Introduction to Computational Learning Theory (PAC)
Learnability and the VC dimension
Sample Compression Schemes
Learning with Membership Queries
(Computational) Hardness of Learning
This Mini-Course

Statistical Learning Theory Framework
Capacity Measures : Rademacher Complexity
Uniform Convergence : Generalisation Bounds
Some Machine Learning Techniques
Algorithmic Stability to prove Generalisation
Outline

Statistical (Supervised) Learning Theory Framework

Linear Regression

Rademacher Complexity

Support Vector Machines

Kernels

Neural Networks

Algorithmic Stability
Input space: $\mathcal{X}$ (most often $\mathcal{X} \subset \mathbb{R}^n$)
Input space : $\mathcal{X}$ (most often $\mathcal{X} \subset \mathbb{R}^n$)

Target values : $\mathcal{Y}$

- $\mathcal{Y} = \{-1, 1\}$ : binary classification
- $\mathcal{Y} = \mathbb{R}$ : regression
Input space: $\mathcal{X}$ (most often $\mathcal{X} \subset \mathbb{R}^n$)

Target values: $\mathcal{Y}$

- $\mathcal{Y} = \{-1, 1\}$: binary classification
- $\mathcal{Y} = \mathbb{R}$: regression

We consider data to be generated from a joint distribution $D$ over $\mathcal{X} \times \mathcal{Y}$
Input space: $\mathcal{X}$ (most often $\mathcal{X} \subset \mathbb{R}^n$)

Target values: $\mathcal{Y}$
- $\mathcal{Y} = \{-1, 1\}$: binary classification
- $\mathcal{Y} = \mathbb{R}$: regression

We consider data to be generated from a joint distribution $D$ over $\mathcal{X} \times \mathcal{Y}$.

Sometimes convenient to factorise: $D(x, y) = D(x)D(y|x)$
Statistical (Supervised) Learning Theory Framework

Input space: \( \mathcal{X} \) (most often \( \mathcal{X} \subset \mathbb{R}^n \))

Target values: \( \mathcal{Y} \)

- \( \mathcal{Y} = \{-1, 1\} \): binary classification
- \( \mathcal{Y} = \mathbb{R} \): regression

We consider data to be generated from a joint distribution \( D \) over \( \mathcal{X} \times \mathcal{Y} \)

Sometimes convenient to factorise: \( D(x, y) = D(x)D(y|x) \)

Make no assumptions about a specific functional relationship between \( x \) and \( y \), a.k.a. agnostic setting \(^{10,13,16}\)
Statistical (Supervised) Learning Theory Framework

Input space: $\mathcal{X}$, target values: $\mathcal{Y}$

Arbitrary data distribution $D$ over $\mathcal{X} \times \mathcal{Y}$ (agnostic setting)
How can we fit a function to the data?

- Classical approach to function approximation: polynomials, trigonometric functions, universality theorems
- These suffer from the curse of dimensionality
Input space: $\mathcal{X}$, target values: $\mathcal{Y}$

Arbitrary data distribution $D$ over $\mathcal{X} \times \mathcal{Y}$ (agnostic setting)

How can we fit a function to the data?

- Classical approach to function approximation: polynomials, trigonometric functions, universality theorems
- These suffer from the curse of dimensionality

Finding any function that fits the observed data may perform arbitrarily badly on unseen points leading to overfitting
Input space: $\mathcal{X}$, target values: $\mathcal{Y}$

Arbitrary data distribution $D$ over $\mathcal{X} \times \mathcal{Y}$ (agnostic setting)

How can we fit a function to the data?

- Classical approach to function approximation: polynomials, trigonometric functions, universality theorems
- These suffer from the curse of dimensionality

Finding any function that fits the observed data may perform arbitrarily badly on unseen points leading to overfitting

We will focus on fitting functions from a class of functions whose “complexity” or “capacity” is bounded
Aside: Connections to classical Statistics/ML

Attempt to explicitly model the distributions $D(x)$ and/or $D(y|x)$
Aside: Connections to classical Statistics/ML

Attempt to explicitly model the distributions $D(x)$ and/or $D(y|x)$

**Generative Models:** Model the full joint distribution $D(x, y)$

- Gaussian Discriminant Analysis, Naïve Bayes
Aside: Connections to classical Statistics/ML

Attempt to explicitly model the distributions $D(x)$ and/or $D(y|x)$

**Generative Models:** Model the full joint distribution $D(x, y)$
- Gaussian Discriminant Analysis, Naïve Bayes

**Discriminative Models:** Model only the conditional distribution $D(y|x)$
- Linear Regression: $y| w_0, w, x \sim w_0 + w \cdot x + \mathcal{N}(0, \sigma^2)$
- Classification: $y| w_0, w, x \sim 2 \cdot \text{Bernoulli}(\text{sigmoid}(w_0 + w \cdot x)) - 1$
Aside: Connections to classical Statistics/ML

Attempt to explicitly model the distributions $D(x)$ and/or $D(y|x)$

**Generative Models:** Model the full joint distribution $D(x, y)$
- Gaussian Discriminant Analysis, Naïve Bayes

**Discriminative Models:** Model only the conditional distribution $D(y|x)$
- Linear Regression: $y|w_0, w, x \sim w_0 + w \cdot x + \mathcal{N}(0, \sigma^2)$
- Classification: $y|w_0, w, x \sim 2 \cdot \text{Bernoulli}(\text{sigmoid}(w_0 + w \cdot x)) - 1$

The (basic) PAC model in CLT assumes a functional form, $y = c(x)$, for some concept $c$ in class $C$, and the VC dimension of $C$ controls learnability.
Statistical (Supervised) Learning Theory Framework

\[ \mathcal{X} \text{ instance space; } \mathcal{Y} \text{ target values} \]

Distribution \( D \) over \( \mathcal{X} \times \mathcal{Y} \)
Statistical (Supervised) Learning Theory Framework

Let $\mathcal{F} \subset \mathcal{Y}^\mathcal{X}$ be a class of functions. A learning algorithm will output some function from the class $\mathcal{F}$.
Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be a class of functions. A learning algorithm will output some function from the class $\mathcal{F}$.

A **cost function** $\gamma : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$

- E.g. $\mathcal{Y} = \{-1, 1\}$, $\gamma(y', y) = \mathbb{I}(y' \neq y)$
- E.g. $\mathcal{Y} = \mathbb{R}$, $\gamma(y', y) = |y' - y|^p$ for $p \geq 1$
Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be a class of functions. A learning algorithm will output some function from the class $\mathcal{F}$.

A cost function $\gamma : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$

- E.g. $\mathcal{Y} = \{-1, 1\}$, $\gamma(y', y) = \mathbb{I}(y' \neq y)$
- E.g. $\mathcal{Y} = \mathbb{R}$, $\gamma(y', y) = |y' - y|^p$ for $p \geq 1$

The loss for $f \in \mathcal{F}$ at point $(x, y)$ is given by

$$\ell(f; x, y) = \gamma(f(x), y)$$
Statistical (Supervised) Learning Theory Framework

\( \mathcal{X} \) instance space; \( \mathcal{Y} \) target values

Distribution \( D \) over \( \mathcal{X} \times \mathcal{Y} \)

Let \( \mathcal{F} \subseteq \mathcal{Y}^\mathcal{X} \) be a class of functions. A learning algorithm will output some function from the class \( \mathcal{F} \).

A cost function \( \gamma : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+ \)
- E.g. \( \mathcal{Y} = \{ -1, 1 \} \), \( \gamma(y', y) = \mathbb{I}(y' \neq y) \)
- E.g. \( \mathcal{Y} = \mathbb{R} \), \( \gamma(y', y) = |y' - y|^p \) for \( p \geq 1 \)

The loss for \( f \in \mathcal{F} \) at point \( (x, y) \) is given by

\[
\ell(f; x, y) = \gamma(f(x), y)
\]

The Risk functional \( R : \mathcal{F} \to \mathbb{R}^+ \) is given by:

\[
R(f) = \mathbb{E}_{(x,y) \sim D} \left[ \ell(f; x, y) \right] = \mathbb{E}_{(x,y) \sim D} \left[ \gamma(f(x), y) \right]
\]
The Risk functional $R : \mathcal{F} \rightarrow \mathbb{R}^+$ is given by:

$$R(f) = \mathbb{E}_{(x, y) \sim D} \left[ \ell(f; x, y) \right] = \mathbb{E}_{(x, y) \sim D} \left[ \gamma(f(x), y) \right]$$
The Risk functional $R : \mathcal{F} \rightarrow \mathbb{R}^+$ is given by:

$$R(f) = \mathbb{E}_{(x,y) \sim D} [\ell(f; x, y)] = \mathbb{E}_{(x,y) \sim D} [\gamma(f(x), y)]$$

Would like to find $f \in \mathcal{F}$ that “minimises” the risk $R$
The Risk functional $R : \mathcal{F} \to \mathbb{R}^+$ is given by:

$$R(f) = \mathbb{E}_{(x,y) \sim D}[\ell(f; x, y)] = \mathbb{E}_{(x,y) \sim D}[\gamma(f(x), y)]$$

Would like to find $f \in \mathcal{F}$ that "minimises" the risk $R$

Even calculating (let alone minimising) the risk is essentially impossible in most cases of interest
Statistical (Supervised) Learning Theory Framework

The Risk functional $R : \mathcal{F} \rightarrow \mathbb{R}^+$ is given by:

$$R(f) = \mathbb{E}_{(x,y) \sim D} [\ell(f; x, y)] = \mathbb{E}_{(x,y) \sim D} [\gamma(f(x), y)]$$

Would like to find $f \in \mathcal{F}$ that “minimises” the risk $R$

Even calculating (let alone minimising) the risk is essentially impossible in most cases of interest

Only have access to $D$ through a sample of size $m$ drawn from $D$ called the training data
Statistical (Supervised) Learning Theory Framework

The Risk functional $R : \mathcal{F} \rightarrow \mathbb{R}^+$ is given by:

$$R(f) = \mathbb{E}_{(x, y) \sim D} [\ell(f; x, y)] = \mathbb{E}_{(x, y) \sim D} [\gamma(f(x), y)]$$

Would like to find $f \in \mathcal{F}$ that "minimises" the risk $R$

Even calculating (let alone minimising) the risk is essentially impossible in most cases of interest

Only have access to $D$ through a sample of size $m$ drawn from $D$ called the training data

Throughout the talk, $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ a sample of size $m$ drawn i.i.d. (independent and identically distributed) from $D$
The Risk functional $R : \mathcal{F} \to \mathbb{R}^+$ is given by:

$$R(f) = \mathbb{E}_{(x,y) \sim D} [\ell(f; x, y)] = \mathbb{E}_{(x,y) \sim D} [\gamma(f(x), y)]$$

Would like to find $f \in \mathcal{F}$ that "minimises" the risk $R$

Even calculating (let alone minimising) the risk is essentially impossible in most cases of interest

Only have access to $D$ through a sample of size $m$ drawn from $D$ called the training data

Throughout the talk, $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ a sample of size $m$ drawn i.i.d. (independent and identically distributed) from $D$

A learning algorithm (possibly randomised) is a map $A$ from $2^{\mathcal{X} \times \mathcal{Y}}$ to $\mathcal{F}$
Statistical (Supervised) Learning Theory Framework

The Risk functional $R : \mathcal{F} \to \mathbb{R}^+$ is given by:

$$R(f) = \mathbb{E}_{(x,y) \sim D} [\ell(f; x, y)] = \mathbb{E}_{(x,y) \sim D} [\gamma(f(x), y)]$$

