theorem we studied above. Since $f'(x^0) = 0$, we have

$$f(x^0 + \Delta x) \approx f(x^0) + \frac{1}{2} f''(x^0) \cdot (\Delta x)^2 \geq f(x^0)$$

This means that irrespective of whether $\Delta x < 0$ or $\Delta x > 0$ (i.e. irrespective of whether we move left or right), the function value always increases. Recall that this is the very definition of a local minimum. Similarly, we can intuitively see that if $f'(x^0) = 0$ and $f''(x^0) < 0$ then x^0 is definitely a local maximum.



If we have $f'(x^0) = 0$ and $f''(x^0) = 0$ at a point then the second derivative test is actually silent and fails to tell us anything informative. The reader is warned that the first and second derivatives both vanishing *does not* mean that the point is a saddle point. For example, consider the case of the function $f(x) = (x-2)^4$. Clearly $x^0 = 2$ is a local (and global) minimum. However, it is also true that $f'(2) = 0 = f''(2)$. In such inconclusive cases, higher order derivatives e.g. $f^{(3)}(x) = f'''(x), f^{(4)}(x)$ have to be used to figure out what is the status of our stationary point.

A.5 Useful Rules for Calculating Derivatives

Several rules exist that can help us calculate the derivative of complex-looking functions with relative ease. These are given below followed by some examples applying them to problems.

1. (Constant Rule) If $h(x) = c$ where c is not a function of x then $h'(x) = 0$
2. (Sum Rule) If $h(x) = f(x) + g(x)$ then $h'(x) = f'(x) + g'(x)$
3. (Scaling Rule) If $h(x) = c \cdot f(x)$ and if c is not a function of x then $h'(x) = c \cdot f'(x)$
4. (Product Rule) If $h(x) = f(x) \cdot g(x)$ then $h'(x) = f'(x) \cdot g(x) + g'(x) \cdot f(x)$
5. (Quotient Rule) If $h(x) = \frac{f(x)}{g(x)}$ then $h'(x) = \frac{f'(x) \cdot g(x) - g'(x) \cdot f(x)}{g^2(x)}$
6. (Chain Rule) If $h(x) = f(g(x)) \triangleq (f \circ g)(x)$, then $h'(x) = f'(g(x)) \cdot g'(x)$

Apart from this, some handy rules exist for polynomial functions e.g. if $f(x) = x^c$ where c is not a function of x, then $f'(x) = c \cdot x^{c-1}$, the logarithmic function i.e. if $f(x) = \ln(x)$ then $f'(x) = \frac{1}{x}$, the exponential function i.e. if $f(x) = \exp(x)$ then $f'(x) = \exp(x)$ and trigonometric functions i.e. if $f(x) =$

$\sin(x)$ then $f'(x) = \cos(x)$ and if $f(x) = \cos(x)$ then $f'(x) = -\sin(x)$. The most common use of the chain rule is finding $f'(x)$ when f is a function of some variable, say t but t itself is a function of x i.e. $t = g(x)$.

Example A.1. Let $\ell(x) = (a \cdot x - b)^2$ where $a, b \in \mathbb{R}$ are constants that do not depend on x. Then we can write $\ell(t) = t^2$ where $t(x) = a \cdot x - b$. Thus, applying the chain rule tells us that $\ell'(x) = \ell'(t) \cdot t'(x)$. By applying the rules above we have $\ell'(t) = 2 \cdot t$ (polynomial rule) and $t'(x) = a$ (constant rule and scaling rule). This gives us $\ell'(x) = 2a \cdot (a \cdot x - b)$.

Example A.2. Let $\sigma(x) = \frac{1}{1+\exp(-B \cdot x)}$ be the sigmoid function where $B \in \mathbb{R}$ is a constant that does not depend on x. Then we can write $\sigma(t) = (t)^{-1}$ where $t(s) = 1 + \exp(s)$ where $s(x) = -B \cdot x$. Thus, applying the chain rule tells us that $\sigma'(x) = \sigma'(t) \cdot t'(s) \cdot s'(x)$. By applying the rules above we have $\sigma'(t) = -\frac{1}{t^2}$ (polynomial rule), $t'(s) = \exp(s)$ (constant rule and exponential rule), $s'(x) = -B$ (scaling rule). This gives us $\sigma'(x) = B\frac{\exp(-B \cdot x)}{(1+\exp(-B \cdot x))^2} = B \cdot \sigma(x)(1 - \sigma(x))$

A.6 Multivariate Functions

In the previous sections we looked at functions of one variable i.e. univariate functions $f : \mathbb{R} \to \mathbb{R}$. We will now extend our intuitions about derivatives to multivariate functions i.e. functions of multiple variables i.e. of the form $f : \mathbb{R}^d \to \mathbb{R}$ which take a d-dimensional vector as input and output a real number.

multivariate functions

A.6.1 First Derivatives

As before, the first derivative tells us how much the function value changes and in what direction, if we move a bit from our current location. Since in d dimensions, there are d directions along which we can move, "moving" means going from $\mathbf{x}^0 \in \mathbb{R}^d$ to a point $\mathbf{x}^0 + \Delta\mathbf{x}$ where $\Delta\mathbf{x} \in \mathbb{R}^d$ (but $\Delta\mathbf{x}$ is "small" i.e. $\|\Delta\mathbf{x}\|_2$ is small). To capture how the function value may change as a result of such movement, the gradient of the function captures how much the function changes if we move just long one of the axes.

More specifically, the gradient of a multivariate function f at a point $\mathbf{x}^0 \in \mathbb{R}^d$ is a vector $\nabla f(\mathbf{x}^0) = \left(\frac{\partial f}{\partial \mathbf{x}_1}, \frac{\partial f}{\partial \mathbf{x}_2}, \ldots, \frac{\partial f}{\partial \mathbf{x}_d}\right)$ where for any $j \in [d]$, $\frac{\partial f}{\partial \mathbf{x}_j}$ indicates whether the function value increases or decreases and by how much, if we keep all coordinates of $\mathbf{x}^0$ fixed except the j^{th} coordinate whih we increase by a small amount i.e. if $\Delta\mathbf{x} = (0, 0, \ldots, 0, \underset{j}{\delta}, 0, \ldots, 0)$, then our friend the Taylor's theorem tells us that

gradient

$$f(\mathbf{x}^0 + \Delta\mathbf{x}) \approx f(\mathbf{x}^0) + \delta \cdot \frac{\partial f}{\partial \mathbf{x}_j}$$

We can use the gradient to find out how much the function value would change if we moved a little bit in a general direction by summing up the individual contributions from all the axes. Suppose we move along $\Delta\mathbf{x}$ where now all

coordinates of $\Delta\mathbf{x}$ may be non-zero (but small), then the following holds

Multivariate Taylor's theorem (first order)

$$\begin{aligned} f(\mathbf{x}^0 + \Delta\mathbf{x}) &\approx f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0)^\top \Delta\mathbf{x} \\ &= f(\mathbf{x}^0) + \sum_{j=1}^{d} \Delta\mathbf{x}_j \cdot \frac{\partial f}{\partial \mathbf{x}_j} \end{aligned}$$

The gradient also has the very useful property of being the direction of steepest ascent. This means that among all the directions in which we could move, if we move along the direction of the gradient, then the function value would experience the maximum amount of increase. However, for machine learning applications, a related property holds more importance: among all the directions in which we could move, if we move along the direction opposite to that of the gradient i.e. we move along $-\nabla f(\mathbf{x}^0)$, then the function value would experience the maximum amount of *decrease* – this means that the direction opposite to the gradient offers the steepest descent.

steepest ascent

steepest descent

A.6.2 Second Derivatives

Second derivatives play a similar role of documenting how the first derivative changes as we move a little bit from point to point. However, since we have d partial derivatives here and d possible axes directions along which to move, the second derivative for multivariate functions is actually a $d \times d$ matrix, called the Hessian and denoted as $\nabla^2 f(\mathbf{x}^0)$.

Hessian

$$\nabla^2 f(\mathbf{x}^0) = \begin{bmatrix} \frac{\partial^2 f}{\partial \mathbf{x}_1^2} & \frac{\partial^2 f}{\partial \mathbf{x}_1 \partial \mathbf{x}_2} & \cdots & \frac{\partial^2 f}{\partial \mathbf{x}_1 \partial \mathbf{x}_d} \\ \frac{\partial^2 f}{\partial \mathbf{x}_2 \partial \mathbf{x}_1} & \frac{\partial^2 f}{\partial \mathbf{x}_2^2} & \cdots & \frac{\partial^2 f}{\partial \mathbf{x}_2 \partial \mathbf{x}_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \mathbf{x}_d \partial \mathbf{x}_1} & \frac{\partial^2 f}{\partial \mathbf{x}_d \partial \mathbf{x}_2} & \cdots & \frac{\partial^2 f}{\partial \mathbf{x}_d^2} \end{bmatrix}$$

Clairaut's theorem tells us that if the function f is "nice" (basically the second order partial derivatives are all continuous), then $\frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j} = \frac{\partial^2 f}{\partial \mathbf{x}_j \partial \mathbf{x}_i}$ i.e. the Hessian matrix is symmetric. The $(i,j)^{\text{th}}$ entry of this Hessian matrix – styled as $\frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j}$ – records how much the i^{th} partial derivative changes if we move a little bit along the j^{th} axis i.e. if $\Delta\mathbf{x} = (0, 0, \ldots, 0, \underset{j}{\delta}, 0, \ldots, 0)$, then

$$\frac{\partial f}{\partial \mathbf{x}_i}(x^0 + \Delta x) \approx \frac{\partial f}{\partial \mathbf{x}_i}(x^0) + \frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j}(x^0) \cdot \Delta\mathbf{x}, \text{if } \Delta\mathbf{x} \text{ is "small".}$$

Just as in the univariate case, the Hessian can be incorporated into the Taylor's theorem to obtain a finer approximation of the change in function value. Denote $H = \nabla^2 f(\mathbf{x}^0)$ for sake of notational simplicity

Multivariate Taylor's theorem (second order)

$$\begin{aligned} f(\mathbf{x}^0 + \Delta\mathbf{x}) &\approx f(\mathbf{x}^0) + \nabla f(\mathbf{x}^0)^\top \Delta\mathbf{x} + (\Delta\mathbf{x})^\top H (\Delta\mathbf{x}) \\ &= f(\mathbf{x}^0) + \sum_{j=1}^{d} \Delta\mathbf{x}_j \cdot \frac{\partial f}{\partial \mathbf{x}_j} + \sum_{i=1}^{d}\sum_{j=1}^{d} \Delta\mathbf{x}_i \Delta\mathbf{x}_j \frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j}(x^0) \end{aligned}$$

A.6.3 Stationary Points

Just as in the univariate case, here also we define stationary points as those where the gradient of the function vanishes i.e. $\nabla f(\mathbf{x}^0) = \mathbf{0}$. As before, stationary points can either be local minima/maxima or else saddle points and the second derivative test is used to decide which is the case. However, the *multivariate second derivative test* looks a bit different.

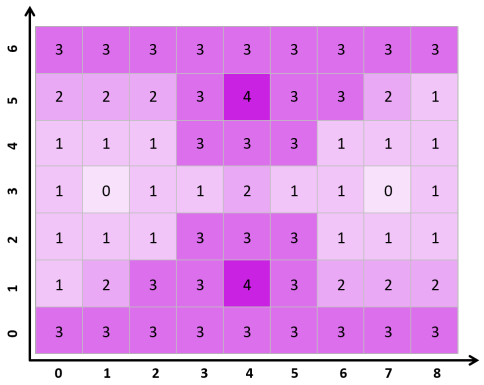
If the Hessian of the function is positive semi definite (PSD) at a stationary point $\mathbf{x}^0$ i.e. $\nabla f(\mathbf{x}^0) = \mathbf{0}$ and $H = \nabla^2 f(\mathbf{x}^0) \succeq 0$ then $\mathbf{x}^0$ is definitely a local minimum. Recall that a square symmetric matrix $A \in \mathbb{R}^{d \times d}$ is called positive semi definite if for all vectors $\mathbf{v} \in \mathbb{R}^d$, we have $\mathbf{v}^\top A \mathbf{v} \geq 0$. As before, this result follows directly from the multivariate second order Taylor's theorem we studied above. Since $\nabla f(\mathbf{x}^0) = \mathbf{0}$, we have

$$f(\mathbf{x}^0 + \Delta\mathbf{x}) \approx f(\mathbf{x}^0) + \frac{1}{2}(\Delta\mathbf{x})^\top H(\Delta\mathbf{x}) \geq f(\mathbf{x}^0)$$

This means that no matter in which direction we move from $\mathbf{x}^0$, the function value always increases. This is the very definition of a local minimum. Similarly, we can intuitively see that if the Hessian of the function is negative semi definite (NSD) at a stationary point $\mathbf{x}^0$ i.e. $\nabla f(\mathbf{x}^0) = \mathbf{0}$ and $\nabla^2 f(\mathbf{x}^0) \preceq 0$ then $\mathbf{x}^0$ is a local maximum. Recall that a square symmetric matrix $A \in \mathbb{R}^{d \times d}$ is called negative semi definite if for all vectors $\mathbf{v} \in \mathbb{R}^d$, we have $\mathbf{v}^\top A \mathbf{v} \leq 0$.

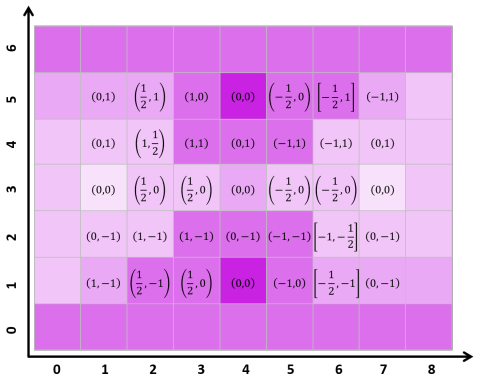
A.7 Visualizing Multivariate Derivatives

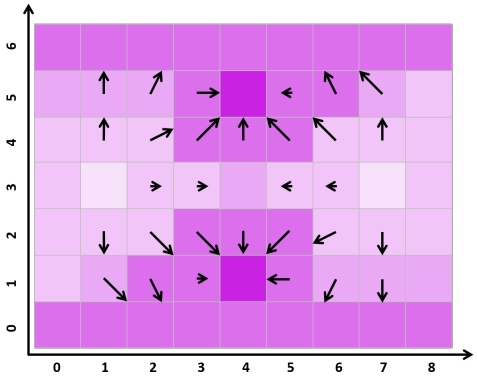
We now take a toy example to help the reader visualize how multivariate derivatives operate. We will take $d = 2$ to allow us to explicitly show gradients and function values on a 2D grid. The function we will study will not be continuous but discrete but will nevertheless allow us to revise the essential aspects of the topics we studied above.



