Statistical Learning: Algorithms and Theory

Sayan Mukherjee
# Contents

**Statistical Learning: Algorithms and Theory** 1

Lecture 1. Course preliminaries and overview 1

Lecture 2. The learning problem 5
  2.1. Key definitions in learning 5
  2.2. Imposing well-posedness 12
  2.3. Voting algorithms 13

Lecture 3. Reproducing Kernel Hilbert Spaces (rkhs) 15
  3.1. Hilbert Spaces 15
  3.2. Reproducing Kernel Hilbert Spaces (rkhs) 16
  3.3. The rkhs norm, complexity, and smoothness 20
  3.4. Gaussian processes and rkhs 21

Lecture 4. Three forms of regularization 23
  4.1. A result of the representer theorem 23
  4.2. Equivalence of the three forms 25

Lecture 5. Algorithms derived from Tikhonov regularization 27
  5.1. Kernel ridge-regression 27
  5.2. Support Vector Machines (SVMs) for classification 30
  5.3. Regularized logistics regression 37
  5.4. Spline models 41
  5.5. Convex Optimization 47

Lecture 6. Voting algorithms 51
  6.1. Boosting 51
  6.2. PAC learning 51
  6.3. The hypothesis boosting problem 53
  6.4. ADaptive BOOSTing (AdaBoost) 56
  6.5. A statistical interpretation of Adaboost 58
  6.6. A margin interpretation of Adaboost 59

Lecture 7. One dimensional concentration inequalities 63
  7.1. Law of Large Numbers 63
  7.2. Polynomial inequalities 63
  7.3. Exponential inequalities 64
  7.4. Martingale inequalities 67

Lecture 8. Vapnik-Červonenkis theory 71
8.1. Uniform law of large numbers 71
8.2. Generalization bound for one function 71
8.3. Generalization bound for a finite number of functions 72
8.4. Generalization bound for compact hypothesis spaces 72
8.5. Generalization bound for hypothesis spaces of indicator functions 76
8.6. Kolmogorov chaining 81
8.7. Covering numbers and VC dimension 84
8.8. Symmetrization and Rademacher complexities 85

Lecture 9. Generalization bounds 89
9.1. Generalization bounds and stability 89
9.2. Uniform stability of Tikhonov regularization 93

Bibliography 99
Course preliminaries and overview

• Course summary

The problem of supervised learning will be developed in the framework of statistical learning theory. Two classes of machine learning algorithms that have been used successfully in a variety of applications will be studied in depth: regularization algorithms and voting algorithms. Support vector machines (SVMs) are an example of a popular regularization algorithm and AdaBoost is an example of a popular voting algorithm. The course will
(1) introduce these two classes of algorithms
(2) illustrate practical uses of the algorithms via problems in computational biology and computer graphics
(3) state theoretical results on the generalization and consistency of these algorithms.

• Grading

Three problem sets for 50% of the grade. A final project for 50% of the grade. Possible final projects are
(1) application of a learning algorithm to data

1Institute of Statistics and Decision Sciences (ISDS) and Institute for Genome Sciences and Policy (IGSP), Duke University, Durham, 27708.
E-mail address: sayan@stat.duke.edu.

November 18, 2004

These lecture notes borrow heavily from the following courses at M.I.T. 9.520 and 18.465.
(2) critical review of topics in the class or related topics
(3) theoretical analysis of an algorithm
The student can pick a topic or select from predefined topics.

• Course outline
  (1) The supervised learning problem
  The problem of supervised learning will be introduced as function approximation given sparse data.
  (2) Regularization algorithms
    (a) Reproducing Kernel Hilbert Spaces
        A function class with some very nice properties.
    (b) Regularization methods
    (c) Algorithms derived from Tikhonov regularization
        (i) Kernel ridge-regression
        (ii) Support vector machines (SVMs)
        (iii) Regularized logistics regression
        (iv) Splines
    (d) Optimization
        Lagrange multipliers and primal/dual problems.
  (3) Voting algorithms
    (a) Examples of voting algorithms
    (b) Probably approximately correct (PAC) framework
        A theoretical framework introduced by Leslie Valiant used extensively in Learning theory.
    (c) Strong and Weak learners and the Boosting hypothesis
        Do there exist two kinds of algorithms: strong and weak algorithms, or are they equivalent in and that one can one boost weak algorithms into strong ones.
    (d) Boosting by majority and Adaptive boosting (AdaBoost).
  (4) Applications
    (a) Computational biology
        (i) Classifying tumors using SVMs
            Expression data from human tumors will be analyzed using SVMs. Both binary and multiclass problems will be addressed. The challenge here is high dimensionality and lack of samples.
        (ii) Predicting genetic regulatory response using boosting
            A boosting algorithm is used to predict gene regulatory response using expression and motif data from yeast.
    (b) Computer graphics
        Trainable videorealistic speech animation system is described that uses Tikhonov regularization to create an animation module. Given video of a pre-determined speech corpus from a human subject the algorithm is capable of synthesizing the human subject’s mouth uttering entirely novel utterances that were not recorded in the original video.
  (5) Theory
(a) Generalization and consistency
   How accurate a function learnt from a finite dataset will be on
   future data and as the number of data go to infinity will the
   optimal function in the class selected.

(b) Technical tools
   (i) One-dimensional concentration inequalities
       (A) Polynomial inequalities
       (B) Exponential inequalities
       (C) Martingale inequalities
   (ii) Vapnik-Červonenkis (VC) theory
       (A) Covering numbers and VC dimension
       (B) Growth functions and metric entropy
       (C) Uniform law of large numbers
       (D) Kolmogorov chaining and Dudley’s entropy integral
       (E) Rademacher averages

(c) Generalization bounds
   (i) Bounds for Tikhonov regularization using algorithmic sta-
       bility
   (ii) Bounds for empirical risk minimization algorithms using
        VC theory
   (iii) Generalization bounds for boosting
LECTURE 2

The learning problem

In this lecture the (supervised) learning problem is presented as the problem of function approximation from sparse data. The key ideas of loss functions, empirical error and expected error are introduced. The Empirical Risk Minimization (ERM) algorithm is introduced and three key requirements on this algorithm are described: generalization, consistency, and well-posedness. We then mention the regularization principle which ensures that the above condition are satisfied. We close with a brief introduction to voting algorithms.

2.1. Key definitions in learning

Our dataset consists of two sets of random variables $X \subseteq \mathbb{R}^d$ and $Y \subseteq \mathbb{R}^k$. Our general assumption will be that $X$ is a compact Euclidean space and $Y$ is closed subset. For most of this class $k = 1$. Given a dataset we would like to learn a function $f : X \rightarrow Y$.

For example the space $X$ can be measurements for a given country:

$$x = (\text{gdp}, \text{polity}, \text{infant mortality}, \text{population density}, \text{estimates of WMD}, \text{oil export}),$$

we will focus on two types of spaces $Y$ in these lectures.

1. Pattern recognition: the space $Y = \{0, 1\}$ corresponds to a binary variable indicating whether the country is bombed,
2. Regression: the space $Y \subseteq \mathbb{R}$ corresponds to a real-valued variable indicating how much the country is bombed (we assume there exists negative bombing here).

The dataset $S$ is often called the “training set” and consists of $n$ samples, $(x, y)$ pairs, drawn i.i.d. from a probability distribution $\mu(z)$, a distribution on the product space of $X$ and $Y$ ($Z = X \times Y$)

$$S = \{z_1 = (x_1, y_1), ..., z_n = (x_n, y_n)\}.$$

An important concept is the conditional probability of $y$ given $x$ $p(y|x)$

$$\mu(z) = p(y|x) \cdot p(x).$$

In the learning problem we assume $\mu(z)$ is fixed but unknown.
2.1.1. Algorithms, hypothesis spaces, and loss functions

**Proposition.** A learning algorithm \( A \) is a map from a data set \( S \) to a function \( f_S \):
\[
A : S \rightarrow f_S.
\]

**Proposition.** A hypothesis space \( \mathcal{H} \) is the space of functions that a learning algorithm \( A \) “searches” over.

The basic goal of supervised learning is to use the training set \( S \) to “learn” a function \( f_S \) that given a value \( x_{\text{new}} \) not in the training set will predict the associated value \( y_{\text{pred}} \):
\[
y_{\text{pred}} = f_S(x_{\text{new}}),
\]
not only for one value \( x_{\text{new}} \) but for a set of these values.

A loss function can be used to decide how well a function “fits” a training set or how well it predicts new observations.

**Definition.** A loss function will be defined as a nonnegative function of two variables \( V(f, z) := V(f(x), y) \):
\[
V : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+.
\]

The loss functions we will encounter will be nondecreasing in either
\[
|f(x) - y| \quad \text{or} \quad -y f(x).
\]

**Examples.** The following are loss functions used in general for regression problems.

1. **Square loss:** the most common loss function goes back to Gauss, sometimes called \( L_2 \) loss
   \[
   V(f(x), y) = (f(x) - y)^2,
   \]
2. **Absolute loss:** this loss function is less sensitive to outliers than the square loss, sometimes called \( L_1 \) loss
   \[
   V(f(x), y) = |f(x) - y|,
   \]
3. **Huber loss:** is less sensitive to outliers like the \( L_1 \) loss but is everywhere differentiable unlike the \( L_1 \) loss
   \[
   V(f(x), y) = \begin{cases} 
   \frac{1}{2}(f(x) - y)^2 & \text{if } |f(x) - y| \leq 2\epsilon, \\
   |f(x) - y| - \epsilon & \text{o.w.}
   \end{cases}
   \]
4. **Vapnik loss:** is less sensitive to outliers like the \( L_1 \) loss and in particular algorithms can lead to sparse solutions
   \[
   V(f(x), y) = \begin{cases} 
   0 & \text{if } |f(x) - y| \leq \epsilon, \\
   |f(x) - y| - \epsilon & \text{o.w.}
   \end{cases}
   \]

**Examples.** The following are loss functions used in general for classification problems.

1. **Indicator loss:** the most intuitive loss for binary classification
   \[
   V(f(x), y) = \Theta(-f(x)y) := \begin{cases} 
   1 & \text{if } -y f(x) \leq 0, \\
   0 & \text{o.w.}
   \end{cases}
   \]
(2) **Hinge loss:** unlike the Indicator loss this loss function is convex and therefore leads to practical algorithms that may have sparse solutions

\[
V(f(x), y) = (1 - f(x)y)_+ := \begin{cases} 
0 & \text{if } yf(x) \geq 1, \\
1 - f(x)y & \text{o.w.}
\end{cases}
\]

(3) **Quadratic hinge loss:** similar to the hinge loss but the deviation is squared

\[
V(f(x), y) = [(1 - f(x)y)_+]^2 := \begin{cases} 
0 & \text{if } yf(x) \geq 1, \\
(1 - f(x)y)^2 & \text{o.w.}
\end{cases}
\]

(4) **Logistic loss:** this is also a convex loss function with the advantage that a probabilistic model can be associated with it

\[
V(f(x), y) = \ln \left(1 + e^{-yf(x)}\right).
\]
Figure 2. Four loss functions for classification: indicator loss, hinge loss, square hinge loss, and logistic loss.
2.1.2. Empirical error, expected error, and generalization

Two key concepts in learning theory are the empirical error of a function and expected error of a function.

Definition. The empirical error of a function $f$ given a loss function $V$ and a training set $S$ of $n$ points is

$$ I_S[f] = n^{-1} \sum_{i=1}^{n} V(f, z_i) . $$

Definition. The expected error of a function $f$ given a loss function $V$ and a distribution $\mu$ is

$$ I[f] = \mathbb{E}_z V(f, z) := \int V(f, z) d\mu(z) . $$

Note that the expected error can almost never be computed since we almost never know $\mu(z)$. However, we often are given $S$ ($n$ points drawn from $\mu(z)$) so we can compute the empirical error. This observation motivates the principal of generalization which is fundamental to learning theory.

Proposition. An algorithm generalizes, $A_{gen}$, if its empirical error is close to its expected error:

$$ A_{gen} = \{ A : |I[f] - I_S[f]| \text{ is small} \} . $$

The advantage of algorithms that generalize is that if the empirical error is small then we have faith that the algorithm will be accurate on future observations. However, an algorithm that generalizes but has a large empirical error is not of much use.

2.1.3. Empirical Risk Minimization

A very common algorithm is the empirical risk minimization (ERM) algorithm.

Definition. Given a hypothesis space $\mathcal{H}$ a function $f_S$ is a minimizer of the empirical risk if

$$ f_S \in \arg\min_{f \in \mathcal{H}} I_S[f] := \arg\min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] . $$

If there exists no minimizer of empirical risk then a variation of the algorithm: almost ERM is used.

Definition. Given a hypothesis space $\mathcal{H}$ a function $f_S$ is an $\epsilon$-minimizer of the empirical risk if for any $\epsilon > 0$

$$ I_S[f_S] \leq \inf_{f \in \mathcal{H}} I_S[f] + \epsilon . $$

2.1.4. Desired properties of ERM

A natural property for an algorithm such as ERM to have is that as the number of observations increase the algorithm should find the best function in the hypothesis space. This property is defined as consistency.
**Definition.** ERM is universally consistent if
\[
\forall \varepsilon > 0 \lim_{n \to \infty} \sup_{\mu} \mathbb{P} \{ I[f_S] > \inf_{f \in \mathcal{H}} I[f] + \varepsilon \} = 0.
\]

One can observe (homework problem) that for ERM universal consistency and distribution independent generalization are equivalent.

**Definition.** ERM has distribution independent generalization if
\[
\forall \varepsilon > 0 \lim_{n \to \infty} \sup_{\mu} \mathbb{P} \{ |I[f_S] - I[f_S]| > \varepsilon \} = 0.
\]

Another set of desirable properties for ERM is that the mapping defined by ERM \((\mathcal{A} : S \to f_S)\) be well-posed. The notion of well-posedness goes back to Hadamard who defined this idea for operators.

**Definition.** A map is well-posed if its solution
1. exists
2. is unique
3. is stable (the function output varies smoothly with respect to perturbations of the input, the training set \(S\)).

The key requirement for ERM is stability since existence is ensured by the definition of ERM or almost ERM and uniqueness is defined modulo empirical error, all functions that have the same empirical error are equivalent.

We will see that the requirements of stability and consistency are complimentary and indeed equivalent for ERM.

The following two examples illustrate these issues.
Example. This example examines the stability of ERM when different function classes are used. The key point here is that ERM with a less complex or simpler function class is more stable.

The dataset is composed of 10 points and we fit the smoothest interpolating polynomial to this data. We then perturb the data slightly and refit the polynomial and note how much the function changes. Repeating this procedure with a second degree polynomial results in a much smaller change in the function.

**Figure 3.** The first figure consists of 10 points sampled from a function. In the second figure we fit the smoothest interpolating polynomial. The third figure displays the original data and the perturbed data. The fourth figure plots the smoothest interpolating polynomial for the two datasets. The fifth figure plots the original dataset with a 2nd order polynomial fit. The sixth figure plots both datasets with fit with 2nd order polynomials.
Example. This example is an illustration of how well-posedness and generalization are related. The key point here is that when ERM is well-posed the expected error will be smaller and in the limit this implies consistency.

Ten points are sampled from a sinusoid. We will use two algorithms to fit the data

1. $A_1$: ERM using a 7th order polynomial
2. $A_2$: ERM using a 15th order polynomial.

$A_1$ is well-posed and $A_2$ is not. Note that the expected error of $A_1$ is much smaller than the expected error of $A_2$.

![Graphs showing sinusoid and polynomial fits to data](image)

Figure 4. The first figure consists of 10 points sampled from a function. The second figure is the underlying function, a sinusoid. The third and fourth figures are a 7th and 15th order polynomial fit to the data.

2.2. Imposing well-posedness

The previous examples illustrated that ERM by itself is not well-posed. A standard approach to imposing well-posed to a procedure is via the principle of regularization. The principle of regularization involves constraining the hypothesis space. Two basic of the three basic methods of imposing constraints are

1. Ivanov regularization: directly constrain the hypothesis space

$$\min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \quad \text{subject to } \Omega(f) \leq \tau$$
(2) Tikhonov regularization: indirectly constrain the hypothesis space by adding a penalty term.

\[
\min_{f \in H} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \lambda \Omega(f) \right].
\]

The function \( \Omega(f) \) is called the regularization or smoothness functional and is typically a Hilbert space norm, see lecture 3, and the parameters \( \tau \) and \( \lambda \) are the regularization parameters that control the trade-off between fitting the data and constraining the hypothesis space.

2.3. Voting algorithms

An alternative to imposing well-posedness using constraints is to use algorithms that by construction are well-posed. Voting algorithms implement this approach, see lecture ??.

The idea behind voting algorithms is that we use a weighted combination of very simple functions called weak learners to build a more complicated function

\[
f(x) = \sum_{i=1}^{p} \alpha_i c_i(x) \hspace{1cm} \text{subject to} \hspace{1cm} \sum_{i=1}^{p} \alpha_i = 1 \text{ and } \alpha_i \geq 0,
\]

the functions \( c_i(x) \) are the weak classifiers and the weights \( \alpha_i \) can be predetermined or set based upon the observed data.
Reproducing Kernel Hilbert Spaces (rkhs)

Reproducing Kernel Hilbert Spaces (rkhs) are hypothesis spaces with some very nice properties. The main property of these spaces is the reproducing property which relates norms in the Hilbert space to linear algebra. This class of functions also has a nice interpretation in the context of Gaussian processes. Thus, they are important for computational, statistical, and functional reasons.

3.1. Hilbert Spaces

A function space is a space of functions where each function can be thought of as a point.

Examples. The following are three examples of function spaces defined on a subset of the real line. In these examples the subset of the real line we consider is $x \in [a, b]$ where for example $a = 0$ and $b = 10$.

1. $C[a, b]$ is the set of all real-valued continuous functions on $x \in [a, b]$. $y = x^3$ is in $C[a, b]$ while $y = [x]$ is not in $C[a, b]$.
2. $L_2[a, b]$ is the set of all square integrable functions on $x \in [a, b]$. If $(\int_a^b |f(x)|^2 \, dx)^{1/2} < \infty$ then $f \in L_2[a, b]$. $y = x^3$ is in $L_2[a, b]$ and so is $y = x^3 + \delta(x - c)$ where $a < c < b$, however the second function is not defined at $x = c$.
3. $L_1[a, b]$ is the set of all functions whose absolute value is integrable on $x \in [a, b]$. $y = x^3$ is in $L_1[a, b]$ and so is $y = x^3 + \delta(x - c)$ where $a < c < b$, however the second function is not defined at $x = c$.

Definition. A normed vector space is a space $\mathcal{F}$ in which a norm is defined. A function $\| \cdot \|$ is a norm iff for all $f, g \in \mathcal{F}$

1. $\|f\| \geq 0$ and $\|f\| = 0$ iff $f = 0$
2. $\|f + g\| \leq \|f\| + \|g\|$
3. $\|\alpha f\| = |\alpha| \|f\|.$

Note, if all conditions are satisfied except $\|f\| = 0$ iff $f = 0$ then the space has a seminorm instead of a norm.
Definition. An inner product space is a linear vector space $\mathcal{E}$ in which an inner product is defined. A real valued function $\langle \cdot, \cdot \rangle$ is an inner product iff \( \forall f, g, h \in \mathcal{E} \) and $\alpha \in \mathbb{R}$

1. $\langle f, g \rangle = \langle g, f \rangle$
2. $\langle f + g, h \rangle = \langle f, h \rangle + \langle g, h \rangle$ and $\langle \alpha f, g \rangle = \alpha \langle f, g \rangle$
3. $\langle f, f \rangle \geq 0$ and $\langle f, f \rangle = 0$ iff $f = 0$.

Given an inner product space the norm is defined as $\|f\| = \sqrt{\langle f, f \rangle}$ and an angle between vectors can be defined.

Definition. For a normed space $\mathcal{A}$ a subspace $\mathcal{B} \subset \mathcal{A}$ is dense in $\mathcal{A}$ iff $\mathcal{A} = \overline{\mathcal{B}}$.
Where $\overline{\mathcal{B}}$ is the closure of the set $\mathcal{B}$.

Definition. A normed space $\mathcal{F}$ is separable iff $\mathcal{F}$ has a countable dense subset.

Example. The set of all rational points is dense in the real line and therefore the real line is separable. Note, the set of rational points is countable.

Counterexample. The space of right continuous functions on $[0, 1]$ with the sup norm is not separable. For example, the step function

$$f(x) = U(x - a) \quad \forall a \in [0, 1]$$

cannot be approximated by a countable family of functions in the sup norm since the jump must occur at $a$ and the set of all $a$ is uncountable.

Definition. A sequence $\{x_n\}$ in a normed space $\mathcal{F}$ is called a Cauchy sequence if $\lim_{n \to \infty} \sup_{m \geq n} \|x_n - x_m\| = 0$.

Definition. A normed space $\mathcal{F}$ is called complete iff every Cauchy sequence in it converges.

Definition. A Hilbert space, $\mathcal{H}$ is an inner product space that is complete, separable, and generally infinite dimensional.
A Hilbert space has a countable basis.

Examples. The following are examples of Hilbert spaces.

1. $\mathbb{R}^n$ is the textbook example of a Hilbert space. Each point in the space $x \in \mathbb{R}^n$ can be represented as a vector $x = \{x_1, ..., x_n\}$ and the metric in this space is $\|x\| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$. The space has a very natural basis composed of the $n$ basis functions $e_1 = \{1, 0, ..., 0\}$, $e_2 = \{0, 1, ..., 0\}$, ..., $e_n = \{0, 0, ..., 1\}$. The inner product between a vector $x$ and a basis vector $e_i$ is simply the projection of $x$ onto the $i$th coordinate $x_i = \langle x, e_i \rangle$.
Note, this is not an infinite dimensional Hilbert space.

2. $L_2$ is also a Hilbert space. This Hilbert space is infinite dimensional.

3.2. Reproducing Kernel Hilbert Spaces (rkhs)

We will use two formulations to describe rkhs. The first is more general, abstract, and elegant. Of course it is less intuitive. The second is less general and constructive. Of course it is more intuitive.

For the remainder of this lecture we constrain the Hilbert spaces to a compact domain $X$. 
3.2.1. Abstract formulation

**Proposition.** A linear evaluation function $L_t$ evaluates each function in a Hilbert space $f \in \mathcal{H}$ at a point $t$. It associates $f \in \mathcal{H}$ to a number $f(t) \in \mathbb{R}$, $L_t[f] = f(t)$.

1. $L_t[f + g] = f(t) + g(t)$
2. $L_t[af] = af(t)$

**Example.** The delta function $\delta(x - t)$ is a linear evaluation function for $C[a, b]$

$$f(t) = \int_a^b f(x)\delta(x - t)dx.$$ 

**Proposition.** A linear evaluation function is bounded if there exists an $M$ such that for all functions in the Hilbert space $f \in \mathcal{H}$

$$|L_t[f]| = |f(t)| \leq M\|f\|,$$

where $\|f\|$ is the Hilbert space norm.

**Example.** For the Hilbert space $C[a, b]$ with the sup norm there exists a bounded linear evaluation function since $|f(x)| \leq M$ for all functions in $C[a, b]$. This is due to continuity and compactness of the domain. The evaluation function is simply $L_t[f]: t \rightarrow f(t)$ and $M = 1$.

**Counterexample.** For the hipert space $L^2[a, b]$ there exists no bounded linear evaluation function. The following function is in $L^2[a, b]$

$$y = x^3 + \delta(x - c) \text{ where } a < c < b.$$ 

At the point $x = c$ there is no $M$ such that $|f(c)| \leq M$ since the function is evaluated as “$\infty$”. This is an example of a function in the space that is not even defined pointwise.

**Definition.** If a Hilbert space has a bounded linear evaluation function, $L_t$, then it is a Reproducing Kernel Hilbert Space (rkhs), $\mathcal{H}_K$.

The following property of a rkhs is very important and is a result of the Riesz representation theorem.

**Proposition.** If $\mathcal{H}_K$ is a rkhs then there exists an element in the space $K_t$ with the property such that for all $f \in \mathcal{H}_K$

$$L_t[f] = \langle K_t, f \rangle = f(t).$$

The inner product is in the rkhs norm and the element $K_t$ is called the representer of evaluation of $t$.

**Remark.** The above property is somewhat amazing in that it says if a Hilbert space has a bounded linear evaluation function then there is an element in this space that evaluates all functions in the space by an inner product.

In the space $L^2[a, b]$ we say that the delta function evaluates all functions in $L^2[a, b]$

$$L_t[f] = \int_a^b f(x)\delta(x - t)dx.$$ 

However, the delta function is not in $L^2[a, b]$. 

**Definition.** The reproducing kernel (rk) is a symmetric real valued function of two variables \( s, t \in X \)

\[
K(s, t) : X \times X \to \mathbb{R}.
\]

In addition \( R(s, t) \) must be positive definite, that is for all real \( a_1, ..., a_n \) and \( t_1, ..., t_n \in X \)

\[
\sum_{i,j=1}^{n} a_ia_jK(t_i, t_j) \geq 0.
\]

If the above inequality is strict then \( K(s, t) \) is strictly positive definite.

**Remark.** Instead of characterizing quadratic forms of the function \( K(s, t) \) one can characterize the matrix \( K \) where \( K_{ij} = K(t_i, t_j) \) and use the notions of positive and positive-semidefinite matrices. The terminologies between analogous concepts for functions versus matrices is unfortunately very confusing.

There is a deep relation between a rkhs and its reproducing kernel. This is characterized by the following theorem.

**Theorem.** For every Reproducing Kernel Hilbert Space (rkhs) there exists a unique reproducing kernel and conversely given a positive definite function \( K \) on \( X \times X \) we can construct a unique rkhs of real valued functions on \( X \) with \( K \) as its reproducing kernel (rk).

**Proof.**

If \( \mathcal{H}_K \) is a rkhs then there exists an element in the rkhs that is the representer evaluation by the Reisz representer theorem. We define the rk

\[
K(s, t) := \langle K_s, K_t \rangle
\]

where \( K_s \) and \( K_t \) are the representers of evaluation at \( s \) and \( t \). The following hold by the properties of Hilbert spaces and the representer property

\[
\left\| \sum_j a_jK_{t_j} \right\|_2^2 \geq 0
\]

\[
\left\| \sum_j a_jK_{t_j} \right\|_2^2 = \sum_{i,j} a_ia_j \langle K_{t_i}, K_{t_j} \rangle
\]

\[
\sum_{i,j} a_ia_jK(t_i, t_j) = \sum_{i,j} a_ia_j \langle K_{t_i}, K_{t_j} \rangle.
\]

Therefore \( K(s, t) \) is positive definite.