Would like to find $f \in \mathcal{F}$ that “minimises” the risk $R$

Even calculating (let alone minimising) the risk is essentially impossible in most cases of interest

Only have access to $D$ through a sample of size $m$ drawn from $D$ called the training data

Throughout the talk, $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ a sample of size $m$ drawn i.i.d. (independent and identically distributed) from $D$

A learning algorithm (possibly randomised) is a map $A$ from $2^{\mathcal{X} \times \mathcal{Y}}$ to $\mathcal{F}$

Goal: To guarantee with high probability (over $S$) that if $\hat{f} = A(S)$, then for some small $\epsilon > 0$:

$$R(\hat{f}) \leq \inf_{f \in \mathcal{F}} R(f) + \epsilon$$
Empirical Risk Minimisation

Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Learning algorithm: $A$ maps $2^{\mathcal{X} \times \mathcal{Y}}$ to $\mathcal{F}$

- Focus mostly on statistical questions
- Computationally ERM is intractable for most problems of interest
- E.g. Find a linear separator that minimises the number of misclassifications
  - Tractable if there exists a separator with no error!
Empirical Risk Minimisation

Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Learning algorithm: $A$ maps $2^{\mathcal{X} \times \mathcal{Y}}$ to $\mathcal{F}$

Define the empirical risk on a sample $S$ as:

$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$
Empirical Risk Minimisation

Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Learning algorithm: $A$ maps $2^X \times Y$ to $F$

Define the **empirical risk** on a sample $S$ as:

$$\hat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$

ERM (Empirical Risk Minimisation) principle suggests that we find $f \in F$ that minimises the **empirical risk**

- Focus mostly on **statistical questions**
- Computationally ERM is intractable for most problems of interest
Empirical Risk Minimisation

Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Learning algorithm: $A$ maps $2^X \times Y$ to $\mathcal{F}$

Define the \textbf{empirical risk} on a sample $S$ as:

$$\hat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$

ERM (Empirical Risk Minimisation) principle suggests that we find $f \in \mathcal{F}$ that minimises the \textbf{empirical risk}

- Focus mostly on \textbf{statistical questions}
- Computationally ERM is intractable for most problems of interest

E.g. Find a linear separator that minimises the number of misclassifications
Empirical Risk Minimisation

Training sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Learning algorithm: $A$ maps $2^{X \times Y}$ to $F$

Define the empirical risk on a sample $S$ as:

$$\hat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$

ERM (Empirical Risk Minimisation) principle suggests that we find $f \in F$ that minimises the empirical risk

- Focus mostly on statistical questions
- Computationally ERM is intractable for most problems of interest

E.g. Find a linear separator that minimises the number of misclassifications

Tractable if there exists a separator with no error!
Empirical Risk Minimisation

ERM Principle: Learning algorithm should pick $f \in \mathcal{F}$ that minimises the empirical risk

$$\hat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$
Empirical Risk Minimisation

ERM Principle: Learning algorithm should pick $f \in \mathcal{F}$ that minimises the empirical risk

$$\hat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$

How do we guarantee that the (actual) risk is close to optimal?
Empirical Risk Minimisation

ERM Principle: Learning algorithm should pick $f \in \mathcal{F}$ that minimises the empirical risk

$$\hat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$

- How do we guarantee that the (actual) risk is close to optimal?
- Focus on classification, i.e. $\mathcal{F} \subset \{-1, 1\}^X$ and suppose $VC(\mathcal{F}) = d < \infty$
- Cost function is $\gamma(y', y) = \mathbb{I}(y' \neq y)$
Empirical Risk Minimisation

**ERM Principle:** Learning algorithm should pick $f \in \mathcal{F}$ that minimises the empirical risk

$$\hat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(x_i), y_i)$$

▶ How do we guarantee that the (actual) risk is close to optimal?
▶ Focus on classification, i.e. $\mathcal{F} \subset \{-1, 1\}^X$ and suppose $VC(\mathcal{F}) = d < \infty$
▶ Cost function is $\gamma(y', y) = I(y' \neq y)$

**Theorem (Vapnik, Chervonenkis)**\(^{14,16}\)

Let $\mathcal{F} \subset \{0, 1\}^X$ with $VC(\mathcal{F}) = d < \infty$. Let $S \sim D^m$ for some distribution $D$ over $X \times \{-1, 1\}$. Then, for every $\delta > 0$, with probability at least $1 - \delta$, for every $f \in \mathcal{F}$,

$$R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)$$
Empirical Risk Minimisation

**Theorem (Vapnik, Chervonenkis)\(^{14,16}\)**

Let \( \mathcal{F} \subset \{0, 1\}^X \) with \( \text{VC}(\mathcal{F}) = d < \infty \). Let \( S \sim D^m \) for some distribution \( D \) over \( X \times \{0, 1\} \). Then, for every \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( f \in \mathcal{F} \),

\[
R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)
\]

Suppose \( f^* \) is the "minimiser" of the true risk \( R \) and \( \hat{f} \) is the minimiser of the empirical risk \( \hat{R}_S \).

Then, we have,

\[
R(\hat{f}) \leq \hat{R}_S(\hat{f}) + \frac{\epsilon}{2}
\]

Using Theorem (flipped)

Where \( \epsilon \) is chosen to be a suitable function of \( \delta \) and \( m \).
Empirical Risk Minimisation

Theorem (Vapnik, Chervonenkis)\textsuperscript{14,16}

Let $\mathcal{F} \subset \{0, 1\}^\mathcal{X}$ with $\text{VC}(\mathcal{F}) = d < \infty$. Let $S \sim D^m$ for some distribution $D$ over $\mathcal{X} \times \{0, 1\}$. Then, for every $\delta > 0$, with probability at least $1 - \delta$, for every $f \in \mathcal{F}$,

$$R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)$$

Suppose $f^*$ is the “minimiser” of the true risk $R$ and $\hat{f}$ is the minimiser of the empirical risk $\hat{R}_S$
Empirical Risk Minimisation

**Theorem (Vapnik, Chervonenkis)**\(^{14,16}\)

Let \( \mathcal{F} \subset \{0, 1\}^X \) with \( \text{VC}(\mathcal{F}) = d < \infty \). Let \( S \sim D^m \) for some distribution \( D \) over \( X \times \{0, 1\} \). Then, for every \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( f \in \mathcal{F} \),

\[
R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)
\]

Suppose \( f^* \) is the "minimiser" of the **true risk** \( R \) and \( \hat{f} \) is the minimiser of the **empirical risk** \( \hat{R}_S \)

Then, we have,

\[
R(\hat{f}) \leq \hat{R}_S(\hat{f}) + \epsilon/2
\]

Using Theorem
Empirical Risk Minimisation

**Theorem (Vapnik, Chervonenkis)**\(^{14,16}\)

Let \( \mathcal{F} \subset \{0, 1\}^\mathcal{X} \) with \( VC(\mathcal{F}) = d < \infty \). Let \( S \sim D^m \) for some distribution \( D \) over \( \mathcal{X} \times \{0, 1\} \). Then, for every \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( f \in \mathcal{F} \),

\[
R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)
\]

Suppose \( f^* \) is the "minimiser" of the true risk \( R \) and \( \hat{f} \) is the minimiser of the empirical risk \( \hat{R}_S \).

Then, we have,

\[
R(\hat{f}) \leq \hat{R}_S(\hat{f}) + \epsilon/2 \quad \text{Using Theorem}
\]

\[
\leq \hat{R}_S(f^*) + \epsilon/2 \quad \text{As } \hat{f} \text{ minimises } \hat{R}_S
\]
Empirical Risk Minimisation

**Theorem (Vapnik, Chervonenkis)\(^{14,16}\)**

Let \( \mathcal{F} \subset \{0, 1\}^X \) with \( \text{VC}(\mathcal{F}) = d < \infty \). Let \( S \sim D^m \) for some distribution \( D \) over \( X \times \{0, 1\} \). Then, for every \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( f \in \mathcal{F} \),

\[
R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)
\]

Suppose \( f^* \) is the "minimiser" of the true risk \( R \) and \( \hat{f} \) is the minimiser of the empirical risk \( \hat{R}_S \).

Then, we have,

\[
R(\hat{f}) \leq \hat{R}_S(\hat{f}) + \epsilon/2 \quad \text{Using Theorem}
\]

\[
\leq \hat{R}_S(f^*) + \epsilon/2 \quad \text{As } \hat{f} \text{ minimises } \hat{R}_S
\]

\[
\leq R(f^*) + \epsilon \quad \text{Using Theorem (flipped)}
\]
### Theorem (Vapnik, Chervonenkis)\textsuperscript{14,16}

Let $\mathcal{F} \subset \{0, 1\}^\mathcal{X}$ with $\text{VC}(\mathcal{F}) = d < \infty$. Let $S \sim D^m$ for some distribution $D$ over $\mathcal{X} \times \{0, 1\}$. Then, for every $\delta > 0$, with probability at least $1 - \delta$, for every $f \in \mathcal{F}$,

$$R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)$$

Suppose $f^*$ is the “minimiser” of the true risk $R$ and $\hat{f}$ is the minimiser of the empirical risk $\hat{R}_S$.

Then, we have,

$$R(\hat{f}) \leq \hat{R}_S(\hat{f}) + \epsilon/2 \quad \text{Using Theorem}$$

$$\leq \hat{R}_S(f^*) + \epsilon/2 \quad \text{As } \hat{f} \text{ minimises } \hat{R}_S$$

$$\leq R(f^*) + \epsilon \quad \text{Using Theorem (flipped)}$$

Where $\epsilon$ is chosen to be a suitable function of $\delta$ and $m$. 
Structural Risk Minimisation

\[ R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right) \]

How should be pick the class of functions \( F \)?

▶ More ''complex'' \( F \) can achieve smaller empirical risk

▶ Difference between true risk and empirical risk (generalisation error) will be higher for more ''complex'' \( F \)

Choose an infinite family of classes \( \{F_d : d = 1, 2, \ldots\} \) and find the minimiser:

\[ \hat{f} = \arg\min_{f \in F_d, d \in \mathbb{N}} \hat{R}_S(f) + \kappa(d,m) \]

where \( \kappa(d,m) \) is a penalty term that depends on the sample size and the ''complexity'' or ''capacity'' measure.
Structural Risk Minimisation

\[ R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right) \]

How should be pick the class of functions \( \mathcal{F} \)?

- More “complex” \( \mathcal{F} \) can achieve smaller empirical risk
- Difference between true risk and empirical risk (generalisation error) will be higher for more “complex” \( \mathcal{F} \)
Structural Risk Minimisation

\[ R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d\log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right) \]

How should be pick the class of functions \( \mathcal{F} \)?