Consider the function $f : [0, 8] \times [0, 6] \to \mathbb{R}$ on the left. The function is discrete – darker shades indicate a higher function value (which is also written inside the boxes) and lighter shades indicate a smaller function value. Since discrete functions are non-differentiable, we will use approximations to calculate the gradient of this function at all the points. Note that the input to this function are two integers (x, y) where $0 \leq x \leq 8$ and $0 \leq y \leq 6$.

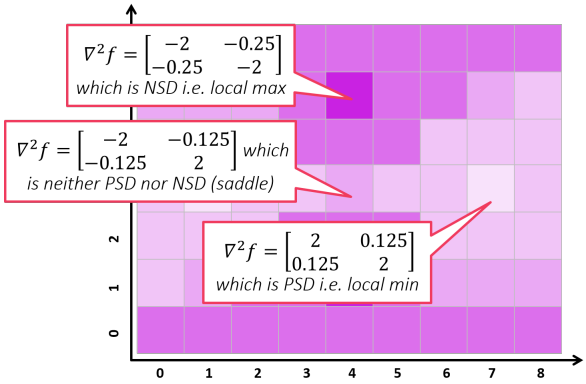
Given this, we may estimate the gradient of the function at a point (x_0, y_0) using the formula $\nabla f(x_0, y_0) = \left(\frac{\Delta f}{\Delta x}, \frac{\Delta f}{\Delta y}\right)$ where





$$\frac{\Delta f}{\Delta x} = \frac{f(x_0+1, y_0) - f(x_0-1, y_0)}{2}$$
$$\frac{\Delta f}{\Delta y} = \frac{f(x_0, y_0+1) - f(x_0, y_0-1)}{2}$$

The values of the gradients calculated using the above formula are shown on the left. Notice that we have five locations where the gradient vanishes (4,5), (1,3), (4,3), (7,3) and (4,1): these are stationary points. It may be more instructive to see the gradients represented as arrows which the figure on the left does. Notice that gradients converge toward the local maxima (4,5) and (4,1) from all directions (this is expected since the point has a greater function value than all its neighbors). Similarly, gradients diverge away from the local minima (1,3) and (7,3) from all directions (this is expected as well since the point has a smaller function value than all its neighbors). However, the point (4,3) being a saddle point, has gradients converging to it in the x direction but diverging away from it in the y direction. In order to verify which of our stationary points are local maxima/minima and which are saddle points, we need to estimate the Hessian of this function.



To do so, we use the following formulae for the approximate Hessian.

$$\nabla^2 f(x_0, y_0) = \begin{bmatrix} \frac{\Delta^2 f}{\Delta x^2} & \frac{\Delta^2 f}{\Delta x \Delta y} \\ \frac{\Delta^2 f}{\Delta x \Delta y} & \frac{\Delta^2 f}{\Delta y^2}, \end{bmatrix}$$

where we calculate each of the mixed partial derivative terms as follows.

$$\frac{\Delta^2 f}{\Delta x^2} = \frac{f(x_0+1, y_0) + f(x_0-1, y_0) - 2f(x_0, y_0)}{1^2}$$
$$\frac{\Delta^2 f}{\Delta y^2} = \frac{f(x_0, y_0+1) + f(x_0, y_0-1) - 2f(x_0, y_0)}{1^2}$$
$$\frac{\Delta^2 f}{\Delta x \Delta y} = \frac{f_{xy} + f_{yx}}{2}$$
$$f_{xy} = \frac{\frac{\Delta f}{\Delta x}(x_0, y_0+1) - \frac{\Delta f}{\Delta x}(x_0, y_0-1)}{2}$$

$$f_{yx} = \frac{\frac{\Delta f}{\Delta y}(x_0 + 1, y_0) - \frac{\Delta f}{\Delta y}(x_0 - 1, y_0)}{2}$$

Deriving these formulae for approximating mixed partial derivatives is relatively simple but we do not do so here. Also, the expression for $\frac{\Delta^2 f}{\Delta x \Delta y}$ which seems needlessly complicated due to the average involved, was made so in order to make sure that we obtain a symmetric matrix as the approximation to the Hessian (since Clairaut's theorem does not apply to our toy example it is not automatically ensured to us). However, any dissatisfaction with formulae aside, we can verify that the Hessian is indeed PSD at the local minima, NSD at the local maxima and neither NSD nor PSD at the saddle point. This verifies our earlier second derivative test rules.

A.8 Useful Rules for Calculating Multivariate Derivatives

The quantities of interest that we wish to calculate in the multivariate setting include derivatives of various orders i.e. gradient (first order derivative) and Hessian (second order derivative). It is notable that the rules that we studied in the context of univariate functions (Constant Rule, Sum Rule, Scaling Rule, Product Rule, Quotient Rule, Chain Rule) continue to apply in the multivariate setting as well. However, we need to be careful while applying them otherwise we may make mistakes and get confusing answers.

A.8.1 Dimensionality rule

A handy rule to remember while taking derivatives of multivariate functions is the dimensionality rule which shows us how to determine the dimensionality of the derivative using the dimensionality of the input and the output of the function. We will have to wait till we study vector valued functions and Jacobians before stating this rule in all its generality. For now, we will simply study special cases of this rule that are needed to calculate gradients and Hessians.

dimensionality rule

1. If $f : \mathbb{R}^d \to \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^d$, then $\frac{df}{d\mathbf{x}}$ (also denoted as the gradient $\nabla f(\mathbf{x})$) must also be a vector of d dimensions such that $\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial \mathbf{x}_1}, \ldots, \frac{\partial f}{\partial \mathbf{x}_d}\right]^\top$ (recall that our vectors are column by convention unless stated otherwise).
2. If $f : \mathbb{R}^d \to \mathbb{R}^d$ is a (vector-valued) function that takes in a d-dimensional vector as input and gives another d-dimensional vector as output, then $\frac{df}{d\mathbf{x}}$ must be a matrix of dimensionality $d \times d$. If we denote f_i as the i^{th} dimension of the output then the $(i,j)^{\text{th}}$ entry of the derivative is given by $\left[\frac{df}{d\mathbf{x}}\right]_{(i,j)} = \frac{\partial f_i}{\partial \mathbf{x}_j}$ i.e. the $(i,j)^{\text{th}}$ entry captures how the i^{th} dimension of the *output* changes as the j^{th} dimension of the *input* is changed.

The above cases may tempt the reader to wonder what happens if we have a function $f : \mathbb{R}^m \to \mathbb{R}^n$ which takes in an m-dimensional vector as input and gives an n-dimensional vector as output. Indeed, the derivative in this case must be an $n \times m$ matrix. Note that all the above cases fit this more general rule. However, we will study this in detail later when we study Jacobians.

A.8.2 Useful Rules for Calculating Gradients

Although carefully and correctly applying all the rules of univariate derivatives, as well as the dimensionality rules stated above, will always give us the right answer, doing so from scratch every time may be time consuming. Thus, we present here a few handy shortcuts for calculating gradients. We stress that every single one of these rules can be derived by simply applying the aforementioned rules carefully. In the following, $\mathbf{x} \in \mathbb{R}^d$ is a vector. Also, all vectors are column vectors unless stated otherwise.

1. (Dot Product Rule) If $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ where $\mathbf{a} \in \mathbb{R}^d$ is a vector that does not depend on $\mathbf{x}$, then $\nabla f(\mathbf{x}) = \mathbf{a}$. This rule can be derived by applying univariate scaling rule repeatedly to each dimension $j \in [d]$.

2. (Sum Rule) If $h(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$ then $\nabla h(\mathbf{x}) = \nabla f(\mathbf{x}) + \nabla g(\mathbf{x})$. This rule can be derived by applying univariate sum rule repeatedly to each dimension $j \in [d]$.

3. (Quadratic Rule) If $f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}$ where $A \in \mathbb{R}^{d\times d}$ is a symmetric matrix that is not a function of $\mathbf{x}$, then $\nabla f(\mathbf{x}) = 2A\mathbf{x}$. If A is not symmetric, then $\nabla f(\mathbf{x}) = A\mathbf{x} + A^\top \mathbf{x}$. This rule can be derived by applying the univariate product rule repeatedly to each dimension $j \in [d]$.

4. (Chain Rule) If $g : \mathbb{R}^d \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$, then if we define $h(\mathbf{x}) = f(g(\mathbf{x}))$, then $\nabla h(\mathbf{x}) = f'(g(\mathbf{x})) \cdot \nabla g(\mathbf{x})$. This rule can be derived by applying the univariate chain rule repeatedly to each dimension $j \in [d]$.

We now illustrate the use of these rules using some examples

Example A.3. Let $f(\mathbf{x}) = \|\mathbf{x}\|_2$. We can rewrite this as $f = \sqrt{t}$, where $t(\mathbf{x}) = \|\mathbf{x}\|_2^2 = \mathbf{x}^\top \mathbf{x} = \mathbf{x}^\top I_d \mathbf{x}$ where I_d is the $d \times d$ identity matrix. Thus, using the chain rule we have $\nabla f(\mathbf{x}) = f'(t) \cdot \nabla t(\mathbf{x})$. Using the polynomial rule we have $f'(t) = \frac{1}{2\sqrt{t}}$, whereas using the quadratic rule, we get $\nabla t(\mathbf{x}) = 2\mathbf{x}$. Thus we have $\nabla f(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. Note that in this case, the gradient is always a unit vector.

Example A.4. Let $\sigma(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{a}^\top \mathbf{x})}$ where $\mathbf{a} \in \mathbb{R}^d$ is a constant vector that does not depend on $\mathbf{x}$. Then we can write $\sigma(t) = (t)^{-1}$ where $t(s) = 1 + \exp(s)$ where $s(x) = -\mathbf{a}^\top \mathbf{x}$. Thus, applying the chain rule tells us that $\nabla\sigma(\mathbf{x}) = \sigma'(t) \cdot t'(s) \cdot \nabla s(\mathbf{x})$. By applying the rules above we have $\sigma'(t) = -\frac{1}{t^2}$ (polynomial rule), $t'(s) = \exp(s)$ (constant rule and exponential rule), $\nabla s(\mathbf{x}) = -\mathbf{a}$ (dot product rule). This gives us $\nabla\sigma(\mathbf{x}) = \frac{\exp(-\mathbf{a}^\top \mathbf{x})}{(1+\exp(-\mathbf{a}^\top \mathbf{x}))^2} \cdot \mathbf{a} = \sigma(\mathbf{x})(1 - \sigma(\mathbf{x})) \cdot \mathbf{a}$.

Example A.5. Let $f(\mathbf{x}) = (\mathbf{a}^\top \mathbf{x} - b)^2$ where $\mathbf{a} \in \mathbb{R}^d$ is a constant vector and $b \in \mathbb{R}$ is a constant scalar. Using the gradient chain rule, we get $\nabla f(\mathbf{x}) = 2(\mathbf{a}^\top \mathbf{x} - b) \cdot \mathbf{a}$.

Example A.6. Let $A \in \mathbb{R}^{n\times d}$ be a constant matrix and $\mathbf{b} \in \mathbb{R}^n$ be a constant vector and define $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|_2^2$. If we let $\mathbf{a}^i \in \mathbb{R}^d$ denote the vector formed out of the i^{th} row of the matrix A, then we can see that we can rewrite the function as $f(\mathbf{x}) = \sum_{i=1}^n (\mathbf{x}^\top \mathbf{a}^i - \mathbf{b}_i)^2$. Using the sum rule for

gradients gives us $\nabla f(\mathbf{x}) = 2\sum_{i=1}^{n}(\mathbf{x}^\top \mathbf{a}^i - \mathbf{b}_i)\cdot \mathbf{a}^i$. Note that this is simply the sum of the vectors $\mathbf{a}^i$ multiplied by the scalar $c_i \triangleq 2(\mathbf{x}^\top \mathbf{a}^i - \mathbf{b}_i)$. If we let $\mathbf{c} \triangleq [c_1, \ldots, c_n]^\top = 2(A\mathbf{x} - \mathbf{b}) \in \mathbb{R}^n$, then we can rewrite the gradient very neatly as $\nabla f(\mathbf{x}) = A^\top \mathbf{c} = 2A^\top(A\mathbf{x} - \mathbf{b})$. Remember, the dimensionality rule tells us that the gradient must be a d-dimensional vector so the gradient cannot be something like $A\mathbf{c}$ which anyway does not make sense.

A.8.3 Useful Rules for Calculating Hessians

The Hessian of function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as the derivative $\nabla^2 f(\mathbf{x}) = \frac{d^2 f}{d\mathbf{x} d\mathbf{x}^\top}$. The $d\mathbf{x}d\mathbf{x}^\top$ expression is merely a stylized way of distinguishing the two applications of derivatives with respect to $\mathbf{x}$. We will find it more convenient to write the Hessian as $\frac{d}{d\mathbf{x}}(\nabla f)$ or else as $\nabla^2 f(\mathbf{x}) = \frac{dg}{d\mathbf{x}}$ where $g(\mathbf{x}) = \nabla f(\mathbf{x})$. Note that $g : \mathbb{R}^d \to \mathbb{R}^d$ is a vector valued function that maps every point $\mathbf{x}^0 \in \mathbb{R}^d$ to the gradient of f at $\mathbf{x}^0$ i.e. $\nabla f(\mathbf{x}^0) \in \mathbb{R}^d$.

The dimensionality rules tell us that since $\nabla^2 f(\mathbf{x}) = \frac{dg}{d\mathbf{x}}$, it must be a $d \times d$ matrix (also symmetric if the function is nice). As before, we present here a few handy shortcuts for calculating Hessians. We remind the reader that every single one of these rules can be derived by simply applying the aforementioned rules of univariate calculus and dimensionality rules carefully.

