Machine Learning and Neural Networks III (MATH3431)

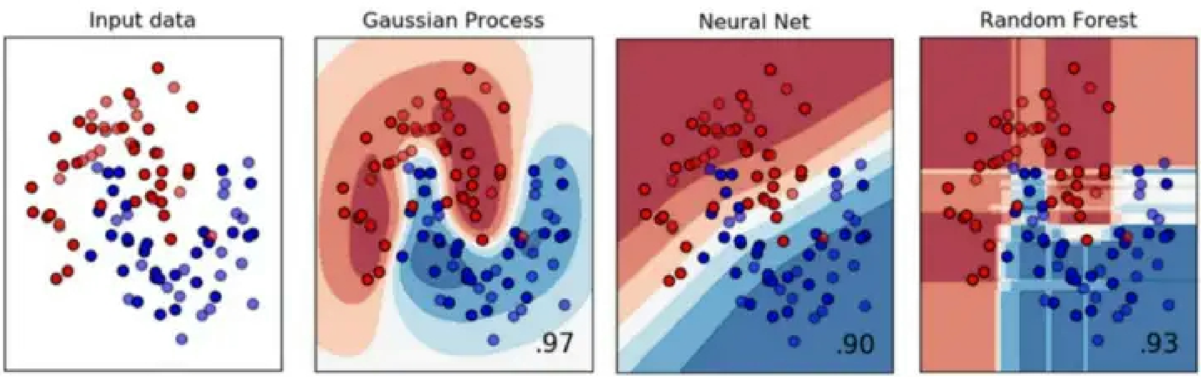
Epiphany term

Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

Department of Mathematical Sciences (Office MCS3088)
Durham University
Stockton Road Durham DH1 3LE UK

2023/01/11 at 00:26:35



Reading list

These lecture Handouts have been derived based on the above reading list.

Main texts:

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.
  - It is a classical textbook in machine learning (ML) methods. It discusses all the concepts introduced in the course (not necessarily in the same depth). It is one of the main textbooks in the module. The level on difficulty is easy.
  - Students who wish to have a textbook covering traditional concepts in machine learning are suggested to get a copy of this textbook. It is available online from the Microsoft's website `https://www.microsoft.com/en-us/research/publication/pattern-recogniti`
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
  - It has several elements of theory about machine learning algorithms. It is one of the main textbooks in the module. The level on difficulty is advanced as it requires moderate knowledge of maths.
- Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.
  - It is a classical textbook about 'traditional' artificial neural networks (ANN). It is very comprehensive (compared to others) and it goes deep enough for the module although it may be a bit outdated. It is one of the main textbooks in the module for ANN. The level on difficulty is moderate.

Supplementary textbooks:

- Ripley, B. D. (2007). Pattern recognition and neural networks. Cambridge university press.
  - A classical textbook in artificial neural networks (ANN) that also covers other machine learning concepts. It contains interesting theory about ANN.
  - It is suggested to be used as a supplementary reading for neural networks as it contains a few interesting theoretical results. The level on difficulty is moderate.
- Williams, C. K., & Rasmussen, C. E. (2006). Gaussian processes for machine learning (Vol. 2, No. 3, p. 4). Cambridge, MA: MIT press.
  - A classic book in Gaussian process regression (GPR) that covers the material we will discuss in the course about GPR. It can be used as a companion textbook with that of (Bishop, C. M., 2006). The level on difficulty is easy.

- Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.
  - A popular textbook in machine learning methods. It discusses all the concepts introduced in the module. It focuses more on the probabilistic/Bayesian framework but not with great detail. It can be used as a comparison textbook for brief reading about ML methods just to see another perspective than that in (Bishop, C. M., 2006). The level on difficulty is easy.
- Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT press.
  - A textbook in machine learning methods. It covers a smaller number of ML concepts than (Murphy, K. P., 2012) but it contains more fancy/popular topics such as deep learning ideas. It is suggested to be used in the same manner as (Murphy, K. P., 2012). The level on difficulty is easy.
- Barber, D. (2012). Bayesian reasoning and machine learning. Cambridge University Press.
  - A textbook in machine learning methods from a Bayesian point of view. It discusses all the concepts introduced apart from ANN and stochastic gradient algorithms. It aims to be more 'statistical' than those of Murphy and Bishop. The level on difficulty is easy.
- Devroye, L., Györfi, L., & Lugosi, G. (2013). A probabilistic theory of pattern recognition (Vol. 31). Springer Science & Business Media.
  - Theoretical aspects about machine learning algorithms. The level on difficulty is advanced as it requires moderate knowledge of probability.

Machine Learning and Neural Networks (MATH3431) Epiphany term, 2023

Handout 0: Learning problem: Definitions, notation, and formulation –A recap

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To get some definitions and set-up about the learning procedure; essentially to formalize what introduced in term 1.

Reading list & references:

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

1. Introductions and loose definitions

Pattern recognition is the automated discovery of patterns and regularities in data $z \in \mathcal{Z}$. **Machine learning (ML)** are statistical procedures for building and understanding probabilistic methods that 'learn'. **ML algorithms** $\mathfrak{A}$ build a (probabilistic/deterministic) model able to make predictions or decisions with minimum human interference and can be used for pattern recognition. **Learning** (or training, estimation) is called the procedure where the ML model is tuned. **Training data** (or observations, sample data set, exemplars) is a set of observables $\{z_i \in \mathcal{Z}\}$ used to tune the parameters of the ML model. **Test set** is a set of available examples/observables $\{z_i'\}$ (different than the training data) used to verify the performance of the ML model for a given a measure of success. **Measure of success** (or performance) is a quantity that indicates how bad the corresponding ML model or Algorithm performs (eg quantifies the failure/error), and can also be used for comparisons among different ML models; eg, **Risk function** or **Empirical Risk Function**. Two main problems in ML are the supervised learning (we focus here) and the unsupervised learning.

Supervised learning problems involve applications where the training data $z \in \mathcal{Z}$ comprises examples of the input vectors $x \in \mathcal{X}$ along with their corresponding target vectors $y \in \mathcal{Y}$; i.e. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. **Classification problems** are those which aim to assign each input vector x to one of a finite number of discrete categories of y. **Regression problems** are those where the output y consists of one or more continuous variables. All in all, the learner wishes to recover an unknown pattern (i.e. functional relationship) between components $x \in \mathcal{X}$ that serves as inputs and components $y \in \mathcal{Y}$ that act as outputs; i.e. $x \longmapsto y$. Hence, $\mathcal{X}$ is the input domain, and $\mathcal{Y}$ is the output domain. The goal of learning is to discover a function which predicts $y \in \mathcal{Y}$ from $x \in \mathcal{X}$.

Unsupervised learning problems involve applications where the training data $z \in \mathcal{Z}$ consists of a set of input vectors $x \in \mathcal{X}$ without any corresponding target values ; i.e. $\mathcal{Z} = \mathcal{X}$. In clustering the goal is to discover groups of similar examples within the data of it is to discover groups of similar examples within the data.

2. (loose) Notation in learning

Definition 1. The learner's output is a function, $h : \mathcal{X} \to \mathcal{Y}$ which predicts $y \in \mathcal{Y}$ from $x \in \mathcal{X}$. It is also called hypothesis, prediction rule, predictor, or classifier.

Notation 2. We often denote the set of hypothesis as $\mathcal{H}$; i.e. $h \in \mathcal{H}$.

Definition 3. Training data set $\mathcal{S}$ of size m is any finite sequence of pairs $((x_i, y_i)\,; i = 1, ..., m)$ in $\mathcal{X} \times \mathcal{Y}$. This is the information that the learner has assess.

Definition 4. Data generation model $g(\cdot)$ is the probability distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, unknown to the learner that has generated the data.