- More “complex” \( \mathcal{F} \) can achieve smaller empirical risk
- Difference between true risk and empirical risk (generalisation error) will be higher for more “complex” \( \mathcal{F} \)

Choose an infinite family of classes \( \{\mathcal{F}_d : d = 1, 2, \ldots\} \) and find the minimiser:

\[ \hat{f} = \arg\min_{f \in \mathcal{F}_d, d \in \mathbb{N}} \hat{R}_S(f) + \kappa(d, m) \]

where \( \kappa(d, m) \) is a penalty term that depends on the sample size and the “complexity” or “capacity” measure.
Structural Risk Minimisation

\[
R(f) \leq \hat{R}_S(f) + \sqrt{\frac{2d \log(em/d)}{m}} + O\left(\sqrt{\frac{\log(1/\delta)}{2m}}\right)
\]

How should be pick the class of functions \( \mathcal{F} \)?
- More “complex” \( \mathcal{F} \) can achieve smaller empirical risk
- Difference between true risk and empirical risk (generalisation error) will be higher for more “complex” \( \mathcal{F} \)

Choose an infinite family of classes \( \{\mathcal{F}_d : d = 1, 2, \ldots\} \) and find the minimiser:

\[
\hat{f} = \arg\min_{f \in \mathcal{F}_d, d \in \mathbb{N}} \hat{R}_S(f) + \kappa(d, m)
\]

where \( \kappa(d, m) \) is a penalty term that depends on the sample size and the “complexity” or “capacity” measure

Related to the more commonly used approach in practice:

\[
\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{R}_S(f) + \lambda \cdot \text{regulariser}(f)
\]
Outline

Statistical (Supervised) Learning Theory Framework

Linear Regression

Rademacher Complexity

Support Vector Machines

Kernels

Neural Networks

Algorithmic Stability
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}$$
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}$$

Consider the squared loss as a cost function:

$$\gamma(y', y) = (y' - y)^2$$
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}$$

Consider the squared loss as a cost function:

$$\gamma(y', y) = (y' - y)^2$$

Let $D$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $g(x) = \mathbb{E}[y \mid x]$
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$\mathcal{F} = \{ \mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \mid \mathbf{w} \in K \}$$

Consider the squared loss as a cost function:

$$\gamma(y', y) = (y' - y)^2$$

Let $D$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $g(x) = \mathbb{E}[y \mid x]$.

For any $h : \mathcal{X} \rightarrow \mathbb{R}$:

$$R(h) = \mathbb{E}_{(x,y) \sim D} \left[ (h(x) - y)^2 \right] = \mathbb{E}_{(x,y) \sim D} \left[ (h(x) - g(x) + g(x) - y)^2 \right]$$
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$\mathcal{F} = \{ \mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \mid \mathbf{w} \in K \}$$

Consider the squared loss as a cost function:

$$\gamma(y', y) = (y' - y)^2$$

Let $D$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $g(x) = \mathbb{E}[y \mid x]$

For any $h : \mathcal{X} \to \mathbb{R}$:

$$R(h) = \mathbb{E}_{(x, y) \sim D} \left[ (h(x) - y)^2 \right] = \mathbb{E}_{(x, y) \sim D} \left[ (h(x) - g(x) + g(x) - y)^2 \right]$$

$$= \mathbb{E}_{(x, y) \sim D} \left[ (h(x) - g(x))^2 \right] + \mathbb{E}_{(x, y) \sim D} \left[ (g(x) - y)^2 \right]$$
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$
\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}
$$

Consider the squared loss as a cost function:

$$
\gamma(y', y) = (y' - y)^2
$$

Let $D$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $g(x) = \mathbb{E}[y \mid x]$

For any $h : \mathcal{X} \to \mathbb{R}$:

$$
R(h) = \mathbb{E}_{(x,y) \sim D} [(h(x) - y)^2] = \mathbb{E}_{(x,y) \sim D} [(h(x) - g(x) + g(x) - y)^2]
$$

$$
= \mathbb{E}_{(x,y) \sim D} [(h(x) - g(x))^2] + \mathbb{E}_{(x,y) \sim D} [(g(x) - y)^2]
$$

$$
+ 2 \mathbb{E}_{(x,y) \sim D} [(h(x) - g(x))(g(x) - y)]
$$

= 0 \text{ as } \mathbb{E}[y \mid x] = g(x)$
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}$$

Consider the squared loss as a cost function:

$$\gamma(y', y) = (y' - y)^2$$

Let $D$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $g(x) = \mathbb{E}[y \mid x]$

For any $h : \mathcal{X} \rightarrow \mathbb{R}$:

$$R(h) = \mathbb{E}_{(x,y) \sim D} \left[ (h(x) - y)^2 \right] = \mathbb{E}_{(x,y) \sim D} \left[ (h(x) - g(x) + g(x) - y)^2 \right]$$

$$= \mathbb{E}_{(x,y) \sim D} \left[ (h(x) - g(x))^2 \right] + \mathbb{E}_{(x,y) \sim D} \left[ (g(x) - y)^2 \right] + 2 \mathbb{E}_{(x,y) \sim D} \left[ (h(x) - g(x))(g(x) - y) \right]$$

$$= \mathbb{E}_{(x,y) \sim D} \left[ (h(x) - g(x))^2 \right] + R(g)$$

If $g \in \mathcal{F}$, we are in the so-called realisable setting.
Linear Regression

Let $K \subset \mathbb{R}^n$. Consider the family of linear functions

$$\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}$$

Consider the squared loss as a cost function:

$$\gamma(y', y) = (y' - y)^2$$

Let $D$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $g(x) = \mathbb{E}[y \mid x]$

For any $h : \mathcal{X} \to \mathbb{R}$:

$$R(h) = \mathbb{E}_{(x,y) \sim D} [(h(x) - y)^2] = \mathbb{E}_{(x,y) \sim D} [(h(x) - g(x) + g(x) - y)^2]$$

$$= \mathbb{E}_{(x,y) \sim D} [(h(x) - g(x))^2] + \mathbb{E}_{(x,y) \sim D} [(g(x) - y)^2]$$

$$+ 2 \mathbb{E}_{(x,y) \sim D} [(h(x) - g(x))(g(x) - y)]$$

$$= \mathbb{E}_{(x,y) \sim D} [(h(x) - g(x))^2] + R(g)$$

If $g \in \mathcal{F}$, we are in the so-called realisable setting
Aside: Maximum Likelihood Principle

**Discriminative Setting:** Model $y \mid w, x \sim w \cdot x + \mathcal{N}(0, \sigma^2)$
Discriminative Setting: Model $y \mid w, x \sim w \cdot x + \mathcal{N}(0, \sigma^2)$

We can define the likelihood of observing the data under this model

$$p(y_1, \ldots, y_m \mid w, x_1, \ldots, x_m) = \frac{1}{(2\pi\sigma^2)^{m/2}} \prod_{i=1}^{m} \exp \left( -\frac{(y_i - w \cdot x_i)^2}{2\sigma^2} \right)$$
Aside: Maximum Likelihood Principle

**Discriminative Setting:** Model $y \mid w, x \sim w \cdot x + \mathcal{N}(0, \sigma^2)$

We can defined the **likelihood** of observing the data under this model

$$p(y_1, \ldots, y_m \mid w, x_1, \ldots, x_m) = \frac{1}{(2\pi\sigma^2)^{m/2}} \prod_{i=1}^{m} \exp \left( -\frac{(y_i - w \cdot x_i)^2}{2\sigma^2} \right)$$

Looking at the log likelihood is slightly simpler

$$\text{LL}(y_1, \ldots, y_m \mid w, x_1, \ldots, x_m) = -\frac{m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - w \cdot x_i)^2$$
Aside: Maximum Likelihood Principle

**Discriminative Setting:** Model \( y \mid w, x \sim w \cdot x + \mathcal{N}(0, \sigma^2) \)

We can defined the **likelihood** of observing the data under this model

\[
p(y_1, \ldots, y_m \mid w, x_1, \ldots, x_m) = \frac{1}{(2\pi\sigma^2)^{m/2}} \prod_{i=1}^{m} \exp \left( -\frac{(y_i - w \cdot x_i)^2}{2\sigma^2} \right)
\]

Looking at the log likelihood is slightly simpler

\[
LL(y_1, \ldots, y_m \mid w, x_1, \ldots, x_m) = -\frac{m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - w \cdot x_i)^2
\]

Finding parameters \( w \) that maximise the (log) likelihood is the same as finding \( w \) that minimises the empirical risk with the **squared error cost**

The method of **least squares** goes back at least 200 years to Gauss, Laplace
Let $K \subset \mathbb{R}^n$, e.g. $K = \{w \mid \|w\|_2 \leq W\}$. Consider the family of linear functions

$$\mathcal{F} = \{x \mapsto w \cdot x \mid w \in K\}$$
Let $K \subset \mathbb{R}^n$, e.g. $K = \{ w \mid \|w\|_2 \leq W \}$. Consider the family of linear functions

$$\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}$$

**ERM for Linear Regression**

$$\hat{w} = \arg\min_{w \in K} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2$$
Linear Regression

Let $K \subset \mathbb{R}^n$, e.g. $K = \{ w \mid \|w\|_2 \leq W \}$. Consider the family of linear functions

$$
\mathcal{F} = \{ x \mapsto w \cdot x \mid w \in K \}
$$

**ERM for Linear Regression**

$$
\hat{w} = \arg\min_{w \in K} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2
$$

How do we argue about the generalisation properties of this algorithm?
Linear Regression

Let $K \subset \mathbb{R}^n$, e.g. $K = \{w \mid \|w\|_2 \leq W\}$. Consider the family of linear functions

$$F = \{x \mapsto w \cdot x \mid w \in K\}$$

**ERM for Linear Regression**

$$\hat{w} = \arg\min_{w \in K} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2$$

How do we argue about the generalisation properties of this algorithm?

Use a different capacity measure

- Rademacher complexity, VC dimension, pseudo-dimension, covering numbers, fat-shattering dimension, ...
Let $K \subset \mathbb{R}^n$, e.g. $K = \{w \mid \|w\|_2 \leq W\}$. Consider the family of linear functions

$$\mathcal{F} = \{x \mapsto w \cdot x \mid w \in K\}$$

**ERM for Linear Regression**

$$\hat{w} = \arg\min_{w \in K} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2$$

How do we argue about the generalisation properties of this algorithm?

Use a different capacity measure

- Rademacher complexity, VC dimension, pseudo-dimension, covering numbers, fat-shattering dimension, ...

We will require some boundedness assumptions on the data and the linear functions
Outline

Statistical (Supervised) Learning Theory Framework

Linear Regression

Rademacher Complexity

Support Vector Machines

Kernels

Neural Networks

Algorithmic Stability
Let $\mathcal{G}$ be a class of functions from $\mathcal{Z} \to [a, b] \subseteq \mathbb{R}$.
Empirical Rademacher Complexity

Let $G$ be a class of functions from $\mathcal{Z} \to [a, b] \subset \mathbb{R}$

$S = \{z_1, \ldots, z_m\} \subset \mathcal{Z}$ be a fixed sample of size $m$
Empirical Rademacher Complexity

Let $G$ be a class of functions from $\mathcal{Z} \to [a, b] \subset \mathbb{R}$

$S = \{z_1, \ldots, z_m\} \subset \mathcal{Z}$ be a fixed sample of size $m$

Then the Empirical Rademacher Complexity of $G$ with respect to $S$ is defined as:

$$\hat{R}_S(G) = \mathbb{E}_{\sigma \sim u\{-1, 1\}^m} \left[ \sup_{g \in G} \frac{1}{m} \sum_{i=1}^m \sigma_i g(z_i) \right]$$

where $(\sigma_1, \ldots, \sigma_m) =: \sigma \sim u\{-1, 1\}^m$ indicates that each $\sigma_i$ is a random variable taking the values $\{-1, 1\}$ with equal probability. These are called Rademacher random variables.
Rademacher Complexity

Let $D$ be a distribution over the set $Z$. Let $G$ be a class of functions from $Z \rightarrow [a,b] \subset \mathbb{R}$. For any $m \geq 1$, the Rademacher complexity of $G$ is the expectation of the empirical Rademacher complexity of $G$ over a sample drawn from $D$:

$$R_m(G) = \mathbb{E}_{\sigma \sim u\{-1,1\}^m} \left[ \sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right]$$
### Empirical Rademacher Complexity

\[
\hat{\mathcal{R}}_S (G) = \mathbb{E}_{\sigma \sim u \{-1, 1\}^m} \left[ \sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right]
\]