We now prove the converse. Given a rk \( K(\cdot, \cdot) \) we construct \( \mathcal{H}_K \). For each \( t \in X \) we define the real valued function

\[
K_t(\cdot) = K(t, \cdot).
\]

We can show that the rkhs is simply the completion of the space of functions spanned by the the set \( K_t \)

\[
\mathcal{H} = \{ f \mid f = \sum_i a_iK_{t_i} \text{ where } a_i \in \mathbb{R}, t_i \in X, \text{ and } i \in \mathbb{N} \}
\]
with the following inner product
\[ \left\langle \sum_i a_i K_{t_i}, \sum_i a_i K_{t_i} \right\rangle = \sum_{i,j} a_i a_j \langle K_{t_i}, K_{t_j} \rangle = \sum_{i,j} a_i a_j K(t_i, t_j). \]

Since \( K(\cdot, \cdot) \) is positive definite the above inner product is well defined. For any \( f \in \mathcal{H}_K \) we can check that
\[ \langle K_{t}, f \rangle = f(t) \]
because for any function in the above linear space norm convergence implies pointwise convergence
\[ |f_n(t) - f(t)| = |\langle f_n - f, K_{t} \rangle| = \|f_n - f\| \|K_{t}\|, \]
the last step is due to Cauchy-Schwartz. Therefore every Cauchy sequence in this space converges and it is complete. □

### 3.2.2. Constructive formulation

The development of rkhs in this subsection is seen in most formulations of Support Vector Machines (SVMs) and Kernel Machines. It is less general in that it relies on the reproducing kernel being a Mercer Kernel. It however requires less knowledge of functional analysis and is more intuitive for most people.

In this formulation we start with a continuous kernel \( K : X \times X \to \mathbb{R} \). We define an integral operator \( L_K : L^2[X] \to C[X] \) by the following integral transform
\[
(3.1) \quad L_K f := \int_X K(s, t)f(t)dt = g(t).
\]

If \( K \) is positive definite then \( L_K \) is positive definite (the converse is also true) and therefore the eigenvalues of (3.1) are nonnegative.

We denote the eigenvalues and eigenvectors of (3.1) as \( \{\lambda_1, ..., \lambda_k\} \) and \( \{\phi_1, ..., \phi_k\} \) respectively, where
\[
\int_X K(s, t)\phi_k(t)dt = \lambda_k \phi_k(t) \quad \forall k.
\]

We now state Mercer’s theorem.

**Theorem.** Given the eigenfunctions and eigenvalues of the integral equation defined by a symmetric positive definite kernel \( K \)
\[
\int_X K(s, t)\phi(s)ds = \lambda \phi(t).
\]
The kernel has the expansion
\[
K(s, t) = \sum_j \lambda_j \phi_j(s)\phi_j(t),
\]
where convergence is in the \( L^2[X] \) norm.

We can define the rkhs as the space of functions spanned by the eigenfunctions of the integral operator defined by the kernel
\[
\mathcal{H}_K = \{ f | f(s) = \sum_k c_k \phi_k(s) \text{ and } \|f\|_{\mathcal{H}_K} < \infty \},
\]
where the rkhs norm $\| \cdot \|_{\mathcal{H}_K}$ is defined as follows

$$
\|f(s)\|^2_{\mathcal{H}_K} = \left\langle \sum_j c_j \phi_j(s), \sum_j c_j \phi_j(s) \right\rangle_{\mathcal{H}_K} := \sum_j c_j^2 \lambda_j.
$$

Similarly the inner product is defined as follows

$$
\langle f, g \rangle_{\mathcal{H}_K} = \left\langle \sum_j c_j \phi_j(s), \sum_j d_j \phi_j(s) \right\rangle_{\mathcal{H}_K} := \sum_j d_j c_j \lambda_j.
$$

Part of a homework problem will be to prove the representer property

$$
\langle f(\cdot, K(\cdot, x)) \rangle_{\mathcal{H}_K} = f(x),
$$

using Mercer’s theorem and the above definition of the rkhs norm.

### 3.2.3 Kernels and feature space

The rkhs concept has been utilized in the SVM and kernel machines literature in what is unfortunately called the kernel trick.

Points in the domains $x \in X \subset \mathbb{R}^d$ are mapped into a higher dimensional space by the eigenvalues and eigenfunctions of the reproducing kernel (the space is of the dimensionality of the number of nonzero eigenvalues of the integral operator defined by the kernel)

$$
x \rightarrow \Phi(x) := \{ \sqrt{\lambda_1} \phi_1(x), ..., \sqrt{\lambda_k} \phi_k(x) \}.
$$

A standard $L_2$ inner product of two points mapped into the feature space can be evaluated by a kernel due to Mercer’s theorem

$$
K(s, t) = \langle \Phi(s), \Phi(t) \rangle_{L_2}.
$$

### 3.3 The rkhs norm, complexity, and smoothness

We will measure the complexity of a hypothesis space using the the rkhs norm. We restrict our function space to consist of functions $f \in \mathcal{H}_K$ where

$$
\|f\|_{\mathcal{H}_K} \leq A.
$$

The next two examples illustrate how restricting the rkhs norm corresponds to enforcing some kind of “simplicity” or smoothness of the functions.

#### Example.

Our function space consists of one dimensional lines

$$
f(x) = wx \text{ and } K(s, t) = st.
$$

For this kernel

$$
\|f\|^2_{\mathcal{H}_K} = \|w\|^2
$$

so our measure of complexity is the slope of the line.

Our objective is to find a function with the smallest slope such that

$$
yf(x) \geq 1
$$

for all observed $(x, y)$ pairs. This can be thought of as separating the samples of the two classes by a margin of 2.

In the three following examples the slope of the function that separates the two classes by a margin of 2 increases as the points of the opposite class get closer.
Three datasets and a line to separate the classes. As the class distinctions become finer a larger slope is required to separate the classes.

\[ \begin{array}{ccc}
\begin{array}{c}
\text{x} \\
\hline
-2 \\
-1.5 \\
-1 \\
-0.5 \\
0 \\
0.5 \\
1 \\
1.5 \\
2
\end{array} & \begin{array}{c}
f(x) \\
\hline
-2 \\
-1.5 \\
-1 \\
-0.5 \\
0 \\
0.5 \\
1 \\
1.5 \\
2
\end{array} & \begin{array}{c}
f(x) \\
\hline
-2 \\
-1.5 \\
-1 \\
-0.5 \\
0 \\
0.5 \\
1 \\
1.5 \\
2
\end{array}
\end{array} \]

\[ x \]

\[ f(x) \]

\[ \begin{array}{c}
-2 \\
-1.5 \\
-1 \\
-0.5 \\
0 \\
0.5 \\
1 \\
1.5 \\
2
\end{array} \]

\[ f(X) \]

\[ \begin{array}{c}
-2 \\
-1.5 \\
-1 \\
-0.5 \\
0 \\
0.5 \\
1 \\
1.5 \\
2
\end{array} \]

**Figure 1.** Three datasets with the points in the two classes getting nearer. Note that the slope of the separating functions get steeper as the two classes approach each other.

**Example.** Consider any function of one variable that is continuous, symmetric and periodic with positive Fourier coefficients over the interval. The rk \( K(s, t) \) can be rewritten as \( K(s - t) = K(z) \) and can be expanded in a uniformly convergent Fourier series (all normalization factors set to 1)

\[
K(z) = \sum_{n=0}^{\infty} \lambda_n \cos(nz)
\]

\[
K(s - t) = 1 + \sum_{p=1}^{\infty} \lambda_p \sin(ps) \sin(pt) + \sum_{p=1}^{\infty} \lambda_p \cos(ps) \cos(pt),
\]

showing that the eigenfunctions of \( K \) are

\[
(1, \sin(z), \cos(z), \sin(2z), \cos(3z), ..., \sin(pz), \cos(pz), ...).
\]

So the rkhs norm of a function is

\[
\|f\|_{\mathcal{H}_K}^2 = \sum_{p=0}^{\infty} \frac{(f, \cos(pz))^2 + (f, \sin(pz))^2}{\lambda_p}.
\]

For the above rkhs norm to be finite the magnitude of the Fourier coefficients of the functions \( f \in \mathcal{H}_K \) has decay fast enough to counteract the decrease in \( \lambda_n \).

The picture below is an illustration of how the smoothness of a function can be characterized by the decay of the Fourier coefficients.

3.4. Gaussian processes and rkhs
Figure 2. The first function is smoother than the second function and the decay of the Fourier coefficients of the first function is quicker.
The three standard approaches to regularization will be described. Setting the regularization functional to a RKHS norm in all three approaches results in the solution having a particular form due to the representer theorem. Lastly the equivalence between of the three forms of regularization is specified.

4.1. A result of the representer theorem

The following are the three standard regularization methods for ERM:

1. Tikhonov regularization: indirectly constrain the hypothesis space by adding a penalty term.

\[
\min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \lambda \Omega(f) \right].
\]

2. Ivanov regularization: directly constrain the hypothesis space

\[
\min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \quad \text{subject to} \quad \Omega(f) \leq \tau.
\]

3. Phillips regularization: directly constrain the hypothesis space

\[
\min_{f \in \mathcal{H}} \Omega(f) \quad \text{subject to} \quad \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \leq \kappa.
\]

Throughout this text the RKHS norm will be used as the regularization functional

\[
\Omega(f) = \|f\|_{\mathcal{H}_K}^2.
\]

This defines the following optimization problems we will consider in this text:

\[
(P1) \quad \min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \lambda \|f\|_{\mathcal{H}_K}^2 \right],
\]

\[
(P2) \quad \min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \quad \text{subject to} \quad \|f\|_{\mathcal{H}_K}^2 \leq \tau,
\]

\[
(P3) \quad \min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}_K}^2 \quad \text{subject to} \quad \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \leq \kappa.
\]
All the above optimization problems above are over spaces of functions that contain an infinite number of functions. Using the formulation in section 3.2.2 we can write any function in the rkhs as

\[ H_K = \left\{ f | f(x) = \sum_k c_k \phi_k(x) \right\}, \]

so the optimization procedure is over the coefficients \( c_k \). The number of nonzero coefficients in the expansion defines the dimensionality of the rkhs and this can be infinite, for example the Gaussian kernel.

One of the amazing aspects of the above optimization problems is that the minimizer satisfies the form

\[ f(x) = \sum_{i=1}^n c_i K(x, x_i). \]

So the optimization procedure is over \( n \) real variables. This is formalized in the following “Representer Theorem.”

**Theorem.** Given a set of points \( \{(x_i, y_i)\}_{i=1}^n \) a function of the form

\[ f(x) = \sum_{i=1}^n c_i K(x, x_i), \]

is a minimizer of the following optimization procedure

\[ c ( (f(x_1), y_1), ..., (f(x_n), y_n) ) + \lambda g(\|f\|_{H_K}), \]

where \( \|f\|_{H_K} \) is a rkhs norm, \( g(\cdot) \) is monotonically increasing, and \( c \) is an arbitrary cost function.

Procedure (P1) is special case of the optimization procedure stated in the above theorem.

**Proof.** For ease of notation all norms and inner products in the proof are rkhs norms and inner products.

Assume that the function \( f \) has the following form

\[ f = \sum_{i=1}^n b_i \phi_i(x_i) + v, \]

where

\[ \langle \phi_i(x_i), v \rangle = 0 \quad \forall i = 1, .., n. \]

The orthogonality condition simple ensures that \( v \) is not in the span of \( \{\phi_i(x_i)\}_{i=1}^n \).

So for any point \( x_j \) \((j = 1, ..., n)\)

\[ f(x_j) = \left( \sum_{i=1}^n b_i \phi(x_i) + v, \phi(x_j) \right) = \sum_{i=1}^n b_i \langle \phi(x_i), \phi(x_j) \rangle, \]

so \( v \) has no effect on the cost function

\[ c ((f(x_1), y_1), ..., (f(x_n), y_n)) \]

We now look at the rkhs norm

\[ g(\|f\|) = g \left( \left\| \sum_{i=1}^n b_i \phi_i(x_i) + v \right\| \right) = g \left( \sqrt{\sum_{i=1}^n b_i \| \phi_i(x_i) \|^2 + \|v\|^2} \right) \geq g \left( \sqrt{\sum_{i=1}^n b_i \| \phi_i(x_i) \|^2} \right). \]
So the extra factor $v$ increases the rkhs norm and has effect on the cost functional and therefore must be zero and the function has the form

$$f = \sum_{i=1}^{n} b_i \phi_i(x_i),$$

and by the reproducing property

$$f(x) = \sum_{i=1}^{n} a_i K(x, x_i). \quad \square$$

Homework: proving a representer theorem for the other two regularization formulations.

### 4.2. Equivalence of the three forms

The three forms of regularization have a certain equivalence. The equivalence is that given a set of points $\{(x_i, y_i)\}_{i=1}^{n}$ the parameters $\lambda, \tau, \text{ and } \kappa$ can be set such that the same function $f(x)$ minimizes (P1), (P2), and (P3). Given this equivalence and the representer theorem for (P1) it is clear that a representer theorem holds for (P2) and (P3).

**Proposition.** Given a convex loss function function the following optimization procedures are equivalent

(P1) \[
\min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \lambda \|f\|_{\mathcal{H}_K}^2 \right],
\]

(P2) \[
\min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \quad \text{subject to } \|f\|_{\mathcal{H}_K}^2 \leq \tau,
\]

(P3) \[
\min_{f \in \mathcal{H}_K} \|f\|_{\mathcal{H}_K}^2 \quad \text{subject to } \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \leq \kappa.
\]

By equivalent we mean that if $f_0(x)$ is a solution of one of the problems then there exist parameters $\tau, \kappa, \lambda$ for which $f_0(x)$ is a of the others.

**Proof.**

Let $f_0$ be the solution of (P2) for a fixed $\tau$ and assume that the constraint under the optimization is tight ($\|f_0\|_{\mathcal{H}_K}^2 = \tau$). Let $\left[ n^{-1} \sum_{i=1}^{n} V(f_0, z_i) \right] = b$. By inspection the solution of (P3) with $\kappa = b$ will be $f_0$.

Let $f_0$ be the solution of (P3) for a fixed $\kappa$ and assume that the constraint under the optimization is tight ($\left[ n^{-1} \sum_{i=1}^{n} V(f_0, z_i) \right] = \kappa$). Let $\|f_0\|_{\mathcal{H}_K}^2 = b$. By inspection the solution of (P2) with $\tau = b$ will be $f_0$.

For both (P2) and (P3) the above argument can be adjusted for the case where the constraints are not tight but the solution $f_0$ is not necessarily unique.

Let $f_0$ be the solution of (P2) for a fixed $\tau$. Using Lagrange multipliers we can rewrite (P2) as

$$\min_{f \in \mathcal{H}_K, \alpha} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] + \alpha \left( \|f\|_{\mathcal{H}_K}^2 - \tau \right),$$

(4.1)
where $\alpha \geq 0$ the optimal $\alpha = \alpha_0$. By the Karush-Kuhn-Tucker (KKT) conditions (complimentary slackness) at optimality

$$
\alpha_0 (\| f_0 \|_{\mathcal{H}_K}^2 - \tau) = 0.
$$

If $\alpha_0 = 0$ then $\| f \|_{\mathcal{H}_K}^2 < \tau$ and we can rewrite equation (4.1) as

$$
\min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right],
$$

which corresponds to (P1) with $\lambda = 0$ and the minima is $f_0$. If $\alpha_0 > 0$ then $\| f \|_{\mathcal{H}_K}^2 = \tau$ and we can rewrite equation (4.1) as the following equivalent optimization procedures

$$
(P2) \quad \min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \alpha_0 (\| f \|_{\mathcal{H}_K}^2 - \tau) \right],
$$

$$
(P2) \quad \min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \alpha_0 \| f \|_{\mathcal{H}_K}^2 \right],
$$

which corresponds to (P1) with $\lambda = \alpha_0$ and the minima is $f_0$.

Let $f_0$ be the solution of (P3) for a fixed $\kappa$. Using Lagrange multipliers we can rewrite (P3) as

(4.2) \begin{align*}
\min_{f \in \mathcal{H}_K, \alpha} \| f \|_{\mathcal{H}_K}^2 + \alpha \left( n^{-1} \sum_{i=1}^{n} V(f, z_i) - \kappa \right),
\end{align*}

where $\alpha \geq 0$ with the optimal $\alpha = \alpha_0$. By the KKT conditions at optimality

$$
\alpha_0 \left( n^{-1} \sum_{i=1}^{n} V(f_0, z_i) - \kappa \right) = 0.
$$

If $\alpha_0 = 0$ then $\left[ n^{-1} \sum_{i=1}^{n} V(f_0, z_i) \right] < \kappa$ and we can rewrite equation (4.2) as

$$
\min_{f \in \mathcal{H}_K} \| f \|_{\mathcal{H}_K}^2,
$$

which corresponds to (P1) with $\lambda = \infty$ and the minima is $f_0$. If $\alpha_0 > 0$ then $\left[ n^{-1} \sum_{i=1}^{n} V(f_0, z_i) \right] = \kappa$ and we can rewrite equation (4.2) as the following equivalent optimization procedures

$$
(P3) \quad \min_{f \in \mathcal{H}_K} \| f \|_{\mathcal{H}_K}^2 + \alpha_0 \left( n^{-1} \sum_{i=1}^{n} V(f, z_i) - \kappa \right),
$$

$$
(P3) \quad \min_{f \in \mathcal{H}_K} \| f \|_{\mathcal{H}_K}^2 + \alpha_0 \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right],
$$

$$
(P3) \quad \min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \frac{1}{\alpha_0} \| f \|_{\mathcal{H}_K}^2 \right],
$$

which corresponds to (P1) with $\lambda = 1/\alpha_0$ and the minima is $f_0$. □
Algorithms derived from Tikhonov regularization

From Tikhonov regularization we will derive three popular and extensively used algorithms: Kernel ridge-regression, Support Vector Machines, and Spline models. For the first two algorithms we will use a RKHS norm as the regularization functional. For the spline models we will use a differential operator as the regularization functional and close with an open-problem regarding the relation between this functional and a RKHS norm.

5.1. Kernel ridge-regression

The Kernel ridge-regression (KRR) algorithm has been invented and reinvented many times and has been called a variety of names such as Regularization networks, Least Square Support Vector Machine (LSSVM), Regularized Least Square Classification (RLSC).

We start with Tikhonov regularization

$$\min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \lambda \Omega(f) \right]$$

and then set the regularization functional to a RKHS norm

$$\Omega(f) = \|f\|_{\mathcal{H}_K}^2$$

and use the square loss functional

$$n^{-1} \sum_{i=1}^{n} V(f, z_i) = n^{-1} \sum_{i=1}^{n} (f(x_i) - y_i)^2.$$ 

The resulting optimization problem is

$$\min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}_K}^2 \right],$$

the minimizer of which we know by the Representer theorem has the form

$$f(x) = \sum_{i=1}^{n} c_i K(x, x_i).$$
This implies that we only need to solve the optimization problem for the \( c_i \). This turns the problem of optimizing over functions which may be infinite-dimensional into a problem of optimizing over \( n \) real numbers.

Using the representer theorem we derive the optimization problem actually solved for Kernel ridge-regression.

We first define some notation. We will use the symbol \( K \) to refer to either the kernel function \( K \) or the \( n \times n \) matrix \( K \) where

\[
K_{ij} \equiv K(x_i, x_j).
\]

Using this definition the function \( f(x) \) evaluated at a training point \( x_j \) can be written in matrix notation as

\[
f(x_j) = \sum_{i=1}^{n} K(x_i, x_j) c_i
= [Kc]_j,
\]

where \([Kc]_j\) is the jth element of the vector obtained in multiplying the kernel matrix \( K \) with the vector \( c \). In this notation we can rewrite equation (5.1) as

\[
\min_{f \in \mathcal{H}_K} \frac{1}{n} \sum_{i} (Kc - y)^2 + \lambda \|f\|^2_K,
\]

where \( y \) is the vector of \( y \) values. Also by the representer theorem the RKHS norm can be evaluated using linear algebra

\[
\|f\|^2_K = c^T Kc,
\]

where \( c^T \) is the transpose of the vector \( c \). Substituting the above norm into equation (5.1) results in an optimization problem on the vector \( c \)

\[
\arg \min_{c \in \mathbb{R}^n} \left[ g(c) := \frac{1}{\ell}(Kc - y)^2 + \lambda c^T Kc \right].
\]

This is a convex, differentiable function of \( c \), so we can minimize it simply by taking the derivative with respect to \( c \), then setting this derivative to 0.

\[
\frac{\partial g(c)}{\partial c} = 2 \frac{1}{\ell} K(Kc - y) + 2\lambda Kc = 0.
\]

We show that the solution of the above equation is the following linear system

\[
c = (K + \lambda I)^{-1} y,
\]

where \( I \) is the identity matrix:

\[
\text{differentiation} \quad \frac{d}{dc} 0 = 2 \frac{1}{\ell} K(Kc - y) + 2\lambda Kc
\]

\[
\text{multiplication} \quad K(Kc) + \lambda \ell Kc = Ky
\]

\[
\text{“left multiplication by } K^{-1}\text{”} \quad (K + \lambda I)c = y
\]

\[
\text{inversion} \quad c = (K + \lambda I)^{-1} y.
\]

The matrix \( K + \lambda I \) is positive definite and will be well-conditioned if \( \lambda \) is not too small.

A few properties of the linear system are:
(1) The matrix $(K + \lambda I)$ is guaranteed to be invertible if $\lambda > 0$. As $\lambda \to 0$, the regularized least-squares solution goes to the standard Gaussian least-squares solution which minimizes the empirical loss. As $\lambda \to \infty$, the solution goes to $f(x) = 0$.

(2) In practice, we don’t actually invert $(K + \lambda I)$, but instead use an algorithm for solving linear systems.

(3) In order to use this approach, we need to compute and store the entire kernel matrix $K$. This makes it impractical for use with very large training sets.

Lastly, there is nothing to stop us for using the above algorithm for classification. By doing so, we are essentially treating our classification problem as a regression problem with $y$ values of 1 or -1.

5.1.1. Solving for $c$

The conjugate gradient (CG) algorithm is a popular algorithm for solving positive definite linear systems. For the purposes of this class, we need to know that CG is an iterative algorithm. The major operation in CG is multiplying a vector $v$ by the matrix $A$. Note that matrix $A$ need not always be supplied explicitly, we just need some way to form a product $Av$.

For ordinary positive semidefinite systems, CG will be competitive with direct methods. CG can be much faster if there is a way to multiply by $A$ quickly.

**Example.** Suppose our kernel $K$ is linear:

$$K(x, y) = \langle x, y \rangle.$$  

Then our solution $x$ can be written as

$$f(x) = \sum c_i \langle x_i, x \rangle = \langle \left( \sum c_i x_i \right), x \rangle := \langle w, x \rangle,$$

and we can apply our function to new examples in time $d$ rather than time $nd$.

This is a general property of Tikhonov regularization with a linear kernel, not related to the use of the square loss.

We can use the CG algorithm to get a huge savings for solving regularized least-squares regression with a linear kernel $(K(x_1, x_2) = x_1 \cdot x_2)$. With an arbitrary kernel, we must form a product $Kv$ explicitly — we multiply a vector by $K$. With the linear kernel, we note that $K = AA^T$, where $A$ is a matrix with the data points as row vectors. Using this:

$$(K + \lambda n I)v = (AA^T + \lambda n I)v = A(A^Tv) + \lambda n Iv.$$  

Suppose we have $n$ points in $d$ dimensions. Forming the kernel matrix $K$ explicitly takes $n^2d$ time, and multiplying a vector by $K$ takes $n^2$ time.

If we use the linear representation, we pay nothing to form the kernel matrix, and multiplying a vector by $K$ takes $2dn$ time.

If $d << n$, we save approximately a factor of $\frac{n}{2d}$ per iteration. The memory savings are even more important, as we cannot store the kernel matrix at all for
large training sets, and if were to recompute the entries of the kernel matrix as needed, each iteration would cost \( n^2 d \) time.

Also note that if the training data are sparse (they consist of a large number of dimensions, but the majority of dimensions for each point are zero), the cost of multiplying a vector by \( K \) can be written as \( 2\bar{d}n \), where \( \bar{d} \) is the average number of nonzero entries per data point.

This is often the case for applications relating to text, where the dimensions will correspond to the words in a “dictionary”. There may be tens of thousands of words, but only a few hundred will appear in any given document.

5.2. Support Vector Machines (SVMs) for classification

SVMs have been used in a multitude of applications and are one of the most popular machine learning algorithms. We will derive the SVM algorithm from two perspectives: Tikhonov regularization, and the more common geometric perspective.

5.2.1. SVMs from Tikhonov regularization

We start with Tikhonov regularization

\[
\min_{f \in H} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) + \lambda \Omega(f) \right]
\]

and then set the regularization functional to a RKHS norm

\[ \Omega(f) = \| f \|^2_{\mathcal{H}_K} \]

and use the hinge loss functional

\[
n^{-1} \sum_{i=1}^{n} V(f, z_i) := n^{-1} \sum_{i=1}^{n} (1 - y_i f(x_i))^+,\]

where \((k)^+ := \max(k, 0)\).

![Figure 1. Hinge loss.](image)

The resulting optimization problem is

\[
(5.2) \quad \min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} (1 - y_i f(x_i))^+ + \lambda \| f \|^2_{\mathcal{H}_K} \right],
\]
which is non-differentiable at \((1 - y_i f(x_i)) = 0\) so we introduce slack variables and write the following constrained optimization problem:

\[
\min_{f \in \mathcal{H}_K} n^{-1} \sum_{i=1}^{n} \xi_i + \lambda ||f||_K^2
\]
subject to:
\[
y_i f(x_i) \geq 1 - \xi_i \quad i = 1, \ldots, n
\]
\[
\xi_i \geq 0 \quad i = 1, \ldots, n.
\]

By the Representer theorem we can rewrite the above constrained optimization problem as a constrained quadratic programming problem

\[
\min_{c \in \mathbb{R}^n} \quad n^{-1} \sum_{i=1}^{n} \xi_i + \lambda c^T K c
\]
subject to:
\[
y_i \sum_{j=1}^{n} c_j K(x_i, x_j) \geq 1 - \xi_i \quad i = 1, \ldots, n
\]
\[
\xi_i \geq 0 \quad i = 1, \ldots, n.
\]

The SVM contains an unregularized bias term \(b\) so the Representer theorem results in a function

\[
f(x) = \sum_{i=1}^{n} c_i K(x, x_i) + b.
\]

Plugging this form into the above constrained quadratic problem results in the “primal” SVM

\[
\min_{c \in \mathbb{R}^n, \xi \in \mathbb{R}^n} \quad n^{-1} \sum_{i=1}^{n} \xi_i + \lambda e^T K c
\]
subject to:
\[
y_i \left( \sum_{j=1}^{n} c_j K(x_i, x_j) + b \right) \geq 1 - \xi_i \quad i = 1, \ldots, n
\]
\[
\xi_i \geq 0 \quad i = 1, \ldots, n.
\]

We now derive the Wolfe dual quadratic program using Lagrange multiplier techniques:

\[
L(c, \xi, b, \alpha, \zeta) = n^{-1} \sum_{i=1}^{n} \xi_i + \lambda e^T K c
\]
\[-\sum_{i=1}^{n} \alpha_i \left( y_i \left( \sum_{j=1}^{n} c_j K(x_i, x_j) + b \right) - 1 + \xi_i \right)
\]
\[-\sum_{i=1}^{n} \zeta_i \xi_i.
\]

We want to minimize \(L\) with respect to \(c, b, \) and \(\xi,\) and maximize \(L\) with respect to \(\alpha\) and \(\zeta,\) subject to the constraints of the primal problem and nonnegativity constraints on \(\alpha\) and \(\zeta.\) We first eliminate \(b\) and \(\xi\) by taking partial derivatives:

\[
\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{n} \alpha_i y_i = 0
\]
\[
\frac{\partial L}{\partial \xi_i} = 0 \implies \frac{1}{n} - \alpha_i - \zeta_i = 0 \implies 0 \leq \alpha_i \leq \frac{1}{n}.
\]
The above two conditions will be constraints that will have to be satisfied at optimality. This results in a reduced Lagrangian:

\[ L^R(c, \alpha) = \lambda c^T Kc - \sum_{i=1}^{n} \alpha_i \left( y_i \sum_{j=1}^{n} c_j K(x_i, x_j) - 1 \right) . \]

We now eliminate \( c \)

\[ \frac{\partial L^R}{\partial c} = 0 \implies 2\lambda Kc - KY\alpha = 0 \implies c_i = \frac{\alpha_i y_i}{2\lambda}, \]

where \( Y \) is a diagonal matrix whose \( i \)th diagonal element is \( y_i \); \( Y\alpha \) is a vector whose \( i \)th element is \( \alpha_i y_i \). Substituting in our expression for \( c \), we are left with the following “dual” program:

\[
\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2\lambda} \alpha^T Q \alpha \\
\text{subject to : } \sum_{i=1}^{n} y_i \alpha_i = 0 \\
\quad \quad 0 \leq \alpha_i \leq \frac{1}{n} \quad i = 1, \ldots, n,
\]

where \( Q \) is the matrix defined by

\[ Q = y^T y \iff Q_{ij} = y_i y_j K(x_i, x_j). \]

In most of the SVM literature, instead of the regularization parameter \( \lambda \), regularization is controlled via a parameter \( C \), defined using the relationship

\[ C = \frac{1}{2\lambda n} . \]

Using this definition (after multiplying our objective function by the constant \( \frac{1}{2n} \)), the basic regularization problem becomes

\[
\min_{f \in H_K} C \sum_{i=1}^{n} V(y_i, f(x_i)) + \frac{1}{2} ||f||^2_K. 
\]

Like \( \lambda \), the parameter \( C \) also controls the trade-off between classification accuracy and the norm of the function. The primal and dual problems become respectively:

\[
\min_{c \in \mathbb{R}^n, \xi \in \mathbb{R}^n} C \sum_{i=1}^{n} \xi_i + \frac{1}{2} c^T Kc \\
\text{subject to : } y_i \left( \sum_{j=1}^{n} c_j K(x_i, x_j) + b \right) \geq 1 - \xi_i \quad i = 1, \ldots, n \\
\quad \quad \xi_i \geq 0 \quad i = 1, \ldots, n
\]

\[
\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \alpha^T Q \alpha \\
\text{subject to : } \sum_{i=1}^{n} y_i \alpha_i = 0 \\
\quad \quad 0 \leq \alpha_i \leq C \quad i = 1, \ldots, n.
\]

### 5.2.2. SVMs from a geometric perspective

The “traditional” approach to developing the mathematics of SVM is to start with the concepts of separating hyperplanes and margin. The theory is usually developed in a linear space, beginning with the idea of a perceptron, a linear hyperplane that separates the positive and the negative examples. Defining the margin as the distance from the hyperplane to the nearest example, the basic observation is that
intuitively, we expect a hyperplane with larger margin to generalize better than one with smaller margin.