Specifically, all we need to do is think of the j^{th} coordinate of the g function (recall that the output of g is a d-dimensional vector) as a univariate function $g_j : \mathbb{R}^d \to \mathbb{R}$, take the gradient of this univariate function using our usual rules, and then set the j^{th} row of the Hessian matrix as $(\nabla g_j)^\top$.

1. (Linear Rule) If $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ where $\mathbf{a}$ is a constant vector, then $g(\mathbf{x}) = \nabla f(\mathbf{x}) = \mathbf{a}$ and thus $\nabla g_j = \mathbf{0}$ for all j (using the constant rule) and thus $\nabla^2 f(\mathbf{x}) = \mathbf{0}\mathbf{0}^\top \in \mathbb{R}^{d\times d}$. Thus, linear (or even affine functions such as $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} + b$ where $b \in \mathbb{R}$ is a constant) have zero Hessians. Note that the dimensionality rule tells us that although the Hessian is zero, it is a zero matrix, not a zero vector or the scalar zero.

2. (Quadratic Rule) If $f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}$ where $A \in \mathbb{R}^{d\times d}$ is a constant symmetric matrix that is not a function of $\mathbf{x}$, then $g(\mathbf{x}) = \nabla f(\mathbf{x}) = 2A\mathbf{x}$. If we let $\mathbf{a}^j$ denote the vector formed out of the j^{th} row of the matrix A, then $g_j(\mathbf{x}) = 2\mathbf{x}^\top \mathbf{a}^i$. Thus, we have $\nabla g_j = 2\mathbf{a}^j$ using the dot product rule for gradients. Applying the dimensionality rule then tells us that $\nabla^2 f(\mathbf{x}) = \frac{dg}{d\mathbf{x}} = 2A \in \mathbb{R}^{d\times d}$. If A is not symmetric, then $\nabla^2 f(\mathbf{x}) = A + A^\top$.

We now illustrate the use of these rules using some examples

Example A.7. Let $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$. We can rewrite this as $f = \frac{1}{2}\mathbf{x}^\top \mathbf{x} = \frac{1}{2}\mathbf{x}^\top I\mathbf{x}$ where I is the $d\times d$ identity matrix. The scaling rule and the Hessian quadratic rule tell us that $\nabla^2 f(\mathbf{x}) = I$ since the identity matrix is symmetric.

Example A.8. Let $f(\mathbf{x}) = (\mathbf{a}^\top \mathbf{x} - b)^2$ where $\mathbf{a} \in \mathbb{R}^d$ is a constant vector and $b \in \mathbb{R}$ is a constant scalar. From Example A.5, we get $g = \nabla f(\mathbf{x}) = 2(\mathbf{a}^\top \mathbf{x} - b)\cdot \mathbf{a}$. Thus, we get $g_j = 2(\mathbf{a}^\top \mathbf{x} - b)\mathbf{a}_j$ where $\mathbf{a}_j$ is the j^{th} coordinate of the vector $\mathbf{a}$. Using the scaling rule and gradient dot product rule we get $\nabla g_j = 2\mathbf{a}_j \cdot \mathbf{a}$.

Thus, the j^{th} row of the Hessian is the vector $\mathbf{a}^\top$ (transposed since it is a row vector) scaled by $2\mathbf{a}_j$. Pondering on this for a moment will tells us that this means that $\nabla^2 f(\mathbf{x}) = 2 \cdot \mathbf{a}\mathbf{a}^\top \in \mathbb{R}^{d\times d}$.

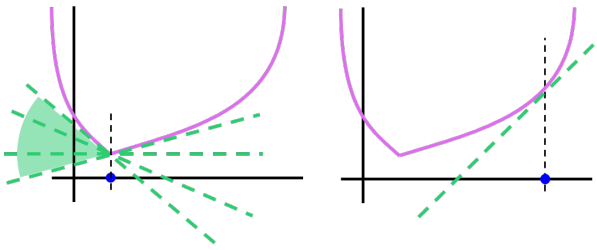
Example A.9. Let $A \in \mathbb{R}^{n\times d}$ be a constant matrix and $\mathbf{b} \in \mathbb{R}^n$ be a constant vector and define $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|_2^2$. From Example A.6, we get $g = \nabla f(\mathbf{x}) = 2\sum_{i=1}^n (\mathbf{x}^\top \mathbf{a}^i - \mathbf{b}_i)\cdot \mathbf{a}^i = A^\top(A\mathbf{x}-\mathbf{b})$, where $\mathbf{a}^i \in \mathbb{R}^d$ denotes the vector formed out of the i^{th} row of the matrix A. It is useful to clarify here that although the vector $\mathbf{a}^i$ was formed out of a row of a matrix, the vector itself is a column vector as per our convention for vectors. Using the sum rule for gradients gives us $\nabla g_j = 2\sum_{i=1}^n \mathbf{a}_j^i \cdot \mathbf{a}^i$. Similarly as in the above example, we can deduce that this means that $\nabla^2 f(\mathbf{x}) = 2\sum_{i=1}^n \mathbf{a}^i(\mathbf{a}^i)^\top = 2 \cdot A^\top A \in \mathbb{R}^{d\times d}$.

A.9 Subdifferential Calculus

The notions of derivatives of various orders discussed above at least assume that the function is differentiable. If that is not the case, then gradients (and by extension Hessians) cannot be defined. However, one can still define some nice extensions of the notion of gradient if the function is nevertheless convex (refer to Chapter B for notions of convexity for non-differentiable functions).

Recall that the tangent definition of convexity (see § B.2.3) demands that the function always lie above all its tangent hyperplanes i.e. $f(\mathbf{x}) \geq \nabla f(\mathbf{x}^0)^\top(\mathbf{x}-\mathbf{x}^0) + f(\mathbf{x}^0)$ for all $\mathbf{x}$. For non-differentiable functions, The key to extending the notion of gradient to non-differentiable convex functions is to take this definition of convexity and turn it on its head.

Note that a hyperplane $t(\mathbf{x})$ that touches the surface of a function f (even if f is non-differentiable) at a point $\mathbf{x}^0$ must necessarily be of the form $t(\mathbf{x}) = \mathbf{g}^\top(\mathbf{x}-\mathbf{x}^0) + f(\mathbf{x}^0)$ (since we must have $t(\mathbf{x}^0) = f(\mathbf{x}^0)$). The trick is to simply uses this to define any vector $\mathbf{g}$ such that $f(\mathbf{x}) \geq \mathbf{g}^\top(\mathbf{x}-\mathbf{x}^0) + f(\mathbf{x}^0)$ for all $\mathbf{x}$ as a subgradient of f at the point $\mathbf{x}^0$. Note that such a definition has some interesting properties. subgradient



Firstly, note that if we apply this definition to differentiable convex functions then we conclude that the gradient $\nabla f(\mathbf{x}^0)$ is a subgradient of f at $\mathbf{x}^0$. In fact it is the *only* subgradient of f at $\mathbf{x}^0$ if f is differentiable at $\mathbf{x}^0$. We emphasize this because if f is not differentiable at $\mathbf{x}^0$, then f may have multiple subgradients (in general an infinite number of subgradients) at $\mathbf{x}^0$. The set of all subgradients at $\mathbf{x}^0$ is called the subdifferential of f at $\mathbf{x}^0$ and denoted as follows subdifferential

$$\partial f(\mathbf{x}^0) = \left\{\mathbf{g} \in \mathbb{R}^d : f(\mathbf{x}) \geq \mathbf{g}^\top(\mathbf{x}-\mathbf{x}^0) + f(\mathbf{x}^0) \text{ for all } \mathbf{x}\right\}$$

Note that the above discussion indicates that if f is differentiable at $\mathbf{x}^0$ then $\partial f(\mathbf{x}^0) = \{\nabla f(\mathbf{x}^0)\}$ i.e. the set has only one element.

A.9.1 Rules of Subgradient Calculus

Several rules exist that can help us calculate the subdifferential of complex-looking non-differentiable functions with relative ease. These (except for the max rule) do share parallels with the rules we saw for regular calculus but with some key differences. These are given below followed by some examples applying them to problems. In the following, $\mathbf{a} \in \mathbb{R}^d$ is a constant vector and $b \in \mathbb{R}$ is a constant scalar.

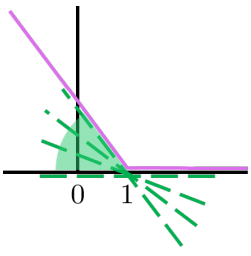
1. (Scaling Rule) If $h(\mathbf{x}) = c \cdot f(\mathbf{x})$ and if c is not a function of $\mathbf{x}$ then $\partial h(\mathbf{x}) = c \cdot \partial f(\mathbf{x}) \triangleq \{c \cdot \mathbf{u} : \mathbf{u} \in \partial f(\mathbf{x})\}$
2. (Sum Rule) If $h(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$ then $\partial h(\mathbf{x}) = \partial f(\mathbf{x}) + \partial g(\mathbf{x}) \triangleq \{\mathbf{u} + \mathbf{v} : \mathbf{u} \in \partial f(\mathbf{x}), \mathbf{v} \in \partial g(\mathbf{x})\}$. Note that here, we are defining the sum of two sets as the Minkowski sum. Minkowski sum
3. (Chain Rule) If $h(\mathbf{x}) = f(\mathbf{a}^\top \mathbf{x} + b)$, then $\partial h(\mathbf{x}) = \partial f(\mathbf{a}^\top \mathbf{x} + b) \cdot \mathbf{a} \triangleq \{c \cdot \mathbf{a} : c \in \partial f(\mathbf{x})\}$
4. (Max Rule) If $h(\mathbf{x}) = \max\{f(\mathbf{x}), g(\mathbf{x})\}$, then the following cases apply:
   (a) If $f(\mathbf{x}) > g(\mathbf{x})$, then $\partial h(\mathbf{x}) = \partial f(\mathbf{x})$
   (b) If $f(\mathbf{x}) < g(\mathbf{x})$, then $\partial h(\mathbf{x}) = \partial g(\mathbf{x})$
   (c) If $f(\mathbf{x}) = g(\mathbf{x})$, then $\partial h(\mathbf{x}) = \{\lambda \cdot \mathbf{u} + (1-\lambda) \cdot \mathbf{v} : \mathbf{u} \in \partial f(\mathbf{x}), \mathbf{v} \in \partial g(\mathbf{x}), \lambda \in [0,1]\}$.

Note that the max rule has no counterpart in regular calculus since functions of the form $h(\mathbf{x}) = f(\mathbf{x}^\top \mathbf{x} + b)$ are usually non-differentiable.

A.9.2 Stationary Points for Non-differentiable Functions

The notion of stationary points does extend to non-differentiable convex functions as well. A point $\mathbf{x}^0$ is called a stationary point for a function f if $\mathbf{0} \in \partial f(\mathbf{x}^0)$ i.e. the zero vector is a part of its subdifferential.

It is noteworthy that even for non-differentiable convex functions, global minima must be stationary points in this sense and vice versa. This is easy to see – suppose that we do have $\mathbf{0} \in \partial f(\mathbf{x}^0)$, then the definition of subgradients for convex function dictates that we must have $f(\mathbf{x}) \geq \mathbf{0}^\top(\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0)$ for all $\mathbf{x}$ which is the same as saying $f(\mathbf{x}) \geq f(\mathbf{x}^0)$ for all $\mathbf{x}$ which is precisely the definition of a global minimum. Thus, $\mathbf{x}^0$ is a global minimum iff $\mathbf{0} \in \partial f(\mathbf{x}^0)$. iff ≡ if and only if



Example A.10. Let $\ell(x) = [1 - x, 0]_+$ denote the hinge loss function. To calculate its subdifferential, we note that we can write this function as $\ell(x) = \max\{f(x), g(x)\}$ where $f(x) = 1 - x$ and $g(x) = 0$. Note that f, g are both differentiable functions. Applying the max rule, we get at the point when $f(x) = g(x)$ i.e. at $x = 1$, we have $\partial \ell(x) = \{\lambda \cdot (-1) + (1-\lambda) \cdot (0) : \lambda \in [0,1]\}$ since $f(x) = -1$

and $g'(x) = 0$ for all x. Thus, we have

$$\partial \ell(x) = \begin{cases} -1 & \text{if } x < 1 \\ 0 & \text{if } x > 1 \\ [-1, 0] & \text{if } x = 1 \end{cases}$$

Example A.11. Let $\ell(\mathbf{w}) = [1 - y \cdot \mathbf{w}^\top \mathbf{x}, 0]_+$ denote the hinge loss function applied to a data point $(\mathbf{x}, y)$ along with the model $\mathbf{w}$. Applying the chain rule gives us

$$\partial \ell(\mathbf{w}) = \begin{cases} -y \cdot \mathbf{x} & \text{if } y \cdot \mathbf{w}^\top \mathbf{x} < 1 \\ \mathbf{0} & \text{if } y \cdot \mathbf{w}^\top \mathbf{x} > 1 \\ cy \cdot \mathbf{x} & \text{if } y \cdot \mathbf{w}^\top \mathbf{x} = 1, \text{ where } c \in [-1, 0] \end{cases}$$

A.10 Exercises

Exercise A.1. Let $f(x) = x^4 - 4x^2 + 4$. Find all stationary points of this function. Which of them are local maxima and minima?

Exercise A.2. Let $g : \mathbb{R}^2 \to \mathbb{R}$ be defined as $g(x, y) = f(x) + f(y) + 8$ where f is defined in the execise above. Find all stationary points of this function. Which of them are local maxima and minima? Which one of these are saddle points?

Exercise A.3. Given a natural number $n \in \mathbb{N}$ e.g. $2, 8, 97$ and a real number $x^0 \in \mathbb{R}$, design a function $f : \mathbb{R} \to \mathbb{R}$ so that $f^{(k)}(x^0) = 0$ for all $k = 1, 2, \ldots, n$. Here $f^{(k)}(x^0)$ denotes the k^{th} order derivative of f at x^0 e.g. $f^{(1)}(x^0) = f'(x^0), f^{(3)}(x^0) = f'''(x^0)$ etc.