### Rademacher Complexity

Let \( D \) be a distribution over the set \( Z \). Let \( G \) be a class of functions from \( Z \to [a, b] \subset \mathbb{R} \). For any \( m \geq 1 \), the Rademacher complexity of \( G \) is the expectation of the empirical Rademacher complexity of \( G \) over a sample drawn from \( D^m \):

\[
\mathcal{R}_m(G) = \mathbb{E}_{S \sim D^m} \left[ \hat{\mathcal{R}}_S(G) \right]
\]
Rademacher Complexity

\[ \hat{R}_S(G) = \mathbb{E}_{\sigma \sim u\{-1,1\}^m} \left[ \sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right] ; \quad R_m(G) = \mathbb{E}_{S \sim D^m} \left[ \hat{R}_S(G) \right] \]
Rademacher Complexity

$$\hat{\mathcal{R}}_{S}(G) = \mathbb{E}_{\sigma \sim u\{-1,1\}^{m}} \left[ \sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_{i} g(z_{i}) \right] ; \quad \mathcal{R}_{m}(G) = \mathbb{E}_{S \sim D^{m}} \left[ \hat{\mathcal{R}}_{S}(G) \right]$$

**Theorem$^{2,14}$**

Let $G$ be a class of functions mapping $\mathcal{Z} \rightarrow [0, 1]$. Let $D$ be a distribution over $\mathcal{Z}$ and suppose that a sample $S$ of size $m$ is drawn from $D^{m}$. Then for every $\delta > 0$, with probability at least $1 - \delta$, the following holds for each $g \in G$:

$$\mathbb{E}_{z \sim D} \left[ g(z) \right] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_{i}) + 2\mathcal{R}_{m}(G) + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right).$$
Rademacher Complexity

\[\hat{R}_S(G) = \mathbb{E}_{\sigma \sim u\{-1,1\}^m} \left[ \sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right] ; \ R_m(G) = \mathbb{E}_{S \sim D^m} \left[ \hat{R}_S(G) \right] \]

**Theorem**

Let \( G \) be a class of functions mapping \( \mathcal{Z} \rightarrow [0, 1] \). Let \( D \) be a distribution over \( \mathcal{Z} \) and suppose that a sample \( S \) of size \( m \) is drawn from \( D^m \). Then for every \( \delta > 0 \), with probability at least \( 1 - \delta \), the following holds for each \( g \in G \):

\[
\mathbb{E}_{z \sim D} [g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2R_m(G) + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right).
\]

Henceforth, for \( S = \{z_1, \ldots, z_m\} \), we will use the notation:

\[
\hat{\mathbb{E}}_{z \sim u S} [g(z)] = \frac{1}{m} \sum_{i=1}^{m} g(z_i)
\]

We will see a full proof of this theorem. First, let’s apply this to linear regression.
Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}, \|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto \mathbf{w} \cdot x \mid \|\mathbf{w}\|_2 \leq W\}$
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}$, $\|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{ x \mapsto w \cdot x \mid \|w\|_2 \leq W \}$

Let $S = \{x_1, \ldots, x_m \}$. Then we have:

$$\hat{\mathcal{R}}_S(\mathcal{F}) = \frac{1}{m} \mathbb{E} \left[ \sup_{w, \|w\|_2 \leq W} \sum_{i=1}^m \sigma_i (w \cdot x_i) \right]$$
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}$, $\|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$

Let $S = \{x_1, \ldots, x_m\}$. Then we have:

$$\hat{\mathcal{R}}_S(\mathcal{F}) = \frac{1}{m \sigma} \mathbb{E} \left[ \sup_{w, \|w\|_2 \leq W} \sum_{i=1}^{m} \sigma_i (w \cdot x_i) \right]$$

$$= \frac{1}{m \sigma} \mathbb{E} \left[ \sup_{w, \|w\|_2 \leq W} w \cdot \sum_{i=1}^{m} \sigma_i x_i \right]$$
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}, \|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$

Let $S = \{x_1, \ldots, x_m\}$. Then we have:

$$\hat{R}_S(\mathcal{F}) = \frac{1}{m \sigma} \mathbb{E} \left[ \sup_{w, \|w\|_2 \leq W} \sum_{i=1}^{m} \sigma_i (w \cdot x_i) \right]$$

$$= \frac{1}{m \sigma} \mathbb{E} \left[ \sup_{w, \|w\|_2 \leq W} w \cdot \sum_{i=1}^{m} \sigma_i x_i \right]$$

$$= \frac{W}{m \sigma} \mathbb{E} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2 \right]$$

The last step follows from (the equality condition of) the Cauchy-Schwartz Inequality
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}, \|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}$, $\|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$

$$\hat{\mathbb{R}}_S(\mathcal{F}) = \frac{W}{m \sigma} \mathbb{E} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2 \right]$$
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}, \|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$

\[
\hat{R}_S(\mathcal{F}) = \frac{W}{m} \mathbb{E} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2 \right] \leq \frac{W}{m} \left( \mathbb{E} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2^2 \right] \right)^{\frac{1}{2}}
\]
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}, \|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x | \|w\|_2 \leq W\}$

$$\hat{R}_S(\mathcal{F}) = \frac{W}{m} \mathbb{E} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2 \right] \leq \frac{W}{m} \left( \mathbb{E} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2^2 \right] \right)^{\frac{1}{2}}$$

$$= \frac{W}{m} \left( \mathbb{E} \left[ \sum_{i=1}^{m} \sigma_i^2 \|x_i\|_2^2 \right] + 2 \sum_{i<j} \sigma_i \sigma_j x_i \cdot x_j \right)^{\frac{1}{2}}$$

= 0 as $\sigma_i$ are independent
Generalisation Bounds for Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}$, $\|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$

$$\hat{\mathcal{R}}_S(\mathcal{F}) = \frac{W}{m} \mathbb{E}_{\sigma} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2 \right] \leq \frac{W}{m} \left( \mathbb{E}_{\sigma} \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_2^2 \right] \right)^{\frac{1}{2}}$$

$$= \frac{W}{m} \left( \mathbb{E}_{\sigma} \left[ \sum_{i=1}^{m} \sigma_i^2 \|x_i\|_2^2 + 2 \sum_{i<j} \sigma_i \sigma_j x_i \cdot x_j \right] \right)^{\frac{1}{2}}$$

$$= \frac{W}{m} \left( \sum_{i=1}^{m} \|x_i\|_2^2 \right)^{\frac{1}{2}} = \frac{WX}{\sqrt{m}}$$
We computed the Rademacher complexity of linear functions, but we’d like to apply the “main theorem” to the true risk
Talagrand’s Lemma

We computed the Rademacher complexity of linear functions, but we’d like to apply the “main theorem” to the true risk

For this we need to look at the composition of the linear function and the loss/cost function
Talagrand’s Lemma

We computed the Rademacher complexity of linear functions, but we’d like to apply the “main theorem” to the true risk.

For this we need to look at the composition of the linear function and the loss/cost function.

Let $G$ be a class of functions from $\mathcal{Z} \rightarrow [a, b]$ and let $\varphi : [a, b] \rightarrow \mathbb{R}$ be $L$-Lipschitz.
We computed the Rademacher complexity of linear functions, but we’d like to apply the “main theorem” to the true risk

For this we need to look at the composition of the linear function and the loss/cost function

Let $\mathcal{G}$ be a class of functions from $\mathbb{Z} \rightarrow [a, b]$ and let $\varphi : [a, b] \rightarrow \mathbb{R}$ be $L$-Lipschitz

Then Talagrand’s Lemma tells us that:

$$\hat{R}_S(\varphi \circ \mathcal{G}) \leq L \cdot \hat{R}_S(\mathcal{G})$$
$$\mathcal{R}_m(\varphi \circ \mathcal{G}) \leq L \cdot \mathcal{R}_m(\mathcal{G})$$
Generalisation of Linear Regression

Instance space \( \mathcal{X} \subset \mathbb{R}^n \), \( \forall x \in \mathcal{X}, \|x\|_2 \leq X \)

Target values \( \mathcal{Y} = [-M, M] \)

Let \( \mathcal{F} = \{ x \mapsto w \cdot x \mid \|w\|_2 \leq W \} \)
Generalisation of Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}, \|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$

Consider the following:

$$\mathcal{H} = \{ (x, y) \mapsto (f(x) - y)^2 \mid x \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F} \}$$

$$\phi: [-(M + WX), (M + WX)] \to \mathbb{R}, \phi(z) = z^2$$
Generalisation of Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}, \|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{ x \mapsto w \cdot x \mid \|w\|_2 \leq W \}$

Consider the following:

$$\mathcal{H} = \{(x, y) \mapsto (f(x) - y)^2 \mid x \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F}\}$$

$$\phi : [-(M + WX), (M + WX)] \to \mathbb{R}, \phi(z) = z^2$$

$\phi$ is $2(M + WX)$-Lipschitz on its domain
Generalisation of Linear Regression

Instance space \( \mathcal{X} \subset \mathbb{R}^n \), \( \forall x \in \mathcal{X}, \|x\|_2 \leq X \)

Target values \( \mathcal{Y} = [-M, M] \)

Let \( \mathcal{F} = \{ x \mapsto w \cdot x \mid \|w\|_2 \leq W \} \)

Consider the following:

\[
\mathcal{H} = \{ (x, y) \mapsto (f(x) - y)^2 \mid x \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F} \}
\]

\[
\phi : [-(M + WX), (M + WX)] \rightarrow \mathbb{R}, \phi(z) = z^2
\]

\( \phi \) is \( 2(M + WX) \)-Lipschitz on its domain

\[
\hat{\mathcal{R}}_S([-M, M]) \leq M/\sqrt{m}
\]
Generalisation of Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}$, $\|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x \mid \|w\|_2 \leq W\}$

Consider the following:

$$\mathcal{H} = \{(x, y) \mapsto (f(x) - y)^2 \mid x \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F}\}$$

$$\phi : [-(M + WX), (M + WX)] \rightarrow \mathbb{R}, \phi(z) = z^2$$

$\phi$ is $2(M + WX)$-Lipschitz on its domain

$$\hat{\mathcal{R}}_S([-M, M]) \leq M/\sqrt{m}$$

Using $\hat{\mathcal{R}}_S(\mathcal{F} + \mathcal{G}) \leq \hat{\mathcal{R}}_S(\mathcal{F}) + \hat{\mathcal{R}}_S(\mathcal{G})$ and Talagrand's Lemma, we get

$$\hat{\mathcal{R}}_S(\mathcal{H}) \leq \frac{2(M + WX)^2}{\sqrt{m}}$$
Generalisation of Linear Regression

Instance space $\mathcal{X} \subset \mathbb{R}^n$, $\forall x \in \mathcal{X}$, $\|x\|_2 \leq X$