![Figure 2](image_url)

Figure 2. Two hyperplanes (a) and (b) perfectly separate the data. However, hyperplane (b) has a larger margin and intuitively would be expected to be more accurate on new observations.

We denote our hyperplane by $w$, and we will classify a new point $x$ via the function

$$f(x) = \text{sign}\left[\langle w, x \rangle\right].$$

Given a separating hyperplane $w$, we let $x$ be a datapoint closest to $w$, and we let $x^w$ be the unique point on $w$ that is closest to $x$. Obviously, finding a maximum margin $w$ is equivalent to maximizing $||x - x^w||$. So for some $k$ (assume $k > 0$ for convenience),

$$\langle w, x \rangle = k$$
$$\langle w, x^w \rangle = 0$$
$$\langle w, (x - x^w) \rangle = k.$$

Noting that the vector $x - x^w$ is parallel to the normal vector $w$,

$$\langle w, (x - x^w) \rangle = \left\langle w, \left(\frac{||x - x^w||}{||w||}w\right) \right\rangle$$
$$= ||w||^2 \frac{||x - x^w||}{||w||}$$
$$= ||w|| \frac{||x - x^w||}{||w||}$$
$$k = ||w|| \frac{||(x - x^w)||}{||w||}$$
$$= ||x - x^w||.$$

$k$ is a “nuisance parameter” and without any loss of generality, we fix $k$ to 1, and see that maximizing $||x - x^w||$ is equivalent to maximizing $\frac{1}{||w||}$, which in turn is equivalent to minimizing $||w||$ or $||w||^2$. We can now define the margin as the distance between the hyperplanes $\langle w, x \rangle = 0$ and $\langle w, x \rangle = 1.$
So if the data is linear separable case and the hyperplanes run through the origin the maximum margin hyperplane is the one for which

\[ \min_{w \in \mathbb{R}^n} ||w||^2 \]

subject to: \[ y_i \langle w, x_i \rangle \geq 1 \quad i = 1, \ldots, n. \]

The SVM introduced by Vapnik includes an unregularized bias term \( b \), leading to classification via a function of the form:

\[ f(x) = \text{sign}(\langle w, x \rangle + b). \]

In addition, we need to work with datasets that are not linearly separable, so we introduce slack variables \( \xi_i \), just as before. We can still define the margin as the distance between the hyperplanes \( \langle w, x \rangle = 0 \) and \( \langle w, x \rangle = 1 \), but the geometric intuition is no longer as clear or compelling.

With the bias term and slack variables the primal SVM problem becomes

\[ \min_{w \in \mathbb{R}^n, b \in \mathbb{R}} C \sum_{i=1}^{n} \xi_i + \frac{1}{2}||w||^2 \]

subject to:

\[ y_i (\langle w, x \rangle + b) \geq 1 - \xi_i \quad i = 1, \ldots, n \]

\[ \xi_i \geq 0 \quad i = 1, \ldots, n. \]

Using Lagrange multipliers we can derive the same dual from in the previous section.

Historically, most developments begin with the geometric form, derived a dual program which was identical to the dual we derived above, and only then observed that the dual program required only dot products and that these dot products could be replaced with a kernel function. In the linearly separable case, we can also derive the separating hyperplane as a vector parallel to the vector connecting the closest two points in the positive and negative classes, passing through the perpendicular bisector of this vector. This was the “Method of Portraits”, derived by Vapnik in the 1970’s, and recently rediscovered (with non-separable extensions) by Keerthi.

### 5.2.3. Optimality conditions

The primal and the dual are both feasible convex quadratic programs. Therefore, they both have optimal solutions, and optimal solutions to the primal and the dual have the same objective value.

We derived the dual from the primal using the (now reparameterized) Lagrangian:

\[
L(c, \xi, b, \alpha, \zeta) = C \sum_{i=1}^{n} \xi_i + c^T K c \\
- \sum_{i=1}^{n} \alpha_i \left( y_i \left( \sum_{j=1}^{n} c_j K(x_i, x_j) + b \right) - 1 + \xi_i \right) \\
- \sum_{i=1}^{n} \zeta_i \xi_i.
\]
We now consider the dual variables associated with the primal constraints:

\[
\alpha_i \quad \text{implies} \quad y_i \left( \sum_{j=1}^{n} c_j K(x_i, x_j) + b \right) - 1 + \xi_i \\
\zeta_i \quad \text{implies} \quad \xi_i \geq 0.
\]

Complementary slackness tells us that at optimality, either the primal inequality is satisfied at equality or the dual variable is zero. In other words, if \( c, \xi, b, \alpha \) and \( \zeta \) are optimal solutions to the primal and dual, then

\[
\alpha_i \left( y_i \left( \sum_{j=1}^{n} c_j K(x_i, x_j) + b \right) - 1 + \xi_i \right) = 0 \\
\zeta_i \xi_i = 0
\]

All optimal solutions must satisfy:

\[
\sum_{j=1}^{n} c_j K(x_i, x_j) - \sum_{j=1}^{n} y_i \alpha_j K(x_i, x_j) = 0 \quad i = 1, \ldots, n \\
\sum_{i=1}^{n} \alpha_i y_i = 0 \\
C - \alpha_i - \zeta_i = 0 \quad i = 1, \ldots, n \\
y_i \left( \sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j) + b \right) - 1 + \xi_i \geq 0 \quad i = 1, \ldots, n \\
\alpha_i \left[ y_i \left( \sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j) + b \right) - 1 + \xi_i \right] = 0 \quad i = 1, \ldots, n \\
\zeta_i \xi_i = 0 \quad i = 1, \ldots, n \\
\xi_i, \alpha_i, \zeta_i \geq 0 \quad i = 1, \ldots, n
\]

The above optimality conditions are both necessary and sufficient. If we have \( c, \xi, b, \alpha \) and \( \zeta \) satisfying the above conditions, we know that they represent optimal solutions to the primal and dual problems. These optimality conditions are also known as the Karush-Kuhn-Tucker (KKT) conditions.

Suppose we have the optimal \( \alpha_i \)'s. Also suppose (this “always” happens in practice”) that there exists an \( i \) satisfying \( 0 < \alpha_i < C \). Then

\[
\alpha_i < C \quad \implies \quad \zeta_i > 0 \\
\implies \quad \zeta_i = 0 \\
\implies \quad y_i \left( \sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j) + b \right) - 1 = 0 \\
\implies \quad b = y_i - \sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j)
\]

So if we know the optimal \( \alpha \)'s, we can determine \( b \).
Defining our classification function \( f(x) \) as

\[
f(x) = \sum_{i=1}^{t} y_i \alpha_i K(x, x_i) + b,
\]

we can derive “reduced” optimality conditions. For example, consider an \( i \) such that \( y_i f(x_i) < 1 \):

\[
y_i f(x_i) < 1 \implies \xi_i > 0 \\
\implies \zeta_i = 0 \\
\implies \alpha_i = C.
\]

Conversely, suppose \( \alpha_i = C \):

\[
\alpha_i = C \implies y_i f(x_i) - 1 + \xi_i = 0 \\
\implies y_i f(x_i) \leq 1.
\]

Figure 3. A geometric interpretation of the reduced optimality conditions. The open squares and circles correspond to cases where \( \alpha_i = 0 \). The dark circles and squares correspond to cases where \( y_i f(x_i) = 1 \) and \( \alpha_i \leq C \), these are samples at the margin. The grey circles and squares correspond to cases where \( y_i f(x_i) < 1 \) and \( \alpha_i = C \).

5.2.4. Solving the SVM optimization problem

Our plan will be to solve the dual problem to find the \( \alpha \)'s, and use that to find \( b \) and our function \( f \). The dual problem is easier to solve the primal problem. It has simple box constraints and a single inequality constraint, even better, we will see that the problem can be decomposed into a sequence of smaller problems.
We can solve QPs using standard software. Many codes are available. Main problem — the \( Q \) matrix is dense, and is \( n \times n \), so we cannot write it down. Standard QP software requires the \( Q \) matrix, so is not suitable for large problems.

To get around this memory issue we partition the dataset into a working set \( W \) and the remaining points \( R \). We can rewrite the dual problem as:

\[
\begin{align*}
\max_{\alpha_W \in \mathbb{R}^{|W|}, \alpha_R \in \mathbb{R}^{|R|}} & \quad \sum_{i \in W} \alpha_i + \sum_{i \in R} \alpha_i - \frac{1}{2} [\alpha_W \alpha_R] \begin{bmatrix} Q_{WW} & Q_{WR} \\ Q_{RW} & Q_{RR} \end{bmatrix} \begin{bmatrix} \alpha_W \\ \alpha_R \end{bmatrix} \\
\text{subject to:} & \quad \sum_{i \in W} y_i \alpha_i + \sum_{i \in R} y_i \alpha_i = 0 \\
& \quad 0 \leq \alpha_i \leq C, \; \forall i.
\end{align*}
\]

Suppose we have a feasible solution \( \alpha \). We can get a better solution by treating the \( \alpha_W \) as variable and the \( \alpha_R \) as constant. We can solve the reduced dual problem:

\[
\begin{align*}
\max_{\alpha_W \in \mathbb{R}^{|W|}} & \quad (1 - Q_{WR} \alpha_R) \alpha_W - \frac{1}{2} \alpha_W Q_{WW} \alpha_W \\
\text{subject to:} & \quad \sum_{i \in W} y_i \alpha_i = - \sum_{i \in R} y_i \alpha_i \\
& \quad 0 \leq \alpha_i \leq C, \; \forall i \in W.
\end{align*}
\]

The reduced problems are fixed size, and can be solved using a standard QP code. Convergence proofs are difficult, but this approach seems to always converge to an optimal solution in practice.

An important issue in the decomposition is selecting the working set. There are many different approaches. The basic idea is to examine points not in the working set, find points which violate the reduced optimality conditions, and add them to the working set. Remove points which are in the working set but are far from violating the optimality conditions.

### 5.3. Regularized logistic regression

One drawback with the SVM is that the method does not explicitly output a probability or likelihood of the labels, instead the output is a real value the magnitude of which should be monotonic with respect to the probability

\[ P(y = \pm 1|x) \propto y f(x). \]

This issue can be addressed by using a loss function based upon logistic or binary regression. The main idea behind logistic regression is that we are trying to model the log likelihood ratio by the function \( f(x) \)

\[ f(x) = \log \left( \frac{P(y = 1|x)}{P(y = -1|x)} \right). \]

Since \( P(y = 1|x) \) is a Bernoulli random variable we can rewrite the above equation as

\[
\begin{align*}
 f(x) &= \log \left( \frac{P(y = 1|x)}{P(y = -1|x)} \right) \\
&= \log \left( \frac{P(y = 1|x)}{1 - P(y = 1|x)} \right).
\end{align*}
\]
which implies
\[
P(y = 1|x) = \frac{1}{1 + \exp(f(x))},
\]
\[
P(y = -1|x) = \frac{1}{1 + \exp(-f(x))},
\]
\[
P(y = \pm 1|x) = \frac{1}{1 + \exp(yf(x))}.
\]

Given a data set \(D = \{(x_i, y_i)\}_{i=1}^n\) and a class of functions \(f \in \mathcal{H}\) the maximum likelihood estimator (MLE) is the function that maximizes the likelihood of observing the data set \(D\)
\[
f_{\text{MLE}}^* := \arg \max_{f \in \mathcal{H}} [P(D|f)] = \arg \max_{f \in \mathcal{H}} \left[ \prod_{i=1}^n \frac{1}{1 + \exp(y_i f(x_i))} \right].
\]

As in the case of Empirical risk minimization the MLE estimate may overfit the data since there is no smoothness or regularization term. A classical way of imposing smoothness in this context is by placing a prior on the functions \(f \in \mathcal{H}\)
\[
P(f) \propto \exp(-\|f\|_H^2).
\]

Given a prior and a likelihood we can use Bayes rule to compute the posterior distribution \(P(f|D)\)
\[
P(f|D) = \frac{P(D|f)P(f)}{P(D)}.
\]

If we plug the prior and likelihood into Bayes rule we can compute the maximum a posteriori (MAP) estimator
\[
f_{\text{MAP}}^* := \arg \max_{f \in \mathcal{H}} \left[ \frac{P(D|f)P(f)}{P(D)} \right]
\]
\[
= \arg \max_{f \in \mathcal{H}} \left[ \prod_{i=1}^n \frac{1}{1 + \exp(y_i f(x_i))} \exp(-\|f\|_H^2) \right]
\]
\[
= \arg \max_{f \in \mathcal{H}} \left[ \sum_{i=1}^n \log \left( \frac{1}{1 + \exp(y_i f(x_i))} \right) - \|f\|_H^2 \right].
\]

With some simple algebra the above MAP estimator can be rewritten in the form of Tikhonov regularization
\[
f_{\text{MAP}}^* = \arg \min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^n \log(1 + \exp(-y_i f(x_i))) + \lambda \|f\|_H^2 \right],
\]
where \(\lambda\) is the regularization parameter. By the representer theorem the above equation has a solution of the form
\[
f^*(x) = \sum_{i=1}^n c_i K(x, x_i).
\]

Given the above representer theorem we can solve for the variables \(c_i\) by the following optimization problem
\[
\min_{c \in \mathbb{R}^n} \left[ n^{-1} \sum_{i=1}^n \log(1 + \exp(-y_i (c^T K)_i)) + \lambda c^T K c \right],
\]
where \((c^T K)_i\) is the \(i\)th element of the vector \(c^T K\). This optimization problem is convex and differentiable so a classical approach for solving for \(c\) is using the Newton-Raphson method.

### 5.3.1. Newton-Raphson

The Newton-Raphson method was initially used to solve for roots of polynomials and the application to optimization problems is fairly straightforward. We first describe the Newton-Raphson method for the case of a scalar, the optimization is in terms of one variable. We then describe the multivariate form and apply this to the optimization problem in logistic regression.

- **Newton’s method for finding roots:** Newton’s method is primarily a method for finding roots of polynomials. It was proposed by Newton around 1669 and Raphson improved on the method in 1690, therefore the Newton-Raphson method. Given a polynomial \(f(x)\) the Taylor series expansion of \(f(x)\) around the point \(x = x_0 + \varepsilon\) is given by

\[
f(x_0 + \varepsilon) = f(x_0) + f'(x_0)\varepsilon + \frac{1}{2}f''(x_0)\varepsilon^2 + \ldots
\]

truncating the expansion after first order terms results in

\[
f(x_0 + \varepsilon) \approx f(x_0) + f'(x_0)\varepsilon.
\]

From the above expression we can estimate the offset \(\varepsilon\) needed to get closer to the root \((x : f(x) = 0)\) starting from the initial guess \(x_0\). This is done by setting \(f(x_0 + \varepsilon) = 0\) and solving for \(\varepsilon\).

\[
0 = f(x_0 + \varepsilon) = f(x_0) + f'(x_0)\varepsilon
\]

\[
0 \approx f(x_0) + f'(x_0)\varepsilon
\]

\[
-f(x_0) \approx f'(x_0)\varepsilon
\]

\[
\varepsilon_0 \approx -\frac{f(x_0)}{f'(x_0)}.
\]

This is the first order or linear adjustment to the root’s position. This can be turned into an iterative procedure by setting \(x_1 = x_0 + \varepsilon_0\), calculating a new \(\varepsilon_1\) and then iterating until convergence:

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.
\]

- **The Newton-Raphson method as an optimization method for scalars:** We are given a convex minimization problem

\[
\min_{x \in (a,b)} g(x),
\]

where \(g(x)\) is a convex function. The extrema of \(g(x)\) will occur at a value of \(x_m\) such that \(g'(x_m) = 0\) and since the function is convex this extrema will be a minima. If \(g(x)\) is a polynomial then \(g'(x)\) is also a polynomial and we can apply Newton’s method for root finding to \(g'(x)\). If \(g(x)\) is not a polynomial then we apply the root finding method to a polynomial approximation of \(g(x)\). We now describe the steps involved.
(1) Taylor expand $g(x)$: A truncated Taylor expansion of $g(x)$ results in a second order polynomial approximation of $g(x)$

$$g(x) \approx g(x_0) + g'(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^2 g''(x_0).$$

(2) Set derivative to zero: Take the derivative of the Taylor expansion and set it equal to zero

$$\frac{dg}{dx} = f(x) = g'(x_0) + g''(x_0)(x - x_0) = 0.$$

This leaves us with a root finding problem, find the root of $f(x)$ for which we use Newton’s method for finding roots.

(3) Update rule: The update rule reduces to

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{g'(x_n)}{g''(x_n)}.$$

A key point in the above procedure is the convexity of $g(x)$. To be sure that the procedure converges the second derivative $g''(x)$ must be positive in the domain of optimization, the interval $[a, b]$. Convexity of $g(x)$ ensures this.

- The Newton-Raphson method as an optimization method for vectors: We are given a convex minimization problem

$$\min_{x \in \mathcal{X}} g(x),$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is convex and $g(x)$ is a convex function. We follow the logic of the scalar case except using vector calculus.

(1) Taylor expand $g(x)$: A truncated Taylor expansion of $g(x)$ results in a second order polynomial approximation of $g(x)$

$$g(x) \approx g(x_0) + (x - x_0)^T \cdot \nabla g(x_0) + \frac{1}{2} (x - x_0)^T \cdot H(x_0) \cdot (x - x_0),$$

where $x$ is a column vector of length $n$, $\nabla g(x_0)$ is the gradient of $g$ evaluated at $x_0$ and is also a column vector of length $n$, $H(x_0)$ is the Hessian matrix evaluated at $x_0$

$$H_{i,j}(x_0) = \frac{\partial^2 g(x_0)}{\partial x^i \partial x^j} \bigg|_{x_0}, \quad i, j = 1, \ldots, n.$$

(2) Set derivative to zero: Take the derivative of the Taylor expansion and set it equal to zero

$$\nabla g(x) = \nabla g(x_0) + \frac{1}{2} H(x_0) \cdot (x - x_0) + \frac{1}{2} (x - x_0)^T \cdot H(x_0) = 0,$$

the Hessian matrix is symmetric and twice differentiable (due to convexity) so we can reduce the above to

$$\nabla g(x) = \nabla g(x_0) + H(x_0) \cdot (x - x_0) = 0.$$

This implies that at a minima $x^*$, the gradient is zero

$$0 = H(x_0) \cdot (x^* - x_0) + \nabla g(x_0).$$
(3) Update rule: Solving the above linear system of equations for $x^*$ leads to the following update rule

$$x_{n+1} = x_n - H^{-1}(x_n) \cdot \nabla g(x_n),$$

where $-H^{-1}(x_n) \cdot \nabla g(x_n)$ is called the Newton direction.

For the above procedure to converge to a minima the Newton direction must be a direction of descent

$$\nabla g^T(x_n) \cdot (x_{n+1} - x_n) < 0.$$

If the Hessian matrix is positive definite then the Newton direction will be a direction of descent, this is the matrix analog of a positive second derivative. Convexity of $g(x)$ in the domain $\mathcal{X}$ ensures that the Hessian is positive definite. If the function $g(x)$ is quadratic the procedure will converge in one iteration.

- The Newton-Raphson method for regularized logistic regression: The optimization problem for regularized logistic regression is

$$f^*_{MAP} = \arg \min_{f \in \mathcal{H}_K} \left[ n^{-1} \sum_{i=1}^{n} \log(1 + \exp(-y_i f(x_i))) + \lambda \|f\|_{\mathcal{H}_K}^2 \right],$$

by the representer theorem

$$f^*(x) = \sum_{i=1}^{n} c_i K(x, x_i) + b,$$

$\|f\|_{\mathcal{H}_K}$ is a seminorm that does not penalize constants, like the SVM case. The optimization problem can be rewritten as

$$\min_{c \in \mathbb{R}^n, b \in \mathbb{R}} \left[ L[c, b] = n^{-1} \sum_{i=1}^{n} \log(1 + \exp(-y_i ((c^T K) + b)) + \lambda c^T K c \right],$$

where $(c^T K)_i$ is the $i$th element of the vector $c^T K$.

### 5.4. Spline models

One can also derive spline models from Tikhonov regularization and often one finds this derivation in the same formulation as that for SVMs or Kernel-ridge regression. This is somewhat misleading in that the two derivations are very similar in spirit there are significant technical differences in the regularization functional used and the mathematical tools.

In general, for spline models the domain of the RKHS is unbounded and therefore a countable basis may not exist. In addition the regularization function is defined as the integral of a differential operator. For example for one-dimensional linear splines the regularization functional is

$$\|f\|_{\mathcal{H}_K}^2 := \int \left| \frac{\partial f(x)}{\partial x} \right|^2 dx,$$

and the kernel function is a piecewise linear function

$$K(x-y) = c_1 |x-y|,$$
where \( c_1 \) is an arbitrary constant and
\[
 f(x) = \sum_{i=1}^{n} c_i |x - x_i| + d_1.
\]

We will show that the regularization functional corresponding to the differential operators can be written as an integral operator with a kernel function and relate the kernel to the differential operator via Fourier analysis.

We assume the kernel is symmetric but defined over an unbounded domain. The eigenvalues of the equation
\[
 \int_{-\infty}^{\infty} K(s, t)\phi(s)ds = \lambda\phi(t)
\]
are not necessarily countable and Mercer’s theorem does not apply. Let us assume that the kernel is translation invariant, or
\[
 K(s, t) = K(s - t).
\]

We will see this implies that we will have to consider Fourier hypotheses spaces and all these spaces will be defined via Fourier transforms.

**Definition.** The Fourier transform of a real valued function \( f \in L_1 \) is the complex valued function \( \hat{f}(\omega) \) defined as
\[
 \hat{f}(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-j\omega x}dx.
\]

**Proposition.** The original function \( f \) can be obtained through the inverse Fourier transform
\[
 f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{j\omega x}d\omega.
\]

**Comment.** Periodic functions can be expanded in a Fourier series
\[
 \hat{f}(\omega) = \sum_{n} \beta_n \delta(\omega - n\omega_0).
\]
This can be shown from the periodicity condition
\[
 f(x + T) = f(x) \quad \text{for } T = \frac{2\pi}{\omega_0}.
\]
Taking the Fourier transform of both side yields
\[
 \hat{f}(\omega) e^{-j\omega T} = \hat{f}(\omega).
\]
This is possible if \( \hat{f}(\omega) \neq 0 \) only when \( \omega = n\omega_0 \). This implies for nontrivial \( f \) that \( \hat{f}(\omega) = \sum_n \beta_n \delta(\omega - n\omega_0) \), which is a Fourier series.

**Proposition.** The eigenfunctions for translation invariant kernels are the Fourier bases and the eigenvalues are the Fourier transform of the kernel.
If \( K(s, t) = K(||s - t||) \) then the solutions of
\[
 \int K(s, t)\phi(s)d\mu(s) = \lambda\phi(t)
\]
have the form
\[
 \phi_{\omega}(s) = e^{-i(s, \omega)},
\]
and
\[ \lambda_\omega = \frac{1}{2\pi} \tilde{K}(\omega), \]
where \( \tilde{K}(\omega) \) is the Fourier transform of \( K(||s-t||) \).

The following theorem relates the Fourier transform of a kernel to it being positive definite.

**Theorem** (Bochner). A function \( K(s-t) \) is positive definite if and only if it is the Fourier transform of a symmetric, positive function \( \tilde{K}(\omega) \) decreasing to 0 at infinity.

We now define the RKHS norm for shift invariant kernels and we will see that this norm will apply to regularizers that are differential operators.

**Definition.** For a positive definite function \( K(s-t) \) we define the RKHS norm using the following inner product:
\[ \langle f(s), g(s) \rangle_{\mathcal{H}_K} := \frac{1}{2\pi} \int \frac{\tilde{f}(\omega) \tilde{g}^*(\omega)}{K(\omega)} d\omega, \]
where \( \tilde{g}^*(\omega) \) is the complex conjugate of the Fourier transform of \( g(s) \).

Give the above definition of the RKHS inner product a direct result is that we can define the RKHS as a subspace of \( L^2(\mathbb{R}^n) \) with \( x \in \mathbb{R}^n \).

**Proposition.** Given a RKHS norm defined by the above scalar product
\[ \|f\|^2_{\mathcal{H}_K} = \frac{1}{2\pi} \int \frac{\left| \tilde{f}(\omega) \right|^2}{K(\omega)} d\omega, \]
where the subspace of \( L^2(\mathbb{R}^n) \) for which the above integral is defined (finite) is the RKHS \( \mathcal{H}_K \).

We now verify the reproducing property
\[ \langle K(s-t), f(s) \rangle_{\mathcal{H}_K} := \frac{1}{2\pi} \int \frac{\tilde{K}(\omega) \tilde{f}^*(\omega) e^{-j\omega t}}{K(\omega)} d\omega = f(t). \]

We now relate the RKHS norm to regularizers that are differential operators sometimes called smoothness functionals.