Exercise A.4. Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ be constant vectors and let $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} \mathbf{x}^\top \mathbf{b}$. Calculate $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$. updated exercise
Hint: write $f(\mathbf{x}) = g(\mathbf{x}) \cdot h(\mathbf{x})$ where $g(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$ and $h(\mathbf{x}) = \mathbf{b}^\top \mathbf{x}$ and apply the product rule.

Exercise A.5. Let $\mathbf{b} \in \mathbb{R}^d$ a constant vector and $A \in \mathbb{R}^{d \times d}$ be a constant symmetric matrix. Let $f(\mathbf{x}) = \mathbf{b}^\top A \mathbf{x}$. Calculate $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$. updated exercise
Hint: write $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}$ where $\mathbf{c} = A^\top \mathbf{b}$.

Exercise A.6. Let $A, B, C \in \mathbb{R}^{d \times d}$ be three symmetric and constant matrices and $\mathbf{p}, \mathbf{q} \in \mathbb{R}^d$ be two constant vectors. Let $f(\mathbf{x}) = (A\mathbf{x} + \mathbf{p})^\top C(B\mathbf{x} + \mathbf{q})$. Calculate $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$. updated exercise

Exercise A.7. Suppose we have n constant vectors $\mathbf{a}^1, \ldots, \mathbf{a}^n \in \mathbb{R}^d$. Let $f(\mathbf{x}) = \sum_{i=1}^n \ln\left(1 + \exp(-\mathbf{x}^\top \mathbf{a}^i)\right)$. Calculate $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$. new exercise

Exercise A.8. Let $\mathbf{a} \in \mathbb{R}^d$ be a constant vector and let $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} \cdot \|\mathbf{x}\|_2^2$. Calculate $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$. new exercise
Hint: the expressions may be more tedious with this one. Be patient and apply the product rule carefully to first calculate the gradient. Then move on to the Hessian by applying the dimensionality rule.

Exercise A.9. Show that for any convex function f (whether differentiable or not), its subdifferential at any point $\mathbf{x}^0$, i.e. $\partial f(\mathbf{x}^0)$, is always a convex set. new exercise

Exercise A.10. For a vector $\mathbf{x} \in \mathbb{R}^d$ its L_1 norm is defined as $\|\mathbf{x}\|_1 \triangleq \sum_{j=1}^d |\mathbf{x}_j|$. Let $f(\mathbf{x}) \triangleq \|\mathbf{x}\|_1 + [1 - \mathbf{x}^\top \mathbf{a}]_+$ where $\mathbf{a} \in \mathbb{R}^d$ is a constant vector. Find the subdifferential $\partial f(\mathbf{x})$. new exercise

Exercise A.11. Let $f(\mathbf{x}) = \max\left\{(\mathbf{x}^\top \mathbf{a} - b)^2, c\right\}$ where $\mathbf{a} \in \mathbb{R}^d$ is a constant vector and $b, c \in \mathbb{R}$ are constant scalars. Find the subdifferential $\partial f(\mathbf{x})$. new exercise

Exercise A.12. Let $\mathbf{x}, \mathbf{a} \in \mathbb{R}^d, b \in \mathbb{R}$ where $\mathbf{a}$ is a constant vector that does not depend on $\mathbf{x}$ and b is a constant real number that does not depend on $\mathbf{x}$. Let $f(\mathbf{x}) = \left|\mathbf{a}^\top \mathbf{x} - b\right|$. Find the subdifferential $\partial f(\mathbf{x})$. new exercise

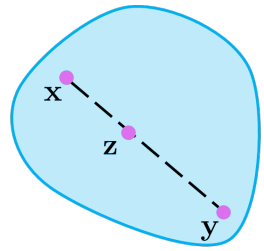
Exercise A.13. Now suppose we have n constant vectors $\mathbf{a}^1, \ldots, \mathbf{a}^n \in \mathbb{R}^d$ and n real constants $b_1, \ldots, b_n \in \mathbb{R}$. Let $f(\mathbf{x}) = \sum_{i=1}^n \left|\mathbf{x}^\top \mathbf{a}^i - b_i\right|$. Find the subdifferential $\partial f(\mathbf{x})$. new exercise

B

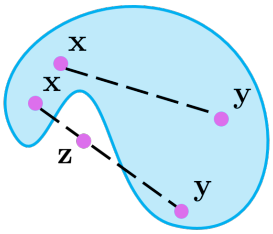
Convex Analysis Refresher

Convex sets and functions remain the favorites of practitioners working on machine learning algorithms since these objects have several beautiful properties that make it simple to design efficient algorithms. Of course, the recent years have seen several strides in *non-convex optimization* as well due to areas such as deep learning, robust learning, sparse learning gaining prominence.

B.1 Convex Set



CONVEX SET



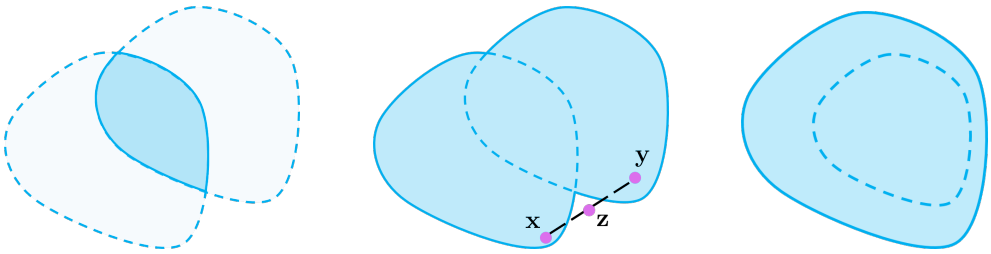
NON-CONVEX SET

Given a set of points (or a region) $\mathcal{C} \subset \mathbb{R}^d$, we call this set or region a convex set if the set contains all line segments that join two points inside that set. More formally, for a set $\mathcal{C}$ to be convex, no matter which two points we take in the set $\mathbf{x}, \mathbf{y} \in \mathcal{C}$, for every $\lambda \in [0, 1]$, we must have $\mathbf{z} \in \mathcal{C}$ where $\mathbf{z} = \lambda \cdot \mathbf{x} + (1 - \lambda) \cdot \mathbf{y}$. It is noteworthy that the vectors $\mathbf{z}$ defined this way completely capture all points on the line segment joining $\mathbf{x}$ and $\mathbf{y}$. Indeed, with $\lambda = 0$, we have $\mathbf{z} = \mathbf{z}$, $\lambda = 1$ gives us $\mathbf{z} = \mathbf{y}$ and $\lambda = 0.5$ gives us the midpoint of the line segment. It is noteworthy however, that we must have $\lambda \in [0, 1]$. If λ starts taking negative values or values greater than 1, then we would start getting points outside the line segment.

convex set

For well behaved sets, in order to confirm convexity, it is sufficient to verify that $\mathbf{z} = \frac{\mathbf{x}+\mathbf{y}}{2} \in \mathcal{C}$ i.e. we need not take the trouble of verifying for all $\lambda \in [0, 1]$ and simply verifying mid-point convexity is enough to verify convexity (we do need to still verify this for all $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ though). It is also important to note that non-convex sets, such as the one depicted in the figure, may contain *some* of the line segments that join points within them – this does not make the set convex! Only if a

mid-point convexity

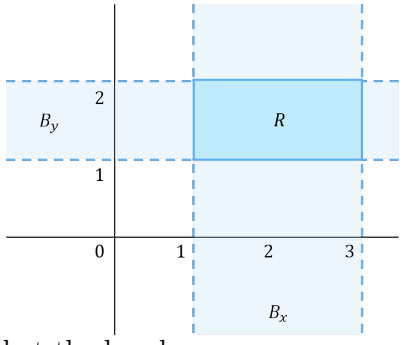


set contains *all* its line segments is it called convex. The reader would have noticed that convex sets *bulge* outwards in all directions. The presence of any inward bulges typically makes a set non-convex.

Theorem B.1. Given two convex sets $\mathcal{C}_1, \mathcal{C}_2 \subset \mathbb{R}^d$, the intersection of these two sets i.e. $\mathcal{C}_1 \cap \mathcal{C}_2$ is always convex. However, the union of these two sets i.e. $\mathcal{C}_1 \cup \mathcal{C}_2$ need not be convex.

Proof. We first deal with the case of intersection. The intersection of two sets (not necessarily convex) is defined to be the set of all points that are contained in both the sets i.e. $\mathcal{C}_1 \cap \mathcal{C}_2 \triangleq \left\{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} \in \mathcal{C}_1 \text{ and } \mathbf{x} \in \mathcal{C}_2\right\}$. Consider two points $\mathbf{x}, \mathbf{y} \in \mathcal{C}_1 \cap \mathcal{C}_2$. Since $\mathbf{x}, \mathbf{y} \in \mathcal{C}_1$, we know that $\mathbf{z} = \frac{\mathbf{x}+\mathbf{y}}{2} \in \mathcal{C}_1$ since $\mathcal{C}_1$ is convex. However, by the same argument, we get that $\mathbf{z} = \frac{\mathbf{x}+\mathbf{y}}{2} \in \mathcal{C}_2$ as well. Since $\mathbf{z} \in \mathcal{C}_1$ and $\mathbf{z} \in \mathcal{C}_2$, we conclude that $\mathbf{z} \in \mathcal{C}_1 \cap \mathcal{C}_2$. This proves that the intersection of any two convex sets must necessarily be convex. The first figure above illustrates the intersection region of two convex sets. intersection of two sets

The union of two sets (not necessarily convex) is defined to be the set of all points that are contained in either of the sets (including points that are present in both sets). More specifically, we define $\mathcal{C}_1 \cap \mathcal{C}_2 \triangleq \left\{\mathbf{x} \in \mathbb{R}^d : \mathbf{x} \in \mathcal{C}_1 \text{ or } \mathbf{x} \in \mathcal{C}_2\right\}$. The second figure above shows that the union of two convex sets may be non-convex. However, the union of two convex sets may be convex in some very special cases, for example, if one set is contained in the other i.e. $\mathcal{C}_1 \subseteq \mathcal{C}_2$ which is illustrated in the third figure. □ union of two sets



Example B.1. Are rectangles convex? Let $R \triangleq \{(x, y) \in \mathbb{R}^2 : x \in [1, 3] \text{ and } y \in [1, 2]\}$ be a rectangle with side lengths 1 and 2. We could show R to be convex directly as we do in the example below. However, there exists a neater way. Consider the bands $B_x \triangleq \{(x, y) \in \mathbb{R}^2 : x \in [1, 3]\}$ and $B_y \triangleq \{(x, y) \in \mathbb{R}^2 : y \in [1, 2]\}$. It is easy to see that $R = B_x \cap B_y$. Thus, if we show that the bands are convex, we could then use Theorem B.1 to show that R is convex too! Showing that B_x is convex is pretty easy: if two points have their x coordinate in the range $[1, 3]$, then the average of those two points clearly satisfies this as well. This establishes that B_x is convex. Similarly B_y is convex which tells us that R is convex.

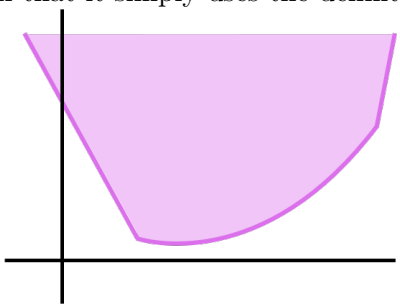
Example B.2. Consider the set of all points which are at a Euclidean distance at most 1 from the origin i.e. the unit ball $\mathcal{B}_2(1) \triangleq \left\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq 1\right\}$. To show that this set is convex, we take $\mathbf{x}, \mathbf{y} \in \mathcal{B}_2(1)$ and consider $\mathbf{z} = \frac{\mathbf{x}+\mathbf{y}}{2}$. Now, instead of showing $\|\mathbf{z}\|_2 \leq 1$ (which will establish convexity), we will instead show $\|\mathbf{z}\|_2^2 \leq 1$ which is equivalent but easier to analyze. We have $\|\mathbf{z}\|_2^2 = \left\|\frac{\mathbf{x}+\mathbf{y}}{2}\right\|_2^2 = \frac{\|\mathbf{x}\|_2^2+\|\mathbf{y}\|_2^2+2\mathbf{x}^\top\mathbf{y}}{4}$. Now, recall that the Cauchy-Schwartz inequality tells us that for any two vectors $\mathbf{a}, \mathbf{b}$ we have $\left|\mathbf{a}^\top\mathbf{b}\right| \leq \|\mathbf{a}\|_2 \|\mathbf{b}\|_2$. Thus, we have $\|\mathbf{z}\|_2^2 \leq \frac{\|\mathbf{x}\|_2^2+\|\mathbf{y}\|_2^2+2\|\mathbf{x}\|_2\|\mathbf{y}\|_2}{4}$. Since $\mathbf{x}, \mathbf{y} \in \mathcal{B}_2(1)$, we have $\|\mathbf{x}\|_2, \|\mathbf{y}\|_2 \leq 1$ which gives us $\|\mathbf{z}\|_2^2 \leq 1$ which establishes the unit ball $\mathcal{B}_2(1)$ is a convex set.

B.2 Convex Functions

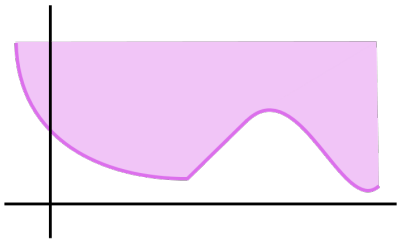
We now move on to convex functions. These functions play an important role in several optimization based machine learning algorithms such as SVMs and logistic regression. There exist several definitions of convex functions, some that apply only to differentiable functions, and others that apply even to non-differentiable functions. We look at these below.

B.2.1 Epigraph Convexity

This is the most fundamental definition of convexity and applies to all functions, whether they are differentiable or not. This definition is also quite neat in that it simply uses the definition of convex sets to define convex functions.