Target values $\mathcal{Y} = [-M, M]$

Let $\mathcal{F} = \{x \mapsto w \cdot x | \|w\|_2 \leq W\}$

Consider the following:

$$\mathcal{H} = \{(x, y) \mapsto (f(x) - y)^2 | x \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F}\}$$

$$\phi : [-(M + WX), (M + WX)] \rightarrow \mathbb{R}, \phi(z) = z^2$$

$\phi$ is $2(M + WX)$-Lipschitz on its domain

$$\hat{\mathcal{R}}_S([-M, M]) \leq M / \sqrt{m}$$

Using $\hat{\mathcal{R}}_S(\mathcal{F} + \mathcal{G}) \leq \hat{\mathcal{R}}_S(\mathcal{F}) + \hat{\mathcal{R}}_S(\mathcal{G})$ and Talagrand’s Lemma, we get

$$\hat{\mathcal{R}}_S(\mathcal{H}) \leq \frac{2(M + WX)^2}{\sqrt{m}}$$

Note that $\mathcal{R}_m(\mathcal{H}) = \mathbb{E}_{S \sim D_m} \left[ \hat{\mathcal{R}}_S(\mathcal{H}) \right] \leq \sup_{S, |S|=m} \hat{\mathcal{R}}_S(\mathcal{H})$
ERM for Linear Regression

\[ J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 \]

\[ \hat{\mathbf{w}} = \text{argmin}_{\mathbf{w}, \|\mathbf{w}\|_2 \leq W} J(\mathbf{w}) \]
Aside: Algorithms for the Linear Regression Model

**ERM for Linear Regression**

\[
J(w) = \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2
\]

\[
\hat{w} = \arg\min_{w, \|w\|_2 \leq W} J(w)
\]

How can we solve this optimisation problem? (without norm constraint there is a closed form solution)
Aside: Algorithms for the Linear Regression Model

**ERM for Linear Regression**

\[
J(w) = \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2
\]

\[
\hat{w} = \arg\min_{w, \|w\|_2 \leq W} J(w)
\]

How can we solve this optimisation problem? (without norm constraint there is a closed form solution)

This convex optimisation problem can solved using projected (stochastic) gradient descent.
ERM for Linear Regression

\[
J(w) = \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2
\]

\[
\hat{w} = \arg\min_{w, \|w\|_2 \leq W} J(w)
\]

How can we solve this optimisation problem? (without norm constraint there is a closed form solution)

This convex optimisation problem can solved using projected (stochastic) gradient descent

Guaranteed to find a near-optimal solution in polynomial time
Aside: Gradient Descent

**Algorithm 1** Projected Gradient Descent

**Inputs:** $\eta, T$

Pick $w_1 \in K$

for $t = 1, \ldots, T$ do

\[
    w_{t+1}' = w_t - \eta \nabla J(w_t)
\]

\[
    w_{t+1} = \Pi_K(w_{t+1}')
\]

end for

**Output:** $\bar{w} = \frac{1}{T} \sum_{t=1}^{T} w_t$

Recall in our case $K = \{w \mid \|w\|_2 \leq W\}$, $\Pi_K(\cdot)$ is the projection operator.

**Informal Theorem 5**

Suppose $\sup_{w, w'} \|w - w'\|_2 \leq R$ and $\sum_{w \in K} \|\nabla J(w)\|_2 \leq L$, then with $\eta = R/(L \sqrt{T})$

$J(w_{t+1}) \leq \min_{w \in K} J(w) + RL \sqrt{T}$
Aside: Gradient Descent

Algorithm 2 Projected Gradient Descent

**Inputs:** $\eta, T$
Pick $w_1 \in K$

for $t = 1, \ldots, T$ do

\[ w'_{t+1} = w_t - \eta \nabla J(w_t) \]

\[ w_{t+1} = \Pi_K(w'_{t+1}) \]

end for

**Output:** $\bar{w} = \frac{1}{T} \sum_{t=1}^{T} w_t$

Recall in our case $K = \{ w \mid \| w \|_2 \leq W \}$, $\Pi_K(\cdot)$ is the projection operator
Algorithm 3 Projected Gradient Descent

**Inputs:** $\eta, T$

Pick $w_1 \in K$

for $t = 1, \ldots, T$ do

\[ w_{t+1}' = w_t - \eta \nabla J(w_t) \]

\[ w_{t+1} = \Pi_K(w_{t+1}') \]

end for

**Output:** $\bar{w} = \frac{1}{T} \sum_{t=1}^{T} w_t$

Recall in our case $K = \{w | \|w\|_2 \leq W\}$, $\Pi_K(\cdot)$ is the projection operator

**Informal Theorem**

Suppose $\sup_{w, w' \in K} \|w - w'\|_2 \leq R$ and $\sum_{w \in K} \|\nabla J(w)\|_2 \leq L$, then with $\eta = R/(L\sqrt{T})$

\[ J(\bar{w}) \leq \min_{w \in K} J(w) + \frac{RL}{\sqrt{T}} \]
Aside: Generalised Linear Models

Can consider more general models called **generalised linear models**

$$\text{GLM} = \{ x \mapsto u(w \cdot x) \mid u \text{ bounded, increasing} \& 1-\text{Lipschitz}, \|w\|_2 \leq W \}$$
Aside: Generalised Linear Models

Can consider more general models called *generalised linear models*

$$\text{GLM} = \{ x \mapsto u(w \cdot x) \mid u \text{ bounded, increasing} \& 1\text{-Lipschitz}, \|w\|_2 \leq W \}$$

We can consider the ERM problem:

$$J(w) = \frac{1}{m} \sum_{i=1}^{m} (u(w \cdot x_i) - y_i)^2; \quad \hat{w} = \arg\min_{w, \|w\|_2 \leq W} J(w)$$
Aside: Generalised Linear Models

Can consider more general models called generalised linear models

$$\text{GLM} = \{ x \mapsto u(w \cdot x) \mid u \text{ bounded, increasing } & \text{ & } 1\text{-Lipschitz}, \|w\|_2 \leq W \}$$

We can consider the ERM problem:

$$J(w) = \frac{1}{m} \sum_{i=1}^{m} (u(w \cdot x_i) - y_i)^2; \quad \hat{w} = \arg\min_{w, \|w\|_2 \leq W} J(w)$$

Can bound Rademacher complexity easily using the boundedness and Lipschitz property of $u$

However, the optimisation problem is now non-convex!
Aside: Generalised Linear Models

Can consider more general models called generalised linear models

\[ \text{GLM} = \{ x \mapsto u(w \cdot x) \mid u \text{ bounded, increasing & } 1\text{-Lipschitz, } \|w\|_2 \leq W \} \]
Aside: Generalised Linear Models

Can consider more general models called generalised linear models

$$\text{GLM} = \{ x \mapsto u(w \cdot x) \mid u \text{ bounded, increasing \& } 1\text{-Lipschitz}, \|w\|_2 \leq W \}$$

Can consider a different cost/loss function:

$$\gamma(y', y) = \int_0^{u^{-1}(y')} (u(z) - y)dz$$

$$\ell(w; x, y) = \int_0^{w \cdot x} (u(z) - y)dz$$
Aside: Generalised Linear Models

Can consider more general models called **generalised linear models**

\[ \text{GLM} = \{ x \mapsto u(w \cdot x) \mid u \text{ bounded, increasing & 1-Lipschitz}, \|w\|_2 \leq W \} \]

Can consider a different cost/loss function:

\[
\gamma(y', y) = \int_0^{u^{-1}(y')} (u(z) - y) \, dz \\
\ell(w; x, y) \int_0^{w \cdot x} (u(z) - y) \, dz
\]

The resulting objective function is convex in \( w \)

\[
\tilde{J}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w; x_i, y_i)
\]
Aside: Generalised Linear Models

Can consider more general models called generalised linear models

\[
\text{GLM} = \{x \mapsto u(w \cdot x) \mid u \text{ bounded, increasing & } 1\text{-Lipschitz, } \|w\|_2 \leq W\}
\]

Can consider a different cost/loss function:

\[
\gamma(y', y) = \int_0^{u^{-1}(y')} (u(z) - y)dz
\]

\[
\ell(w; x, y) = \int_0^w (u(z) - y)dz
\]

The resulting objective function is convex in \(w\)

\[
\tilde{J}(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w; x_i, y_i)
\]

In the realisable setting, i.e. \(\mathbb{E}[y \mid x] = u(w \cdot x)\), the global minimisers of \(J(w)\) (squared error) and \(\tilde{J}(w)\) coincide, yielding computationally and statistically efficient algorithms.\(^{12}\)
Rademacher Complexity : Main Result

Let $G$ be a class of functions mapping $\mathbb{Z} \rightarrow [0, 1]$. Let $D$ be a distribution over $\mathbb{Z}$ and suppose that a sample $S$ of size $m$ is drawn from $D^m$. Then for every $\delta > 0$, with probability at least $1 - \delta$, the following holds for each $g \in G$:

$$\mathbb{E}_{z \sim D}[g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2R_m(G) + O(\sqrt{\log(1/\delta)/m}).$$

We will make use of a concentration of measure inequality, called McDiarmid's inequality.

**McDiarmid's Inequality**

Let $Z$ be a set and let $f : Z^m \rightarrow \mathbb{R}$ be a function such that, for all $i$,

$$\forall z_1, \ldots, z_i, \ldots, z_m, z_i', \quad |f(z_1, \ldots, z_i, \ldots, z_m) - f(z_1, \ldots, z_i', \ldots, z_m)| \leq c_i.$$

Let $Z_1, \ldots, Z_m$ be i.i.d. random variables taking values in $Z$, then for all $\varepsilon > 0$,

$$\mathbb{P}[f(Z_1, \ldots, Z_m) \geq \mathbb{E}[f(Z_1, \ldots, Z_m)] + \varepsilon] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{m} c_i^2}\right).$$
Rademacher Complexity: Main Result

**Theorem**\textsuperscript{2,14}

Let $\mathcal{G}$ be a class of functions mapping $\mathcal{Z} \to [0, 1]$. Let $D$ be a distribution over $\mathcal{Z}$ and suppose that a sample $S$ of size $m$ is drawn from $D^m$. Then for every $\delta > 0$, with probability at least $1 - \delta$, the following holds for each $g \in \mathcal{G}$:

\[
\mathbb{E}_{z \sim D} [g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(\mathcal{G}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right).
\]
Rademacher Complexity: Main Result

**Theorem**\(^{2,14}\)

Let \( \mathcal{G} \) be a class of functions mapping \( \mathcal{Z} \rightarrow [0, 1] \). Let \( D \) be a distribution over \( \mathcal{Z} \) and suppose that a sample \( S \) of size \( m \) is drawn from \( D^m \). Then for every \( \delta > 0 \), with probability at least \( 1 - \delta \), the following holds for each \( g \in \mathcal{G} \):

\[
\mathbb{E}_{z \sim D} [ g(z) ] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2 \mathcal{R}_m(\mathcal{G}) + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right).
\]

We will make use of a concentration of measure inequality, called McDiarmid’s inequality.
Rademacher Complexity: Main Result

**Theorem**

Let $\mathcal{G}$ be a class of functions mapping $\mathcal{Z} \to [0, 1]$. Let $D$ be a distribution over $\mathcal{Z}$ and suppose that a sample $S$ of size $m$ is drawn from $D^m$. Then for every $\delta > 0$, with probability at least $1 - \delta$, the following holds for each $g \in \mathcal{G}$:

$$
\mathbb{E}_{z \sim D} [g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\mathcal{R}_m(\mathcal{G}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right).
$$

We will make use of a concentration of measure inequality, called McDiarmid’s inequality.