**Example.** The following differential operators
\[ \Phi_1[f] = \int_{-\infty}^{+\infty} |f'(x)|^2 dx = \frac{1}{2\pi} \int \omega^2 |\tilde{f}(\omega)|^2 d\omega, \]
\[ \Phi_2[f] = \int_{-\infty}^{+\infty} |f''(x)|^2 dx = \frac{1}{2\pi} \int \omega^4 |\tilde{f}(\omega)|^2 d\omega \]
are an RKHS norm of the form
\[ \|f\|^2_{\mathcal{H}_K} = \frac{1}{2\pi} \int \frac{\left| \tilde{f}(\omega) \right|^2}{K(\omega)} d\omega, \]
where
\[ \tilde{K}(\omega) = 1/\omega^2 \text{ for } \Phi_1, \]
\[ \tilde{K}(\omega) = 1/\omega^4 \text{ for } \Phi_2. \]
Notice, that $K(\omega)$ is a positive symmetric function decreasing to zero at infinity. Given the Fourier domain representation of the kernel taking the inverse Fourier transform gives us the reproducing kernel

\[ K(x) = -|x|/2 \iff \tilde{K}(\omega) = 1/\omega^2, \]
\[ K(x) = -|x|^3/12 \iff \tilde{K}(\omega) = 1/\omega^4. \]

For both kernels, the singularity of the Fourier transform for $\omega = 0$ is due to the seminorm property and the fact that the kernel is only conditionally positive definite.

The fact that both functionals

\[ \Phi_1[f] = \int_{-\infty}^{+\infty} |f'(x)|^2 dx = \frac{1}{2\pi} \int \omega^2 |\hat{f}(\omega)|^2 d\omega, \]
\[ \Phi_2[f] = \int_{-\infty}^{+\infty} |f''(x)|^2 dx = \frac{1}{2\pi} \int \omega^4 |\hat{f}(\omega)|^2 d\omega \]

are seminorms is obvious since $f(x) = c$ will result in a zero norm for both functionals.

**Examples.** Other possible kernel functions and their Fourier transforms are

\[ K(x) = e^{-x^2/2\sigma^2} \iff \tilde{K}(\omega) = e^{-\omega^2\sigma^2/2}, \]
\[ K(x) = \frac{1}{2}e^{-\gamma|x|} \iff \tilde{K}(\omega) = \frac{1}{1 + \omega^2}, \]
\[ K(x) = \frac{\sin(\Omega x)}{\pi x} \iff \tilde{K}(\omega) = U(\omega + \Omega) - U(\omega - \Omega). \]

**Example.** The Gaussian kernel $K(x) = e^{-x^2/2\sigma^2}$ corresponds to the following differential operator

\[ \Phi[f] = 1 + \int_{-\infty}^{+\infty} \sum_{n=1}^{\infty} \frac{|f'(x)|^{2n}}{2n!} dx. \]

We now state the representer theorem for shift invariant kernels on unbounded domains. Note we cannot use the previous proof of the representer theorem since these kernels are not Mercer kernels.

**Theorem.** Let $\tilde{K}(\omega)$ be the Fourier transform of a kernel function $K(x)$. The function minimizing the functional

\[ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \frac{\lambda}{2\pi} \int \frac{|\hat{f}(\omega)|^2}{K(\omega)} d\omega \]

has the form

\[ f(x) = \sum_{i=1}^{n} c_i K(x - x_i). \]

**Proof.**

We rewrite the functional in terms of the Fourier transform $\hat{f}$ and obtain

\[ \frac{1}{n} \sum_{i=1}^{n} \left( y_i - \frac{1}{2\pi} \int \hat{f}(\omega) e^{i\omega x_i} d\omega \right)^2 + \lambda \frac{1}{2\pi} \int \frac{\hat{f}(-\omega) \hat{f}(\omega)}{K(\omega)} d\omega. \]
Taking the functional derivative w.r.t $\tilde{f}(\xi)$ gives

$$-\frac{1}{n\pi} \sum_{i=1}^{n} (y_i - f(x_i)) \int \frac{D\tilde{f}(\omega)}{D\tilde{f}(\xi)} e^{j\omega x_i} d\omega + \frac{2}{2\pi} \lambda \int \frac{\tilde{f}(\omega)}{K(\omega)} \frac{D\tilde{f}(\omega)}{D\tilde{f}(\xi)} d\omega$$

$$= -\frac{1}{n\pi} \sum_{i=1}^{n} (y_i - f(x_i)) \int \delta(\omega - \xi) e^{j\omega x_i} d\omega + \frac{1}{\pi} \lambda \int \tilde{f}(\omega) \frac{1}{K(\omega)} \delta(\omega - \xi) d\omega.$$ 

From the definition of $\delta$ we have

$$-\frac{1}{n\pi} \sum_{i=1}^{n} (y_i - f(x_i)) e^{j\xi x_i} + \frac{1}{\pi} \lambda \tilde{f}(\xi).$$

Setting the derivative to zero and changing the sign of $\xi$

$$\tilde{f}_\lambda(\xi) = \tilde{K}(\xi) \sum_{i=1}^{n} \frac{y_i - f(x_i)}{n\lambda} e^{-j\xi x_i}.$$ 

Defining the coefficients

$$c_i = \frac{y_i - f(x_i)}{n\lambda},$$

taking the inverse Fourier transform, we obtain

$$f(x) = \sum_{i=1}^{n} c_i K(x - x_i). \Box$$

We noted that some of the smoothness functionals were seminorms and therefore led to conditionally positive definite functions. We now formalize and explore this issue.

**Definition.** Let $r = \|x\|$ with $x \in \mathbb{R}^n$. A continuous function $K = K(r)$ is conditionally positive definite of order $m$ on $\mathbb{R}^n$, if and only if for any distinct points $t_1, t_2, ..., t_\ell \in \mathbb{R}^n$ and scalars $c_1, c_2, ..., c_\ell$ such that $\sum_{i=1}^{\ell} c_i p(t_i) = 0$ for all $p \in \pi_{m-1}(\mathbb{R}^n)$, the quadratic form is nonnegative

$$\sum_{i=1}^{\ell} \sum_{j=1}^{\ell} c_i c_j K(\|t_i - t_j\|) \geq 0.$$ 

In the case of a strict inequality the function is strictly conditionally positive definite.

**Example.** The following class of functionals are conditionally positive definite of order $m$

$$K(x) = -\frac{|x|^{2m-1}}{2m(2m-1)} \rightarrow \tilde{K}(\omega) = 1/\omega^{2m}.$$ 

**Proposition.** If $K$ is a conditionally positive definite function of order $m$, then

$$\Phi[f] = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{|\tilde{f}(\omega)|^2}{K(\omega)}$$

is a seminorm whose null space is the set of polynomials of degree $m - 1$. If $K$ is strictly positive definite, then $\Phi$ is a norm.
For a positive definite kernel, we have shown that
\[ f(x) = \sum_{i=1}^{n} c_i K(x, x_i), \]
where the coefficients \( c_i \) can be found by solving the linear system
\[ (K + n\lambda I) c = y. \]

For a conditionally positive definite kernel of order \( m \), it can be shown that
\[ f(x) = \sum_{i=1}^{n} c_i K(x, x_i) + \sum_{k=1}^{m-1} d_k \gamma_k(x), \]
where the coefficients \( c \) and \( d = (d_1, \ldots, d_m) \) are found by solving the linear system
\[ (K + n\lambda I)c + \Gamma^\top d = y \]
\[ \Gamma c = 0. \]
with \( \Gamma_{ik} = \gamma_k(x_i) \).

**Examples.** We state a few regularization functionals and their appropriate solutions.

1) **1-D Linear Splines.** The solution is the space of piecewise linear polynomials:
\[ \Phi[f] = \|f\|_K^2 = \int |f'(x)|^2 dx = \frac{1}{2\pi} \int |\tilde{f}(\omega)|^2 d\omega \]
\[ \tilde{K}(\omega) = \frac{1}{\omega^2} \]
\[ K(x-y) \propto |x-y| \]
\[ f(x) = \sum_{i=1}^{n} c_i |x-x_i| + d_1. \]

2) **1-D Cubic Splines.** The solution is the space of piecewise cubic polynomials.
\[ \Phi[f] = \|f\|_K^2 = \int |f''(x)|^2 dx = \frac{1}{2\pi} \int |\tilde{f}(\omega)|^2 d\omega \]
\[ \tilde{K}(\omega) = \frac{1}{\omega^4} \]
\[ K(x-y) \propto |x-y|^3 \]
\[ f(x) = \sum_{i=1}^{n} c_i |x-x_i|^3 + d_2 x + d_1. \]

3) **2-D Thin Plate Splines.**
\[ \Phi[f] = \|f\|_K^2 = \int \int \left[ \left( \frac{\partial^2 f}{\partial x_1^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left( \frac{\partial^2 f}{\partial x_2^2} \right)^2 \right] dx_1 dx_2 \]
\[ = \frac{1}{(2\pi)^2} \int d\omega \|\omega^4 \tilde{f}(\omega)\|^2 \]
47
\[ \tilde{K}(\omega) = \frac{1}{||\omega||^4} \]
\[ K(x) \propto ||x||^2 \ln ||x|| \]
\[ f(x) = \sum_{i=1}^{n} c_i ||x - x_i||^2 \ln ||x - x_i|| + \langle d_2, x \rangle + d_1. \]

4) Gaussian Radial Basis Functions.
\[ \Phi[f] = ||f||^2_K \]
\[ = \frac{1}{2\pi} \int e^{-||\omega||^2/2} \tilde{f}(\omega)^2 d\omega \]
\[ \tilde{K}(\omega) = e^{-||\omega||^2/2} \]
\[ K(x) \propto e^{-||x||^2/2\sigma^2} \]
\[ f(x) = \sum_{i=1}^{n} c_i e^{-||x - x_i||^2/2\sigma^2} \]

5.5. Convex Optimization

Concepts from convex optimization such as Karush-Kuhn-Tucker (KKT) conditions were used in the previous sections of this lecture. In this section we give a brief introduction and derivation of these conditions.

Definition. A set \( \mathcal{X} \in \mathbb{R}^n \) is convex if
\[ \forall x_1, x_2 \in \mathcal{X}, \forall \lambda \in [0, 1], \lambda x_1 + (1 - \lambda)x_2 \in \mathcal{X}. \]

A set is convex if, given any two points in the set, the line segment connecting them lies entirely inside the set.

Definition. A function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is convex if:
1. For any \( x_1 \) and \( x_2 \) in the domain of \( f \), for any \( \lambda \in [0, 1] \),
\[ f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2). \]
2. The line segment connecting two points \( f(x_1) \) and \( f(x_2) \) lies entirely on or above the function \( f \).
3. The set of points lying on or above the function \( f \) is convex.

A function is strictly convex if we replace “on or above” with “above”, or replace “\( \leq \)” with “\(<\)”.

Definition. A point \( x^* \) is called a local minimum of \( f \) if there exists \( \varepsilon > 0 \) such that \( f(x^*) < f(x) \) for all \( ||x - x^*|| \leq \varepsilon. \)

Definition. A point \( x^* \) is called a global minimum of \( f \) if \( f(x^*) < f(x) \) for all feasible \( x \).

Unconstrained convex functions (convex functions where the domain is all of \( \mathbb{R}^n \)) are easy to minimize. Convex functions are differentiable almost everywhere. Directional derivatives always exist. If we cannot improve our solution by moving locally, we are at the optimum. If we cannot find a direction that improves our solution, we are at the optimum.

Convex functions over convex sets (a convex domain) are also easy to minimize. If the set and the functions are both convex, if we cannot find a direction which we
are able to move in which decreases the function, we are done. Local optima are global optima.

**Example.** *Linear programming is always a convex problem*

\[
\begin{align*}
\min_c & \quad \langle c, x \rangle \\
\text{subject to} & \quad Ax = b \\
& \quad Cx \leq d.
\end{align*}
\]

**Example.** *Quadratic programming is a convex problem iff the matrix Q is positive semidefinite*

\[
\begin{align*}
\min & \quad x'Qx + \langle c, x \rangle \\
\text{subject to} & \quad Ax = b \\
& \quad Cx \leq d.
\end{align*}
\]

**Definition.** *The following constrained optimization problem P will be called the primal problem*

\[
\begin{align*}
\min & \quad f(x) \\
\text{subject to} & \quad g_i(x) \geq 0 \quad i = 1, \ldots, m \\
& \quad h_i(x) = 0 \quad i = 1, \ldots, n \\
& \quad x \in X.
\end{align*}
\]

*Here, f is our objective function, the g_i are inequality constraints, the h_i are equality constraints, and X is some set.*
Figure 5. The top two figures are convex functions. The first function is strictly convex. Bottom figures are nonconvex functions.

Figure 6. Optimizing a convex function of convex and nonconvex sets. In the example on the left the set is convex and the function is convex so a local minima corresponds to a global minima. In the example on the right the set is nonconvex and the function is convex one can find local minima that are not global minima.

Definition. We define a Lagrangian dual problem \( D \):

\[
\max \Theta(u, v) \\
\text{subject to:} \quad u \geq 0
\]
where \( \Theta(u, v) := \inf \left\{ f(x) - \sum_{i=1}^{m} u_i g_i(x) - \sum_{j=1}^{n} v_i h_i(x) : x \in \mathcal{X} \right\} \).

**Theorem (Weak Duality).** Suppose \( x \) is a feasible solution of \( P \). Then \( x \in \mathcal{X}, g_i(x) \leq 0 \forall i, h_i(x) = 0 \forall i \). Suppose \( u, v \) are a feasible solution of \( D \). Then for all \( u \geq 0 \)
\[
f(x) \geq \Theta(u, v).
\]

**Proof.**
\[
\Theta(u, v) = \inf \left\{ f(y) - \sum_{i=1}^{m} u_i g_i(y) - \sum_{j=1}^{n} v_i h_i(y) : y \in \mathcal{X} \right\}
\leq f(x) - \sum_{i=1}^{m} u_i g_i(x) - \sum_{i=1}^{n} v_i h_i(x)
\leq f(x).
\]

Weak duality says that every feasible solution to \( P \) is at least as expensive as every feasible solution to \( D \). It is a very general property of duality, and we did not rely on any convexity assumptions to show it.

**Definition.** Strong duality holds when the optima of the primal and dual problems are equivalent \( \text{Opt}(P) = \text{Opt}(D) \).

If strong duality does not hold, we have the possibility of a duality gap. Strong duality is very useful, because it usually means that we may solve whichever of the dual or primal is more convenient computationally, and we can usually obtain the solution of one from the solution of the other.

**Proposition.** If the objective function \( f \) is convex, and the feasible region is convex, under mild technical we have strong duality.

We now look at what are called saddle points of the Lagrangian function. We defined the Lagrangian function as the dual problem
\[
L(x, u, v) = f(x) - \sum_{i=1}^{m} u_i g_i(x) - \sum_{j=1}^{n} v_j h_j(x).
\]
A set \((x^*, u^*, v^*)\) of feasible solutions to \( P \) and \( D \) is called a saddle point of the Lagrangian if
\[
L(x^*, u, v) \leq L(x, u, v) \leq L(x, u^*, v^*) \forall x \in \mathcal{X}, \forall u, v \geq 0
\]
x* minimizes \( L \) if \( u \) and \( v \) are fixed at \( u^* \) and \( v^* \), and \( u^* \) and \( v^* \) maximize \( L \) if \( x \) is fixed at \( x^* \).

**Definition.** The points \((x^*, u^*, v^*)\) satisfy the Karush Kuhn Tucker (KKT) conditions or are KKT points if they are feasible to \( P \) and \( D \) and
\[
\nabla f(x^*) - \nabla g(x^*)'u^* - \nabla h(x^*)'v = 0
\]
\[
u^* g(x^*) = 0.
\]
In a convex, differentiable problem, with some minor technical conditions, points that satisfy the KKT conditions are equivalent to saddle points of the Lagrangian.
LECTURE 6
Voting algorithms

Voting algorithms or algorithms where the final classification or regression function is a weighted combination of “simpler” or “weaker” classifiers have been used extensively in a variety of applications.

We will study two examples of voting algorithms in greater depth: Bootstrap AGGREGATING (BAGGING) and boosting.

6.1. Boosting

Boosting algorithms especially AdaBoost (adaptive boosting) have had a significant impact on a variety of practical algorithms and also have been the focus of theoretical investigation for a variety of fields. The formal term boosting and the first boosting algorithm came out of the field of computational complexity in theoretical Computer Science. In particular, learning as formulated by boosting came from the concept of Probably Approximatley Correct (PAC) learning.

6.2. PAC learning

The idea of Probably Approximatley Correct (PAC) learning was formulated in 1984 by Leslie Valiant as an attempt to characterize what is learnable. Let \( \mathcal{X} \) be a set. This set contains encodings of all objects of interest in the learning problem. The goal of the learning algorithm is to infer some unknown subset of \( \mathcal{X} \), called a concept, from a known class of concepts, \( \mathcal{C} \). Unlike the previous statistical formulations of the learning problem, the issue of representation arises in this formulation due to computational issues.

- Concept classes A representation class over \( \mathcal{X} \) is a pair \((\sigma,\mathcal{C})\), where \( \mathcal{C} \subseteq \{0,1\}^* \) and \( \sigma : \mathcal{C} \to 2^\mathcal{X} \). For \( c \in \mathcal{C} \), \( \sigma(c) \) is a concept over \( \mathcal{X} \) and the image space \( \sigma(\mathcal{C}) \) is the concept class represented by \((\sigma,\mathcal{C})\). For \( c \in \mathcal{C} \) the positive examples are \( \text{pos}(c) = \sigma(c) \) and the negative examples are \( \text{neg}(c) = \mathcal{X} - \sigma(c) \). The notations \( c(x) = 1 \) is equivalent to \( x \in \text{pos}(c) \) and \( c(x) = 0 \) is equivalent to \( x \in \text{neg}(c) \). We assume that domain points \( x \in \mathcal{X} \) and representations \( c \in \mathcal{C} \) are efficiently encoded with by codings of length \(|x|\) and \(|c|\) respectively.
Parameterized representation We will study representation classes parameterized by an index $n$ resulting in the domain $\mathcal{X} = \cup_{n \geq 1} \mathcal{X}_n$ and representation class $\mathcal{C} = \cup_{n \geq 1} \mathcal{C}_n$. The index $n$ serves as a measure of the complexity of concepts in $\mathcal{C}$. For example, $\mathcal{X}$ may be the set $\{0, 1\}^n$ and $\mathcal{C}$ the set of all Boolean formulae over $n$ variables.

Efficient evaluation of representations If $\mathcal{C}$ is a representation class over $\mathcal{X}$, then $\mathcal{C}$ is polynomially evaluable if there is a (probabilistic) polynomial-time evaluation algorithm $A$ that given a representation $c \in \mathcal{C}$ and domain point $x \in \mathcal{X}$ outputs $c(x)$.

Samples A labeled example from a domain $\mathcal{X}$ is a pair $< x, b >$ where $x \in \mathcal{X}$ and $b \in \{0, 1\}$. A sample $S = (< x_1, b_1 >, ..., < x_m, b_m >)$ is a finite sequence of labeled examples. A labeled example of $c \in \mathcal{C}$ has the form $< x, c(x) >$. A representation $h$ and an example $< x, b >$ agree if $h(x) = b$. A representation $h$ and a sample $S$ are consistent if $h$ agrees with each example in $S$.

Distributions on examples A learning algorithm for a representation class $\mathcal{C}$ will receive examples from a single representation $c \in \mathcal{C}$ which we call the target representation. Examples of the target representation are generated probabilistically: $D_c$ is a fixed but arbitrary distribution over $\text{pos}(c)$ and $\text{neg}(c)$. This is the target distribution. The learning algorithm will be given access to an oracle $EX$ which returns in unit time an example of the target representation drawn according to the target distribution $D_c$.

Measure of error Given a target representation $c \in \mathcal{C}$ and a target distribution $D$ the error of a representation $h \in \mathcal{H}$ is

\[ e_c(h) = D(h(x) \neq c(x)) . \]

In the above formulation $\mathcal{C}$ is the target class. In the above formulation $\mathcal{H}$ is the hypothesis class. The algorithm $A$ is a learning algorithm for $\mathcal{C}$ and the output $h_A \in \mathcal{H}$ is the hypothesis of $A$.

We can now define learnability:

**Definition** (Strong learning). Let $\mathcal{C}$ and $\mathcal{H}$ be representation classes over $\mathcal{X}$ that are polynomially evaluable. Then $\mathcal{C}$ is polynomially learnable by $\mathcal{H}$ if there is a (probabilistic) algorithm $A$ with access to $EX$, taking inputs $\varepsilon, \delta$ with the property that for any target representation $c \in \mathcal{C}$, for any target distribution $D$, and for any input values $0 < \varepsilon, \delta < 1$, algorithm $A$ halts in time polynomial in $\frac{1}{\varepsilon}, \frac{1}{\delta}, |c|, n$ and outputs a representation $h_A \in \mathcal{H}$ that with probability greater than $1 - \delta$ satisfies $e_c(h) < \varepsilon$.

The parameter $\varepsilon$ is the accuracy parameter and the parameter $\delta$ is the confidence parameter. These two parameters characterize the name Probably ($\delta$) Approximately ($\varepsilon$) Correct ($e_c(h)$). The above definition is sometimes called distribution free learning since the property holds over an target representation and target distribution.

Considerable research in PAC learning has focused on which representation classes $\mathcal{C}$ are polynomially learnable.

So far we have defined learning as approximating arbitrarily close the target concept. Another model of learning called weak learning considers the case where the learning algorithm is required to perform slightly better than chance.
Definition (Weak learning). Let \( C \) and \( H \) be representation classes over \( X \) that are polynomially evaluable. Then \( C \) is polynomially weak learnable by \( H \) if there is a (probabilistic) algorithm \( A \) with access to \( EX \), taking inputs \( \varepsilon, \delta \) with the property that for any target representation \( c \in C \), for any target distribution \( D \), and for any input values \( 0 < \varepsilon, \delta < 1 \), algorithm \( A \) halts in time polynomial in \( \frac{1}{\varepsilon}, \frac{1}{\delta}, |c|, n \) and outputs a representation \( h_A \in H \) that with probability greater than \( 1 - \delta \) satisfies \( e_c(h) < \frac{1}{2} - \frac{1}{p(|c|)} \), where \( p \) is a polynomial.

Definition (Sample complexity). Given a learning algorithm \( A \) for a representation class \( C \). The number of calls \( s_A(\varepsilon, \delta) \) to the oracle \( EX \) made by \( A \) on inputs \( \varepsilon, \delta \) for the worst-case measure over all target representations \( c \in C \) and target distributions \( D \) is the sample complexity of the algorithm \( A \).

We now state some Boolean classes whose learnability we will state as positive or negative examples of learnability.

- The class \( M_n \) consists of monomials over the Boolean variables \( x_1, \ldots, x_n \).
- For a constant \( k \), the class \( kCNF_n \) (conjunctive normal forms) consists of Boolean formulae of the form \( C_1 \land \cdots \land C_l \) where each \( C_i \) is a disjunction of at most \( k \) monomials over the Boolean variables \( x_1, \ldots, x_n \).
- For a constant \( k \), the class \( kDNF_n \) (disjunctive normal forms) consists of Boolean formulae of the form \( T_1 \lor \cdots \lor T_l \) where each \( T_i \) is a conjunction of at most \( k \) monomials over the Boolean variables \( x_1, \ldots, x_n \).
- Boolean threshold functions \( I(\sum_{i=1}^n w_ix_i > t) \) where \( w_i \in \{0, 1\} \) and \( I \) is the indicator function.

Definition (Empirical risk minimization, ERM). A consistent algorithm \( A \) is one that outputs hypotheses \( h \) that are consistent with the sample \( S \) and the range over possible hypotheses for \( A \) is \( h \in C \).

The above algorithm is ERM in the case of zero error with the target concept in the hypothesis space.

Theorem. If the hypothesis class is finite then \( C \) is learnable by the consistent algorithm \( A \).

Theorem. Boolean threshold functions are not learnable.

Theorem. \( \{ f \lor g : f \in kCNF, g \in kDNF \} \) is learnable.

Theorem. \( \{ f \land g : f \in kDNF, g \in kCNF \} \) is learnable.

Theorem. Let \( C \) be a concept class with finite VC dimension \( VC(C) = d < \infty \). Then \( C \) is learnable by the consistent algorithm \( A \).

6.3. The hypothesis boosting problem

An important question both theoretically and practically in the late 1980’s was whether strong learnability and weak learnability were equivalent. This was the hypothesis boosting problem:

Conjecture. A concept class \( C \) is weakly learnable if and only if it is strongly learnable.

The above conjecture was proven true in 1990 by Robert Schapire.
Theorem. A concept class $C$ is weakly learnable if and only if it is strongly learnable.

The proof of the above theorem was based upon a particular algorithm. The following algorithm takes as input a weaklearner, an error parameter $\varepsilon$, a confidence parameter $\delta$, an oracle $EX$, and outputs a strong learner. At each iteration of the algorithm a weaklearner with error rate $\varepsilon$ gets boosted so that its error rate decreases to $3\varepsilon^2 - 2\varepsilon^3$.

**Algorithm 1**: Learn($\varepsilon, \delta, EX$)

- **input**: error parameter $\varepsilon$, confidence parameter $\delta$, examples oracle $EX$
- **return**: $h$ that is $\varepsilon$ close to the target concept $c$ with probability $\geq 1 - \delta$

  if $\varepsilon \geq 1/2 - 1/p(n, s)$ then return WeakLearn($\delta, EX$);

  $\alpha \leftarrow g^{-1}(\varepsilon)$ : where $g(x) = 3x^2 - 2x^3$;

  $EX_1 \leftarrow EX$;

  $h_1 \leftarrow$ Learn($\alpha, \delta/5, EX_1$);

  $\tau_1 \leftarrow \varepsilon/3$;

  let $\hat{a}_1$ be an estimate of $a_1 = \Pr_{x \sim D}[h_1(x) \neq c(x)]$;

  choose a sample sufficiently large that

  $|a_1 - \hat{a}_1| \leq \tau_1$ with probability $\geq 1 - \delta/5$;

  if $\hat{a}_1 \leq \varepsilon - \tau_1$ then return $h_1$;

  defun $EX_2()$

  {flip coin

    if heads then return first $x : h_1(x) = c(x)$;

    else tails return first $x : h_1(x) \neq c(x)$}

  $h_2 \leftarrow$ Learn($\alpha, \delta/5, EX_2$);

  $\tau_2 \leftarrow (1 - 2\alpha)\varepsilon/9$;

  let $\hat{e}$ be an estimate of $e = \Pr_{x \sim D}[h_2(x) \neq c(x)]$;

  choose a sample sufficiently large that

  $|e - \hat{e}| \leq \tau_2$ with probability $\geq 1 - \delta/5$;

  if $\hat{e} \leq \varepsilon - \tau_2$ then return $h_2$;

  defun $EX_3()$

  {return first $x : h_1(x) \neq h_2(x)$};

  $h_3 \leftarrow$ Learn($\alpha, \delta/5, EX_3$);

  defun $h(x)$

  { $b_1 \leftarrow h_1(x)$, $b_2 \leftarrow h_2(x)$

    if $b_1 = b_2$ then return $b_1$

    else return $h_3(x)$}

  return $h$

The above algorithm can be summarized as follows:

(1) Learn an initial classifier $h_1$ on the first $N$ training points
(2) Learn \( h_2 \) on a new sample of \( N \) points, half of which are misclassified by \( h_1 \).