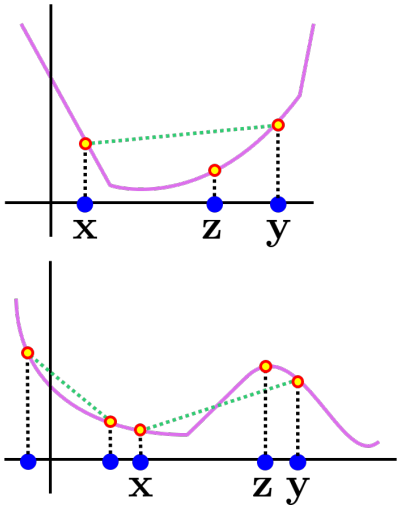
Definition B.1 (Epigraph)**.** Given a function $f : \mathbb{R}^d \to \mathbb{R}$, its epigraph is defined as the set of points that lie on or above the graph of the function i.e. $\text{epi}(f) \triangleq \left\{(\mathbf{x}, y) \in \mathbb{R}^{d+1} : y \geq f(\mathbf{x})\right\}$. Note that the epigraph is a $d+1$-dimensional set and not a d dimensional set. Epigraph



Definition B.2 (Epigraph Convexity)**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is defined to be convex if its epigraph $\text{epi}(f) \in \mathbb{R}^{d+1}$ is a convex set. On the left, we have a non-convex function whose epigraph is a non-convex set (notice the inward bulge) whereas in the figure above, we have a convex function whose epigraph is a convex set. Epigraph Convexity

B.2.2 Chord Convexity

The above definition, although fundamental, is not used quite often since there exist simpler definitions. One of these definitions exploits the fact that convexity of the epigraph set need only be verified at the lower boundary of the set i.e. at the surface of the function graph. Applying the mid-point definition of convex sets then gives us this new definition of convex functions.



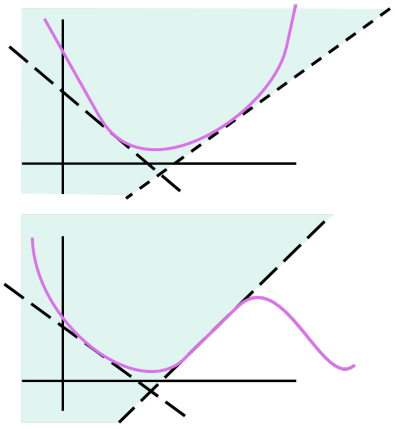
Definition B.3 (Chord). Given a function $f : \mathbb{R}^d \to \mathbb{R}$ and any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the line segment joining the two points $(\mathbf{x}, f(\mathbf{x}))$ and $(\mathbf{y}, f(\mathbf{y}))$ is called a chord of this function. Chord

Definition B.4 (Chord Convexity). A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if the function graph lies below all its chords. Using the mid-point definition, this is equivalent to saying that a function is convex if and only if for any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ we have $f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \leq \frac{f(\mathbf{x})+f(\mathbf{y})}{2}$. Chord Convexity

The figures depict a convex function that lies above all its chords and a non-convex function which does not do so. It is also important to note that a non-convex function may lie below *some* of its chords (as the figure on the bottom shows) – this does not make the function convex! Only if a function lies below *all* its chords is it called convex.

B.2.3 Tangent Convexity

This definition holds true only for differentiable functions but is usually easier to apply when checking whether a function is convex or not.



Definition B.5 (Tangent). Given a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, the tangent of the function at a point $\mathbf{x}^0 \in \mathbb{R}^d$ is the hyperplane $\nabla f(\mathbf{x}^0)^\top(\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0) = 0$ i.e. of the form $\mathbf{w}^\top\mathbf{x} + b$ where $\mathbf{w} = \nabla f(\mathbf{x}^0)$ and $b = f(\mathbf{x}^0) - \nabla f(\mathbf{x}^0)^\top\mathbf{x}^0$. Tangent

Definition B.6 (Tangent Convexity). A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if the function graph lies above all its tangents i.e. for all $\mathbf{x}^0, \mathbf{x} \in \mathbb{R}^d$, we have $f(\mathbf{x}) \geq \nabla f(\mathbf{x}^0)^\top(\mathbf{x} - \mathbf{x}^0) + f(\mathbf{x}^0)$. Tangent Convexity

Note that the point $(\mathbf{x}^0, f(\mathbf{x}^0))$ always lies on the tangent hyperplane at $\mathbf{x}^0$. The figures above depict a convex function that lies above all its tangents and a non-convex function which fails to lie above at least one of its tangents. It is important to note that non-convex functions may lie above *some* of their tangenst (as the figure on the bottom shows) – this does not make the function convex! Only if a functions lies above *all* its tangents is it called convex.

It is also useful to clarify that the epigraph and chord definitions of convexity continue to apply here as well. Its is just that the tangent definition is easier to use in several cases. A rough analogy is that of deciding the income of individuals – although we can find out the total income of any citizen of India, it may be tedious to do so. However, it is much easier to find the income

of a person if that person files income tax returns (truthfully, of course).

B.2.4 Hessian Convexity

Hessian Convexity

For doubly differentiable functions, we have an even simpler definition of convexity. A doubly differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if its Hessian is positive semi-definite at all points i.e. $\nabla^2 f(\mathbf{x}^0) \succeq 0$ for all $\mathbf{x}^0 \in \mathbb{R}^d$. Recall that this implies that for all $\mathbf{v} \in \mathbb{R}^d$, we have $\mathbf{v}^\top \nabla^2 f(\mathbf{x}^0)\mathbf{v} \geq 0$.

B.2.5 Concave Functions

Concave functions

Concave functions are defined as those whose negative is a convex function i.e. $f : \mathbb{R}^d \to \mathbb{R}$ is defined to be concave if the function $-f$ is convex. Convex functions typically look like upturned cups (think of the function $f(x) = x^2$ which is convex and looks like a right-side-up cup). Concave functions on the other hand look like inverted cups, for example $f(x) = -x^2$. To check whether a function is concave or not, we need to simply check (using the epigraph, chord, tangent, or Hessian methods) whether the negative of that function is convex or not.

Example B.3. Let us look at the example of the Euclidean norm $f(\mathbf{x}) = \|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}}$. This function is non-differentiable at the origin i.e. at $\mathbf{x} = \mathbf{0}$ so we have to use the chord definition of convexity. Given two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have $f\left(\frac{\mathbf{x}+vy}{2}\right) = \left\|\frac{\mathbf{x}+\mathbf{y}}{2}\right\| = \frac{1}{2}\|\mathbf{x}+\mathbf{y}\|$ by using the homogeneity property of the Euclidean distance (if we halve a vector, its length gets halved too). However, recall that the triangle inequality tells us that for any two vectors $\mathbf{p}, \mathbf{q}$, we have $\|\mathbf{p}+\mathbf{q}\|_2 \leq \|\mathbf{p}\|_2 + \|\mathbf{q}\|_2$. This gives us $f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \leq \frac{\|\mathbf{x}\|_2+\|\mathbf{y}\|_2}{2} = \frac{f(\mathbf{x})+f(\mathbf{y})}{2}$ which proves the convexity of the norm.

B.3 Operations with Convex Functions

We can take convex functions and manipulate them to obtain new convex functions. Here we explore some such operations that are useful in machine learning applications.

1. Affine functions are always convex[1].
2. Scaling a convex function by a positive scale factor always yields a convex function[2].
3. The sum of two convex functions is always convex (see Theorem B.2).
4. If $g : \mathbb{R}^d \to \mathbb{R}$ is a (multivariate) convex function and $f : \mathbb{R} \to \mathbb{R}$ is a (univariate) convex and non-decreasing function i.e. $a \leq b \Leftrightarrow f(a) \leq f(b)$, then the function $h \triangleq f \circ g$ i.e. $h(\mathbf{x}) = f(g(\mathbf{x}))$ is also convex (see Theorem B.3).

[1] See Exercise B.6.
[2] See Exercise B.7.

5. If $f : \mathbb{R} \to \mathbb{R}$ is a (univariate) convex function then for any $\mathbf{a} \in \mathbb{R}^d, b \in \mathbb{R}$, the (multivariate) function $g : \mathbb{R}^d \to \mathbb{R}$ defined as $g(\mathbf{x}) = f(\mathbf{a}^\top \mathbf{x} + b)$ is always convex (see Theorem B.4).

6. If $f, g : \mathbb{R}^d \to \mathbb{R}$ are two (multivariate) convex functions then the function $h \triangleq \max\{f, g\}$ is also convex (see Theorem B.5).

Theorem B.2 (Sum of Convex Functions)**.** Given two convex functions $f, g : \mathbb{R}^d \to \mathbb{R}$, the function $h \triangleq f + g$ is always convex.

Proof. We will use the chord definition of convexity here since there is no surety that f and g are differentiable. Consider two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. We have

$$\begin{aligned} h\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) &= f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) + g\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \\ &\leq \frac{f(\mathbf{x}) + f(\mathbf{y})}{2} + \frac{g(\mathbf{x}) + g(\mathbf{y})}{2} \\ &= \frac{(f(\mathbf{x}) + g(\mathbf{x})) + (f(\mathbf{y}) + g(\mathbf{y}))}{2} = \frac{h(\mathbf{x}) + h(\mathbf{y})}{2} \end{aligned}$$

where in the second step, we used the fact that f and g are both convex. This proves that h is convex by the chord definition of convexity. □

Theorem B.3 (Composition of Convex Functions)**.** Suppose $g : \mathbb{R}^d \to \mathbb{R}$ is a (multivariate) convex function and $f : \mathbb{R} \to \mathbb{R}$ is a (univariate) convex and non-decreasing function i.e. $a \leq b \Leftrightarrow f(a) \leq f(b)$, then the function $h \triangleq f \circ g$ i.e. $h(\mathbf{x}) = f(g(\mathbf{x}))$ is always convex.

Proof. We will use the chord definition of convexity here since there is no surety that f and g are differentiable. Consider two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. We have

$$h\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) = f\left(g\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)\right)$$

Now, since g is convex, we have

$$g\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \leq \frac{g(\mathbf{x}) + g(\mathbf{y})}{2}$$

Let us denote the left hand side of the above inequality by p and the right hand side by q for sake of notational simplicity. Thus, the above inquality tells us that $p \leq q$. However, since f is non-decreasing, we get $f(p) \leq f(q)$ i.e.

$$f\left(g\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)\right) \leq f\left(\frac{g(\mathbf{x}) + g(\mathbf{y})}{2}\right)$$

Let us denote $u \triangleq g(\mathbf{x})$ and $v \triangleq g(\mathbf{y})$ for sake of notational simplicity. Since f is convex, we have

$$f\left(\frac{u+v}{2}\right) \leq \frac{f(u) + f(v)}{2}$$

This is the same as saying

$$f\left(\frac{g(\mathbf{x}) + g(\mathbf{y})}{2}\right) \leq \frac{f(g(\mathbf{x})) + f(g(\mathbf{y}))}{2} = \frac{h(\mathbf{x}) + h(\mathbf{y})}{2}$$

Thus, with the chain of inequalities established above, we have shown that

$$h\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) \leq \frac{h(\mathbf{x})+h(\mathbf{y})}{2}$$

which proves that h is a convex function. □

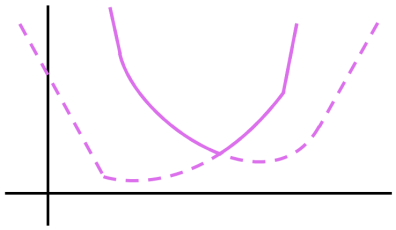
Theorem B.4 (Convex Wrappers over Affine Functions)**.** If $f : \mathbb{R} \to \mathbb{R}$ is a (univariate) convex function then for any $\mathbf{a} \in \mathbb{R}^d, b \in \mathbb{R}$, the (multivariate) function $g : \mathbb{R}^d \to \mathbb{R}$ defined as $g(\mathbf{x}) = f(\mathbf{a}^\top \mathbf{x} + b)$ is always convex.

Proof. We will yet again use the chord definition of convexity. Consider two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. We have

$$\begin{aligned} g\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) &= f\left(\mathbf{a}^\top\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)+b\right) \\ &= f\left(\frac{(\mathbf{a}^\top\mathbf{x}+b)+(\mathbf{a}^\top\mathbf{y}+b)}{2}\right) \\ &\leq \frac{f(\mathbf{a}^\top\mathbf{x}+b)+f(\mathbf{a}^\top\mathbf{y}+b)}{2} = \frac{g(\mathbf{x})+g(\mathbf{y})}{2} \end{aligned}$$

where in the second step we used the linearity of the dot product i.e. $\mathbf{c}^\top(\mathbf{a}+\mathbf{b}) = \mathbf{c}^\top\mathbf{a}+\mathbf{c}^\top\mathbf{b}$ and in the third step we used convexity of f. This shows that the function g is convex. □

Theorem B.5 (Maximum of Convex Functions)**.** If $f, g : \mathbb{R}^d \to \mathbb{R}$ are two (multivariate) convex functions then the function $h \triangleq \max\{f, g\}$ is also convex.



Proof. To prove this result, we will need the following simple monotonicity property of the max function: Let $a, b, c, d \in \mathbb{R}$ be four real numbers such that $a \leq c$ and $b \leq d$. Then we must have $\max\{a, b\} \leq \max\{c, d\}$. This can be shown in a stepwise manner ($\max\{a, b\} \leq \max\{c, b\} \leq \max\{c, d\}$) Now, consider two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. We have

$$\begin{aligned} &h\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right) = \max\left\{f\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right), g\left(\frac{\mathbf{x}+\mathbf{y}}{2}\right)\right\} \\ &\leq \max\left\{\frac{f(\mathbf{x})+f(\mathbf{y})}{2}, \frac{g(\mathbf{x})+g(\mathbf{y})}{2}\right\} \\ &\leq \max\left\{\frac{\max\{f(\mathbf{x}),g(\mathbf{x})\}+\max\{f(\mathbf{y}),g(\mathbf{y})\}}{2}, \frac{g(\mathbf{x})+g(\mathbf{y})}{2}\right\} \\ &\leq \max\left\{\frac{\max\{f(\mathbf{x}),g(\mathbf{x})\}+\max\{f(\mathbf{y}),g(\mathbf{y})\}}{2}, \frac{\max\{f(\mathbf{x}),g(\mathbf{x})\}+\max\{f(\mathbf{y}),g(\mathbf{y})\}}{2}\right\} \\ &= \frac{\max\{f(\mathbf{x}),g(\mathbf{x})\}+\max\{f(\mathbf{y}),g(\mathbf{y})\}}{2} = \frac{h(\mathbf{x})+h(\mathbf{y})}{2} \end{aligned}$$

where in the second step, we used the fact that f, g are convex functions and the monotonicity property of the max function. The third and the fourth steps also use the monotonicity property. The fifth step uses the fact that $\max\{a, a\} = a$. This proves that h is a convex function. □

Example B.4. The functions $f(x) = \ln(x)$ and $g(x) = \sqrt{x}$ are concave. Since both of these are doubly differentiable functions, we may use the Hessian definition to decide their concavity. Recall that a function is concave if and only if its negation is convex. Let $p(x) = -\ln(x)$. Then $p''(x) = \frac{1}{x^2} \geq 0$ for all $x > 0$. This confirms that $p(x)$ is convex and that $\ln(x)$ is concave. Similarly, define $q(x) = -\sqrt{x}$. Then $q''(x) = \frac{1}{4x\sqrt{x}} \geq 0$ for all $x \geq 0$ which confirms that $q(x)$ is convex and that $\sqrt{x}$ is concave.