**McDiarmid’s Inequality**

Let $\mathcal{Z}$ be a set and let $f : \mathcal{Z}^m \to \mathbb{R}$ be a function such that, $\forall i, \exists c_i > 0, \forall z_1, \ldots, z_m, z'_i,$

$$
|f(z_1, \ldots, z_i, \ldots, z_m) - f(z_1, \ldots, z'_i, \ldots, z_m)| \leq c_i.
$$

Let $Z_1, \ldots, Z_m$ be i.i.d. random variables taking values in $\mathcal{Z}$, then $\forall \varepsilon > 0$,

$$
\mathbb{P} \left[ f(Z_1, \ldots, Z_m) \geq \mathbb{E} [f(Z_1, \ldots, Z_m)] + \varepsilon \right] \leq \exp \left( -\frac{2\varepsilon^2}{\sum_i c_i^2} \right).
$$
Proof of Main Result

Let $S = \{z_1, \ldots, z_m\}$, $S' = \{z'_1, \ldots, z'_m\} \sim D^m$
Proof of Main Result

Let $S = \{z_1, \ldots, z_m\}, S' = \{z'_1, \ldots, z'_m\} \sim D^m$

For $S \subset \mathcal{Z}$, define the function:

$$
\Phi(S) = \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \hat{\mathbb{E}}_{z \sim u \mid S} [g(z)] \right)
$$
Proof of Main Result

Let \( S = \{z_1, \ldots, z_m\}, S' = \{z'_1, \ldots, z'_m\} \sim D^m \)

For \( S \subset \mathcal{Z} \), define the function:

\[
\Phi(S) = \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \mathbb{E}_{z \sim u S} [g(z)] \right)
\]

Let \( S^i = \{z_1, \ldots, z_{i-1}, z'_i, z_{i+1}, \ldots, z_m\} \), and consider,

\[
|\Phi(S) - \Phi(S')| \leq \frac{1}{m} |g(z_i) - g(z'_i)| \leq \frac{1}{m}
\]
Proof of Main Result

Let $S = \{z_1, \ldots, z_m\}, S' = \{z'_1, \ldots, z'_m\} \sim D^m$

For $S \subset \mathcal{Z}$, define the function:

$$\Phi(S) = \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \mathbb{E}_{z \sim u_S} [g(z)] \right)$$

Let $S^i = \{z_1, \ldots, z_{i-1}, z'_i, z_{i+1}, \ldots, z_m\}$, and consider,

$$\left| \Phi(S) - \Phi(S') \right| \leq \frac{1}{m} |g(z_i) - g(z'_i)| \leq \frac{1}{m}$$

McDiarmid’s Inequality

Let $\mathcal{Z}$ be a set and let $f : \mathcal{Z}^m \to \mathbb{R}$ be a function such that, $\forall i, \exists c_i > 0, \forall z_1, \ldots, z_m, z'_i,$

$$|f(z_1, \ldots, z_i, \ldots, z_m) - f(z_1, \ldots, z'_i, \ldots, z_m)| \leq c_i.$$ 

Let $Z_1, \ldots, Z_m$ be i.i.d. random variables taking values in $\mathcal{Z}$, then $\forall \varepsilon > 0,$

$$\mathbb{P} \left[ f(Z_1, \ldots, Z_m) \geq \mathbb{E} [f(Z_1, \ldots, Z_m)] + \varepsilon \right] \leq \exp \left( -\frac{2\varepsilon^2}{\sum_i c_i^2} \right)$$
Proof of Main Result

McDiarmid’s Inequality

\[
P \left[ f(Z_1, \ldots, Z_m) \geq \mathbb{E} \left[ f(Z_1, \ldots, Z_m) \right] + \varepsilon \right] \leq \exp \left( -\frac{2\varepsilon^2}{\sum_i c_i^2} \right)
\]
Proof of Main Result

McDiarmid’s Inequality

\[ \mathbb{P} \left[ f(Z_1, \ldots, Z_m) \geq \mathbb{E} \left[ f(Z_1, \ldots, Z_m) \right] + \varepsilon \right] \leq \exp \left( -\frac{2\varepsilon^2}{\sum_i c_i^2} \right) \]

Let \( S = \{z_1, \ldots, z_m\}, S^i = \{z_1, \ldots, z_{i-1}, z'_i, z_{i+1}, \ldots, z_m\} \), we have

\[ \left| \Phi(S) - \Phi(S') \right| \leq \frac{1}{m} |g(z_i) - g(z'_i)| \leq \frac{1}{m} \]
Proof of Main Result

<table>
<thead>
<tr>
<th>McDiarmid’s Inequality</th>
</tr>
</thead>
</table>
| \[
\mathbb{P} \left[ f(Z_1, \ldots, Z_m) \geq \mathbb{E} \left[ f(Z_1, \ldots, Z_m) \right] + \varepsilon \right] \leq \exp \left( -\frac{2\varepsilon^2}{\sum_i c_i^2} \right)
\]|

Let \( S = \{z_1, \ldots, z_m\} \), \( S^i = \{z_1, \ldots, z_{i-1}, z'_i, z_{i+1}, \ldots, z_m\} \), we have

\[
\frac{1}{m} |g(z_i) - g(z'_i)| \leq \frac{1}{m}
\]

Applying McDiarmid’s inequality with \( c_i = 1/m \) for all \( i \),

\[
\mathbb{P} \left[ \Phi(S) \geq \mathbb{E}_{\sim D^m} [\Phi(S')] + \varepsilon \right] \leq \exp(-2\varepsilon^2 m)
\]
### Proof of Main Result

**McDiarmid’s Inequality**

\[
P \left[ f(Z_1, \ldots, Z_m) \geq \mathbb{E} \left[ f(Z_1, \ldots, Z_m) \right] + \varepsilon \right] \leq \exp \left( -\frac{2\varepsilon^2}{\sum_i c_i^2} \right)
\]

Let \( S = \{z_1, \ldots, z_m\} \), \( S^i = \{z_1, \ldots, z_{i-1}, z_i', z_{i+1}, \ldots, z_m\} \), we have

\[
\left| \Phi(S) - \Phi(S') \right| \leq \frac{1}{m} |g(z_i) - g(z_i')| \leq \frac{1}{m}
\]

Applying McDiarmid’s inequality with \( c_i = 1/m \) for all \( i \),

\[
P \left[ \Phi(S) \geq \mathbb{E}_{S \sim \mathcal{D}^m} [\Phi(S')] + \varepsilon \right] \leq \exp(-2\varepsilon^2 m)
\]

Alternatively, for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[
\Phi(S) \leq \mathbb{E}_{S \sim \mathcal{D}^m} [\Phi(S')] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right)
\]
Proof of Main Result

\[
\Phi(S) = \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \hat{\mathbb{E}}_{z \sim u_S} [g(z)] \right)
\]

Alternatively, for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[
\Phi(S) \leq \mathbb{E}_{S \sim D^m} [\Phi(S)] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right)
\]
Proof of Main Result

\[ \Phi(S) = \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \hat{\mathbb{E}}_{z \sim u S} [g(z)] \right) \]

Alternatively, for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[ \Phi(S) \leq \mathbb{E}_{S \sim D^m} \left[ \Phi(S) \right] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Thus, for any \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( g \in G \),

\[ \mathbb{E}_{z \sim D} [g(z)] \leq \hat{\mathbb{E}}_{z \sim u S} [g(z)] + \mathbb{E}_{S \sim D^m} \left[ \Phi(S) \right] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]
Proof of Main Result

\[ \Phi(S) = \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \hat{\mathbb{E}}_{z \sim uS} [g(z)] \right) \]

Alternatively, for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[ \Phi(S) \leq \mathbb{E}_{S \sim D^m} [\Phi(S)] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Thus, for any \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( g \in G \),

\[ \mathbb{E}_{z \sim D} [g(z)] \leq \hat{\mathbb{E}}_{z \sim uS} [g(z)] + \mathbb{E}_{S \sim D^m} [\Phi(S)] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Recall that,

\[ \hat{\mathbb{E}}_{z \sim uS} [g(z)] = \frac{1}{m} \sum_{i=1}^{m} g(z_i) \]
Proof of Main Result

\[ \Phi(S) = \sup_{g \in \mathcal{G}} \left( \mathbb{E}_{z \sim D} [g(z)] - \mathbb{E}_{z \sim uS} [g(z)] \right) \]

Alternatively, for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[ \Phi(S) \leq \mathbb{E}_{S \sim D^m} [\Phi(S)] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Thus, for any \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( g \in \mathcal{G} \),

\[ \mathbb{E}_{z \sim D} [g(z)] \leq \mathbb{E}_{z \sim uS} [g(z)] + \mathbb{E}_{S \sim D^m} [\Phi(S)] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Recall that,

\[ \mathbb{E}_{z \sim uS} [g(z)] = \frac{1}{m} \sum_{i=1}^{m} g(z_i) \]

Want to show

\[ \mathbb{E}_{z \sim D} [g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\mathcal{R}_m(\mathcal{G}) + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right). \]
Proof of Main Result

\[ \Phi(S) = \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \hat{\mathbb{E}}_{z \sim uS} [g(z)] \right) \]

Alternatively, for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[ \Phi(S) \leq \mathbb{E}_{S \sim D^m} [\Phi(S)] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Thus, for any \( \delta > 0 \), with probability at least \( 1 - \delta \), for every \( g \in G \),

\[ \mathbb{E}_{z \sim D} [g(z)] \leq \hat{\mathbb{E}}_{z \sim uS} [g(z)] + \mathbb{E}_{S \sim D^m} [\Phi(S)] + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Recall that,

\[ \hat{\mathbb{E}}_{z \sim uS} [g(z)] = \frac{1}{m} \sum_{i=1}^{m} g(z_i) \]

Want to show

\[ \mathbb{E}_{z \sim D} [g(z)] \leq \frac{1}{m} \sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(G) + O \left( \sqrt{\frac{\log(1/\delta)}{m}} \right) \]

Wouldn’t it be nice if \( \mathbb{E}_{S \sim D^m} [\Phi(S)] \leq 2\mathfrak{R}_m(G) \)?
Proof of Main Result

All that remains to show is that $\mathbb{E}_{S \sim D_m} [\Phi(S)] \leq 2\mathcal{R}_m(\mathcal{G})$.
Proof of Main Result

All that remains to show is that \( \mathbb{E}_{S \sim D_m} [\Phi(S)] \leq 2\mathcal{R}_m(G) \)

Consider

\[
\mathbb{E}_{S \sim D_m} [\Phi(S)] = \mathbb{E}_{S \sim D_m} \left[ \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \hat{\mathbb{E}}_{z \sim u | S} [g(z)] \right) \right]
\]
Proof of Main Result

All that remains to show is that \( \mathbb{E}_{S \sim D^m} [\Phi(S)] \leq 2\mathcal{R}_m(G) \)

Consider

\[
\mathbb{E}_{S \sim D^m} [\Phi(S)] = \mathbb{E}_{S \sim D^m} \left[ \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \hat{\mathbb{E}}_{z \sim u S} [g(z)] \right) \right]
\]

Introduce a fresh sample \( S' \sim D^m \)

\[
\mathbb{E}_{S \sim D^m} [\Phi(S')] = \mathbb{E}_{S \sim D^m} \left[ \sup_{g \in G} \left( \mathbb{E}_{S' \sim D^m} \left[ \hat{\mathbb{E}}_{z \sim u S'} [g(z)] \right] - \hat{\mathbb{E}}_{z \sim u S} [g(z)] \right) \right]
\]
Proof of Main Result

All that remains to show is that \( \mathbb{E}_{S \sim D^m} [\Phi(S)] \leq 2\mathcal{R}_m(G) \)