(3) Learn \( h_3 \) on \( N \) points for which \( h_1 \) and \( h_2 \) disagree.

(4) The boosted classifier \( h = \text{Majority vote}(h_1, h_2, h_3) \).

The basic result is that if the individual classifiers \( h_1, h_2, \) and \( h_3 \) have error \( \varepsilon \) the boosted classifier has error \( 2\varepsilon^2 - 3\varepsilon^3 \).

To prove the theorem one needs to show that the algorithm is correct in the following sense.

**Theorem.** For \( 0 < \varepsilon < 1/2 \) and for \( 0 < \delta < 1 \), the hypothesis returned by calling \( \text{Learn}(\varepsilon, \delta, EX) \) is \( \varepsilon \) close to the target concept with probability at least \( 1 - \delta \).

We first define a few quantities

\[
\begin{align*}
p_i &= \Pr_{x \sim D}[h_i(x) = c(x)] \\
q &= \Pr_{x \sim D}[h_1(x) \neq h_2(x)] \\
w &= \Pr_{x \sim D}[h_2(x) \neq h_1(x) = c(x)] \\
v &= \Pr_{x \sim D}[h_1(x) = h_2(x) = c(x)] \\
y &= \Pr_{x \sim D}[h_1(x) \neq h_2(x) = c(x)] \\
z &= \Pr_{x \sim D}[h_1(x) \neq h_2(x) \neq c(x)].
\end{align*}
\]

Given the above quantities

\[
\begin{align}
w + v &= \Pr_{x \sim D}[h_1(x) = c(x)] = 1 - a_1 \quad (6.1) \\
y + z &= \Pr_{x \sim D}[h_1(x) \neq c(x)] = a_1. \quad (6.2)
\end{align}
\]

We can explicitly express the chance that \( EX_i \) returns an instance \( x \) in terms of the above variable:

\[
\begin{align}
D_1(x) &= D(x) \\
D_2(x) &= \frac{D(x)}{2} \left( \frac{p_1(x)}{a_1} + \frac{1 - p_1(x)}{1 - a_1} \right) \\
D_3(x) &= \frac{D(x)q(x)}{w + y}.
\end{align}
\]

From equation (6.3) we have

\[
1 - a_2 = \sum_{x \in \mathcal{X}_n} D_2(x)(1 - p_2(x))
\]

\[
= \frac{1}{2a_1} \sum_{x \in \mathcal{X}_n} D(x)p_1(x)(1 - p_2(x)) + \frac{1}{2(1 - a_1)} \sum_{x \in \mathcal{X}_n} D(x)(1 - p_1(x))(1 - p_2(x))
\]

\[
= \frac{y}{2a_1} + \frac{z}{2(1 - a_1)}.
\]

Combining the above equation with equations (6.1) and (6.2) we can solve for \( w \) and \( z \) in terms of \( y, a_1, a_2 \).

\[
w = (2a_2 - 1)(1 - a_1) + \frac{y(1 - a_1)}{a_1}
\]

\[
z = a_1 - y.
\]
We now control the quantity
\[
\Pr_{x \sim D}[h(x) \neq c(x)] = \Pr_{x \sim D}[(h_1(x) = h_2(x)) \lor (h_1(x) \neq h_2(x) \land h_3(x) \neq c(x))]
\]
\[
= z + \sum_{x \in X_n} D(x)q(x)p_3(x)
\]
\[
= z + \sum_{x \in X_n} (w + y)D_3(x)p_3(x)
\]
\[
= z + a_3(w + y)
\]
\[
\leq z + \alpha(w + y)
\]
\[
= \alpha(2a_2 - 1)(1 - a_1) + a_1 + \frac{y(\alpha - a_1)}{a_1}
\]
\[
\leq \alpha(2a_2 - 1)(1 - a_1) + \alpha
\]
\[
\leq \alpha(2\alpha - 1)(1 - \alpha) + \alpha = 3\alpha^2 - 2\alpha^3 = \varepsilon,
\]
the inequalities follow from the fact that \( a_i \leq \alpha < 1/2 \) and \( y \leq a_1 \). \( \square \)

One also needs to show that the algorithm runs in polynomial time. The following lemma implies this. The proof of the lemma is beyond the scope of the lecture notes.

**Lemma.** On a good run the expected execution time of Learn(\( \varepsilon, \delta/2, EX \)) is polynomial in \( m, 1/\delta, 1/\varepsilon \).

![Figure 1. A plot of the boosted error rate as a function of the initial error for different numbers of boosting rounds.](image)

### 6.4. ADaptive BOOSTing (AdaBoost)

We will call the above formulation of boosting the boost-by-majority algorithm. The formulation of boosting by majority by Schapire involved boosting by filtering
since one weak learner served as a filter for the other. Another formulation of boost by majority was developed by Yoav Freund also based upon filtering. Both of these algorithms were later adjusted so that sampling weights could be instead of filtering. However, all of these algorithms had the problem that the strength $1/2 - \gamma$ of the weak learner had to be known a priori.

Freund and Schapire developed the following adaptive boosting algorithm, AdaBoost, to address these issues.

**Algorithm 2: AdaBoost**

```plaintext
input: samples $S = \{(x_i, y_i)\}_{i=1}^N$, weak learner, number of iterations $T$

return: $h(x) = \text{sign} \left[ \sum_{i=1}^T \alpha_i h_i(x) \right]$

for $i=1$ to $N$ do $w_i^0 = 1/N$;

for $t=1$ to $T$ do
  $h_t \leftarrow \text{Call WeakLearn with weights } w^t$;
  $\varepsilon_t = \sum_{j=1}^N w_j^t I\{y_j \neq h_t(x_j)\}$;
  $\alpha_t = \log((1 - \varepsilon_t)/\varepsilon_t)$;
  for $j=1$ to $N$ do $w_j^{t+1} = w_j^t \exp(-\alpha_t y_j h_t(x_j))$;
  $Z_t = \sum_{j=1}^N w_j^{t+1}$;
  for $j=1$ to $N$ do $w_j^{t+1} = w_j^{t+1}/Z_t$;
```

For the above algorithm we can prove that the training error will decrease over boosting iterations. The advantage of the above algorithm is we don’t need a uniform $\gamma$ over all rounds. All we need is for each boosting round there exists a $\gamma_t > 0$.

**Theorem.** Suppose WeakLearn when called by AdaBoost generates hypotheses with errors $\varepsilon_1, \ldots, \varepsilon_T$. Assume each $\varepsilon_i \leq 1/2$ and let $\gamma_i = 1/2 - \varepsilon_i$ then the following upper bound holds on the hypothesis $h$

$$\frac{\left| \left\{ j : h(x_j) \neq y_j \right\} \right|}{N} \leq \prod_{i=1}^T \sqrt{1 - 4\gamma_i^2} \leq \exp \left( -2 \sum_{i=1}^T \gamma_i^2 \right).$$

**Proof.**

If $y_i \neq h(x_i)$ then $y_i h(x_i) \leq 0$ and $e^{-y_i h(x_i)} \geq 1$. Therefore

$$\frac{\left| \left\{ j : h(x_j) \neq y_j \right\} \right|}{N} \leq \frac{1}{N} \sum_{i=1}^N e^{-y_i h(x_i)},$$

$$= \sum_{i=1}^N w_i^{T+1} \prod_{t=1}^T Z_t = \prod_{t=1}^T Z_t.$$

In addition, since $\alpha_t = \log((1 - \varepsilon_t)/\varepsilon_t)$ and $1 + x \leq e^x$

$$Z_t = 2\sqrt{\varepsilon_t(1 - \varepsilon_t)} = \sqrt{1 - 4\gamma_t^2} \leq e^{-2\gamma_t}. \quad \square$$
6.5. A statistical interpretation of Adaboost

In this section we will reinterpret boosting as a greedy algorithm to fit an additive model. We first define our weak learners as a parameterized class of functions \( h_\theta(x) = h(x; \theta) \) where \( \theta \in \Theta \). If we think of each weak learner as a basis function then the boosted hypothesis \( h(x) \) can be thought of as a linear combination of the weak learners

\[
h(x) = \sum_{i=1}^{T} \alpha_i h_{\theta_i}(x),
\]

where the \( h_{\theta_i}(x) \) is the \( i \)th weak learner parameterized by \( \theta_i \). One approach to setting the parameters \( \theta_i \) and weights \( \alpha_i \) is called forward stagewise modelling. In this approach we sequentially add new basis functions or weak learners without adjusting the parameters and coefficients of the current solution. The following algorithm implements forward stagewise additive modeling.

**Algorithm 3**: Forward stagewise additive modeling

```
input : samples \( S = (x_i, y_i)_{i=1}^{N} \), weak learner, number of iterations \( T \), loss function \( L \)
return: \( h(x) = \left[ \sum_{i=1}^{T} \alpha_i h_{\theta_i}(x) \right] \)

\( h_0(x) = 0; \)
for \( i = 1 \) to \( T \) do
  \( (\alpha_i, \theta_i) = \arg \min_{\alpha \in \mathbb{R}^+, \theta \in \Theta} \sum_{i=1}^{N} L(y_i, h_{i-1}(x_i) + \alpha h_{\theta}(x)); \)
  \( h_i(x) = h_{i-1}(x) + \alpha_i h_{\theta_i}(x); \)
```

We will now show that the above algorithm with exponential loss

\[
L(y, f(x)) = e^{-yf(x)}
\]

is equivalent to AdaBoost.

At each iteration the following minimization is performed

\[
(\alpha_t, \theta_t) = \arg \min_{\alpha \in \mathbb{R}^+, \theta \in \Theta} \sum_{i=1}^{N} \exp[-y_i(h_{i-1}(x_i) + \alpha h_{\theta}(x))],
\]

\[
(\alpha_t, \theta_t) = \arg \min_{\alpha \in \mathbb{R}^+, \theta \in \Theta} \sum_{i=1}^{N} \exp[-y_i h_{i-1}(x_i)] \exp[-y_i \alpha h_{\theta}(x)],
\]

(6.4) \( (\alpha_t, \theta_t) = \arg \min_{\alpha \in \mathbb{R}^+, \theta \in \Theta} \sum_{i=1}^{N} w_i^t \exp[-y_i \alpha h_{\theta}(x)], \)
where \(w_t^i = \exp[-y_i h_{t-1}(x_i)]\) does not effect the optimization functional. For any \(\alpha > 0\) the objective function in equation (6.4) can be rewritten as

\[
\theta_t = \arg\min_{\theta \in \Theta} \left[ e^{-\alpha} \sum_{y_i = h_\theta(x_i)} w_t^i + e^\alpha \sum_{y_i \neq h_\theta(x_i)} w_t^i \right],
\]

\[
\theta_t = \arg\min_{\theta \in \Theta} \left[ (e^{-\alpha} - e^\alpha) \sum_{i=1}^{N} w_t^i I_{\{y_i \neq h_\theta(x_i)\}} + e^\alpha \sum_{i=1}^{N} w_t^i \right],
\]

\[
\theta_t = \arg\min_{\theta \in \Theta} \sum_{i=1}^{N} w_t^i I_{\{y_i \neq h_\theta(x_i)\}}.
\]

Therefore the weak learner that minimizes equation (6.4) will minimize the weighted error rate which if we plug back into equation (6.4) we can solve for \(\alpha_t\) which is

\[
\alpha_t = \frac{1}{2} \log \frac{1 - \varepsilon_t}{\varepsilon_t},
\]

where

\[
\varepsilon_t = \sum_{i=1}^{N} w_t^i I_{\{y_i \neq h_t(x_i)\}}.
\]

The last thing to show is the updating of the linear model

\[
h_t(x) = h_{t-1}(x) + \alpha_t h_t(x),
\]

is equivalent to the reweighting used in AdaBoost. Due to the exponential loss function and the additive updating at each iteration the above sum can be rewritten as

\[
w_{t+1}^i = w_t^i e^{-\alpha y_i h_t(x_i)}.
\]

So AdaBoost can be interpreted as an algorithm that minimizes the exponential loss criterion via forward stagewise additive modeling.

We now give some motivation for why the exponential loss is a reasonable loss function in the classification problem. The first argument is that like the hinge loss for SVM classification the exponential loss serves as an upper bound on the misclassification loss (see figure 2).

Another simple motivation for using the exponential loss is the minimizer of the expected loss with respect to some function class \(H\)

\[
f^*(x) = \arg\min_{f \in H} \mathbb{E}_{Y|x} \left[ e^{-Yf(x)} \right] = \frac{1}{2} \log \frac{Pr(Y = 1|x)}{Pr(Y = -1|x)},
\]

estimates one-half the log-odds ratio

\[
Pr(Y = 1|x) = \frac{1}{1 + e^{-2f^*(x)}}.
\]

### 6.6. A margin interpretation of Adaboost

We developed a geometric formulation of support vector machines in the separable case via maximizing the margin. We will formulate AdaBoost as a margin maximization problem.
Recall that for the linear separable SVM with points in $\mathbb{R}^d$ given a dataset $S$ the following optimization problem characterizes the maximal margin classifier

$$\hat{w} = \arg \max_{w \in \mathbb{R}^d} \min_{x_i \in S} \frac{y_i \langle w, x_i \rangle}{||w||_{L_2}}.$$ 

In the case of AdaBoost we can construct a coordinate space with as many dimensions as weak classifiers, $T$, $u \in \mathbb{R}^T$, where the elements of $\{u_1, ..., u_T\}$ correspond to the outputs of the weak classifiers $\{u_1 = f_1(x), ..., u_T = f_T(x)\}$. We can show that AdaBoost is an iterative way to solve the following mini-max problem

$$\hat{w} = \arg \max_{w \in \mathbb{R}^T} \min_{u_i \in S} \frac{y_i \langle w, u_i \rangle}{||w||_{L_1}},$$

where $u_i = \{f_1(x_i), ..., f_T(x_i)\}$ and the final classifier has the form

$$h_T(x) = \sum_{i=1}^{T} \hat{w}_i^T f_i(x).$$

This follows immediately from the forward additive stagewise modeling interpretation since under separability the addition at each iteration of a weak classifier to the linear expansion will result in a boosted hypothesis $h_t$ that as a function of $t$ will be nondecreasing in $y_i h_t(x_i)$ for with the $L_1$ norm on $w^t$ constrained, $||w||_{L_1} = 1$, following from the fact that the weights at each iteration must satisfy the distribution requirement.

An interesting geometry arises from the two different norms on the weights $w$ in the two different optimization problems. The main idea is that we want to relate the norm on $w$ to properties of norms on points in either $\mathbb{R}^d$ in the SVM case or $\mathbb{R}^T$ in the boosting case. By Hölder’s inequality for the dual norms $||x||_{L_q}$ and
$||w||_{L_p}$ with $\frac{1}{p} + \frac{1}{q} = 1$ and $p, q \in [1, \infty]$ the following holds

$$|\langle x, w \rangle| \leq ||x||_{L_q} ||w||_{L_p}.$$  

The above inequality implies that minimizing the $L_2$ norm on $w$ is equivalent to maximizing the $L_2$ distance between the hyperplane and the data. Similarly, minimizing the $L_1$ norm on $w$ is equivalent to maximizing the $L_\infty$ norm between the hyperplane and the data.
LECTURE 7
One dimensional concentration inequalities

7.1. Law of Large Numbers

In this lecture, we will look at concentration inequalities or law of large numbers for a fixed function. Let \((\Omega, \mathcal{L}, \mu)\) be a probability space. Let \(x_1, \ldots, x_n\) be real random variables on \(\Omega\). A sequence of random variables \(y_n\) converges almost surely to a random variable \(Y\) iff \(P(\lim_{n \to \infty} y_n = Y) = 1\). A sequence of random variables \(y_n\) converges in probability to a random variable \(Y\) iff for every \(\epsilon > 0\), \(\lim_{n \to \infty} P(|y_n - Y| > \epsilon) = 0\). Let \(\hat{\mu}_n := \frac{1}{n} \sum_{i=1}^{n} x_n\). The sequence \(x_1, \ldots, x_n\) satisfies the strong law of large numbers if for some constant \(c\), \(\hat{\mu}_n\) converges to \(c\) almost surely. The sequence \(x_1, \ldots, x_n\) satisfies the weak law of large numbers iff for some constant \(c\), \(\hat{\mu}_n\) converges to \(c\) in probability. In general the constant \(c\) will be the expectation of the random variable \(\mathbb{E}x\).

A given function \(f(x)\) of random variables \(x\) concentrates if the deviation between its empirical average, \(\frac{1}{n} \sum_{i=1}^{n} f(x_i)\) and expectation, \(\mathbb{E}f(x)\), goes to zero as \(n\) goes to infinity. That is \(f(x)\) satisfies the law of large numbers.

7.2. Polynomial inequalities

**Theorem** (Jensen). If \(\phi\) is a convex function then \(\phi(\mathbb{E}x) \leq \mathbb{E}\phi(x)\).

**Theorem** (Bienaymé-Chebyshev). For any random variable \(x\), \(\epsilon > 0\)
\[
\mathbb{P}(|x| \geq \epsilon) \leq \frac{\mathbb{E}x^2}{\epsilon^2}.
\]

**Proof.**
\[
\mathbb{E}x^2 \geq E(x^2 I_{|x| \geq \epsilon}) \geq \epsilon^2 \mathbb{P}(|x| > \epsilon). \quad \square
\]

**Theorem** (Markov). For any random variable \(x\), \(\epsilon > 0\)
\[
\mathbb{P}(|x| \geq \epsilon) \leq \frac{\mathbb{E}e^{\lambda x}}{e^{\lambda \epsilon}}
\]
and
\[
\mathbb{P}(|x| \geq \epsilon) \leq \inf_{\lambda < 0} e^{-\lambda \epsilon} \mathbb{E}e^{\lambda x}.
\]
Proof:
\[ \Pr(x > \epsilon) = \Pr(e^{\lambda x} > e^{\lambda \epsilon}) \leq \frac{\mathbb{E}e^{\lambda x}}{e^{\lambda \epsilon}}. \]

7.3. Exponential inequalities

For the sums or averages of independent random variables the above bounds can be improved from polynomial in \(1/\epsilon\) to exponential in \(\epsilon\).

**Theorem (Bennet).** Let \(x_1, \ldots, x_n\) be independent random variables with \(\mathbb{E}x = 0\), \(\mathbb{E}x^2 = \sigma^2\), and \(|x_i| \leq M\). For \(\epsilon > 0\)

\[
\Pr\left(\sum_{i=1}^{n} x_i > \epsilon\right) \leq 2e^{-\frac{\sigma^2}{2M^2} \phi\left(\frac{\epsilon M}{n\sigma^2}\right)},
\]

where
\[ \phi(z) = (1 + z) \log(1 + z) - z. \]

**Proof.** We will prove a bound on one-side of the above theorem

\[
\Pr\left(\sum_{i=1}^{n} x_i > \epsilon\right) \leq e^{-\lambda \epsilon} \mathbb{E}e^{\lambda \sum x_i} = e^{-\lambda \epsilon} \prod_{i=1}^{n} \mathbb{E}e^{\lambda x_i},
\]

where
\[ \mathbb{E}e^{\lambda x} = \mathbb{E} \sum_{k=0}^{\infty} \frac{(\lambda x)^k}{k!} = \sum_{k=0}^{\infty} \frac{(\lambda x)^k}{k!} \mathbb{E}x^k
\]

\[ = 1 + \sum_{k=2}^{\infty} \frac{\lambda^k}{k!} \mathbb{E}x^2 x^{k-2} \leq 1 + \sum_{k=2}^{\infty} \frac{\lambda^k}{k!} M^{k-2} \sigma^2
\]

\[ = 1 + \frac{\sigma^2}{M^2} \sum_{k=2}^{\infty} \frac{\lambda^k M^k}{k!} = 1 + \frac{\sigma^2}{M^2} (e^{\lambda M} - 1 - \lambda M)
\]

The last line holds since \(1 + x \leq e^x\).

Therefore,

\[
(7.1) \quad \Pr\left(\sum_{i=1}^{n} x_i > \epsilon\right) \leq e^{-\lambda \epsilon} \frac{\sigma^2}{M^2} (e^{\lambda M} - 1 - \lambda M).
\]

We now optimize with respect to \(\lambda\) by taking the derivative with respect to \(\lambda\)

\[
0 = -\epsilon + \frac{\sigma^2}{M^2} (Me^{\lambda M} - M),
\]

\[ e^{\lambda M} = \frac{\epsilon M}{\sigma^2} + 1,
\]

\[ \lambda = \frac{1}{M} \log \left(1 + \frac{\epsilon M}{\sigma^2}\right). \]
The theorem is proven by substituting $\lambda$ into equation (7.1). □

The problem with Bennet’s inequality is that it is hard to get a simple expression for $\epsilon$ as a function of the probability of the sum exceeding $\epsilon$.

**Theorem (Bernstein).** Let $x_1, \ldots, x_n$ be independent random variables with $\mathbb{E}x = 0$, $\mathbb{E}x^2 = \sigma^2$, and $|x_i| \leq M$. For $\epsilon > 0$

$$\mathbb{P}\left(\left|\sum_{i=1}^{n} x_i\right| > \epsilon\right) \leq 2e^{-\frac{\epsilon^2}{2n\sigma^2 + \frac{2}{3}M}}.$$ 

**Proof.**

Take the proof of Bennet’s inequality and notice

$$\phi(z) \geq \frac{z^2}{2 + \frac{2}{3}z}.$$ □

**Remark.** With Bernstein’s inequality a simple expression for $\epsilon$ as a function of the probability of the sum exceeding $\epsilon$ can be computed

$$\sum_{i=1}^{n} x_i \leq \frac{2}{3}uM + \sqrt{2n\sigma^2u}.$$ 

**Outline.**

$$\mathbb{P}\left(\sum_{i=1}^{n} x_i > \epsilon\right) \leq 2e^{-\frac{\epsilon^2}{2n\sigma^2 + \frac{2}{3}M}} = e^{-u},$$

where

$$u = \frac{\epsilon^2}{2n\sigma^2 + \frac{2}{3}M}.$$ 

we now solve for $\epsilon$

$$\epsilon^2 - \frac{2}{3}\epsilon M - 2n\sigma^2\epsilon = 0$$

and

$$\epsilon = \frac{1}{3}uM + \sqrt{\frac{u^2M^2}{9} + 2n\sigma^2u}.$$ 

Since $\sqrt{a} + b \leq \sqrt{a} + \sqrt{b}$

$$\epsilon = \frac{2}{3}uM + \sqrt{2n\sigma^2u}.$$ 

So with large probability

$$\sum_{i=1}^{n} x_i \leq \frac{2}{3}uM + \sqrt{2n\sigma^2u}. \quad \triangle$$ 

If we want to bound

$$|n^{-1}\sum_{i=1}^{n} f(x_i) - \mathbb{E}f(x)|$$

we consider

$$|f(x_i) - \mathbb{E}f(x)| \leq 2M.$$ 

Therefore

$$\sum_{i=1}^{n} (f(x_i) - \mathbb{E}f(x)) \leq \frac{4}{3}uM + \sqrt{2n\sigma^2u}.$$
and
\[ n^{-1} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f(x) \leq \frac{4uM}{3n} + \sqrt{\frac{2\sigma^2 u}{n}}. \]

Similarly,
\[ \mathbb{E} f(x) - n^{-1} \sum_{i=1}^{n} f(x_i) \geq \frac{4uM}{3n} + \sqrt{\frac{2\sigma^2 u}{n}}. \]

In the above bound
\[ \sqrt{2\frac{\sigma^2 u}{n}} \geq \frac{4uM}{n} \]
which implies \( u \leq \frac{n\sigma^2}{8M^2} \), and therefore
\[ |n^{-1} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f(x)| \lesssim \sqrt{\frac{2\sigma^2 u}{n}} \quad \text{for} \quad u \lesssim n\sigma^2, \]
which corresponds to the tail probability for a Gaussian random variable and is predicted by the Central Limit Theorem (CLT) Condition that \( \lim_{n \to \infty} n\sigma^2 \to \infty \).

If \( \lim_{n \to \infty} n\sigma^2 = C \), where \( C \) is a fixed constant, then
\[ |n^{-1} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f(x)| \lesssim \frac{C}{n} \]
which corresponds to the tail probability for a Poisson random variable.

We now look at an even simpler exponential inequality where we do not need information on the variance.

**Theorem** (Hoeffding). Let \( x_1, ..., x_n \) be independent random variables with \( \mathbb{E} x = 0 \) and \( |x_i| \leq M_i \). For \( \epsilon > 0 \)
\[ \mathbb{P} \left( \left| \sum_{i=1}^{n} x_i \right| > \epsilon \right) \leq 2e^{-\frac{\epsilon^2}{2\sum_{i=1}^{n} M_i^2}}. \]

**Proof.**
\[ \mathbb{P} \left( \sum_{i=1}^{n} x_i > \epsilon \right) \leq e^{-\lambda \epsilon} \mathbb{E} e^{\lambda \sum_{i=1}^{n} x_i} = e^{-\lambda \epsilon} \prod_{i=1}^{n} \mathbb{E} e^{\lambda x_i}. \]
It can be shown (Homework problem)
\[ \mathbb{E} (e^{\lambda x_i}) \leq e^{\frac{\lambda^2 M_i^2}{2}}. \]
The bound is proven by optimizing the following with respect to \( \lambda \)
\[ e^{-\lambda \epsilon} \prod_{i=1}^{n} e^{\frac{\lambda^2 M_i^2}{2}}. \]

Applying Hoeffding’s inequality to
\[ n^{-1} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f(x) \]
we can state that with probability \( 1 - e^{-u} \)
\[ n^{-1} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f(x) \leq \sqrt{\frac{2Mu}{n}}, \]
which is a sub-Gaussian as in the CLT but without the variance information we can never achieve the $\frac{1}{n}$ rate we achieved when the random variable has a Poisson tail distribution.

We will use the following version of Hoeffding’s inequality in later lectures on Kolmogorov chaining and the Dudley’s entropy integral.

**Theorem** (Hoeffding). Let $x_1, ..., x_n$ be independent random variables with $\mathbb{P}(x_i = M_i) = 1/2$ and $\mathbb{P}(x_i = -M_i) = 1/2$. For $\epsilon > 0$

$$\mathbb{P} \left( \left| \sum_{i=1}^{n} x_i \right| > \epsilon \right) \leq 2e^{-\frac{2\epsilon^2}{\sum_{i=1}^{n} M_i^2}}.$$

**Proof.**

$$\mathbb{P} \left( \sum_{i=1}^{n} x_i > \epsilon \right) \leq e^{-\lambda \epsilon} \mathbb{E}e^\lambda \sum_{i=1}^{n} x_i = e^{-\lambda \epsilon} \prod_{i=1}^{n} \mathbb{E}e^{\lambda x_i}.$$

$$\mathbb{E}(e^{\lambda x_i}) = \frac{1}{2} e^{\lambda M_i} + \frac{1}{2} e^{-\lambda M_i},$$

$$\frac{1}{2} e^{\lambda M_i} + \frac{1}{2} e^{-\lambda M_i} = \sum_{k=0}^{\infty} \frac{(M_i \lambda)^{2k}}{(2k)!} \leq e^{\frac{\lambda^2 M_i^2}{2}}.$$

Optimize the following with respect to $\lambda$

$$e^{-\lambda \epsilon} \prod_{i=1}^{n} e^{\frac{\lambda^2 M_i^2}{2}}. \square$$

### 7.4. Martingale inequalities

In the previous section we stated some concentration inequalities for sums of independent random variables. We now look at more complicated functions of independent random variables and introduce a particular Martingale inequality to prove concentration.