Example B.5. Let us show that squared Euclidean norm i.e. the function $h(\mathbf{x}) = \|\mathbf{x}\|_2^2$ is convex. We have already shown above that the function $g(\mathbf{x}) = \|\mathbf{x}\|_2$ is convex. We can write $h(\mathbf{x}) = f(g(\mathbf{x}))$ where $f(t) = t^2$. Now, $f''(t) = 1 > 0$ i.e. f is convex by applying the Hessian rule for convexity. Also, $\|\mathbf{x}\|_2 \geq 0$ for all $\mathbf{x} \in \mathbb{R}^d$ and the function f is indeed an increasing function on the positive half of the real line. Thus, Theorem B.3 tells us that $h(\mathbf{x})$ is convex.

Example B.6. Let us show that the hinge loss is a convex function $\ell_{\text{hinge}}(t) = \max\{1 - t, 0\}$. Note that the hinge loss function is treated as a univariate function here i.e. $\ell_{\text{hinge}} : \mathbb{R} \to \mathbb{R}$. Exercise B.6 shows us that affine functions are convex. Thus $f(x) = 1 - x$ and $g(x) = 0$ are both convex functions. Thus, by applying Theorem B.5, we conclude that the hinge loss function is convex.

Example B.7. We will now show that the objective function used in the C-SVM formulation is a convex function of the model vector $\mathbf{w}$. For sake of simplicity, we will show this result without the bias parameter b although we stress that the result holds even if the bias parameter is present (recall that the bias can always be hidden inside the model by adding a fake dimension into the data). Let $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$ be n data points with $\mathbf{x}^i \in \mathbb{R}^d$ and $y^i \in \{-1, 1\}$. Denote $\mathbf{z}^i \triangleq y^i \cdot \mathbf{x}^i$ for sake of notational simplicity. The the C-SVM objective function is reproduced below:

$$f_{\text{C-SVM}}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \ell_{\text{hinge}}(y^i \cdot \mathbf{w}^\top \mathbf{x}^i) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \ell_{\text{hinge}}(\mathbf{w}^\top \mathbf{z}^i)$$

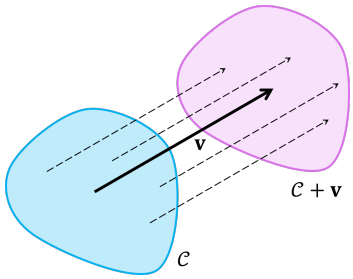
Note that the feature vectors $\mathbf{x}^i, i = 1, \ldots, n$ and the labels $y^i, i = i, \ldots, n$ (and hence the vectors $\mathbf{z}^i$) are treated as constants since we cannot change our training data. The only variable here is the model vector $\mathbf{w}$ which we learn using the training data. We have already shown that $\|\mathbf{w}\|_2^2$ is a convex function of $\mathbf{w}$, Exercise B.7 shows that $\frac{1}{2}\|\mathbf{w}\|_2^2$ is convex too. We showed above that ℓ_{hinge} is a convex function and thus, Theorem B.4 shows that $h_i(\mathbf{w}) = \ell_{\text{hinge}}(\mathbf{w}^\top \mathbf{z}^i)$ is a convex function of $\mathbf{w}$ for every i. Theorem B.2 shows that the sum of convex functions is convex which shows that $f_{\text{C-SVM}}(\mathbf{w})$ is a convex function.

B.4 Exercises

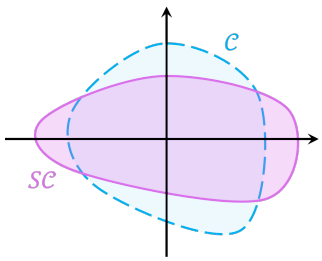
Exercise B.1. Let A be a positive semi-definite matrix and let us define a Mahalanobis distance using A as $d_A(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top A(\mathbf{x} - \mathbf{y})}$. Consider the unit ball according to this distance i.e. the set of all points that are

less than or equal to unit Mahalanobis distance from the origin i.e. $\mathcal{B}_A(1) \triangleq \left\{\mathbf{x} \in \mathbb{R}^d : d_A(\mathbf{x}, \mathbf{0}) \leq 1\right\}$. Show that $\mathcal{B}_A(1)$ is a convex set.

Exercise B.2. Consider the hyperplane given by the equation $\mathbf{w}^\top \mathbf{x} + b = 0$ i.e. $H \triangleq \left\{\mathbf{x} \in \mathbb{R}^d : \mathbf{w}^\top \mathbf{x} + b = 0\right\}$ where $\mathbf{w}$ is the normal vector to the hyperplane and b is the bias term. Show that H is a convex set.



Exercise B.3. If I take a convex set and shift it, does it remain convex? Let $\mathcal{C} \subset \mathbb{R}$ be a convex set and let $\mathbf{v} \in \mathbb{R}^d$ be any vector (whether "small" or "large"). Define $\mathcal{C} + \mathbf{v} \triangleq \{\mathbf{x} : \mathbf{x} = \mathbf{z} + \mathbf{v} \text{ for some } \mathbf{z} \in \mathcal{C}\}$. Show that the set $\mathcal{C}+\mathbf{v}$ will always be convex, no matter what $\mathbf{v}$ or convex set $\mathcal{C}$ we choose.



Exercise B.4. If I take a convex set and scale it, does it remain convex? Let $\mathcal{C} \subset \mathbb{R}$ be a convex set and let me scale dimension j using a scaling factor $s_j > 0$ i.e. for a vector $\mathbf{x}$, the scaled vector is $\tilde{\mathbf{x}}$ where $\tilde{\mathbf{x}}_j = s_j \cdot \mathbf{x}_j$. We can represent this operation using a diagonal matrix $S \in \mathbb{R}^{d \times d}$ where $S_{ii} = s_i$ and $S_{ij} = 0$ if $i \neq j$ i.e. $\tilde{\mathbf{x}} = S\mathbf{x}$. In the figure to the left, the x axis has been scaled up (expanded) 33% i.e. $s_1 = 1.333$ and the y axis has been scaled down (shrunk) by 33% i.e. $s_2 = 0.667$. Thus, in this example $S = \begin{bmatrix} 1.333 & 0 \\ 0 & 0.667 \end{bmatrix}$. Define $S\mathcal{C} \triangleq \{\mathbf{x} : \mathbf{x} = S\mathbf{z} \text{ for some } \mathbf{z} \in \mathcal{C}\}$. Show that the set $S\mathcal{C}$ will always be convex, no matter what positive scaling factors or convex set $\mathcal{C}$ we choose. Does this result hold even if (some of) the scaling factors are negative? What if some of the scaling factors are zero?

Exercise B.5. Above, we saw two operations (shifting a.k.a *translation* and scaling) that keep convex sets convex. However, can these operations turn a non-convex set into a convex set i.e. can there exist a non-convex set $\mathcal{C}$ such that $\mathcal{C}+\mathbf{v}$ is convex or else $S\mathcal{C}$ is convex when all scaling factors are non-zero? What if some (or all) scaling factors are zero?

Exercise B.6. Show that affine functions are always convex i.e. for any $\mathbf{a} \in \mathbb{R}^d, b \in \mathbb{R}$, the function $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x} + b$ is a convex function. Next, show that affine functions are always concave as well. In fact, affine functions are the only functions that are both convex as well as concave.

Exercise B.7. Show that affine functions when scaled by a positive constant, remain convex i.e. for convex function $f : \mathbb{R}^d \to \mathbb{R}$ and any $c > 0$, the function $g = c \cdot f$ is also convex. Next, show that if $c < 0$ then g is concave. What happens if $c = 0$? Does g become convex or concave?

Exercise B.8. The logistic loss function is very popular in machine learning and is defined as $\ell_{\text{logistic}}(t) = \ln(1 + \exp(-t))$. Show that given data points $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$ (which are to be treated as constants) the function $f(\mathbf{w}) \triangleq \sum_{i=1}^n \ell_{\text{logistic}}(y^i \cdot \mathbf{w}^\top \mathbf{x}^i)$ is a convex function of $\mathbf{w}$.

C

Probability Theory Refresher

C.1 Empirical Median

Here is a proof that shows that if we have a set of n independent samples $x_1, \ldots, x_n$ of a random variable X, then the solution to the following optimization problem (P) does indeed give us a value for the empirical median (also sometimes called the *sample median*).

$$\hat{\text{med}}\{x_1, \ldots, x_n\} = \arg\min_{x \in \mathbb{R}} \sum_{i=1}^{n} |x_i - x| \qquad (P)$$

We present the proof below for the case when n is odd – the case when n is even, admits a similar proof. Without loss of generality, assume that we have reordered the samples so that they are now in increasing order (even though when we received them, they need not have been in any particular order) i.e. from now on we assume $x_1 \leq x_2 \leq \ldots \leq x_n$. In this case, the definition of the empirical median tells us that the median must be $m \triangleq x_{(n+1)/2}$ i.e. the "middle" element when samples are arranged in increasing order.

Suppose there are n_1 elements in the sample set that are *strictly* less than m and n_2 elements in the sample set that are *strictly* greater than m. Note that this means that we are admitting that $n - n_1 - n_2$ elements in the set are equal to m – this can very well happen if some element in the support of X gets sampled more than once. We will show below that m indeed optimizes (P) by analyzing a few simple cases regarding the values n_1 and n_2 can take.

The proof will basically show that no matter whether we move to the left or right of m (by a tiny bit), the objective value of (P) always increases. However, this means m is a local optimum. Since (P) has a convex objective and has no constraints, this would imply that m is a global optimum as well. This will conclude the proof.

Note that we must always have $n_1 < \frac{n}{2}$ as well as $n_2 < \frac{n}{2}$. If this were not the case, for instance if $n_1 \geq \frac{n}{2}$, then since n is odd and n_1 must be an integer, then we would have $n_1 \geq \frac{n+1}{2}$. However this contradicts the fact that

by definition $m = x_{(n+1)/2}$ and n_1 was defined as the number of elements *strictly* smaller than m. Similarly, we can show that $n_2 < \frac{n}{2}$.

Case 1: $n_1 = 0 = n_2$. This is the trivial case where all the n samples are m itself. In this case m clearly optimizes (P).

Case 2: $n_2 = 0$ but $n_1 > 0$. This means that m is the largest value in the sample i.e. $m = \max_{i\in[n]} x_i$. In this case, the objective value of the problem (P) at m can be written as

$$\sum_{i\le n_1}(m - x_i) + (n - n_1)(m - m) = v^*$$

Let $\Delta = m - x_{n_1}$ denote the gap between the median and the next largest element. Now if we move a tiny bit to the right from m by an amount $\delta < \Delta$ i.e. let $\tilde{m} = m + \delta$, then the objective value of the problem (P) at $\tilde{m}$ is

$$\sum_{i\le n_1}(\tilde{m} - x_i) + (n - n_1)(\tilde{m} - m) = v^* + n\delta > v^*$$

If we instead move a tiny bit to the left from m by an amount $\delta < \Delta$ i.e. let $\tilde{m} = m - \delta$, then the objective value of the problem (P) at $\tilde{m}$ is

$$\sum_{i\le n_1}(\tilde{m} - x_i) + (n - n_1)(m - \tilde{m}) = v^* + (n - 2n_1)\delta > v^*$$

Note that since we chose $\delta < \Delta$ we will still have $\tilde{m} > x_i$ for all $i \le n_1$. However, since $n_1 < \frac{n}{2}$, we have $n - 2n_1 > 0$ which justifies the last step in the above line. This shows that whether we move to the left or right of m by a small amount, the objective value always increases i.e. m is a local optimum.

Case 3: $n_1 = 0$ but $n_2 > 0$. Similar proof as Case 2.

Case 4: $n_1 > 0$ and $n_2 > 0$. In this case, the objective value of the problem (P) at m can be written as

$$\sum_{i\le n_1}(m - x_i) + (n - n_1 - n_2)(m - m) + \sum_{i\ge n-n_1+1}(x_i - m) =: v^*$$

Let $\Delta := \min\{m - x_{n_1}, x_{n-n_2+1} - m\}$ be the distance of the median m from its closest non-median neighbor. Now suppose I move a tiny bit to the right of the median – specifically, I choose some $\delta < \Delta$ and set $\tilde{m} = m+\delta$. Note that elements that were strictly greater (respectively strictly smaller) than m are still strictly greater (respectively strictly smaller) than $\tilde{m}$ since we chose $\delta < \Delta$. The objective value of the problem (P) at m can be written as

$$\begin{aligned}
&\sum_{i\le n_1}(\tilde{m} - x_i) + (n - n_1 - n_2)(\tilde{m} - m) + \sum_{i\ge n-n_2+1}(x_i - \tilde{m}) \\
&= v^* + n_1\cdot\delta + (n - n_1 - n_2)\cdot\delta - n_2\cdot\delta \\
&= v^* + (n - 2n_2)\cdot\delta \\
&> v^*,
\end{aligned}$$

where the last statement holds since we have already established that $n_2 < \frac{n}{2}$ i.e. $n - 2n_2 > 0$. This shows that the objective value of (P) goes up if we move right a bit. A similar argument shows that we cannot move left a bit either without increasing the objective value. This establishes that m is a local optimum in this case as well.

The above case analyses establish that the median is indeed the optimum solution to the problem (P).