Consider

\[
\mathbb{E}_{S \sim D^m} [\Phi(S)] = \mathbb{E}_{S \sim D^m} \left[ \sup_{g \in G} \left( \mathbb{E}_{z \sim D} [g(z)] - \mathbb{E}_{z \sim u S} [g(z)] \right) \right]
\]

Introduce a fresh sample \( S' \sim D^m \)

\[
\mathbb{E}_{S \sim D^m} [\Phi(S)] = \mathbb{E}_{S \sim D^m} \left[ \sup_{g \in G} \left( \mathbb{E}_{S' \sim D^m} \left[ \mathbb{E}_{z \sim u S'} [g(z)] \right] - \mathbb{E}_{z \sim u S} [g(z)] \right) \right]
\]

Pushing the \( \sup \) inside the expectation

\[
\mathbb{E}_{S \sim D^m} [\Phi(S)] \leq \mathbb{E}_{S \sim D^m, S' \sim D^m} \left[ \sup_{g \in G} \left( \mathbb{E}_{z \sim u S'} [g(z)] - \mathbb{E}_{z \sim u S} [g(z)] \right) \right]
\]
Proof of Main Result

Pushing the $\sup$ inside the expectation

$$
\mathbb{E}_{S \sim D^m} [\Phi(S)] \leq \mathbb{E}_{S \sim D^m, S' \sim D^m} \left[ \sup_{g \in \mathcal{G}} \left( \hat{\mathbb{E}}_{z \sim u} [g(z)] - \hat{\mathbb{E}}_{z \sim u} [g(z)] \right) \right]
$$
Proof of Main Result

Pushing the $\sup$ inside the expectation

$$\mathbb{E}_{S \sim D^m} [\Phi(S')] \leq \mathbb{E}_{S \sim D^m, S' \sim D^m} \left[ \sup_{g \in G} \left( \mathbb{E}_{z \sim_u S'} [g(z)] - \mathbb{E}_{z \sim_u S} [g(z)] \right) \right]$$

$S$ and $S'$ are identically distributed, so their elements can be swapped by introducing Rademacher random variables $\sigma_i \in \{-1, 1\}$

$$\mathbb{E}_{S \sim D^m} [\Phi(S')] \leq \mathbb{E}_{S \sim D^m, S' \sim D^m, \sigma} \left[ \sup_{g \in G} \frac{1}{m} \sigma_i (g(z'_i) - g(z_i)) \right]$$

$$\leq 2 \mathbb{E}_{S \sim D^m, \sigma} \left[ \sup_{g \in G} \frac{1}{m} \sigma_i g(z_i) \right] = 2 \mathcal{R}_m(\mathcal{G})$$
Outline

Statistical (Supervised) Learning Theory Framework

Linear Regression

Rademacher Complexity

Support Vector Machines

Kernels

Neural Networks

Algorithmic Stability
Support Vector Machines: Binary Classification

Goal: Find a linear separator

Data is **linearly separable** if there exists a linear separator that classifies all points correctly.
Support Vector Machines: Binary Classification

Goal: Find a linear separator

Data is **linearly separable** if there exists a linear separator that classifies all points correctly

Which separator should be picked?
Support Vector Machines: Maximum Margin Principle

Maximise the distance of the closest point from the decision boundary

Points that are closest to the decision boundary are support vectors
Support Vector Machines: Geometric View

Given a hyperplane: \( H \equiv \mathbf{w} \cdot \mathbf{x} + w_0 = 0 \) and a point \( \mathbf{x} \in \mathbb{R}^n \), how far is \( \mathbf{x} \) from \( H \)?
Consider the hyperplane: \( H \equiv \mathbf{w} \cdot \mathbf{x} + w_0 = 0 \)

The distance of point \( \mathbf{x} \) from \( H \) is given by

\[
\frac{|\mathbf{w} \cdot \mathbf{x} + w_0|}{\|\mathbf{w}\|_2}
\]

All points on one side of the hyperplane satisfy (labelled \( y = +1 \))

\[
\mathbf{w} \cdot \mathbf{x} + w_0 \geq 0
\]

and points on the other side satisfy (labelled \( y = -1 \))

\[
\mathbf{w} \cdot \mathbf{x} + w_0 < 0
\]
SVM Formulation: Separable Case

minimise: \( \frac{1}{2} \|w\|_2^2 \)

subject to:
\[ y_i(w \cdot x_i + w_0) \geq 1 \]
for \( i = 1, \ldots, m \)

Here \( y_i \in \{-1, 1\} \)

If data is separable, then we find a classifier with no classification error on the training set.
The margin of the classifier is \( \frac{1}{\|w^*\|_2} \) if \( w^* \) is the optimal solution.
This is a convex quadratic program and hence can be solved efficiently.
SVM Formulation: The Dual

minimise: \( \frac{1}{2} ||w||^2 \)

subject to:
\[ y_i (w \cdot x_i + w_0) - 1 \geq 0 \]
for \( i = 1, \ldots, m \)

Here \( y_i \in \{-1, 1\} \)

maximise \( \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \)

subject to:
\[ \sum_{i=1}^{m} \alpha_i y_i = 0 \]
\[ 0 \leq \alpha_i \]
for \( i = 1, \ldots, m \)

Lagrange Function

\[ \Lambda(w, w_0; \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{m} \alpha_i (y_i (w \cdot x_i + w_0) - 1) \]

Complementary Slackness

\[ \alpha_i (y_i (w \cdot x_i + w_0) - 1) = 0, \quad i = 1, \ldots, m \]
SVM Formulation: Non-Separable Case

minimise: \( \frac{1}{2} \| \mathbf{w} \|_2^2 + C \sum_{i=1}^{m} \zeta_i \)

subject to:
\[
y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 - \zeta_i
\]
\[\zeta_i \geq 0\]
for \( i = 1, \ldots, m \)

Here \( y_i \in \{-1, 1\} \)
SVM Formulation: Loss Function

minimise: \[ \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{m} \zeta_i \]

subject to:
\[ y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 - \zeta_i \]
\[ \zeta_i \geq 0 \]
for \( i = 1, \ldots, m \)

Here \( y_i \in \{-1, 1\} \)
SVM Formulation : Loss Function

minimise: \[ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \zeta_i \]

subject to:

\[ y_i(w \cdot x_i + w_0) \geq 1 - \zeta_i \]

\[ \zeta_i \geq 0 \]

for \( i = 1, \ldots, m \)

Here \( y_i \in \{-1, 1\} \)
SVM Formulation: Loss Function

minimise: \[ \frac{1}{2} \| w \|_2^2 + C \sum_{i=1}^{m} \xi_i \]

subject to:
\[ y_i(w \cdot x_i + w_0) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]
for \( i = 1, \ldots, m \)

Here \( y_i \in \{-1, 1\} \)

Note that for the optimal solution, \( \xi_i = \max\{0, 1 - y_i(w \cdot x_i + w_0)\} \)

Thus, SVM can be viewed as minimising the hinge loss with regularisation
SVM : Deriving the Dual

minimise: \( \frac{1}{2} \| \mathbf{w} \|_2^2 + C \sum_{i=1}^{m} \zeta_i \)

subject to:
\[
y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) - (1 - \zeta_i) \geq 0
\]
\[
\zeta_i \geq 0
\]
for \( i = 1, \ldots, m \)

Here \( y_i \in \{-1, 1\} \)

Lagrange Function

\[
\Lambda(\mathbf{w}, w_0, \zeta; \alpha, \mu) = \frac{1}{2} \| \mathbf{w} \|_2^2 + C \sum_{i=1}^{m} \zeta_i - \sum_{i=1}^{m} \alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) - (1 - \zeta_i)) - \sum_{i=1}^{m} \mu_i \zeta_i
\]
SVM: Deriving the Dual

Lagrange Function

$$\Lambda(w, w_0, \zeta; \alpha, \mu) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \zeta_i - \sum_{i=1}^{m} \alpha_i (y_i (w \cdot x_i + w_0) - (1 - \zeta_i)) - \sum_{i=1}^{m} \mu_i \zeta_i$$

We write derivatives with respect to $w$, $w_0$ and $\zeta_i$,

$$\frac{\partial \Lambda}{\partial w_0} = - \sum_{i=1}^{m} \alpha_i y_i$$

$$\frac{\partial \Lambda}{\partial \zeta_i} = C - \alpha_i - \mu_i$$

$$\nabla_w \Lambda = w - \sum_{i=1}^{m} \alpha_i y_i x_i$$

For (KKT) dual feasibility constraints, we require $\alpha_i \geq 0, \mu_i \geq 0$
SVM: Deriving the Dual

Setting the derivatives to 0, substituting the resulting expressions in $\Lambda$ (and simplifying), we get a function $g(\alpha)$ and some constraints

$$g(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

Constraints

$$0 \leq \alpha_i \leq C \quad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

Finding critical points of $\Lambda$ satisfying the KKT conditions corresponds to finding the maximum of $g(\alpha)$ subject to the above constraints
### SVM: Primal and Dual Formulations

#### Primal Form

| minimise: $\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \zeta_i$ |
| subject to: $y_i (w \cdot x_i + w_0) \geq (1 - \zeta_i)$ |
| $\zeta_i \geq 0$ |
| for $i = 1, \ldots, m$ |

#### Dual Form

| maximise $\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$ |
| subject to: $\sum_{i=1}^{m} \alpha_i y_i = 0$ |
| $0 \leq \alpha_i \leq C$ |
| for $i = 1, \ldots, m$ |
KKT Complementary Slackness Conditions

For all $i$, $\alpha_i \left( y_i (w \cdot x_i + w_0) - (1 - \zeta_i) \right) = 0$

If $\alpha_i > 0$, $y_i (w \cdot x_i + w_0) = 1 - \zeta_i$
For all \( i \), \( \alpha_i \left( y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) - (1 - \zeta_i) \right) = 0 \)

If \( \alpha_i > 0 \), \( y_i (\mathbf{w} \cdot \mathbf{x}_i + w_0) = 1 - \zeta_i \)

Recall the form of the solution: \( \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \)

Thus, only those datapoints \( \mathbf{x}_i \) for which \( \alpha_i > 0 \), determine the solution

This is why they are called support vectors
Support Vectors
Suppose we solve the SVM objective by constraining \( \mathbf{w} \) to be in the set
\[
\{ \mathbf{w} | \| \mathbf{w} \|_2 \leq W \}
\]
Suppose we solve the SVM objective by constraining \( w \) to be in the set \( \{ w \mid \|w\|_2 \leq W\} \)

Consider the cost function \( \gamma_\rho : \mathbb{R} \times \{-1, 1\} \to [0, 1] \) defined as \( \gamma_\rho(y', y) = \varphi_\rho(yy') \), where \( \varphi_\rho : \mathbb{R} \to [0, 1] \) is defined as:

\[
\varphi_\rho(z) = \begin{cases} 
0 & \text{if } \rho \leq z \\
1 - z/\rho & \text{if } 0 \leq z \leq \rho \\
1 & \text{if } z \leq 0 
\end{cases}
\]
Generalisation Bounds Based on Margin

Suppose we solve the SVM objective by constraining $\mathbf{w}$ to be in the set $
\{ \mathbf{w} \mid \| \mathbf{w} \|_2 \leq W \}$

Consider the cost function $\gamma_\rho : \mathbb{R} \times \{-1, 1\} \to [0, 1]$ defined as $\gamma_\rho(y', y) = \varphi_\rho(yy')$, where $\varphi_\rho : \mathbb{R} \to [0, 1]$ is defined as:

$$\varphi_\rho(z) = \begin{cases} 
0 & \text{if } \rho \leq z \\
1 - z/\rho & \text{if } 0 \leq z \leq \rho \\
1 & \text{if } z \leq 0
\end{cases}$$

Let $\mathcal{H} = \{ \mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \mid \| \mathbf{w} \|_2 \leq W \}$ and let $\| \mathbf{x} \|_2 \leq X$ for all $\mathbf{x} \in X$, as $\varphi_\rho$ is $1/\rho$-Lipschitz by Talagrand’s Lemma we have

$$\hat{\mathcal{R}}(\varphi_\rho \circ \mathcal{H}) \leq \frac{WX}{\rho \sqrt{m}}$$
Generalisation Bounds Based on Margin

Let \( \gamma(y', y) = I(\text{sign}(y') \neq y) \) (zero-one loss) and \( \gamma_\rho(y', y) = \varphi_\rho(y'y) \). Observe that \( \gamma(y', y) \leq \gamma_\rho(y', y) \).
Generalisation Bounds Based on Margin

Let $\gamma(y', y) = \1(\text{sign}(y') \neq y)$ (zero-one loss) and $\gamma_\rho(y', y) = \varphi_\rho(y'y)$. Observe that $\gamma(y', y) \leq \gamma_\rho(y', y)$.