Definition 5. We denote as $\mathfrak{A}(\mathcal{S})$ the hypothesis (outcome) that a learning algorithm $\mathfrak{A}$ returns given training sample S.

Definition 6. (Loss function) Given any set of hypothesis $\mathcal{H}$ and some domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, a loss function $\ell(\cdot)$ is any function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$. The purpose of loss function $\ell(h, z)$ is to quantify the "error" for a given hypothesis h and example z –the greater the error the greater its value of the loss.

Example 7. In binary classification problems where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{Y} = \{0, 1\}$ is discrete, a loss function can be

$$\ell_{0-1}(h, (x, y)) = 1(h(x) = y)\,,$$

Example 8. In regression problems $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{Y}$ is uncountable, a loss function can be

$$\ell_{\text{sq}}(h, (x, y)) = (h(x) - y)^2$$

Definition 9. (Risk function) The risk function $R_g(h)$ of h is the expected loss of the hypothesis $h \in \mathcal{H}$, w.r.t. probability distribution g over domain Z; i.e.

$$R_g(h) = \mathrm{E}_{z \sim g}(\ell(h, z)) \tag{2.1}$$

Remark 10. In learning, an ideal way to obtain an optimal predictor h^* is to compute the risk minimizer

$$h^* = \arg\min_{\forall}(R_g(h))$$

Example 11. (Cont. Ex. 8)The risk function is $R_g(h) = \mathrm{E}_{z \sim g}(h(x) - y)^2$, and it measures the quality of the hypothesis function $h : \mathcal{X} \to \mathcal{Y}$ as the expected square difference between the predicted values h and the true target values y at every x.

Remark 12. Computing the risk minimizer may be practically challenging due to the integration w.r.t. the unknown data generation model g involved in the expectation (2.1). Suboptimally, one may resort to the Empirical risk function.

Definition 13. (Empirical risk function) The empirical risk function $\hat{R}_S(h)$ of h is the expectation of loss of h over a given sample $S = (z_1, ..., z_m) \in \mathcal{Z}^m$; i.e.

$$\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i)\,.$$

Example 14. (Cont. Example 11) Given given sample $S = \{(x_i, y_i); i = 1, ..., m\}$ the empirical risk function is $\hat{R}_S(h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)^2$.

Example 15. Consider a learning problem where the true data generation distribution (unknown to the learner) is $g(z)$, the statistical model (known to the learner) is given by a sampling distribution $f(y|\theta)$ where the parameter θ is unknown. The goal is to learn θ. If we assume loss function

$$\ell(\theta, z) = \log\left(\frac{g(z)}{f_\theta(z)}\right)$$

then the risk is

$$R_g(\theta) = \mathrm{E}_{z\sim g}\left(\log\left(\frac{g(z)}{f_\theta(z)}\right)\right) = \mathrm{E}_{z\sim g}(\log(g(z))) - \mathrm{E}_{z\sim g}(\log(f_\theta(z))) \tag{2.2}$$

whose minimizer is

$$\theta^* = \arg\min_{\forall\theta}(R_g(\theta)) = \arg\min_{\forall\theta}(\mathrm{E}_{z\sim g}(-\log(f_\theta(z))))$$

as the first term in (2.2) is constant. Note that in the Maximum Likelihood Estimation technique the MLE θ_{MLE} is the minimizer of

$$\theta_{\mathrm{MLE}} = \arg\min_{\forall\theta}\left(\frac{1}{m}\sum_{i=1}^{m}(-\log(f_\theta(z_i)))\right)$$

where $S = \{y_1, ..., y_m\}$ is an IID sample from g. Hence, MLE θ_{MLE} can be considered as the minimizer of the empirical risk $R_S(\theta) = \frac{1}{m}\sum_{i=1}^{m}(-\log(f_\theta(z_i)))$.

Example 16. (Linear Regression) Consider the multiple linear regression problem $x \longmapsto y$ where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y \in \mathcal{Y} \subseteq \mathbb{R}^d$.