Let $(\Omega, \mathcal{L}, \mu)$ be a probability space. Let $x_1, ..., x_n$ be real random variables on $\Omega$. Let the function $Z(x_1, ..., x_n) : \Omega^n \rightarrow \mathbb{R}$ be a map from the random variables to a real number.

The function $Z$ concentrates if the deviation between $Z(x_1, ..., x_n)$ and $\mathbb{E}_{x_1, ..., x_n} Z(x_1, .., x_n)$ goes to zero as $n$ goes to infinity.

**Theorem** (McDiarmid). Let $x_1, ..., x_n$ be independent random variables let $Z(x_1, ..., x_n) : \Omega^n \rightarrow \mathbb{R}$ such that

$$\forall x_1, ..., x_n, x'_1, ..., x'_n \ |Z(x_1, .., x_n) - Z(x_1, ..., x_{i-1}, x'_i, x_{i+1}, x_n)| \leq c_i,$$

then

$$\mathbb{P} (Z - \mathbb{E}Z > \epsilon) \leq e^{-\frac{\epsilon^2}{\sum_{i=1}^{n} c_i^2}}.$$

**Proof.**

$$\mathbb{P}(Z - \mathbb{E}Z > \epsilon) = \mathbb{P}(e^{\lambda(Z - \mathbb{E}Z)} > e^{\lambda \epsilon}) \leq e^{-\lambda \epsilon} \mathbb{E}e^{\lambda(Z - \mathbb{E}Z)}.$$
We will use the following very useful decomposition
\[
Z(x_1, ..., x_n) - \mathbb{E}_{x'_1, ..., x'_n}Z(x'_1, ..., x'_n) = [Z(x_1, ..., x_n) - E_{x'_1}Z(x'_1, x_2, ..., x_n)]
+ [E_{x'_1}Z(x'_1, x_2, ..., x_n) - E_{x'_1,x'_2}Z(x'_1, x'_2, x_3, ..., x_n)]
+ ...
+ [E_{x'_1, ..., x'_n} Z(x'_1, x'_2, ..., x'_{n-1}, x_n) - E_{x'_1, ..., x'_n} Z(x'_1, ..., x'_n)].
\]
We denote the random variable
\[
z_i(x_1, ..., x_n) := \mathbb{E}_{x'_1, ..., x'_{i-1}}Z(x'_1, ..., x'_{i-1}, x_i, ..., x_n) - \mathbb{E}_{x'_1, ..., x'_n}Z(x'_1, ..., x'_i, x_{i+1}, ..., x_n),
\]
and
\[
Z(x_1, ..., x_n) - \mathbb{E}_{x'_1, ..., x'_n} Z(x'_1, ..., x'_n) = z_1 + ... + z_n.
\]
The following inequality is true (see the following Lemma for a proof)
\[
\mathbb{E}_{x_i} e^{\lambda z_i} \leq e^{\lambda^2 c_i^2/2} \forall \lambda \in \mathbb{R}.
\]
by induction
\[
\mathbb{E} e^{\lambda (Z - \mathbb{E}Z)} \leq e^{\lambda^2 \sum_{i=1}^n c_i^2/2}.
\]
To derive the bound we optimize with respect to \( \lambda \)
\[
e^{-\lambda t + \lambda^2 \sum_{i=1}^n c_i^2/2}. \]
**Lemma.** For all \( \lambda \in \mathbb{R} \)
\[
\mathbb{E}_{x_i} e^{\lambda z_i} \leq e^{\lambda^2 c_i^2/2}.
\]
**Proof.**
For any \( t \in [-1, 1] \) the function \( e^{\lambda t} \) is convex with respect to \( \lambda \).
\[
e^{\lambda t} = e^{\lambda \frac{1+t}{2} - \lambda \frac{1-t}{2}} \leq \frac{1}{2} e^{\lambda} + \frac{1-t}{2} e^{-\lambda} = \frac{e^{\lambda} + e^{-\lambda}}{2} + \frac{t}{2} (e^{\lambda} - e^{-\lambda}) \leq e^{\lambda^2/2} + t \text{sh}(\lambda).
\]
Set \( t = \frac{z_i}{c_i} \) and notice that \( \frac{z_i}{c_i} \in [-1, 1] \) so,
\[
e^{\lambda z_i} = e^{\lambda c_i \frac{z_i}{c_i}} \leq e^{\lambda^2 c_i^2/2} + \frac{z_i}{c_i} \text{sh}(\lambda c_i),
\]
and
\[
\mathbb{E}_{x_i} e^{\lambda z_i} \leq e^{\lambda^2 c_i^2/2}.
\]
**Example.** We can use McDiarmid’s inequality to prove the concentration of the empirical minima. Given a dataset \( \{v_1 = (x_1, y_1), ..., v_n = (x_n, y_n)\} \) the empirical minima is
\[
Z(v_1, ..., v_n) = \min_{f \in \mathcal{H}_K} n^{-1} \sum_{i=1}^n V(f(x_i), y_i).
\]
If the loss function is bounded one can show that for all \((v_1, ..., v_n, v_i')\)
\[
|Z(v_1, ..., v_n) - Z(v_1, ..., v_{i-1}, v_i', ..., v_n)| \leq \frac{k}{n}.
\]
Therefore with probability \(1 - e^{-u}\)
\[
|Z - \mathbb{E}Z| \leq \sqrt{\frac{2ku}{n}}.
\]
So the empirical minima concentrates.
Vapnik-Červonenkis theory

8.1. Uniform law of large numbers

In the previous lecture we considered law of large numbers for a single or fixed function. We termed this as one dimensional concentration inequalities. We now look at uniform law of large numbers, that is a law of large numbers that holds uniformly over a class of functions.

The point of these uniform limit theorems is that if the law of large numbers holds for all functions in a hypothesis space then it holds for the empirical minimizer.

The reason this chapter is called Vapnik-Červonenkis theory is that they provided some of the basic tools to study these classes.

8.2. Generalization bound for one function

Before looking at uniform results we prove generalization results when the hypothesis space \( \mathcal{H} \) consists of one function, \( f_1 \).

In this case the empirical risk minimizer is \( f_1 = f_S := \arg\min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^{n} V(f, z_i) \right] \).

**Theorem.** Given \( 0 \leq V(f_1, z) \leq M \) for all \( z \) and \( S = \{z_i\}_{i=1}^{n} \) drawn i.i.d. then with probability at least \( 1 - e^{-t} \) (\( t > 0 \))

\[
\mathbb{E}_z V(f_1, z) \leq n^{-1} \sum_{i=1}^{n} V(f_1, z_i) + \sqrt{\frac{M^2 t}{n}}.
\]

**Proof.**

By Hoeffding’s inequality

\[
\mathbb{P} \left( \mathbb{E}_z V(f_1, z) - n^{-1} \sum_{i=1}^{n} V(f_1, z_i) > \varepsilon \right) \leq e^{-n\varepsilon^2/M^2}
\]

so

\[
\mathbb{P} \left( \mathbb{E}_z V(f_1, z) - n^{-1} \sum_{i=1}^{n} V(f_1, z_i) \leq \varepsilon \right) > 1 - e^{-n\varepsilon^2/M^2}.
\]

Set \( t = n\varepsilon^2/M^2 \) and the result follows. \( \square \)
8.3. Generalization bound for a finite number of functions

We now look at the case of ERM on a hypothesis space $\mathcal{H}$ with a finite number of functions, $k = |\mathcal{H}|$. In this case, the empirical minimizer will be one of the $k$ functions.

**Theorem.** Given $0 \leq V(f_j, z) \leq M$ for all $f_j \in \mathcal{H}$, $z$ and $S = \{z_i\}_{i=1}^n$ drawn i.i.d. then with probability at least $1 - e^{-t}$ ($t > 0$) for the empirical minimizer, $f_S$,

$$
\mathbb{E}_z V(f_S, z) < n^{-1} \sum_{i=1}^n V(f_S, z_i) + \sqrt{\frac{M^2(\log K + t)}{n}}.
$$

**Proof.** The follow implication of events holds

$$
\left\{ \max_{f_j \in \mathcal{H}} \mathbb{E}_z V(f_j, z) - n^{-1} \sum_{i=1}^n V(f_j, z_i) < \varepsilon \right\} \Rightarrow \left\{ \mathbb{E}_z V(f_S, z) - n^{-1} \sum_{i=1}^n V(f_S, z_i) < \varepsilon \right\}.
$$

$$
\mathbb{P} \left( \max_{f_j \in \mathcal{H}} \mathbb{E}_z V(f_j, z) - n^{-1} \sum_{i=1}^n V(f_j, z_i) \geq \varepsilon \right)
\leq \sum_{f_j \in \mathcal{H}} \mathbb{P} \left( \mathbb{E}_z V(f_j, z) - n^{-1} \sum_{i=1}^n V(f_j, z_i) \geq \varepsilon \right)
\leq k e^{-ne^2/M^2},
$$

the last step comes from our single function result. Set $e^{-t} = ke^{-ne^2/M^2}$ and the result follows. □

8.4. Generalization bound for compact hypothesis spaces

We now prove a sufficient condition for the generalization of hypothesis spaces with an infinite number of functions and then give some examples of such spaces.

We first assume that our hypothesis space is a subset of the space of continuous functions, $\mathcal{H} \subset C(X)$.

**Definition.** A metric space is compact if and only if it is totally bounded and complete.

**Definition.** Let $R$ be a metric space and $\varepsilon$ any positive number. Then a set $A \subset R$ is said to be an $\varepsilon$-net for a set $M \subset R$ if for every $x \in M$, there is at least one point $a \in A$ such that $\rho(x, a) < \varepsilon$. Here $\rho(\cdot, \cdot)$ is a norm.

**Definition.** Given a metric space $R$ and a subset $M \subset R$ suppose $M$ has a finite $\varepsilon$-net for every $\varepsilon > 0$. Then $M$ is said to be totally bounded.

**Proposition.** A compact space has a finite $\varepsilon$-net for all $\varepsilon > 0$.

For the remainder of this section we will use the supnorm,

$$
\rho(a, b) = \sup_{x \in X} |a(x) - b(x)|.
$$
Definition. Given a hypothesis space \( \mathcal{H} \) and the supnorm, the covering number \( N(\mathcal{H}, \epsilon) \) is the minimal number \( \ell \in \mathbb{N} \) such that for every \( f \in \mathcal{H} \) there exists functions \( \{g_i\}_{i=1}^\ell \) such that
\[
\sup_{x \in \mathcal{X}} |f(x) - g_i(x)| \leq \epsilon \text{ for some } i.
\]

We now state a generalization bound for this case. In the bound we assume \( V(f, z) = (f(x) - y)^2 \) but the result can be easily extended for any Lipschitz loss
\[
|V(f_1, z) - V(f_2, z)| \leq C||f_1(x) - f_2(x)||_\infty \forall z.
\]

Theorem. Let \( \mathcal{H} \) be a compact subset of \( \mathcal{C}(\mathcal{X}) \). Given \( 0 \leq |f(x) - y| \leq M \) for all \( f \in \mathcal{H}, z \) and \( S = \{z_i\}_{i=1}^n \) drawn i.i.d. then with probability at least \( 1 - e^{-t} \) (\( t > 0 \)) for the empirical minimizer, \( f_S \),
\[
\mathbb{E}_{x,y}(f_S(x) - y)^2 < n^{-1} \sum_{i=1}^n (f_S(x_i) - y_i)^2 + \sqrt{\frac{M^2(\log N(\mathcal{H}, \epsilon/8M) + t)}{n}}.
\]

We first prove two useful lemmas. Define
\[
D(f, S) := \mathbb{E}_{x,y}(f(x) - y)^2 - n^{-1} \sum_{i=1}^n (f(x_i) - y_i)^2.
\]

Lemma. If \( |f_j(x) - y| \leq M \) for \( j = 1, 2 \) then
\[
|D(f_1, S) - D(f_2, S)| \leq 4M||f_1 - f_2||_\infty.
\]

Proof. Note that
\[
(f_1(x) - y)^2 - (f_2(x) - y)^2 = (f_1(x) - f_2(x))(f_1(x) + f_2(x) - 2y)
\]
so
\[
\mathbb{E}_{x,y}(f_1(x) - y)^2 - \mathbb{E}_{x,y}(f_2(x) - y)^2 = \left| \int (f_1(x) - f_2(x))(f_1(x) + f_2(x) - 2y)d\mu(x, y) \right|
\]
\[
\leq ||f_1 - f_2||_\infty \int |f_1(x) - y + f_2(x) - y|d\mu(x, y)
\]
\[
\leq 2M||f_1 - f_2||_\infty,
\]
and
\[
|n^{-1} \sum_{i=1}^n [(f_1(x_i) - y_i)^2 - (f_2(x_i) - y_i)^2]| = n^{-1} \left| \sum_{i=1}^n (f_1(x_i) - f_2(x_i))(f_1(x_i) + f_2(x_i) - 2y) \right|
\]
\[
\leq ||f_1 - f_2||_\infty \frac{1}{n} \sum_{i=1}^n |f_1(x_i) - y_i + f_2(x_i) - y_i|
\]
\[
\leq 2M||f_1 - f_2||_\infty.
\]
The result follows from the above inequalities. \( \square \)

Lemma. Let \( \mathcal{H} = B_1 \cup \ldots \cup B_\ell \) and \( \epsilon > 0 \). Then
\[
\mathbb{P} \left( \sup_{f \in \mathcal{H}} D(f, S) \right) \leq \sum_{j=1}^\ell \mathbb{P} \left( \sup_{f \in B_j} D(f, S) \right).
\]
We now prove Theorem 8.4.

Let \( \ell = N(\mathcal{H}, \frac{\epsilon}{4M}) \) and the functions \( \{g_j\}_{j=1}^\ell \) have the property that the disks \( B_j \) centered at \( f_j \) with radius \( \frac{\epsilon}{4M} \) cover \( \mathcal{H} \). By the first lemma for all \( f \in B_j \)
\[
\left| D(f, S) - D(f_j, S) \right| \leq 4M \left\| f - f_j \right\|_\infty \leq 4M \frac{\epsilon}{4M} = \epsilon,
\]
this implies that for all \( f \in B_j \)
\[
\sup_{f \in B_j} \left| D(f, S) \right| \geq 2\epsilon \Rightarrow \left| D(f_j, S) \right| \geq \epsilon.
\]
So
\[
\mathbb{P} \left( \sup_{f \in B_j} \left| D(f, S) \right| \geq 2\epsilon \right) \leq \mathbb{P} \left( \left| D(f_j, S) \right| \geq \epsilon \right) \leq 2e^{-\epsilon^2 n/M^2}.
\]

This combined with the second lemma implies
\[
\mathbb{P} \left( \sup_{f \in \mathcal{H}} D(f, S) \right) \leq N \left( \mathcal{H}, \frac{\epsilon}{8M} \right) e^{-\epsilon^2 n/M^2}.
\]
Since the following implication of events holds
\[
\left\{ \sup_{f \in \mathcal{H}} \mathbb{E}_x V(f_j, z) - n^{-1} \sum_{i=1}^n V(f_j, z_i) < \epsilon \right\} \Rightarrow \left\{ \mathbb{E}_x V(f_S, z) - n^{-1} \sum_{i=1}^n V(f_s, z_i) < \epsilon \right\}
\]
the result is obtained by setting \( e^{-t} = N \left( \mathcal{H}, \frac{\epsilon}{8M} \right) e^{-n\epsilon^2/M^2} \). □

A result of the above theorem is the following sufficient condition for uniform convergence and consistency of ERM.

**Corollary.** For a Lipschitz loss function ERM is consistent if for all \( \epsilon > 0 \)
\[
\lim_{n \to \infty} \frac{\log N(\mathcal{H}, \epsilon)}{n} = 0.
\]

**Proof.**

This follows directly from the statement
\[
\mathbb{P} \left( \sup_{f \in \mathcal{H}} D(f, S) \right) \leq N \left( \mathcal{H}, \frac{\epsilon}{8M} \right) e^{-\epsilon^2 n/M^2}. \quad \square
\]

We now compute covering numbers for a few types of hypothesis spaces.
We also need the definition of packing numbers.

**Definition.** Given a hypothesis space \( \mathcal{H} \) and the supnorm, \( \ell \) functions \( \{g_i\}_{i=1}^\ell \) are \( \epsilon \)-separated if
\[
\sup_{x \in \mathcal{X}} \left| g_j(x) - g_i(x) \right| > \epsilon \quad \forall i \neq j.
\]
The packing number \( \mathcal{P}(\mathcal{H}, \epsilon) \) is the maximum cardinality of \( \epsilon \)-separated sets.

The following relationship between packing and covering numbers is very useful.
Lemma. Given a metric space \((A, \rho)\). Then for all \(\epsilon > 0\) and for every \(W \subset A\), the covering numbers and packing numbers satisfy

\[
\mathcal{P}(W, 2\epsilon, \rho) \leq N(W, \epsilon, \rho) \leq \mathcal{P}(W, \epsilon, \rho).
\]

Proof.

For the second inequality suppose \(P\) is an \(\epsilon\)-packing of maximal cardinality, \(\mathcal{P}(W, \epsilon, d)\). Then for any \(w \in W\) there must be a \(u \in P\) with \(\rho(u, w) < \epsilon\), otherwise \(w\) is not an element of \(P\) and \(P \cup w\) is an \(\epsilon\)-packing. This contradicts the assumption that \(P\) is a maximal \(\epsilon\)-packing. So any maximal \(\epsilon\) packing is an \(\epsilon\)-cover.

For the first inequality suppose \(C\) is an \(\epsilon\)-cover for \(W\) and that \(P\) is a \(2\epsilon\)-packing of \(W\) with maximum cardinality \(\mathcal{P}(W, \epsilon, d)\). We will show that \(|P| \leq |C|\).

Assume that \(|C| > |P|\). Then for two points \(w_1, w_2 \in P\) and one point \(u \in C\) the following will hold

\[
\rho(w_1, u) \leq \epsilon \text{ and } \rho(w_2, u) \leq \epsilon \implies \rho(w_1, w_2) \leq 2\epsilon.
\]

This contradicts the fact that the points in \(P\) are \(2\epsilon\)-separated. \(\square\)

In general we will compute packing numbers for hypothesis spaces and use the above lemma to obtain the covering number.

The following proposition will be useful.

Proposition. Given \(x \in \mathbb{R}^d\), the restriction the space to the unit ball \(B = \{x : ||x|| \leq M\}\), and the standard Euclidean metric \(\rho(x, y) = ||x - y||\), then for \(\epsilon \leq M\)

\[
\mathcal{P}(B, \epsilon, \rho) \leq \left(\frac{3M}{\epsilon}\right)^d.
\]

Proof.

The \(\ell\) points \(w_1, .., w_\ell\) form an optimal \(\epsilon\)-packing so

\[
\text{Vol}\left(M + \frac{\epsilon}{2}\right) = C_d \left(M + \frac{\epsilon}{2}\right)^d
\]

\[
\text{Vol}\left(\frac{\epsilon}{2}\right) = C_d \left(\frac{\epsilon}{2}\right)^d
\]

\[
\ell [C_d \left(\frac{\epsilon}{2}\right)^d] = C_d \left(M + \frac{\epsilon}{2}\right)^d
\]

\[
\ell \leq \left(\frac{2M + \epsilon}{\epsilon}\right)^d
\]

\[
\leq \left(\frac{3M}{\epsilon}\right)^d \text{ for all } \epsilon \leq M. \quad \square
\]

Example. Covering numbers for a finite dimensional RKHS.

For a finite dimensional bounded RKHS

\[
\mathcal{H}_K = \left\{ f : f(x) = \sum_{p=1}^{m} c_p \phi_p(x) \right\},
\]

with \(\|f\|_K^2 \leq M\).
By the reproducing property and Cauchy-Schwartz inequality, the supnorm can be bound by the RKHS norm:

\[
\|f\|_\infty = \|\langle K(x, \cdot), f(\cdot) \rangle_K\|_\infty \\
\leq \|K(x, \cdot)\|_K \|f\|_K \\
= \sqrt{\langle K(x, \cdot), K(x, \cdot) \rangle} \|f\|_K \\
= \kappa \|f\|_K
\]

This means that if we can cover with the RKHS norm we can cover with the supnorm.

Each function in our cover, \(\{g_i\}_{i=1}^\ell\) can be written as

\[
g_i(x) = \sum_{p=1}^m d_p \phi_p(x)
\]

So if we find \(\ell\) vectors \(d_i\) for which for all \(c : \sum_{p=1}^m c^2_p \leq M\) there exists a \(d_i\) such that

\[
\sum_{p=1}^m \frac{(c_p - d_{ip})^2}{\lambda_p} < \epsilon^2,
\]

we have a cover at scale \(\epsilon\). The above is simply a weighted Euclidean norm and can be reduced to the problem of covering a ball of radius \(M\) in \(\mathbb{R}^m\) using the Euclidean metric. Using proposition 8.4 we can bound the packing number with the RKHS norm and the supnorm

\[
P(H, \epsilon, \|\cdot\|_{\mathcal{H}_K}) \leq \left(\frac{3M}{\epsilon \kappa} \right)^m,
\]

\[
P(H, \epsilon, \|\cdot\|_\infty) \leq \left(\frac{3M}{\kappa \epsilon} \right)^m.
\]

Using lemma 8.4 we can get a bound on the covering number

\[
\mathcal{N}(H, \epsilon, \|\cdot\|_\infty) \leq \left(\frac{3M}{\kappa \epsilon} \right)^m.
\]

We have shown that for \(H \subset C(X)\) that is compact with respect to the supnorm we have uniform convergence. This requirement is too strict to determine necessary conditions. A large class of functions that these conditions do not apply to are indicator functions \(f(x) \in \{0, 1\}\).

### 8.5. Generalization bound for hypothesis spaces of indicator functions

In this section we derive necessary and sufficient conditions for uniform convergence of indicator functions and as a result generalization bounds for indicator functions, \(f(x) \in \{0, 1\}\).

As in the case of compact functions we will take a class of indicator functions \(H\) and reduce this to some finite set of functions. In the case of indicator functions this is done via the notion of a growth function which we now define.
Definition. Given a set of $n$ points $\{x_i\}_{i=1}^n$ and a class of indicator functions $\mathcal{H}$ we say that a function $f \in \mathcal{H}$ picks out a certain subset of $\{x_i\}_{i=1}^n$ if this set can be formed by the operation $f \cap \{x_i\}_{i=1}^n$. The cardinality of the number of subsets that can be picked out is called the growth function:

$$\triangle_n(\mathcal{H}, \{x_i\}_{i=1}^n) = \# \{ f \cap \{x_i\}_{i=1}^n : f \in \mathcal{H} \}.$$ 

We will now state a lemma which will look very much like the generalization of the growth function. We will remove this dependence soon.

We first prove two useful lemmas. Define

$$D(f, S) := \mathbb{E}_{x, y} I_{f(x) \neq y} - \frac{1}{n} \sum_{i=1}^{n} I_{f(x_i) \neq y_i}.$$ 

The first lemma is based upon the idea of symmetrization and replaces the deviation between the empirical and expected error to the difference between two empirical errors.

Lemma. Given two independent copies of the data $S = \{z_i\}_{i=1}^n$ and $S' = \{z_i'\}_{i=1}^n$, then for any fixed $f \in \mathcal{H}$ if $n \geq 2/\epsilon^2$

$$\mathbb{P}( |D(f, S) - D(f, S')| \geq \epsilon) \leq 2 \mathbb{P}( |D(f, S) - D(f, S')| > \epsilon/2 ),$$

where

$$|D(f, S) - D(f, S')| = \frac{1}{n} \sum_{i=1}^{n} I_{f(x_i) \neq y_i} - \frac{1}{n} \sum_{i=1}^{n} I_{f(x_i') \neq y_i'}.$$ 

Proof. We first assume that

$$\mathbb{P}( |D(f, S')| \leq \epsilon/2 \mid S) \geq 1/2,$$

where we have conditioned on $S$. Since $S$ and $S'$ are independent we can integrate out

$$\frac{1}{2} \mathbb{P}( |D(f, S')| > \epsilon) \leq \mathbb{P}( |D(f, S')| \leq \epsilon/2, |D(f, S')| > \epsilon).$$

By the triangle inequality $|D(f, S)| > \epsilon$ and $|D(f, S')| \leq \epsilon/2$ implies $|D(f, S) - D(f, S')| \geq \epsilon/2$, so

$$\mathbb{P}( |D(f, S')| \leq \epsilon/2, |D(f, S')| > \epsilon) \leq \mathbb{P}( |D(f, S) - D(f, S')| \geq \epsilon/2 ).$$

To complete the proof we need to show our initial assumption holds. Since $\mathcal{H}$ is a class of indicator functions the elements in the sum are binomial random variables and the variance of $n$ of them will be at most $1/4n$. So by the Bienaymé-Chebyshev inequality

$$\mathbb{P}( |D(f, S')| > \epsilon/2) \geq 1/4ne^2,$$
which implies the initial assumption when $n \geq 2/\epsilon^2$. □

By symmetrizing we now have a term that depends only on samples. The problem is that it depends on the samples we have but also an independent compy. This nuisance is removed by a second step of symmetrization.

**Lemma.** Let $\sigma_i$ be a Rademacher random variable ($\mathbb{P}(\sigma_i = \pm 1) = 1/2$) then

$$\mathbb{P}(|D(f, S) - D(f, S')| > \epsilon/2) \leq 2\mathbb{P}\left(\left| n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x_i) \neq y_i)} \right| > \epsilon/4 \right).$$

*Proof.*

\[\begin{align*}
\mathbb{P}(|D(f, S) - D(f, S')| > \epsilon/2) &= \mathbb{P}\left(\left| n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x_i) \neq y_i)} - n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x'_i) \neq y'_i)} \right| > \epsilon/2 \right) \\
&\leq \mathbb{P}\left(\left| n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x_i) \neq y_i)} \right| > \epsilon/4 \right) + \\
&\quad \mathbb{P}\left(\left| n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x'_i) \neq y'_i)} \right| > \epsilon/4 \right) \\
&\leq 2\mathbb{P}\left(\left| n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x_i) \neq y_i)} \right| > \epsilon/4 \right). \quad \Box
\end{align*}\]

The second symmetrization step allows is to bound the deviation between the empirical and expected errors based upon a quantity computed just on the empirical data.

We now prove Lemma 8.5.

By the symmetrization lemmas for $n \geq 8/\epsilon^2$

$$\mathbb{P}(|D(f, S)| > \epsilon) \leq 4\mathbb{P}\left(\left| n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x_i) \neq y_i)} \right| > \epsilon/4 \right).$$

By the Rademacher version of Hoeffdings inequality

$$\mathbb{P}\left(\left| n^{-1} \sum_{i=1}^{n} \sigma_i I_{(f(x_i) \neq y_i)} \right| > \epsilon \right) \leq 2e^{-2\epsilon^2}.$$ 

Combining the above for a single function

$$\mathbb{P}(|D(f, S)| > \epsilon) \leq 8e^{-\epsilon^2/n}.$$

Given data $S$ the growth function characterizes the cardinality of subsets that can be “picked out” which is a bound on the number of possible labellings or realizable functions, $\ell = \triangle_n(H, S)$. We index the possible labelings by $f_j$ where $j = 1, \ldots, \ell$. 