Example C.1. *Dr. Strange is trying to analyze all possible outcomes of the Infinity War using the Time Stone but there are way too many of them so he analyzes only 75% of the n possible outcomes. The Avengers then sit down to discuss the outcomes. Yet again, since there are so many of them, they discuss only 25% of the n outcomes (much fewer outcomes are discussed by the Avengers than were analyzed by Dr Strange since by now Thanos has snatched away the Time Stone). They may discuss some outcomes that Dr. Strange has already analyzed as well as some outcomes that he has not analyzed. It is known that of the outcomes that were analyzed by Dr. Strange, only 30% got discussed. Given this, can we find out what fraction of the discussed outcomes were previously analyzed by Dr. Strange?*

This problem may not seem like a probability problem to begin with. However, if we recall the interpretation of probability in terms of proportions, then it is easy to see this as a probability problem and also apply powerful tools from probability theory. When we say that Dr. Strange analyzes only 75% of the outcomes, this is just another way of saying that if we pick one of the n outcomes uniformly at random (i.e. each outcome gets picked with equal probability $\frac{1}{n}$), then there is a $\frac{3}{4}$ probability that it would be an outcome that was analyzed by Dr. Strange.

With this realization in mind, we set up our probability space properly. Our sample space is $[n]$ consisting of the n possible outcomes. Each outcome is equally likely to be picked i.e. each outcome gets picked with probability $\frac{1}{n}$. We now define two indicator variables A and D. $A = 1$ if the chosen outcome was analyzed by Dr. Strange and $A = 0$ otherwise. $D = 1$ if the chosen outcome was discussed by the Avengers and $D = 0$ otherwise.

The problem statement tells us that $\mathbb{P}[A = 1] = \frac{3}{4}$ (since 75% outcomes were analyzed), $\mathbb{P}[D = 1] = \frac{1}{4}$ (since 25% outcomes were analyzed). The problem statement also tells us that of the analyzed outcomes, only 30% were discussed which means that the number of outcomes that were both discussed and analyzed, divided by the number of outcomes that were analyzed is $\frac{3}{10}$ i.e. $\frac{\mathbb{P}[D=1,A=1]}{\mathbb{P}[A=1]} = \frac{3}{10}$. This is just another way of saying that $\mathbb{P}[D = 1 \mid A = 1] = \frac{3}{10}$.

The problem asks us to find the fraction of discussed outcomes that were analyzed i.e. we want the number of outcomes that were both analyzed and discussed, divided by the number of discussed outcomes. This is nothing but $\mathbb{P}[A = 1 \mid D = 1]$. Applying the Bayes theorem now tells us

$$\mathbb{P}[A = 1 \mid D = 1] = \frac{\mathbb{P}[D = 1 \mid A = 1] \cdot \mathbb{P}[A = 1]}{\mathbb{P}[D = 1]} = \frac{\frac{3}{10} \cdot \frac{3}{4}}{\frac{1}{4}} = \frac{9}{10}$$

which means that 90% of the discussed outcomes were previously analyzed by Dr. Strange. This is not surprising given that Dr. Strange analyzed so many more outcomes than the Avengers were able to discuss.

Example C.2 (Problem 6.4 from Deisenroth et al. (2019)). *There are two bags. The first bag contains four mangoes and two apples; the second bag contains four mangoes and four apples. We also have a biased coin, which shows "heads" with probability 0.6 and "tails" with probability 0.4. If the coin shows "heads", we pick a fruit at random from bag 1; otherwise we pick a fruit at random from bag 2. Your friend flips the coin (you cannot see the result), picks a fruit* uniformly *at random from the corresponding bag, and presents you a mango. What is the probability that the mango was picked from bag 2?*[1]

To solve the above problem, let us define some random variables. Let $B \in \{1, 2\}$ denote the bag from which the fruit is picked and let $F \in \{M, A\}$ denote which fruit is selected[2]. The problem statement tells us the following: $\mathbb{P}[B = 1] = 0.6$ and $\mathbb{P}[B = 2] = 0.4$ since the outcome of the coin flip completely decides which bag we choose. Now, suppose we knew that the fruit was being sampled from bag 1, then interpreting probabilities as proportions (since fruits are chosen uniformly at random from a bag), we have $\mathbb{P}[F = M \mid B = 1] = \frac{4}{4+2} = \frac{2}{3}$ and $\mathbb{P}[F = A \mid B = 1] = 1 - \mathbb{P}[F = M \mid B = 1] = \frac{1}{3}$. Similarly, we have $\mathbb{P}[F = M \mid B = 2] = \frac{4}{4+4} = \frac{1}{2} = \mathbb{P}[F = A \mid B = 2]$.

We are told that the fruit that was picked was indeed a mango and are interested in knowing the chances that it was picked from bag 2. Thus, we are interested in $\mathbb{P}[B = 2 \mid F = M]$. Applying the Bayes theorem gives us

$$\mathbb{P}[B = 2 \mid F = M] = \frac{\mathbb{P}[F = M \mid B = 2] \cdot \mathbb{P}[B = 2]}{\mathbb{P}[F = M]}$$

We directly have values for the two terms in the numerator. The denominator, however, will have to be calculated by deriving the marginal probability $\mathbb{P}[F = M]$ from the joint probability distribution $\mathbb{P}_{F,B}$ using the sum rule (law of total probability) and the product rule

$$\begin{aligned}\mathbb{P}[F = M] &= \mathbb{P}[F = M, B = 1] + \mathbb{P}[F = M, B = 2] \\ &= \mathbb{P}[F = M \mid B = 1] \cdot \mathbb{P}[B = 1] + \mathbb{P}[F = M \mid B = 2] \cdot \mathbb{P}[B = 2] \\ &= \frac{2}{3} \cdot \frac{3}{5} + \frac{1}{2} \cdot \frac{2}{5} = \frac{3}{5},\end{aligned}$$

where in the first step we used the sum rule and in the second step we used the product rule. Putting things together gives us

$$\mathbb{P}[B = 2 \mid F = M] = \frac{\frac{1}{2} \cdot \frac{2}{5}}{\frac{3}{5}} = \frac{1}{3}.$$

Thus, there is a $\frac{1}{3}$ probability that the mango we got was picked from bag 2. The complement rule for conditional probability tell us that $\mathbb{P}[B = 1 \mid F = M] =$

[1] In this statement, the word *uniformly* was added (it was not present in Deisenroth et al. (2019)) to emphasize the nature of the random choice of fruit from a bag.

[2] Note that we used a non-numeric support $\{M, A\}$ for the random variable F merely for sake of easy identification. We can easily make this support numeric say, by mapping $M = 1$ and $A = 0$.

$1 - \mathbb{P}[B = 2 \mid F = M] = \frac{2}{3}$, i.e. there is a much larger, $\frac{2}{3}$ probability that the mango we got was picked from bag 1. This is to be expected since not only is bag 1 more likely to be picked up than bag 2, bag 1 also has a much larger proportion of mangoes than bag 2 which means that if we got a mango, it is more likely that it came from bag 1.

Example C.3. *Let A, B denote two events such that $\mathbb{P}[A \mid B] = 0.36 = \mathbb{P}[A]$. Can we find $\mathbb{P}[A \mid \neg B]$ i.e. the probability that event A will take place given that event B has* not *taken place?*

Let us abuse notation to let A, B also denote the indicator variables for the events i.e. $A = 1$ if A takes place and $A = 0$ otherwise and similarly for B. Note that the problem statement has essentially told us that A and B are independent events. Since $\mathbb{P}[A \mid B] = \mathbb{P}[A]$, by abusing notation we get

$$\frac{\mathbb{P}[A = 1, B = 1]}{\mathbb{P}[B = 1]} = \mathbb{P}[A = 1],$$

i.e. $\mathbb{P}[A = 1, B = 1] = \mathbb{P}[A = 1] \cdot \mathbb{P}[B = 1]$. Now, we are interested in the probability $\mathbb{P}[A = 1 \mid B = 0] = \frac{\mathbb{P}[A=1, B=0]}{\mathbb{P}[B=0]}$. Since we have not been given $\mathbb{P}[B = 0]$ directly, we try to massage the numerator to see if we can get hold of something.

$$\begin{aligned}
\mathbb{P}[A = 1, B = 0] &= \mathbb{P}[A = 1] - \mathbb{P}[A = 1, B = 1] \\
&= \mathbb{P}[A = 1] - \mathbb{P}[A = 1] \cdot \mathbb{P}[B = 1] \\
&= \mathbb{P}[A = 1](1 - \mathbb{P}[B = 1]) \\
&= \mathbb{P}[A = 1]\,\mathbb{P}[B = 0],
\end{aligned}$$

where in the first step we used the sum rule (law of total probability), in the second step, we exploited the independence of the two events and in the last step, we used the complement rule. This gives us

$$\mathbb{P}[A = 1 \mid B = 0] = \frac{\mathbb{P}[A = 1, B = 0]}{\mathbb{P}[B = 0]} = \frac{\mathbb{P}[A = 1]\,\mathbb{P}[B = 0]}{\mathbb{P}[B = 0]} = \mathbb{P}[A = 1] = 0.36,$$

since the problem statement already tells us that $\mathbb{P}[A = 1] = 0.36$.

Example C.4. *Timmy is is trying to kill some free time by typing random letters on his keyboard. He types 7 random capital alphabet letters (A - Z) on his keyboard. Each letter is chosen uniformly randomly from the 26 letters and each choice is completely independent of the other choices. Can we find the probability that Timmy will end up typing the word COVFEFE?*

Let $L_i, i \in [7]$ denote the random variable that tells us which letter was chosen at the i^{th} location in the word. As before, we will let the support of the random variables L_i be the words of the English alphabet rather than numbers, for sake of easy identification. We can readily map the letters of the alphabet to [26] to have numerical supports instead. We are interested in

$$\mathbb{P}[L_1 = C, L_2 = O, L_3 = V, L_4 = F, L_5 = E, L_6 = F, L_7 = E]$$

However, since the choices were made independently, applying the product rule for expectations tells us that the above is equal to

$$\mathbb{P}[L_1 = C] \cdot \mathbb{P}[L_2 = O] \cdot \mathbb{P}[L_3 = V] \cdot \mathbb{P}[L_4 = F] \cdot \mathbb{P}[L_5 = E] \cdot \mathbb{P}[L_6 = F] \cdot \mathbb{P}[L_7 = E]$$

Since each letter is chosen uniformly at random from the 26 letters of the alphabet, the above probability is simply $\left(\frac{1}{26}\right)^7$.

C.2 Exercises

Exercise C.1. Let Z denote a random variable which takes value 1 with probability p and value 2 with probability $1-p$. For what value of $p \in [0,1]$ does this random variable have the highest variance?

Exercise C.2. Summers are here and people are buying soft drinks, say from three brands – ThumsUp, Pepsi and Coke. Suppose the total sales of all the brands put together is exactly 14 million units every day. Now, the market share of the individual brands changes from day to day. However, it is also known that ThumsUp sells on an average of 8 million units per day and Coke sells 4 million units per day on average. How many units does Pepsi sell per day on an average?

Exercise C.3. Suppose A and B are two events such that $\mathbb{P}[A] = 0.125$ whereas B is an almost sure event i.e. $\mathbb{P}[B] = 1$. What is $\mathbb{P}[A \mid B]$? What is $\mathbb{P}[B \mid A]$?

Exercise C.4. The probability of life on a planet given that there is water on it is 80%. If there is no water on a planet then life cannot exist on that planet. The probability of finding water on a planet is 50%. Planet A has life on it whereas Planet B does not have life on it. What is the probability that Planet A has water? What is the probability that Planet B has water on it?

Exercise C.5. My sister and I both try to sell comic books to our classmates at school. Since our target audience is limited, it is known that on any given day, the two of us together sell at most 10 comic books. It is known that I sell 5 comic books on average everyday. It is also known that my sister sells 3 comic books on average everyday. How many comic books do the two of us sell together on average everyday?

Exercise C.6. Martin has submitted his thesis proposal. The probability that professor X will approve the proposal is 70%. The probability that professor Y will approve the proposal is 50%. The probability that professor Z will approve the proposal is 40%. The approvals of the three professors are entirely independent of one another. Suppose Martin has to get approval from at least two of the three professors to pass his proposal, what is the probability that his proposal will pass?

D

Linear Algebra Refresher

Example D.1. Let $A \in \mathbb{R}^{d \times d}$ be a square symmetric matrix with real entries. Then the following optimization problem will always recover a leading eigenvector of the matrix (recall that symmetric matrices always posses an eigendecomposition)

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^d} \quad & \mathbf{x}^\top A \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{x}\|_2^2 = 1 \end{aligned}$$

We have seen how to solve optimization problems by deriving Lagrangian duals when the problems have inequality constraints. To handle equality constraints, one option is to convert the equality constraint into two inequality constraints i.e. $\|\mathbf{x}\|_2^2 \leq 1$ and $\|\mathbf{x}\|_2^2 \geq 1$ and proceed.

However, there exists a simpler alternative – corresponding to each equality constraint, introduce a single Lagrangian variable which is constrained take non-zero value (i.e. it is allowed to take positive or negative values, just not zero). Doing so (after converting the problem into a minimization problem by affixing a negative sign) gives us the Lagrangian

$$\mathcal{L}(\mathbf{x}, \alpha) = -\mathbf{x}^\top A \mathbf{x} + \alpha(\|\mathbf{x}\|_2^2 - 1),$$

and the corresponding dual problem

$$\max_{\alpha \neq 0} \ \left\{ \min_{\mathbf{x}} \ \left\{ -\mathbf{x}^\top A \mathbf{x} + \alpha(\|\mathbf{x}\|_2^2 - 1) \right\} \right\}$$

Since the inner problem is unconstrained, applying first order optimality gives

$$-A\mathbf{x} + 2\alpha \cdot \mathbf{x} = 0,$$

or in other words

$$A\mathbf{x} = 2\alpha \cdot \mathbf{x}.$$

Thus, the solution $\mathbf{x}$ must be an eigenvector of A. Since the original problem is a maximization problem, it is readily apparent that the solution is a leading

eigenvector. Certain matrices may have infinitely many eigenvectors w.r.t their largest eigenvalue which is why we have always said in this example that the solution is *a* leading eigenvector and not *the* leading eigenvector.