- The hypothesis is a linear function $h : \mathcal{X} \to \mathcal{Y}$ (that learner wishes to learn) to approximate mapping $x \longmapsto y$. The hypothesis set $\mathcal{H} = \left\{< w, x >\longmapsto y : w \in \mathbb{R}^d\right\}$. We can use the loss $\ell(h, (x, y)) = (h(x) - y)^2$.
- Equivalently, learning problem can be set differently because the predictor (linear function) is parametrized by $w \in \mathbb{R}^d$ as $< w, x >\longmapsto y$. Set $\mathcal{H} = \left\{w \in \mathbb{R}^d\right\}$. The set of examples is $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The loss is $\ell(w, (x, y)) = (< w, x > -y)^2$.

Machine Learning and Neural Networks (MATH3431) Epiphany term, 2023

Handout 1: Elements of convex learning problems

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To introduce elements of convexity, Lipschitzbness, and smoothmess that can be used for the analysis of stochastic gradient related learning algorithms.

Reading list & references:

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

1. Motivations

Note 1. Introducing convexity and smoothness in the learning problems makes easier the (theoretical) analysis of the problem and its solution.

Note 2. Most of the ML problems discussed in the course (eg, Artificial neural networks, Gaussian process regression) are usually non-convex.

Note 3. Extensions of handle non-convex problems as below can be done via surrogates –to be discussed.

2. Convex learning problem

Definition 4. Convex learning problem is a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$ is the learning problem that the hypothesis class $\mathcal{H}$ is a convex set, and the loss function ℓ is a convex function for each example $z \in \mathcal{Z}$.

Example 5. Multiple linear regression $< w, x >\rightarrow y$ with $y \in \mathbb{R}$, hypothesis class $\mathcal{H} = \left\{w \in \mathbb{R}^d\right\}$ and loss $\ell\left(w, (x, y)\right) = \left(< w, x > -y\right)^2$ with

$$w^* = \arg\min_{\forall w} \mathrm{E}\left(< w, x > -y\right)^2$$

or

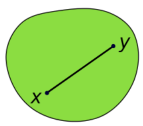
$$w^{**} = \arg\min_{\forall w} \frac{1}{m}\sum_{i=1}^{m}\left(< w, x_i > -y\right)^2$$

is a convex learning problem for arguments discussed below.

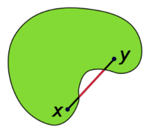
3. Convexity

Definition 6. A set C is convex if for any $u, v \in C$, the line segment between u and v is contained in C. Namely,

- for any $u, v \in C$ and for any $\alpha \in [0, 1]$ we have that $\alpha u + (1 - \alpha) v \in C$.



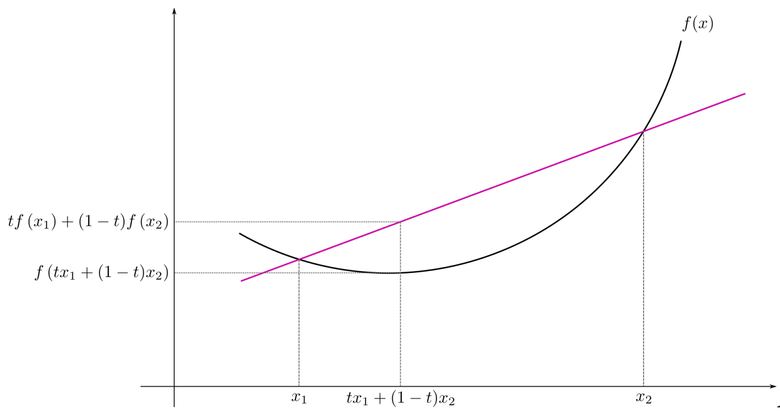
A convex set



A non-convex set

Definition 7. Let C be a convex set. A function $f : C \to R$ is convex function if for any $u, v \in C$ and for any $\alpha \in [0, 1]$

$$f(\alpha u + (1-\alpha) v) \leq \alpha f(u) + (1-\alpha) f(v)$$



A convex function

Example 8. The function $f : \mathbb{R} \to \mathbb{R}_+$ with $f(x) = x^2$ is convex function. For any $u, v \in C$ and for any $\alpha \in [0, 1]$ it is

$$(\alpha u + (1-\alpha) v)^2 \leq \alpha^2 (u)^2 + (1-\alpha)^2 (v)^2 + 2\alpha u (1-\alpha) v \leq \alpha (u)^2 + (1-\alpha)(v)^2$$