We now proceed as in the case of a finite number of functions

\[
\mathbb{P}\left(\sup_{f \in \mathcal{H}} |D(f, S)| \geq \epsilon \right) = \mathbb{P}\left(\bigcup_{f \in \mathcal{H}} |D(f, S)| \geq \epsilon \right) \\
\leq \sum_{i=1}^{\ell} \mathbb{P}\left(|D(f_i, S)| \geq \epsilon \right) \\
\leq 8 \Delta_n(\mathcal{H}, S) e^{-n\epsilon^2/8}.
\]

Setting \( e^{-t/8} = 8 \Delta_n(\mathcal{H}, S) e^{-n\epsilon^2/8} \) completes the proof. □

This bound is not uniform since the growth function depends on the data \( S \). We can make the bound uniform by defining a uniform notion of the growth function.

**Definition.** The uniform growth function is

\[
\Delta_n(\mathcal{H}) = \max_{x_1, \ldots, x_n} \Delta_n(\mathcal{H}, \{x_i\}_{i=1}^n).
\]

**Corollary.** Let \( \mathcal{H} \) be a class of indicator functions and \( S = \{z_i\}_{i=1}^n \) drawn i.i.d. then with probability at least \( 1 - e^{-t/8} \) (\( t > 0 \)) for the empirical minimizer, \( f_S \),

\[
\mathbb{E}_{x,y} I_{\{f_S(x) \neq y\}} < n^{-1} \sum_{i=1}^{n} I_{\{f_S(x_i) \neq y_i\}} + \sqrt{\frac{(\log 8 \Delta_n(\mathcal{H}) + t)}{n}},
\]

where \( \Delta_n(\mathcal{H}) \) is the uniform growth function.

**Corollary.** For a class of indicator functions ERM is consistent if and only if for all \( \epsilon > 0 \)

\[
\lim_{n \to \infty} \frac{8 \log \Delta_n(\mathcal{H})}{n} = 0.
\]

We now characterize conditions under which the uniform growth function grows polynomially. To do this we need a few definitions.

**Definition.** A hypothesis space, \( \mathcal{H} \), shatters a set \( \{x_1, \ldots, x_n\} \) if each of its \( 2^n \) subsets can be “picked out” by \( \mathcal{H} \). The Vapnik-Červonenkis (VC) dimension, \( v(\mathcal{H}) \), of a hypothesis space is the largest \( n \) for which all sets of size \( n \) are shattered by \( \mathcal{H} \)

\[
v(\mathcal{H}) = \sup \{ n : \Delta_n(\mathcal{H}) = 2^n \},
\]

if the exists no such \( n \) then the VC dimension is infinite.

**Definition.** A hypothesis space of indicator functions \( \mathcal{H} \) is a VC class if and only if it has finite VC dimension.

**Examples.**

The VC dimension controls the growth function via the following lemma.

**Lemma.** For a hypothesis space \( \mathcal{H} \) with VC dimension \( d \) and \( n > d \)

\[
\Delta_n(\mathcal{H}) \leq \sum_{i=1}^{d} \binom{n}{i}.
\]
Proof.

The proof will be by induction on \( n + d \). We define \( \binom{n}{i} := 0 \) if \( i < 0 \) or \( i > n \). In addition one can check
\[
\binom{n}{i} = \binom{n-1}{i-1} + \binom{n-1}{i}.
\]

When \( d = 0 \) \(|\mathcal{H}| = 1\) since no points can be shattered so for all \( n \)
\[\triangle_n(\mathcal{H}) = 1 = \binom{n}{0} = \Phi_0(n).\]

When \( n = 0 \) there is only one way to label 0 examples so
\[\triangle_0(\mathcal{H}) = 1 = \sum_{i=1}^{d} \binom{0}{i} = \Phi_d(0).\]

Assume the lemma to hold for \( n', d' \) such that \( n' + d' < n + d \).

Given \( S = \{x_1, ..., x_n\} \) and \( S_n = \{x_1, ..., x_{n-1}\} \). We now define three hypothesis spaces \( \mathcal{H}_0, \mathcal{H}_1, \) and \( \mathcal{H}_2 \):
\[
\mathcal{H}_0 := \{ f_i : i = 1, ..., \triangle_n(\mathcal{H}, S) \}\]
\[
\mathcal{H}_1 := \{ f_i : i = 1, ..., \triangle_{n-1}(\mathcal{H}, S_n) \}\]
\[
\mathcal{H}_2 := \mathcal{H}_0 - \mathcal{H}_1,
\]
where each \( f_i \) in set \( \mathcal{H}_0 \) is a possible labeling of \( S \) by \( \mathcal{H} \), each \( f_i \) in set \( \mathcal{H}_1 \) is a possible labeling of \( S_n \) by \( \mathcal{H} \).

For the set \( \mathcal{H}_1 \) over \( S_n \): \( n_1 = n-1 \) since there is one fewer sample and \( v(\mathcal{H}_1) \leq d \) since reducing the number of hypotheses cannot increase the VC dimension.

For the set \( \mathcal{H}_2 \) over \( S_n \): \( n_1 = n-1 \) since there is one fewer sample and \( v(\mathcal{H}_2) \leq d - 1 \). If \( S' \subseteq S_n \) is shattered by \( \mathcal{H}_2 \) then all labellings of \( S' \) must occur both in \( \mathcal{H}_1 \) and \( \mathcal{H}_2 \) but with different labels on \( x_n \). So \( S' \cup \{x_n\} \) which has cardinality \(|S'| + 1\) is shattered by \( \mathcal{H} \) and so \(|S'|\) cannot be more than \( d - 1 \). By induction \( \triangle_{n-1}(\mathcal{H}_1, S_n) \leq \Phi_d(m-1) \) and \( \triangle_{n-1}(\mathcal{H}_2, S_n) \leq \Phi_{d-1}(m-1) \). By construction
\[
\triangle_n(\mathcal{H}, S) = |\mathcal{H}_1| + |\mathcal{H}_2| = \triangle_{n-1}(\mathcal{H}_1, S_n) + \triangle_{n-1}(\mathcal{H}_2, S_n)
\leq \Phi_d(n-1) + \Phi_{d-1}(n-1)
= \sum_{i=0}^{d} \binom{n-1}{i} + \sum_{i=0}^{d-1} \binom{n-1}{i}
= \sum_{i=0}^{d} \binom{n-1}{i} + \sum_{i=0}^{d-1} \binom{n-1}{i-1}
= \sum_{i=0}^{d} \left[ \binom{n-1}{i} + \binom{n-1}{i-1} \right]
= \sum_{i=0}^{d} \binom{n}{i}. \quad \square
\]

Lemma. For \( n \geq d \geq 1 \)
\[
\sum_{i=1}^{d} \binom{n}{i} < \left( \frac{en}{d} \right)^d.
\]
Proof.
For $0 \leq i \leq d$ and $n \geq d$

$$(m/d)^d(d/m)^i \geq 1,$$

so

$$\sum_{i=1}^{d} \binom{n}{i} \leq (n/d)^d \sum_{i=1}^{d} \binom{n}{i} (d/n)^i \leq (n/d)^d(1 + d/n)^n < (ne/d)^d.$$  □

This now lets state the generalization bound in terms of VC dimension.

**Theorem.** Let $\mathcal{H}$ be a class of indicator functions with VC dimension $d$ and $S = \{z_i\}_{i=1}^{n}$ drawn i.i.d. then with probability at least $1 - e^{-t/8}$ ($t > 0$) for the empirical minimizer, $f_S$,\n
$$\mathbb{E}_{x,y} I_{\{f_S(x) \neq y\}} < n^{-1} \sum_{i=1}^{n} I_{\{f_S(x) \neq y_i\}} + 2 \sqrt{\frac{(8d \log(8en/d) + t)}{n}}.$$  

**Proof.** From the proof of lemma we have

$$\mathbb{P} \left( \sup_{f \in \mathcal{H}} |D(f, S)| \geq \epsilon \right) \leq 8 \Delta_n(\mathcal{H}, S) e^{-n\epsilon^2/8},$$

therefore since $\Delta_n(\mathcal{H}, S) \leq \left(\frac{en}{d}\right)^d$, we have

$$\mathbb{P} \left( \sup_{f \in \mathcal{H}} |D(f, S)| \geq \epsilon \right) \leq 8 \left(\frac{en}{d}\right)^d e^{-n\epsilon^2/8},$$

and setting $e^{-t/8} = 8 \left(\frac{en}{d}\right)^d e^{-n\epsilon^2/8}$ gives us\n
$$\mathbb{E}_{x,y} I_{\{f_S(x) \neq y\}} < n^{-1} \sum_{i=1}^{n} I_{\{f_S(x) \neq y_i\}} + \sqrt{\frac{(8d \log(8en/d) + t + 8 \log 8)}{n}},$$

for $n > 2$ and $d > 1$ $8 \log 8 < 8d \log(en/d)$ so\n
$$\sqrt{\frac{(8d \log(8en/d) + t + 8 \log 8)}{n}} < 2 \sqrt{\frac{(8d \log(8en/d) + t)}{n}},$$

which proves the theorem. □

**Theorem.** For a class of indicator functions ERM the following are equivalent

1. ERM is consistent
2. for all $\epsilon > 0$
   $$\lim_{n \to \infty} \frac{8 \log \Delta_n(\mathcal{H})}{n} = 0.$$
3. the VC dimension $v(\mathcal{H})$ is finite.

### 8.6. Kolmogorov chaining

In this section we introduce Kolmogorov chaining which is a much more efficient way of constructing a cover. In the process we derive Dudley’s entropy integral.

We first define an empirical norm.
Definition. Given \( S = \{x_1, ..., x_n\} \) the empirical \( \ell_2 \) norm is
\[
\rho_S(f, g) = \left( n^{-1} \sum_{i=1}^{n} (f(x_i) - g(x_i))^2 \right)^{1/2}.
\]

We can define a cover given the empirical norm

Definition. Given a hypothesis space \( \mathcal{H} \) and the above norm, the covering number \( N(\mathcal{H}, \epsilon, \rho_S) \) is the minimal number \( \ell \in \mathbb{N} \) such that for every \( f \in \mathcal{H} \) there exists functions \( \{g_i\}_{i=1}^{\ell} \) such that
\[
\rho_S(f, g_i) \leq \epsilon.
\]

The proof of the following theorem is identical to the proof of lemma 8.5.

Theorem. Given the square loss and \( H \) be a functions such that \(-1 \leq f(x) \leq 1 \), \( y \in [-1, 1] \) and \( S = \{z_i\}_{i=1}^{n} \) drawn i.i.d. then with probability at least \( 1 - e^{-t/8} \) for the empirical minimizer, \( f_S \),
\[
\mathbb{E}_{x,y}(f_S(x) - y)^2 < n^{-1} \sum_{i=1}^{n} (f_S(x_i) - y_i)^2 + \sqrt{\frac{(8 \log N(\mathcal{H}, \epsilon/8M, \rho_S) + t)}{n}},
\]
where \( N(\mathcal{H}, \epsilon/8M, \rho_S) \) is the empirical cover.

The key idea in the proof of both lemma 8.5 and the above theorem is that
\[
\mathbb{P}(|D(f, S)| > \epsilon) \leq 4 \mathbb{P} \left( n^{-1} \sum_{i=1}^{n} \sigma_i f(x_i) > \epsilon/4 \right),
\]
where
\[
D(f, S) := \mathbb{E}_{x,y}(f(x) - y)^2 - n^{-1} \sum_{i=1}^{n} (f(x_i) - y_i)^2,
\]
and \( \sigma_i \) is a Rademacher random variable.

We now prove the chaining theorem.

Theorem. Given a hypothesis space \( \mathcal{H} \) where for all \( f \in \mathcal{H} -1 \leq f(x) \leq 1 \) if we define
\[
R(f) = n^{-1} \sum_{i=1}^{n} \sigma_i f(x_i),
\]
then
\[
\mathbb{P} \left( \forall f \in \mathcal{H}, \ R(f) \leq \frac{2^{9/2}}{\sqrt{n}} \int_0^{d(0,f)} \log^{1/2} \mathcal{P}(\mathcal{H}, \epsilon, \rho_S) d\epsilon + 2^{7/2} d(0,f) \sqrt{\frac{\epsilon}{n}} \right) \geq 1 - e^{-u},
\]
where \( \mathcal{P}(\mathcal{H}, \epsilon, \rho_S) \) is the empirical packing number and
\[
\int_0^{d(0,f)} \log^{1/2} \mathcal{P}(\mathcal{H}, \epsilon, \rho_S) d\epsilon
\]
is Dudley’s entropy integral.

Proof.

Without loss of generality we will assume that the zero function \( \{0\} \) is in \( \mathcal{H} \).

We will construct a nested sets of functions
\[
\{0\} = \mathcal{H}_0 \subseteq \mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq ... \subseteq \mathcal{H}_j \subseteq ... \mathcal{H}.
\]

These subsets will have the following properties
Given a set \( \mathcal{H} \) we can construct \( \mathcal{H}_{j+1} \) via the following procedure:

1. \( \mathcal{H}_{j+1} := \mathcal{H}_j \)
2. Find all \( f \in \mathcal{H} \) such that for all \( g \in \mathcal{H}_{j+1} \) \( \rho_S(f, g) > 2^{-(j+1)} \)
3. Add the above \( f \) to \( \mathcal{H}_{j+1} \).

We now define a projection operation \( \pi_j : \mathcal{H} \to \mathcal{H}_j \) where given \( f \in \mathcal{H} \)

\[
\pi_j(f) := g \quad \text{where } g \in \mathcal{H}_j \text{ such that } \rho_S(g, f) \leq 2^{-j}.
\]

For all \( f \in \mathcal{H} \) the following chaining holds

\[
f = \pi_0(f) + (\pi_1(f) - \pi_0(f)) + (\pi_2(f) - \pi_1(f)) + \ldots
\]

\[
= \sum_{j=1}^{\infty} (\pi_j(f) - \pi_{j-1}(f)),
\]

and

\[
\rho_S(\pi_{j-1}(f), \pi_j(f)) \leq \rho(\pi_{j-1}(f), f) + \rho_S(\pi_j(f), f)
\]

\[
\leq 2^{-(j-1)} + 2^{-j} = 3 \cdot 2^{-j} \leq 2^{-j+2}.
\]

\( R(f) \) is a linear function, so

\[
R(f) = \sum_{j=1}^{\infty} (\pi_j(f) - \pi_{j-1}(f)).
\]

The set of links in the chain between two levels are defined as follows

\[
L_{j-1,j} := \{ f - g : f \in \mathcal{H}_j, g \in \mathcal{H}_{j-1} \text{ and } \rho_S(f, g) \leq 2^{-j+2} \}.
\]

For a fixed link \( \ell \in L_{j-1,j} \)

\[
R(\ell) = n^{-1} \sum_{i=1}^{n} \sigma_\ell(x_i),
\]

and \( |\ell(x_i)| \leq 2^{-j+2} \) so by Hoeffding’s inequality

\[
\mathbb{P}(R(\ell) \geq t) \leq e^{-nt^2/(2 \sum_{i=1}^{n} \ell^2(x_i))} \leq e^{-nt^2/(2 \cdot 2^{-2j+4})}.
\]

The cardinality of the set of links is

\[
|L_{j-1,j}| \leq |\mathcal{H}_j| \cdot |\mathcal{H}_{j-1}| \leq (|\mathcal{H}_j|)^2.
\]

So

\[
\mathbb{P}(\forall \ell \in L_{j-1,j}, R(\ell) \leq t) \geq 1 - (|\mathcal{H}_j|)^2 e^{-nt^2/2^{-2j+5}},
\]

setting

\[
t = \sqrt{\frac{2^{-2j+5}}{n} \left( 4 \log |\mathcal{H}_j| + u \right)} \leq \sqrt{\frac{2^{-2j+5}}{n} 4 \log |\mathcal{H}_j| + \sqrt{\frac{2^{-2j+5}u}{n}}},
\]

gives us

\[
\mathbb{P} \left( \forall \ell \in L_{j-1,j}, R(\ell) \leq \frac{2^{7/2} 2^{-j} \log^{1/2} |\mathcal{H}_j|}{\sqrt{n}} + 2^{5/2} 2^{-j} \sqrt{\frac{u}{n}} \right) \geq 1 - \frac{1}{|\mathcal{H}_j|} e^{-u}.
\]
If $\mathcal{H}_{j-1} = \mathcal{H}_j$ then

$$\pi_{j-1}(f) = \pi_j(f)$$

and $L_{j-1,j} = \{0\}$.

So over all levels and links with probability at least $1 - \sum_{j=1}^{\infty} \frac{1}{|H_j|} e^{-u}$

$$\forall j \geq 1, \forall \ell \in L_{j-1,j}, R(\ell) \leq \frac{2^{7/2} 2^{-j} \log^{1/2} |\mathcal{H}_j|}{\sqrt{n}} + 2^{5/2} 2^{-j} \sqrt{\frac{u}{n}},$$

and

$$1 - \sum_{j=1}^{\infty} \frac{1}{|H_j|} e^{-u} \geq 1 - \sum_{j=1}^{\infty} \frac{1}{2^j} e^{-u} = 1 - \left( \frac{\pi^2}{6} - 1 \right) e^{-u} \geq 1 - e^{-u}.$$

For some level $k$

$$2^{-(k+1)} \leq d(0, f) \leq 2^{-k}$$

and

$$0 = \pi_0(f) = \pi_1(f) = \cdots = \pi_k(f).$$

So

$$R(f) = \sum_{j=k+1}^{\infty} R(\pi_j(f) - \pi_{j-1}(f))$$

$$\leq \sum_{j=k+1}^{\infty} \left( \frac{2^{7/2} 2^{-j}}{\sqrt{n}} \log^{1/2} |\mathcal{H}_j| + 2^{5/2} 2^{-j} \sqrt{\frac{u}{n}} \right)$$

$$\leq \sum_{j=k+1}^{\infty} \left( \frac{2^{7/2} 2^{-j}}{\sqrt{n}} \log^{1/2} \mathcal{P}(\mathcal{H}, 2^{-j}, \rho_S) \right) + 2^{5/2} 2^{-k} \sqrt{\frac{u}{n}}.$$
Definition. A subgraph of function \( f(x) \) where \( f : \mathcal{X} \rightarrow \mathbb{R} \) is the set 
\[ \mathcal{F}_f = \{(x, t) \in \mathcal{X} \times \mathbb{R} : 0 \leq t \leq f(x) \text{ or } f(x) \leq t \leq 0\}. \]

Definition. The subgraph of a class of functions \( \mathcal{H} \) are the sets 
\[ \mathcal{F} = \{ \mathcal{F}_f : f \in \mathcal{H} \}. \]

Definition. If \( \mathcal{F} \) is a VC class of sets then \( \mathcal{H} \) is a VC subgraph class of functions 
and \( v(\mathcal{H}) = v(\mathcal{F}) \).

We now show that we can upper-bound the covering number with the empirical \( \ell_1 \) norm with a function of the VC dimension for a hypothesis spaces with finite VC dimension.

Theorem. Given a VC subgraph class \( \mathcal{H} \) where \(-1 \leq f(x) \leq 1 \) \forall f \in \mathcal{H} and \( x \in \mathcal{X} \) with \( v(\mathcal{H}) = d \) and \( \rho_S(f, g) = n^{-1} \sum_{i=1}^{n} |f(x_i) - g(x_i)| \) then 
\[ \mathcal{P}(\mathcal{H}, \varepsilon, \rho_s) \leq \left( \frac{8e}{\varepsilon \log \frac{7}{\varepsilon}} \right)^d. \]

The bound in the above theorem can be improved to 
\[ \mathcal{P}(\mathcal{H}, \varepsilon, \rho_s) \leq \left( \frac{K}{\varepsilon} \right)^d, \]
however, the proof is more complicated so we prove the weaker statement.

Proof. 
Set \( m = \mathcal{P}(\mathcal{H}, \varepsilon, \rho_s) \) so \( \{f_1, ..., f_m\} \) are \( \varepsilon \)-separated and each function \( f_k \) has its respective subgraph \( \mathcal{F}_{f_k} \).

Sample uniformly from \( \{x_1, ..., x_n\} \) \( k \) elements \( \{z_1, ..., z_k\} \) and uniformly on \([-1, 1] \) \( k \) elements \( \{t_1, ..., t_k\} \).

We now bound the probability that the subgraphs of two \( \varepsilon \)-separated functions pick out different subsets of \( \{(z_1, t_1), ..., (z_k, t_k)\} \)
\[ \mathbb{P}(\mathcal{F}_{f_k} \text{ and } \mathcal{F}_{f_l} \text{ pick out different subsets of } \{(z_1, t_1), ..., (z_k, t_k)\}) \]
\[ = \mathbb{P}(\text{at least one } (z_i, t_i) \text{ is picked out by either } \mathcal{F}_{f_k} \text{ or } \mathcal{F}_{f_l} \text{ but not the other}) \]
\[ = 1 - \mathbb{P}(\text{all } (z_i, t_i) \text{ are picked out by both or none}). \]

The probability that \( (z_i, t_i) \) is either picked out by either both \( \mathcal{F}_{f_k}, \mathcal{F}_{f_l} \) or by neither

8.8. Symmetrization and Rademacher complexities

In the previous lectures we have considered various complexity measures, such as covering numbers. But what is the right notion of complexity for the learning problem we posed? Consider the covering numbers for a moment. Take a small function class and take its convex hull. The resulting class can be extremely large. Nevertheless, the supremum of the difference of expected and empirical errors will be attained at the vertices, i.e. at the base class. In some sense, the “inside” of the class does not matter. The covering numbers take into account the whole class, and therefore become very large for the convex hull, even though the essential complexity is that of the base class. This suggests that the covering numbers are not the ideal complexity measure. In this lecture we introduce another notion (Rademacher averages), which can be claimed to be the “correct” one. In particular,
the Rademacher averages of a convex hull will be equal to those of the base class. This notion of complexity will be shown to have other nice properties.

Instead of jumping right to the definition of Rademacher Averages, we will take a longer route and show how these averages arise. Results on this topic can be found in the Theory of Empirical Processes, and so we will give some definitions from it.

Let $\mathcal{F}$ be a class of functions. Then $(Z_i)_{i \in \mathcal{I}}$ is a random process indexed by $\mathcal{F}$ if $Z_i(f)$ is a random variable for any $i$.

As before, $\mu$ is a probability measure on $\Omega$, and data $x_1, ..., x_n \sim \mu$. Then $\mu_n$ is the empirical measure supported on $x_1, ..., x_n$:

$$\mu_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}.$$  

Define $Z_i(\cdot) = (\delta_{x_i} - \mu)(\cdot)$, i.e.

$$Z_i(f) = f(x_i) - \mathbb{E}_\mu(f).$$

Then $Z_1, ..., Z_n$ is an i.i.d. process with 0 mean.

In the previous lectures we looked at the quantity

$$(8.1) \quad \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f \right|,$$

which can be written as $n \sup_{f \in \mathcal{F}} |\sum_{i=1}^{n} Z_i(f)|$.

Recall that the difficulty with (8.1) is that we do not know $\mu$ and therefore cannot calculate $\mathbb{E} f$. The classical approach of covering $\mathcal{F}$ and using the union bound is too loose.

**Proposition.** Symmetrization: If $\frac{1}{n} \sum_{i=1}^{n} f(x_i)$ is close to $\mathbb{E} f$ for data $x_1, ..., x_n$, then $\frac{1}{n} \sum_{i=1}^{n} f(x_i)$ is close to $\frac{1}{n} \sum_{i=1}^{n} f(x_i')$, the empirical average on $x_1', ..., x_n'$ (an independent copy of $x_1, ..., x_n$). Therefore, if the two empirical averages are far from each other, then empirical error is far from expected error.

Now fix one function $f$. Let $\epsilon_1, ..., \epsilon_n$ be i.i.d. Rademacher random variables (taking on values 0 or 1 with probability 1/2). Then

$$\mathbb{P} \left[ \left| \sum_{i=1}^{n} (f(x_i) - f(x_i')) \right| \geq t \right] = \mathbb{P} \left[ \left| \sum_{i=1}^{n} \epsilon_i (f(x_i) - f(x_i')) \right| \geq t \right] \leq \mathbb{P} \left[ \sum_{i=1}^{n} \epsilon_i f(x_i) \geq t/2 \right] + \mathbb{P} \left[ \sum_{i=1}^{n} \epsilon_i f(x_i') \geq t/2 \right] = 2 \mathbb{P} \left[ \sum_{i=1}^{n} \epsilon_i f(x_i) \geq t/2 \right].$$

Together with symmetrization, this suggests that controlling $\mathbb{P} \left( \left| \sum_{i=1}^{n} \epsilon_i f(x_i) \right| \geq t/2 \right)$ is enough to control $\mathbb{P} \left( \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \mathbb{E} f \right| \geq t \right)$. Of course, this is a very simple example. Can we do the same with quantities that are uniform over the class?

**Definition.** Suprema of an Empirical process:

$$Z(x_1, ..., x_n) = \sup_{f \in \mathcal{F}} \left[ \mathbb{E} f - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right].$$
Definition. Suprema of a Rademacher Process:

\[ R(x_1, \ldots, x_n, \epsilon_1, \ldots, \epsilon_n) = \sup_{f \in \mathcal{F}} \left[ \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(x_i) \right] . \]

Proposition. The expectation of the Rademacher process bounds the expectation of the empirical process:

\[ \mathbb{E} Z \leq 2 \mathbb{E} R. \]

Proof.

\[
\mathbb{E} Z = \mathbb{E}_x \sup_{f \in \mathcal{F}} \left[ \mathbb{E} f - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right] \\
= \mathbb{E}_x \sup_{f \in \mathcal{F}} \left[ \mathbb{E}_{x'} \left( \frac{1}{n} \sum_{i=1}^{n} f(x'_i) \right) - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right] \\
\leq \mathbb{E}_{x,x'} \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (f(x'_i) - f(x_i)) \\
= \mathbb{E}_{x,x',\epsilon} \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i (f(x'_i) - f(x_i)) \\
\leq \mathbb{E}_{x,x',\epsilon} \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(x'_i) + \mathbb{E}_{x,x',\epsilon} \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (-\epsilon_i) f(x_i) \\
= 2 \mathbb{E} R. \]

As we discussed previously, we would like to bound the empirical process since this will imply “generalization” for any function in \( \mathcal{F} \). We will bound \( Z \) by the Rademacher average \( \mathbb{E} R \) which we will see has some nice properties.