Example D.2. Suppose we know that A is a 3×3 symmetric matrix with eigenvalues $1, 0, -1$ (symmetric matrices always have real eigenvalues but they can be negative). Can we find the value of $|2 \cdot I + A^{100}|$ where $I = I_3$ is the 3×3 identity matrix and $|\cdot|$ denotes the determinant?

This problem may seem daunting at first but is actually solved very easily by invoking the eigendecomposition of A and writing $A = QSQ^\top$ where $Q \in \mathbb{R}^{3\times 3}$ is an orthonormal matrix, and $S = \text{diag}\,(1, 0, -1)$. Now, orthonormality gives us $Q^\top Q = I$ and thus, we have $A^2 = QSQ^\top QSQ^\top = QS^2Q^\top$ (just as we saw while deriving the power method). By induction, we deduce that $A^{100} = QS^{100}Q^\top$.

However, orthornormality also gives us $QQ^\top = I$ which can be thought of as saying $A^0 = QS^0Q^\top = I$. Thus, we can write $2 \cdot I + A^{100} = QQ^\top + QS^{100}Q^\top = Q(2 \cdot I + S^{100})Q^\top$ which tells us that the eigenvalues of the matrix $2 \cdot I + A^{100}$ are $(2 + 1^{100}), (2 + 0^{100}), (2 + (-1)^{100})$ i.e. $3, 2, 3$. The determinant of a square symmetric matrix is always the product of its eigenvalues which tells us that $|2 \cdot I + A^{100}| = 18$.

Example D.3. Suppose we have a matrix $X \in \mathbb{R}^{n\times d}$ where $n > d$. Suppose we select some d rows in this matrix, say $S = \{i_1, \ldots, i_d\} \subset [n]$ and construct a new matrix $X^S \in \mathbb{R}^{d\times d}$ using these rows. Suppose X^S is full rank i.e. $\text{rank}(X^S) = d$. Then this implies that X itself was full rank[1].

To see this, assume the contrapositive i.e. assume that X is not full rank. Since the rank of a matrix is the same as its column rank, this means that some column of X, say the j^{th} column X_j is expressible as a linear combination of the other columns. i.e. $X_j = \sum_{k\neq j} c_k \cdot X_k$. However, note that this also means that $X^S_j = \sum_{k\neq j} c_k \cdot X^S_k$ i.e. the same column in the matrix X^S would also be expressible as the same linear combination of the other columns in X^S.

The above is a contradiction since we know that X^S is full rank. This means that the presence of any $d \times d$ *sub-matrix* within X that is full rank, makes the entire matrix full rank.

Example D.4. We have seen that linear maps can be used to change (increase/decrease) the dimensionality of data as well as scale and rotate data. Given all this power, it may be tempting to think that given a classification problem that is not linearly separable, it should be possible to linearly project the data features onto a very high dimensional space where the points do become linearly separable.

All such hope is futile. Linear maps cannot add to the power of linear classifiers. More specifically, given a feature matrix $X \in \mathbb{R}^{n\times d}$ and a linear map $f : \mathbb{R}^d \mapsto \mathbb{R}^D$ specified by a matrix $A \in \mathbb{R}^{d\times D}$ (where D can be any strictly positive integer, possibly greater or smaller than d), let the matrix of the mapped features be called $\tilde{X} \triangleq XA \in \mathbb{R}^{n\times D}$.

[1] It should be evident that the rows we select must be distinct otherwise the matrix X^S can clearly not be full rank if two rows in X^S are the same

Then, for any D-dimensional linear model $\tilde{\mathbf{w}} \in \mathbb{R}^D$, there always exists a d-dimensional model $\mathbf{w} \in \mathbb{R}^d$ such that $\tilde{X}\tilde{\mathbf{w}} = X\mathbf{w}$. To see this, simply use $\mathbf{w} = A\tilde{\mathbf{w}}$. This means that there is nothing to be gained by changing the dimensionality of the data (except perhaps increasing algorithm speed or reducing storage costs by reducing the dimensionality). Whatever we can do with the D-dimensional mapped features using D-dimensional linear models, we could already have done with the original d-dimensional features using d-dimensional linear models.

Example D.5. Suppose we draw a 3-dimensional random vector $\mathbf{x}$ from the standard Gaussian distribution i.e. $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I_3)$. We are also given a 3×3 symmetric matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 5 \\ 3 & 5 & 3 \end{bmatrix}$. Can we find the value of $\mathbb{E}\left[\mathbf{x}^\top A\mathbf{x}\right]$?

Although it does not seem so, to solve this problem, we would greatly benefit from using properties of the trace operator. Namely we would need the following three properties of the trace operator

1. For any scalar $v \in \mathbb{R}$, we have $\text{trace}(v) = v$
2. The trace has a cyclic property – for any two (possibly rectangular or non symmetric) matrices A, B such that the products AB and BA both make sense dimension-wise, we always have $\text{trace}(AB) = \text{trace}(BA)$.
3. The trace obeys linearity of expectation i.e. if A is a constant matrix and X is a random matrix (where X and A may have different dimensionalities) then, if AX makes sense dimension wise, then we have $\mathbb{E}\left[\text{trace}(AX)\right] = \text{trace}(A\mathbb{E}\left[X\right])$. Similarly, if XA makes sense dimension wise, then we have $\mathbb{E}\left[\text{trace}(XA)\right] = \text{trace}(\mathbb{E}\left[X\right]A)$.

Using the above facts we have

$$\mathbb{E}\left[\mathbf{x}^\top A\mathbf{x}\right] = \mathbb{E}\left[\text{trace}(\mathbf{x}^\top A\mathbf{x})\right] = \mathbb{E}\left[\text{trace}(\mathbf{x}\mathbf{x}^\top A)\right] = \text{trace}(\mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]A)$$

However, since $\mathbb{E}\left[\mathbf{x}\right] = \mathbf{0}$, we conclude that $\mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]$ is the covariance matrix of the random variable $\mathbf{x}$ and thus $\mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right] = I$. Thus, we get

$$\mathbb{E}\left[\mathbf{x}^\top A\mathbf{x}\right] = \text{trace}(\mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]A) = \text{trace}(IA) = \text{trace}(A) = 4$$

D.1 Exercises

Exercise D.1. Suppose we have two d-dimensional (column) vectors $\mathbf{u}$ an $\mathbf{v}$ that are orthogonal to each other. Let $A = \mathbf{u}\mathbf{v}^\top$ be the $d \times d$ matrix created out of these two vectors. Find $\text{trace}(A)$.

Exercise D.2. Prove that a matrix $A \in \mathbb{R}^{m \times n}$ is unit rank if and only if (often stylized as *iff*) $A = uw^\top$ for some vectors $u \in \mathbb{R}^m, w \in \mathbb{R}^n$. To prove a matrix to be unit rank, show either its column rank or its row rank, to be unity. Do not invoke the SVD theorem to prove this result (you should be able to show this without invoking a powerful result like the SVD theorem).

Exercise D.3. Let $A_{m\times n}$ denote a matrix with m, n rows and columns resp. Using the associativity property of matrix multiplication, you can compute the triple matrix product ABC as either $(AB)C$ or $A(BC)$. Suppose we have the sizes of these three matrices as $A_{40\times 3}$, $B_{3\times 10}$, and $C_{10\times 2}$. Which multiplication technique would you prefer to use to compute the product and why ? (a) $(A_{40\times 3}B_{3\times 10})C_{10\times 2}$, or (b) $A_{40\times 3}(B_{3\times 10}C_{10\times 2})$.

Exercise D.4. Example D.3 shows that the presence of any full rank submatrix makes a matrix full rank. Is the converse true as well? More specifically, suppose we know that a matrix $X \in \mathbb{R}^{n\times d}$, with $n > d$, is full rank i.e. $\text{rank}(X) = d$. Does this mean that there must exist some set of d rows $S \subset [n], |S| = d$, such that the matrix X^S is full rank? If your answer is *yes*, prove it. If your answer is *no*, give a counter example of a $n \times d$ matrix with $n > d$ that is full rank but no $d \times d$ submatrix is full rank.

Exercise D.5. Taking the previous exercise (i.e. Exercise D.4) forward, suppose we know that a matrix $X \in \mathbb{R}^{n\times d}$, with $n > d$, is full rank i.e. $\text{rank}(X) = d$. Does this mean that for *every* set of d rows $S \subset [n], |S| = d$, the matrix X^S must be full rank? If your answer is *yes*, prove it. If your answer is *no*, give a counter example of a $n \times d$ matrix with $n > d$ that is full rank but at least one $d \times d$ submatrix not full rank.

Exercise D.6. Consider n vectors $\mathbf{x}^1, \ldots, \mathbf{x}^n \in \mathbb{R}^d$. Let us define an $n \times n$ matrix $G = [G_{ij}]$ where $G_{ij} = \exp(-\gamma \cdot \left\|\mathbf{x}^i - \mathbf{x}^j\right\|_2^2)$. Comment on what will happen to the trace and rank of G when $\gamma \to \infty$, as well as when $\gamma \to 0$.

Exercise D.7. We have three $n \times n$ square symmetric matrices A, B, and C. All the eigenvalues of A are zero whereas all the eigenvalues of B are one. C has one eigenvalue equal to one and the rest equal to zero. We also have with us an n-dimensional vector $\mathbf{x} \in \mathbb{R}^n$. What is the maximum and minimum possible value of $\left\|A\mathbf{x}\right\|_2, \left\|B\mathbf{x}\right\|_2, \left\|C\mathbf{x}\right\|_2$?

Exercise D.8. For a (possibly rectangular) matrix A, its *spectral norm* is defined as

$$\left\|A\right\|_2 \triangleq \max_{\left\|\mathbf{x}\right\|_2 \leq 1} \left\|A\mathbf{x}\right\|_2,$$

Suppose $\mathbf{u} \in \mathbb{R}^k, \mathbf{v} \in \mathbb{R}^d$ are two vectors with Euclidean norms 5 and 6 respectively. Find the spectral norm of the matrix $B \triangleq \mathbf{u}\mathbf{v}^\top \in \mathbb{R}^{k\times d}$. Suppose now that $k = d$. What can you say about the spectral norm of B^t for some integer $t > 0$?

Exercise D.9. Suppose X is a random variable that can take m possible values $u_1, \ldots, u_m$ and Y is another random variable that can take n possible values $v_1, \ldots, v_n$. All values that X, Y take are necessarily integers and $m \geq n$ but we may have $m \neq n$. Suppose we write down a matrix $A = [A_{ij}] \in \mathbb{R}^{m\times n}$ where $A_{ij} = \mathbb{P}\left[X = u_i \wedge Y = v_j\right]$. What can you say about $\text{rank}(A)$? Can A be full rank?

Exercise D.10. Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ be two orthonormal vectors, i.e. $\mathbf{u}^\top\mathbf{v} = 0$ and $\left\|\mathbf{u}\right\|_2 = 1 = \left\|\mathbf{v}\right\|_2$. Let $A \triangleq \mathbf{u}\mathbf{u}^\top + 0.5\mathbf{v}\mathbf{v}^\top$. Is A always a diagonal matrix? Is

A always symmetric? What is the rank of A? Find an expression for A^t for any integer $t > 0$.

Exercise D.11. Consider the two vectors

$$\mathbf{x} = \begin{bmatrix} -2 \\ 3 \\ 4 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}$$

Suppose A is a 3×3 matrix that is symmetric as well as rank one. Also, we know that

$$A\mathbf{x} = \begin{bmatrix} 4 \\ -6 \\ -8 \end{bmatrix}$$

Find out the vector $A\mathbf{y}$.

Exercise D.12. Jack has prepared a square (not necessarily symmetric) matrix $A \in \mathbb{R}^{n \times n}$ that is full rank i.e. $\text{rank}(A) = n$. However, Jill is upto mischief. She picks two columns of this matrix uniformly at random without replacement, say $i \neq j$ and replaces the i^{th} column A_i with the j^{th} column A_j. When Jack finds this out, he tries to undo this by picking yet another two columns of this matrix uniformly at random without replacement, say $p \neq q$ and replacing the p^{th} column A_p with the q^{th} column A_q. What are the possible ranks of the matrix after the above operations and with what probabilities does the final matrix have those ranks?

Exercise D.13. Consider a real matrix $A \in \mathbb{R}^{d \times d}$ with the curious property that for all vectors $\mathbf{x} \in \mathbb{R}^d$, we have $\mathbf{x}^\top A\mathbf{x} = 0$. Prove that we must have $A = -A^\top$. Also, convince yourself that there do indeed exist non-zero matrices with this property i.e. it is not necessary that $A = \mathbf{0}\mathbf{0}^\top$ for $\mathbf{x}^\top A\mathbf{x} = 0$ for all $\mathbf{x} \in \mathbb{R}^d$. In fact it is true that all and only *anti-symmetric* matrices (i.e. for which $A = -A^\top$) have this curious property – can you show this iff statement too?

Exercise D.14. Consider the matrix

$$\begin{bmatrix} 2 & 0 & -2 \\ 0 & 3 & 0 \\ -2 & 0 & 4 \end{bmatrix}$$

Suppose we are told that two of the eigenvalues of this matrix are $3 \pm \sqrt{5}$, find the third eigenvalue without explicitly performing SVD/PCA on this matrix.

Exercise D.15. Calculate the rank of the following matrix:

$$\begin{bmatrix} 4 & -6 & 0 \\ -6 & 0 & 1 \\ 0 & 9 & -1 \\ 0 & 1 & 4 \end{bmatrix}$$

Exercise D.16. Agent Romanov needs to unlock the Tesseract to defeat the Chitauri. The Tesseract can only be unlocked using a matrix A given to her by Nick Fury. Unfortunately, Loki erased some of the entries of the matrix and she is in a fix! However, she remembers that Nick Fury told her while giving her the matrix that the matrix is special – it has unit rank and trace equal to negative 10 i.e. -10. Can you help Natasha find the missing entries in the matrix? Fill in your entries in the space provided.

$$\begin{bmatrix} \square & 4 & -2 \\ 7.5 & \square & 2.5 \\ \square & \square & \square \end{bmatrix}$$

References

Deisenroth, M. P., A. A. Faisal, and C. S. On (2019), 'Mathematics for Machine Learning'. https://mml-book.com/.