Proposition 9. *Every local minimum of a convex function is the global minimum.*

Proposition 10. *Let $f : C \to \mathbb{R}$ be convex function. The tangent of f at $w \in C$ is below f, namely*

$$\forall u \in C \quad f(u) \geq f(w) + < \nabla f(w), u - w >$$

Proposition 11. *Let $f : \mathbb{R}^d \to \mathbb{R}$ such that $f(w) = g(< w, x > + y)$ or some $x \in \mathbb{R}^d$, $y \in \mathbb{R}$. If $g$is convex function then $f$is convex function.*

Proof. See Exercise 1 in the Exercise sheet. □

Example 12. Consider the regression problem $x \mapsto y$ with $x \in \mathbb{R}^d$, $y \in \mathbb{R}$ and predictor $h(x) =< w, x >$. The risk $R(w) = (< w, x > + y)^2$ because $g(a) = (a)^2$ is convex and Proposition 11.

Example 13. Let $f_j : \mathbb{R}^d \to \mathbb{R}$ convex functions for $j = 1, ..., r$. Then:

(1) $g(x) = \max_{\forall j}(f_j(x))$ is a convex function
(2) $g(x) = \sum_{j=1}^{r} w_j f_j(x)$ is a convex function where $w_j > 0$

Solution.

(1) For any $u, v \in \mathbb{R}^d$ and for any $\alpha \in [0,1]$

$$\begin{aligned}
g(\alpha u + (1-\alpha) v) &= \max_{\forall j} (f_j (\alpha u + (1-\alpha) v)) \\
&\leq \max_{\forall j} (\alpha f_j (u) + (1-\alpha) f_j (v)) && (f_j \text{ is convex}) \\
&\leq \alpha \max_{\forall j} (f_j (u)) + (1-\alpha) \max_{\forall j} (f_j (v)) && (\max(\cdot) \text{ is convex}) \\
&\leq \alpha g(u) + (1-\alpha) g(v)
\end{aligned}$$

(2) For any $u, v \in \mathbb{R}^d$ and for any $\alpha \in [0,1]$

$$\begin{aligned}
g(\alpha u + (1-\alpha) v) &= \sum_{j=1}^{r} w_j f_j (\alpha u + (1-\alpha) v) \\
&\leq \alpha \sum_{j=1}^{r} w_j f_j (u) + (1-\alpha) \sum_{j=1}^{r} w_j f_j (v) && (f_j \text{ is convex}) \\
&\leq \alpha g(u) + (1-\alpha) g(v)
\end{aligned}$$

Example 14. $g(x) = |x|$ is convex according to Example 13, as $g(x) = |x| = \max(-x, x)$.

4. Lipschitzbness

Machine Learning and Neural Networks (MATH3431) Epiphany term, 2023

Handout 0: Learning problem: Definitions, notation, and formulation –A recap

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To get some definitions and set-up about the learning procedure; essentially to formalize what introduced in term 1.