Theorem. If the functions in \( \mathcal{F} \) are uniformly bounded between \([a, b] \) then with probability \( 1 - e^{-u} \)

\[ Z \leq 2 \mathbb{E} R + \sqrt{\frac{2u(b-a)}{n}}. \]

Proof. The inequality involves two steps

(1) the concentration of \( Z \) around its mean \( \mathbb{E} Z \)

(2) applying the bound \( \mathbb{E} Z \leq 2 \mathbb{E} R \)

We will use McDiarmid’s inequality for the first step. We define the following two variables \( Z := Z(x_1, \ldots, x_i, \ldots, x_n) \) and \( Z^i := Z(x_1, \ldots, x_i, \ldots, x_n) \). Since \( a \leq f(x) \leq b \) for all \( x \) and \( f \in \mathcal{F} \):

\[
|Z^i - Z| = \left| \sup_{f \in \mathcal{F}} |\mathbb{E} f - n^{-1} \sum_{j=1}^{n} f(x_j) + (n^{-1} f(x_i) - n^{-1} f(x'_i))| - \sup_{f \in \mathcal{F}} |\mathbb{E} f - n^{-1} \sum_{j=1}^{n} f(x_j)| \right| \\
\leq \sup_{f \in \mathcal{F}} \frac{1}{n} |f(x_i) - f(x'_i)| \leq \frac{b-a}{n} = c_i.
\]

1The quantity \( \mathbb{E} R \) is called a Rademacher average.
This bounds the Martingale difference for the empirical process. Given the difference bound McDiarmid’s inequality states

\[ \Pr(\mathbf{Z} - \mathbb{E}\mathbf{Z} > t) \leq \exp\left(\frac{-t^2}{\sum_{i=1}^{n} \frac{(b-a)^2}{n^2}}\right) = \exp\left(\frac{-nt^2}{2(b-a)^2}\right) . \]

Therefore, with probability at least \(1 - \epsilon^{-u}\),

\[ Z - \mathbb{E}Z < \sqrt{\frac{2u(b-a)}{n}} . \]

So as the number of samples, \(n\), grows, \(Z\) becomes more and more concentrated around \(\mathbb{E}Z\).

Applying symmetrization proves the theorem. With probability at least \(1 - \epsilon^{-u}\).

\[ Z \leq \mathbb{E}Z + \sqrt{\frac{2u(b-a)}{n}} \leq 2\mathbb{E}R + \sqrt{\frac{2u(b-a)}{n}} . \quad \square \]

McDiarmid’s inequality does not incorporate a notion of variance so it is possible to obtain a sharper inequality using see Talagrand’s inequality for the suprema of empirical processes.

We are now left with bounding the Rademacher average. Implicit in the previous lecture on on Kolmogorov chaining was such a bound. Before we restate that result and give some examples we state some nice and useful properties of Rademacher averages.

**Properties.** Let \(\mathcal{F}, \mathcal{G}\) be classes of real-valued functions. Then for any \(n\),

1. If \(\mathcal{F} \subseteq \mathcal{G}\), then \(\mathbb{E}R(\mathcal{F}) \leq \mathbb{E}R(\mathcal{G})\)
2. \(\mathbb{E}R(\mathcal{F}) = \mathbb{E}R(\text{conv}\mathcal{F})\)
3. \(\forall c \in \mathbb{R}, \mathbb{E}R(c\mathcal{F}) = |c|\mathbb{E}R(\mathcal{F})\)
4. If \(\phi : \mathbb{R} \rightarrow \mathbb{R}\) is \(L\)-Lipschitz and \(\phi(0) = 0\), then \(\mathbb{E}R(\phi(\mathcal{F})) \leq 2L\mathbb{E}R(\mathcal{F})\)
5. For RKHS balls, \(c(\sum_{i=1}^{\infty} \lambda_i)^{1/2} \leq \mathbb{E}R(\mathcal{F}_k) \leq C(\sum_{i=1}^{\infty} \lambda_i)^{1/2}\), where \(\lambda_i\)'s are eigenvalues of the corresponding linear operator in the RKHS.

**Theorem.** The Rademacher average is bounded by Dudley’s entropy integral

\[ \mathbb{E}_\epsilon R \leq c \frac{1}{\sqrt{n}} \int_0^D \sqrt{\log N(\epsilon, \mathcal{F}, L_2(\mu_n))} d\epsilon, \]

where \(N\) denotes the covering number.

**Example.** Let \(\mathcal{F}\) be a class with finite VC-dimension \(V\). Then

\[ N(\epsilon, \mathcal{F}, L_2(\mu_n)) \leq \left( \frac{2}{\epsilon} \right)^{kV} , \]

for some constant \(k\). The entropy integral above is bounded as

\[ \int_0^1 \sqrt{\log N(\epsilon, \mathcal{F}, L_2(\mu_n))} d\epsilon \leq \int_0^1 \sqrt{kV\log 2/\epsilon} d\epsilon \]
\[ \leq k'\sqrt{V} \int_0^1 \sqrt{\log 2/\epsilon} d\epsilon \leq k\sqrt{V} . \]

Therefore, \(\mathbb{E}_\epsilon R \leq k\sqrt{\frac{V}{n}}\) for some constant \(k\).
LECTURE 9
Generalization bounds

9.1. Generalization bounds and stability

We define an algorithm $\mathcal{A}$ to be a mapping from a training set $S = \{z_1, \ldots, z_n\}$ to a function $f_S$. Here, $z_i := (x_i, y_i)$.

How do we measure the performance of $\mathcal{A}$? First, we introduce a loss function $V$, so that $V(f(x), y)$ measures the penalty of predicting the value $f(x)$ at point $x$ while the true value is $y$. One goal of learning is to minimize the expected error at a new point:

$$I[f] := \mathbb{E}_{(x,y)}[V(f(x), y)] = \int V(f(x), y) d\mu(x, y).$$

Unfortunately, we cannot compute the above quantity because the measure $\mu$ is unknown. We therefore try to upper bound it. The natural approach is to approximate the expectation by the average (empirical error) on the given sample $S$:

$$I_S[f] = \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i).$$

Generalization bounds are probabilistic bounds on the difference $I[f] - I_S[f]$.

But how far is $I_S[f]$ from $I[f]$? For a fixed function $f$, the difference between these two quantities is small (law of large numbers). What if the algorithm chooses different $f$? Then $f$ is itself a random function, as denoted by $f_S$, and in general we cannot control the difference $I_S[f_S] - I[f_S]$.

The classical uniform generalization bounds use some notion of the “size” of the function class and hold for all functions in the class:

$$\mathbb{P}_S\{\sup_{f \in \mathcal{F}} |I[f] - I_S[f]| > \epsilon\} \leq \phi(\mathcal{F}, n, \epsilon).$$

It doesn’t matter what the algorithm is doing because these bounds give a guarantee for all of them at the same time! These types of bounds ignore the fact that we are dealing with a specific algorithm. The only knowledge about the algorithm used in these types of bounds is “functions are chosen from a fixed

---

1The true and empirical risks are denoted in Bousquet & Elisseeff as $R(\mathcal{A}, S)$ and $\hat{R}(\mathcal{A}, S)$, respectively, to emphasize the algorithm that produced $f_S$. 89
function class $\mathcal{F}$. Once the function class is fixed, no matter what algorithm we are using, uniform bounds would provide the same bound on the generalization error.

Now, imagine that the algorithm is actually outputting only one function. The generalization bound then follows, as mentioned above, from the law of large numbers, and is tighter than the uniform bound over a large function class. Of course, this is an extreme example, but the main message is: to get better bounds on the performance of the algorithm, one should use the knowledge about the specific algorithm.

Such knowledge can come in different forms. One useful (as we will see now) assumption is “algorithmic stability”. It turns out that if the function output by algorithm does not change much when one training point is perturbed, generalization bounds follow quite easily.

9.1.1. Uniform stability

In this section we introduce the notion of Uniform Stability and show how generalization bounds can be obtained for uniformly stable algorithms. In the next section we will show that Tikhonov regularization algorithm exhibits such stability, and therefore we get guarantees on its performance.

We assume that $A$ is deterministic, and that $A$ does not depend on the ordering of the points in the training set.

Define $D[S] = I[f_S] - I[f_S]$, the defect, or generalization error. It measures the discrepancy between the expected loss and the empirical estimate. Since we can measure $I[f_S]$, bounds on $D[S]$ can be translated into bounds on $I[f_S]$. We would like to show that with high probability $D[S]$ is small. Then if we can observe that $I[f_S]$ is small, it will follow that $I[f_S]$ must be small. Note that in this approach we are not concerned with “good performance” of all functions, but only the one produced by our algorithm:

$$\mathbb{P}_S \left( |I[f_S] - I[f_S]| > \epsilon \right) < \delta$$

Given a training set $S$, we define $S^{i,z}$ to be the new training set obtained when point $i$ of $S$ is replaced by the new point $z \in Z$. We will overload the notation by writing the loss function as $V(f, z)$ instead of $V(f(x), y)$, where $z = (x, y)$. To obtain the results of this section, we also need to assume that the loss function $V$ is positive and upper-bounded by $M$.

**Definition.** We say that an algorithm $A$ has uniform stability $\beta$ (is $\beta$-stable) if

$$\forall (S, z) \in Z^{n+1}, \forall i, \sup_u |V(f_S, u) - V(f_{S^{-i}}, u)| \leq \beta.$$ 

An algorithm is $\beta$-stable if, for any possible training set, we can replace an arbitrary training point with any other possible training point, and the loss at any point will change by no more than $\beta$. Uniform stability is a strong requirement because it ignores the fact that the points are drawn from a probability distribution. For uniform stability, the function still has to change very little even when a very unlikely (“bad”) training set is drawn.

In general, the stability $\beta$ is a function of $n$, and should perhaps be written $\beta_n$.

Given that an algorithm $A$ has stability $\beta$, how can we get bounds on its performance? The answer is: we will use Concentration Inequalities, introduced in
the previous lectures. In particular, McDiarmid’s Inequality will prove to be useful for obtaining exponential bounds on the defect \( D[S] \).

Recall McDiarmid’s Inequality:

Given random variables \( \{ z_1, \ldots, z_n \} = S \), and a function \( F: z^n \rightarrow \mathbb{R} \) satisfying

\[
\sup_{z_1, \ldots, z_n, z_i'} |F(z_1, \ldots, z_n) - F(z_1, \ldots, z_i', z_{i+1}, \ldots, z_n)| \leq c_i,
\]

the following statement holds:

\[
\Pr \left( |F(z_1, \ldots, z_n) - \mathbb{E}_S(F(z_1, \ldots, z_n))| > \epsilon \right) \leq 2 \exp \left( -\frac{2\epsilon^2}{\sum_{i=1}^{n} c_i^2} \right).
\]

We will apply McDiarmid’s inequality to the function \( F(z_1, \ldots, z_n) = D[S] = I[f_S] - I_S[f_S] \). We will show two things: that \( D[S] \) is close to its expectation \( \mathbb{E}_S D[S] \), and that the expectation is small. Both of these will follow from uniform \( \beta \)-stability.

\[
\mathbb{E}_S D[f_S] = \mathbb{E}_S [I_S[f_S] - I[f_S]]
\]

\[
= \mathbb{E}_{S, z} \left[ \frac{1}{n} \sum_{i=1}^{n} V(f_S(x_i), y_i) - V(f_S(x), y) \right]
\]

\[
= \mathbb{E}_{S, z} \left[ \frac{1}{n} \sum_{i=1}^{n} V(f_{S \cdot i}(x), y) - V(f_S(x), y) \right] \leq \beta.
\]

The second equality follows by exploiting the “symmetry” of expectation: The expected value of a training set on a training point doesn’t change when we “rename” the points.

Now we need to show that the requirements of McDiarmid’s theorem are satisfied:

\[
|D[f_S] - D[f_{S \cdot i}]| = |I_S[f_S] - I[f_S] - I_{S \cdot i}[f_S] + I_{S \cdot i}[f_{S \cdot i}]| \leq |I[f_S] - I[f_{S \cdot i}]| + |I_S[f_S] - I_{S \cdot i}[f_{S \cdot i}]| \leq \beta + \frac{1}{n} |V(f_S(x_i), y_i) - V(f_{S \cdot i}(x), y)| + \frac{1}{n} \sum_{j \neq i} |V(f_S(x_j), y_j) - V(f_{S \cdot i}(x_j), y_j)| \leq \beta + \frac{M}{n} + \beta \leq 2 \beta + \frac{M}{n}.
\]

By McDiarmid’s Inequality, for any \( \epsilon \),

\[
\Pr \left( |D[f_S] - \mathbb{E}D[f_S]| > \epsilon \right) \leq 2 \exp \left( -\frac{2\epsilon^2}{\sum_{i=1}^{n} (2(\beta + \frac{M}{n}))^2} \right) = 2 \exp \left( -\frac{\epsilon^2}{2n(\beta + \frac{M}{n})^2} \right) = 2 \exp \left( -\frac{n\epsilon^2}{2(n\beta + M)^2} \right).
\]
Note that
\[ \mathbb{P}(D[f_S] > \beta + \epsilon) = \mathbb{P}(D[f_S] - \mathbb{E}[D[f_S]] > \epsilon) \]
\[ \leq \mathbb{P}(|D[f_S] - \mathbb{E}[D[f_S]]| > \epsilon) \]
Hence,
\[ \mathbb{P}(I_S[f_S] - I[f_S] > \beta + \epsilon) \leq 2 \exp \left( -\frac{n\epsilon^2}{2(n\beta + M)^2} \right) \]
If we define
\[ \delta := 2 \exp \left( -\frac{n\epsilon^2}{2(n\beta + M)^2} \right) \]
Solving for \( \epsilon \) in terms of \( \delta \), we find that
\[ \epsilon = (n\beta + M) \sqrt{\frac{2\ln(2/\delta)}{n}}. \]
By varying \( \delta \) (and \( \epsilon \)), we can say that for any \( \delta \in (0, 1) \), with probability \( 1 - \delta \),
\[ I_S[f_S] \leq I_S[f_S] + \beta + (n\beta + M) \sqrt{\frac{2\ln(2/\delta)}{n}}. \]
Note that if \( \beta = \frac{k}{n} \) for some \( k \), we can restate our bounds as
\[ P \left( |I[f_S] - I_S[f_S]| \geq \frac{k}{n} + \epsilon \right) \leq 2 \exp \left( -\frac{n\epsilon^2}{2(k + M)^2} \right), \]
and with probability \( 1 - \delta \),
\[ I[f_S] \leq I_S[f_S] + \frac{k}{n} + (2k + M) \sqrt{\frac{2\ln(2/\delta)}{n}}. \]
What is the best rate of convergence that can be achieved with this method? Obviously, the best possible stability would be \( \beta = 0 \) — the function can’t change at all when you change the training set. An algorithm that always picks the same function, regardless of its training set, is maximally stable and has \( \beta = 0 \). Using \( \beta = 0 \) in the last bound, with probability \( 1 - \delta \),
\[ I[f_S] \leq I_S[f_S] + M \sqrt{\frac{2\ln(2/\delta)}{n}}. \]
The convergence is still \( O \left( \frac{1}{\sqrt{n}} \right) \). So once \( \beta = O \left( \frac{1}{n} \right) \), further increases in stability don’t change the rate of convergence.

With only relatively few papers on stability in Machine Learning, there is a proliferation of names and kinds of stabilities, which can cause lots of confusion. Here we mention a few to give a taste of kinds of perturbations considered in the literature.

**Definition.** An algorithm is \((\beta, \delta)\) error stable if
\[ \forall S, u, \forall i, |I[f_S] - I[f_{S^i,u}]| \leq \beta. \]

**Definition.** An algorithm is \( \beta \) \( L_1 \)-stable at \( S \):
\[ \forall i, u \in \mathcal{Z}, \mathbb{E}_z ||V(f_S, z) - V(f_{S^i,u}, z)|| \leq \beta. \]

**Definition.** An algorithm is \((\beta, \delta)\) cross-validation stable:
\[ \forall S, u \in \mathcal{Z}^{n+1}, \forall i, |V(f_S, u) - V(f_{S^i,u}, u)| \leq \beta. \]
We have used McDiarmid’s inequality to prove a generalization bound for a uniformly $\beta$-stable algorithm. Note that this bound cannot tell us that the expected error will be low a priori, it can only tell us that with high probability, the expected error will be close to the empirical error. We have to actually observe a low empirical error to conclude that we have a low expected error.

Uniform stability of $O(\frac{1}{n})$ seems to be a strong requirement. In the next section, we will show that Tikhonov regularization possesses this property.

### 9.2. Uniform stability of Tikhonov regularization

Recall the definition of Tikhonov Regularization:

$$f^*_S = \arg \min_{f \in \mathcal{H}} \left[ n^{-1} \sum_{i=1}^n V(f(x_i), y_i) + \lambda \|f\|^2_K \right],$$

where $\mathcal{H}$ is an RKHS with kernel $K$.

**Proposition.** Under mild conditions Tikhonov regularization is is uniformly stable with

$$\beta = \frac{L^2 \kappa^2}{\lambda n},$$

where the constants $L$ and $\kappa$ will be defined with respect to the mild conditions.

Given the above stability condition and the results from the previous section, we have the following bound:

$$\mathbb{P} \left( |I[f^*_S] - I_S[f^*_S]| \geq \frac{k}{n} + \epsilon \right) \leq 2 \exp \left( -\frac{n\epsilon^2}{2(k + M)^2} \right).$$

Equivalently, with probability $1 - \delta$,

$$I[f^*_S] \leq I_S[f^*_S] + \frac{k}{n} + (2k + M)\sqrt{\frac{2 \ln(2/\delta)}{n}}.$$

We now prove the proposition and state the mild conditions.

**Proof.**

The first condition is on the loss function:

**Definition.** A loss function (over a possibly bounded domain $\mathcal{X}$) is Lipschitz with Lipschitz constant $L$ if

$$\forall y_1, y_2, y' \in \mathcal{Y}, \ |V(y_1, y') - V(y_2, y')| \leq L|y_1 - y_2|.$$

Put differently, if we have two functions $f_1$ and $f_2$, under an $L$-Lipschitz loss function,

$$\sup_{(x, y)} |V(f_1(x), y) - V(f_2(x), y)| \leq L|f_1(x) - f_2(x)|.$$

Yet another way to write it:

$$|V(f_1, \cdot) - V(f_2, \cdot)|_\infty \leq L|f_1(\cdot) - f_2(\cdot)|_\infty.$$

If a loss function is $L$-Lipschitz, then closeness of two functions (in $L_\infty$ norm) implies that they are close in loss. The converse is false — it is possible for the difference in loss of two functions to be small, yet the functions to be far apart (in $L_\infty$):

**Example.** Consider constant loss $V = \text{const}$. The difference of losses of any two functions is zero while the functions can be far apart from each other.
The hinge loss and the \( \epsilon \)-insensitive loss are both \( L \)-Lipschitz with \( L = 1 \). The square loss function is \( L \) Lipschitz if we can bound the \( y \) values and the \( f(x) \) values generated. The \( 0 - 1 \) loss function is not \( L \)-Lipschitz at all — an arbitrarily small change in the function can change the loss by 1:

\[
f_1 = 0, \ f_2 = \epsilon, \ V(f_1(x), 0) = 0, \ V(f_2(x), 0) = 1.
\]

Assuming \( L \)-Lipschitz loss, we have transformed the problem of bounding

\[
\sup_{u \in Z} |V(f_S, u) - V(f_{S^{i,z}}, u)|
\]

into a problem of bounding

\[
|f_S - f_{S^{i,z}}|_{\infty}.
\]

As the next step, we bound the \( L_{\infty} \) norm by the norm in the RKHS associated with the kernel \( K \). We now impose the second condition: there exists a \( \kappa \) satisfying

\[
\forall x \in \mathcal{X}, \sqrt{K(x,x)} \leq \kappa.
\]

Under the above assumption, using the reproducing property and the Cauchy-Schwartz inequality, we can derive the following:

\[
\forall x \quad |f(x)| = |(K(x,\cdot), f(\cdot))_K| \\
\leq ||K(x,\cdot)||_K ||f||_K \\
= \sqrt{(K(x,\cdot), K(x,\cdot))_K} ||f||_K \\
= \sqrt{K(x,x)} ||f||_K \\
\leq \kappa ||f||_K
\]

Since above inequality holds for all \( x \), we have \( |f|_\infty \leq ||f||_K \). Hence, if we can bound the RKHS norm, we can bound the \( L_{\infty} \) norm. We have now reduced the problem to bounding \( ||f_S - f_{S^{i,z}}||_K \).

**Lemma.** For Tikhonov regularization under the above mild conditions

\[
||f_S - f_{S^{i,z}}||_K^2 \leq \frac{L||f_S - f_{S^{i,z}}||_{\infty}}{\lambda n}.
\]

This lemma says that when we replace a point in the training set, the change in the RKHS norm (squared) of the difference between the two functions cannot be too large compared to the change in \( L_{\infty} \). Using the lemma and the relation between \( L_K \) and \( L_{\infty} \),

\[
||f_S - f_{S^{i,z}}||_K^2 \leq \frac{L||f_S - f_{S^{i,z}}||_{\infty}}{\lambda n} \\
\leq \frac{L\kappa ||f_S - f_{S^{i,z}}||_K}{\lambda n}
\]

Dividing through by \( ||f_S - f_{S^{i,z}}||_K \), we derive

\[
||f_S - f_{S^{i,z}}||_K \leq \frac{\kappa L}{\lambda n}.
\]
Using again the relationship between $L_K$ and $L_\infty$, and the $L$ Lipschitz condition,

\[
\sup |V(f^*_S(\cdot), \cdot) - V(f^*_{S^z}(\cdot), \cdot)| \leq L|f^*_S - f^*_{S^z}|_\infty
\leq L\kappa |f^*_S - f^*_{S^z}|_K
\leq \frac{L^2\kappa^2}{\lambda n}
= \beta.
\]

We now prove lemma 9.2.

Proof. The proof involves comparing norms of $f^*_S$ and $f^*_{S^z}$ and uses the notion of divergences.

Suppose we have a convex, differentiable function $F$, and we know $F(f_1)$ for some $f_1$. We can “guess” $F(f_2)$ by considering a linear approximation to $F$ at $f_1$:

\[
\hat{F}(f_2) = F(f_1) + (f_2 - f_1, \nabla F(f_1)).
\]

The Bregman divergence is the error in this linearized approximation:

\[
d_F(f_2, f_1) = F(f_2) - F(f_1) - (f_2 - f_1, \nabla F(f_1)).
\]

We will need the following key facts about divergences:

- $d_F(f_2, f_1) \geq 0$
- If $f_1$ minimizes $F$, then the gradient is zero, and $d_F(f_2, f_1) = F(f_2) - F(f_1)$.
- If $F = A + B$, where $A$ and $B$ are also convex and differentiable, then $d_F(f_2, f_1) = d_A(f_2, f_1) + d_B(f_2, f_1)$ (the derivatives add).

Consider the Tikhonov functional

\[
T_S(f) = \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda \|f\|_K^2,
\]

as well as the component functionals

\[
V_S(f) = \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)
\]
and

\[ N(f) = \|f\|_K^2. \]

Hence, \( T_S(f) = V_S(f) + \lambda N(f) \). If the loss function is convex (in the first variable), then all three functionals are convex.

This picture illustrates the relation between the various functionals on the minimizers over the original set \( S \) and the perturbed set \( S^{i,z} = S' \), where \((x_i, y_i)\) is replaced by a new point \( z = (x, y) \). Let \( f_S \) be the minimizer of \( T_S \), and let \( f_{S^{i,z}} \) be the minimizer of \( T_{S^{i,z}} \). Then

\[
\lambda (d_N(f_{S^{i,z}}, f_S) + d_N(f_S, f_{S^{i,z}})) \leq \\
d_{T_S}(f_{S^{i,z}}, f_S) + d_{T_{S^{i,z}}}(f_S, f_{S^{i,z}}) = \\
\frac{1}{n} (V(f_{S^{i,z}}, z_i) - V(f_S, z_i) + V(f_S, z) - V(f_{S^{i,z}}, z)) \leq \\
\frac{2L|f_S - f_{S^{i,z}}|_\infty}{n}.
\]

We conclude that

\[
d_N(f_{S^{i,z}}, f_S) + d_N(f_S, f_{S^{i,z}}) \leq \frac{2L|f_S - f_{S^{i,z}}|_\infty}{\lambda n}.
\]

But what is \( d_N(f_{S^{i,z}}, f_S) \)? Let’s express our functions as the sum of orthogonal eigenfunctions in the RKHS:

\[
f_S(x) = \sum_{n=1}^{\infty} c_n \phi_n(x)
\]

\[
f_{S^{i,z}}(x) = \sum_{n=1}^{\infty} c'_n \phi_n(x)
\]
Once we express a function in this form, we recall that
\[ \|f\|_K^2 = \sum_{n=1}^{\infty} \frac{c_n^2}{\lambda_n} \]

Using this notation, we reexpress the divergence in terms of the \(c_i\) and \(c'_i\):
\[ d_N(f_{S^{i,z}}, f_S) = ||f_{S^{i,z}}||_K^2 - ||f_S||_K^2 - \langle f_{S^{i,z}} - f_S, \nabla ||f_S||_K^2 \rangle \]
\[ = \sum_{n=1}^{\infty} \frac{c_n'^2}{\lambda_n} - \sum_{n=1}^{\infty} \frac{c_n^2}{\lambda_n} - \sum_{i=1}^{\infty} (c'_n - c_n)(\frac{2c_n}{\lambda_n}) \]
\[ = \sum_{n=1}^{\infty} \frac{c_n'^2 + c_n^2 - 2c'_n c_n}{\lambda_n} \]
\[ = \sum_{n=1}^{\infty} \frac{(c'_n - c_n)^2}{\lambda_n} \]
\[ = ||f_{S^{i,z}} - f_S||_K^2 \]

We conclude that
\[ d_N(f_{S^{i,z}}, f_S) + d_N(f_S, f_{S^{i,z}}) = 2||f_{S^{i,z}} - f_S||_K^2 \]

Combining these results proves our Lemma:
\[ ||f_{S^{i,z}} - f_S||_K^2 = \frac{d_N(f_{S^{i,z}}, f_S) + d_N(f_S, f_{S^{i,z}})}{2} \leq \frac{2L|f_S - f_{S^{i,z}}|_{\infty}}{\lambda n} \]

We have shown that Tikhonov regularization with an \(L\)-Lipschitz loss is \(\beta\)-stable with \(\beta = \frac{L^2}{2\kappa^2}\). If we want to actually apply the theorems and get the generalization bound, we need to bound the loss (note that this was a requirement in the previous section). So, is the loss bounded?

Let \(C_0\) be the maximum value of the loss when we predict a value of zero. If we have bounds on \(X\) and \(Y\), we can find \(C_0\). Noting that the “all 0” function \(\tilde{0}\) is always in the RKHS, we see that
\[ \lambda ||f_S||_K^2 \leq T(f_S) \leq T(\tilde{0}) \]
\[ = \frac{1}{n} \sum_{i=1}^{n} V(\tilde{0}(x_i), y_i) \]
\[ \leq C_0. \]

Therefore,
\[ ||f_S||_K^2 \leq \frac{C_0}{\lambda} \]
\[ \Rightarrow |f_S|_{\infty} \leq \kappa ||f_S||_K \leq \kappa \sqrt{\frac{C_0}{\lambda}} \]

Since the loss is \(L\)-Lipschitz, a bound on \(|f_S|_{\infty}\) implies boundedness of the loss function.

A few final notes:
• if we keep $\lambda$ fixed as we increase $n$, the generalization bound will tighten as $O\left(\frac{1}{\sqrt{n}}\right)$. However, keeping $\lambda$ fixed is equivalent to keeping our hypothesis space fixed. As we get more data, we want $\lambda$ to get smaller. If $\lambda$ gets smaller too fast, the bounds become trivial.

• It is worth noting that Ivanov regularization

$$\hat{f}_{H,S} = \arg\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} V(f(x_i), y_i)$$

s.t. $\|f\|_{K}^2 \leq \tau$

is not uniformly stable with $\beta = O\left(\frac{1}{n}\right)$. This is an important distinction between Tikhonov and Ivanov regularization.
BIBLIOGRAPHY