Reading list & references:

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York: Springer.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

1. Introductions and loose definitions

Pattern recognition is the automated discovery of patterns and regularities in data $z \in \mathcal{Z}$. **Machine learning (ML)** are statistical procedures for building and understanding probabilistic methods that 'learn'. **ML algorithms** $\mathfrak{A}$ build a (probabilistic/deterministic) model able to make predictions or decisions with minimum human interference and can be used for pattern recognition. **Learning** (or training, estimation) is called the procedure where the ML model is tuned. **Training data** (or observations, sample data set, exemplars) is a set of observables $\{z_i \in \mathcal{Z}\}$ used to tune the parameters of the ML model. **Test set** is a set of available examples/observables $\{z_i'\}$ (different than the training data) used to verify the performance of the ML model for a given a measure of success. **Measure of success** (or performance) is a quantity that indicates how bad the corresponding ML model or Algorithm performs (eg quantifies the failure/error), and can also be used for comparisons among different ML models; eg, **Risk function** or **Empirical Risk Function**. Two main problems in ML are the supervised learning (we focus here) and the unsupervised learning.

Supervised learning problems involve applications where the training data $z \in \mathcal{Z}$ comprises examples of the input vectors $x \in \mathcal{X}$ along with their corresponding target vectors $y \in \mathcal{Y}$; i.e. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. **Classification problems** are those which aim to assign each input vector x to one of a finite number of discrete categories of y. **Regression problems** are those where the output y consists of one or more continuous variables. All in all, the learner wishes to recover an unknown pattern (i.e. functional relationship) between components $x \in \mathcal{X}$ that serves as inputs and components $y \in \mathcal{Y}$ that act as outputs; i.e. $x \longmapsto y$. Hence, $\mathcal{X}$ is the input domain, and $\mathcal{Y}$ is the output domain. The goal of learning is to discover a function which predicts $y \in \mathcal{Y}$ from $x \in \mathcal{X}$.

Unsupervised learning problems involve applications where the training data $z \in \mathcal{Z}$ consists of a set of input vectors $x \in \mathcal{X}$ without any corresponding target values ; i.e. $\mathcal{Z} = \mathcal{X}$. In clustering the goal is to discover groups of similar examples within the data of it is to discover groups of similar examples within the data.

2. (loose) Notation in learning

Definition 1. The learner's output is a function, $h : \mathcal{X} \to \mathcal{Y}$ which predicts $y \in \mathcal{Y}$ from $x \in \mathcal{X}$. It is also called hypothesis, prediction rule, predictor, or classifier.

Notation 2. We often denote the set of hypothesis as $\mathcal{H}$; i.e. $h \in \mathcal{H}$.

Definition 3. Training data set $\mathcal{S}$ of size m is any finite sequence of pairs $((x_i, y_i); i = 1, ..., m)$ in $\mathcal{X} \times \mathcal{Y}$. This is the information that the learner has assess.

Definition 4. Data generation model $g(\cdot)$ is the probability distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, unknown to the learner that has generated the data.

Definition 5. We denote as $\mathfrak{A}(\mathcal{S})$ the hypothesis (outcome) that a learning algorithm $\mathfrak{A}$ returns given training sample S.

Definition 6. (Loss function) Given any set of hypothesis $\mathcal{H}$ and some domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, a loss function $\ell(\cdot)$ is any function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$. The purpose of loss function $\ell(h, z)$ is to quantify the "error" for a given hypothesis h and example z –the greater the error the greater its value of the loss.

Example 7. In binary classification problems where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{Y} = \{0, 1\}$ is discrete, a loss function can be

$$\ell_{0-1}(h, (x, y)) = 1(h(x) = y),$$

Example 8. In regression problems $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\mathcal{Y}$ is uncountable, a loss function can be

$$\ell_{\text{sq}}(h, (x, y)) = (h(x) - y)^2$$

Definition 9. (Risk function) The risk function $R_g(h)$ of h is the expected loss of the hypothesis $h \in \mathcal{H}$, w.r.t. probability distribution g over domain Z; i.e.

$$R_g(h) = \mathrm{E}_{z \sim g}(\ell(h, z)) \tag{2.1}$$

Remark 10. In learning, an ideal way to obtain an optimal predictor h^* is to compute the risk minimizer

$$h^* = \arg\min_{\forall}(R_g(h))$$

Example 11. (Cont. Ex. 8)The risk function is $R_g(h) = \mathrm{E}_{z \sim g}(h(x) - y)^2$, and it measures the quality of the hypothesis function $h : \mathcal{X} \to \mathcal{Y}$ as the expected square difference between the predicted values h and the true target values y at every x.

Remark 12. Computing the risk minimizer may be practically challenging due to the integration w.r.t. the unknown data generation model g involved in the expectation (2.1). Suboptimally, one may resort to the Empirical risk function.

Definition 13. (Empirical risk function) The empirical risk function $\hat{R}_S(h)$ of h is the expectation of loss of h over a given sample $S = (z_1, ..., z_m) \in \mathcal{Z}^m$; i.e.

$$\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i).$$

Example 14. (Cont. Example 11) Given given sample $S=\{(x_i,y_i)\,;i=1,...,m\}$ the empirical risk function is $\hat{R}_S\left(h\right)=\frac{1}{m}\sum_{i=1}^{m}\left(h(x_i)-y_i\right)^2$.

Example 15. Consider a learning problem where the true data generation distribution (unknown to the learner) is $g(z)$, the statistical model (known to the learner) is given by a sampling distribution $f\left(y|\theta\right)$ where the parameter θ is unknown. The goal is to learn θ. If we assume loss function

$$\ell\left(\theta,z\right)=\log\left(\frac{g\left(z\right)}{f_\theta\left(z\right)}\right)$$

then the risk is

$$R_g\left(\theta\right)=\mathrm{E}_{z\sim g}\left(\log\left(\frac{g\left(z\right)}{f_\theta\left(z\right)}\right)\right)=\mathrm{E}_{z\sim g}\left(\log\left(g\left(z\right)\right)\right)-\mathrm{E}_{z\sim g}\left(\log\left(f_\theta\left(z\right)\right)\right) \tag{2.2}$$

whose minimizer is

$$\theta^*=\arg\min_{\forall\theta}\left(R_g\left(\theta\right)\right)=\arg\min_{\forall\theta}\left(\mathrm{E}_{z\sim g}\left(-\log\left(f_\theta\left(z\right)\right)\right)\right)$$

as the first term in (2.2) is constant. Note that in the Maximum Likelihood Estimation technique the MLE θ_{MLE} is the minimizer of

$$\theta_{\mathrm{MLE}}=\arg\min_{\forall\theta}\left(\frac{1}{m}\sum_{i=1}^{m}\left(-\log\left(f_\theta\left(z_i\right)\right)\right)\right)$$

where $S=\{y_1,...,y_m\}$ is an IID sample from g. Hence, MLE θ_{MLE} can be considered as the minimizer of the empirical risk $R_S\left(\theta\right)=\frac{1}{m}\sum_{i=1}^{m}\left(-\log\left(f_\theta\left(z_i\right)\right)\right)$.

Example 16. (Linear Regression) Consider the multiple linear regression problem $x\longmapsto y$ where $x\in\mathcal{X}\subseteq\mathbb{R}^d$ and $y\in\mathcal{Y}\subseteq\mathbb{R}^d$.

- The hypothesis is a linear function $h:\mathcal{X}\to\mathcal{Y}$ (that learner wishes to learn) to approximate mapping $x\longmapsto y$. The hypothesis set $\mathcal{H}=\left\{<w,x>\longmapsto y:w\in\mathbb{R}^d\right\}$. We can use the loss $\ell\left(h,(x,y)\right)=\left(h(x)-y\right)^2$.
- Equivalently, learning problem can be set differently because the predictor (linear function) is parametrized by $w\in\mathbb{R}^d$ as $<w,x>\longmapsto y$. Set $\mathcal{H}=\left\{w\in\mathbb{R}^d\right\}$. The set of examples is $\mathcal{Z}=\mathcal{X}\times\mathcal{Y}$. The loss is $\ell\left(w,(x,y)\right)=\left(<w,x>-y\right)^2$.