Let $R(h_w) = \mathbb{E}_{(x,y) \sim D} [\gamma(\text{sign}(w \cdot x), y)]$ and let $R_\rho(h_w) = \mathbb{E}_{(x,y) \sim D} [\gamma_\rho(w \cdot x, y)]$. Let $\hat{R}$ and $\hat{R}_\rho$ denote the corresponding empirical risks.
Generalisation Bounds Based on Margin

Let \(\gamma(y', y) = \mathbb{I}(\text{sign}(y') \neq y)\) (zero-one loss) and \(\gamma_\rho(y', y) = \varphi_\rho(y'y)\).

Observe that \(\gamma(y', y) \leq \gamma_\rho(y', y)\)

Let \(R(h_w) = \mathbb{E}_{(x,y) \sim D} [\gamma(\text{sign}(w \cdot x), y)]\) and let \(R_\rho(h_w) = \mathbb{E}_{(x,y) \sim D} [\gamma_\rho(w \cdot x, y)]\).

Let \(\hat{R}\) and \(\hat{R}_\rho\) denote the corresponding empirical risks.

Then, we have

\[
R(h) \leq R_\rho(h) \leq \hat{R}_\rho(h) + 2\hat{R}(\phi \circ \mathcal{H}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)
\]
Generalisation Bounds Based on Margin

Let $\gamma(y', y) = I(\text{sign}(y') \neq y)$ (zero-one loss) and $\gamma_\rho(y', y) = \varphi_\rho(y'y)$. Observe that $\gamma(y', y) \leq \gamma_\rho(y', y)$

Let $R(h_w) = \mathbb{E}_{(x,y) \sim D} [\gamma(\text{sign}(w \cdot x), y)]$ and let $R_\rho(h_w) = \mathbb{E}_{(x,y) \sim D} [\gamma_\rho(w \cdot x, y)]$. Let $\hat{R}$ and $\hat{R}_\rho$ denote the corresponding empirical risks

Then, we have

$$R(h) \leq R_\rho(h) \leq \hat{R}_\rho(h) + 2\hat{\mathcal{R}}(\phi \circ \mathcal{H}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$$

As $\hat{\mathcal{R}}(\phi \circ \mathcal{H}) = O(XW/\rho \sqrt{m})$, a sample size of $m = O(W^2X^2/(\rho \epsilon)^2)$ is sufficient to get $\epsilon$ excess risk (over $\hat{R}_\rho(h)$)
Generalisation Bounds Based on Margin

Let $\gamma(y', y) = \mathbb{I}(\text{sign}(y') \neq y)$ (zero-one loss) and $\gamma_\rho(y', y) = \varphi_\rho(y'y)$. Observe that $\gamma(y', y) \leq \gamma_\rho(y', y)$.

Let $R(h_w) = \mathbb{E}_{(x, y) \sim D} [\gamma(\text{sign}(w \cdot x), y)]$ and let $R_\rho(h_w) = \mathbb{E}_{(x, y) \sim D} [\gamma_\rho(w \cdot x, y)]$. Let $\widehat{R}$ and $\widehat{R}_\rho$ denote the corresponding empirical risks.

Then, we have

$$R(h) \leq R_\rho(h) \leq \widehat{R}_\rho(h) + 2\widehat{\mathcal{R}}(\phi \circ \mathcal{H}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)$$

As $\widehat{\mathcal{R}}(\phi \circ \mathcal{H}) = O(XW/\sqrt{m})$, a sample size of $m = O(W^2X^2/(\rho\epsilon)^2)$ is sufficient to get $\epsilon$ excess risk (over $\widehat{R}_\rho(h)$).

Note that solving the SVM objective is not guaranteed to give $h$ that has the smallest $\widehat{R}(h)$ (the problem of minimising disagreements with a linear separator is NP-hard).
Outline

Statistical (Supervised) Learning Theory Framework

Linear Regression

Rademacher Complexity

Support Vector Machines

Kernels

Neural Networks

Algorithmic Stability
If we put the inputs in matrix $X$, where the $i^{th}$ row of $X$ is $x_i^T$.

$$K = XX^T = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_m \\ x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_m \\ \vdots & \vdots & \ddots & \vdots \\ x_m^T x_1 & x_m^T x_2 & \cdots & x_m^T x_m \end{bmatrix}$$
Gram Matrix

If we put the inputs in matrix $X$, where the $i^{th}$ row of $X$ is $x_i^T$.

$$K = XX^T = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_m \\ x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_m \\ \vdots & \vdots & \ddots & \vdots \\ x_m^T x_1 & x_m^T x_2 & \cdots & x_m^T x_m \end{bmatrix}$$

The matrix $K$ is positive semi-definite
Gram Matrix

If we put the inputs in matrix $X$, where the $i^{th}$ row of $X$ is $x_i^T$.

$$K = XX^T = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_m \\ x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_m \\ \vdots & \vdots & \ddots & \vdots \\ x_m^T x_1 & x_m^T x_2 & \cdots & x_m^T x_m \end{bmatrix}$$

The matrix $K$ is positive semi-definite

If we perform basis expansion

$$\phi : \mathbb{R}^n \rightarrow \mathbb{R}^N$$

then replace entries by $\phi(x_i)^T \phi(x_j)$
Gram Matrix

If we put the inputs in matrix $X$, where the $i^{th}$ row of $X$ is $x_i^T$.

$$K = XX^T = \begin{bmatrix}
    x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_m \\
    x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_m \\
    \vdots & \vdots & \ddots & \vdots \\
    x_m^T x_1 & x_m^T x_2 & \cdots & x_m^T x_m
\end{bmatrix}$$

The matrix $K$ is positive semi-definite

If we perform basis expansion

$$\phi : \mathbb{R}^n \rightarrow \mathbb{R}^N$$

then replace entries by $\phi(x_i)^T \phi(x_j)$

We only need the ability to compute inner products to use (dual version of) SVM
Kernel Trick

Suppose, \( x \in \mathbb{R}^2 \) and we perform degree 2 polynomial expansion, we could use the map:

\[
\psi(x) = \left[ 1, x_1, x_2, x_1^2, x_2^2, x_1 x_2 \right]^T
\]
Kernel Trick

Suppose, $x \in \mathbb{R}^2$ and we perform degree 2 polynomial expansion, we could use the map:

$$
\psi(x) = \left[ 1, x_1, x_2, x_1^2, x_2^2, x_1x_2 \right]^T
$$

But, we could also use the map:

$$
\phi(x) = \left[ 1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2 \right]^T
$$
Kernel Trick

Suppose, $x \in \mathbb{R}^2$ and we perform degree 2 polynomial expansion, we could use the map:

$$
\psi(x) = \left[ 1, x_1, x_2, x_1^2, x_2^2, x_1x_2 \right]^T
$$

But, we could also use the map:

$$
\phi(x) = \left[ 1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2 \right]^T
$$

If $x = [x_1, x_2]^T$ and $x' = [x'_1, x'_2]^T$, then

$$
\phi(x)^T \phi(x') = 1 + 2x_1x'_1 + 2x_2x'_2 + x_1^2(x'_1)^2 + x_2^2(x'_2)^2 + 2x_1x_2x'_1x'_2 \\
= (1 + x_1x'_1 + x_2x'_2)^2 = (1 + x \cdot x')^2
$$
Kernel Trick

Suppose, \( x \in \mathbb{R}^2 \) and we perform degree 2 polynomial expansion, we could use the map:

\[
\psi(x) = \begin{bmatrix} 1, x_1, x_2, x_1^2, x_2^2, x_1 x_2 \end{bmatrix}^T
\]

But, we could also use the map:

\[
\phi(x) = \begin{bmatrix} 1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2 \end{bmatrix}^T
\]

If \( x = [x_1, x_2]^T \) and \( x' = [x'_1, x'_2]^T \), then

\[
\phi(x)^T \phi(x') = 1 + 2x_1 x'_1 + 2x_2 x'_2 + x_1^2 (x'_1)^2 + x_2^2 (x'_2)^2 + 2x_1 x_2 x'_1 x'_2
\]

\[
= (1 + x_1 x'_1 + x_2 x'_2)^2 = (1 + x \cdot x')^2
\]

Instead of spending \( \approx n^d \) time to compute inner products after degree \( d \) polynomial basis expansion, we only need \( O(n) \) time
Kernel Trick

We can use a symmetric positive semi-definite kernel (Mercer Kernels)

\[ K = \begin{bmatrix}
\kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots & \kappa(x_1, x_m) \\
\kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots & \kappa(x_2, x_m) \\
\vdots & \vdots & \ddots & \vdots \\
\kappa(x_m, x_1) & \kappa(x_m, x_2) & \cdots & \kappa(x_m, x_m)
\end{bmatrix} \]

Here \( \kappa(x, x') \) is some measure of similarity between \( x \) and \( x' \)
Kernel Trick

We can use a symmetric positive semi-definite kernel (Mercer Kernels)

\[
K = \begin{bmatrix}
\kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots & \kappa(x_1, x_m) \\
\kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots & \kappa(x_2, x_m) \\
\vdots & \vdots & \ddots & \vdots \\
\kappa(x_m, x_1) & \kappa(x_m, x_2) & \cdots & \kappa(x_m, x_m) 
\end{bmatrix}
\]

Here \( \kappa(x, x') \) is some measure of similarity between \( x \) and \( x' \)

The dual program becomes

\[
\text{maximise } \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K_{i,j}
\]

subject to: \( 0 \leq \alpha_i \leq C \) and \( \sum_{i=1}^{m} \alpha_i y_i = 0 \)
Kernel Trick

We can use a symmetric positive semi-definite kernel (Mercer Kernels)

\[
K = \begin{bmatrix}
\kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots & \kappa(x_1, x_m) \\
\kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots & \kappa(x_2, x_m) \\
\vdots & \vdots & \ddots & \vdots \\
\kappa(x_m, x_1) & \kappa(x_m, x_2) & \cdots & \kappa(x_m, x_m)
\end{bmatrix}
\]

Here \(\kappa(x, x')\) is some measure of similarity between \(x\) and \(x'\)

The dual program becomes

\[
\text{maximise} \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K_{i,j}
\]

subject to: \(0 \leq \alpha_i \leq C\) and \(\sum_{i=1}^{m} \alpha_i y_i = 0\)

To make prediction on new \(x_{\text{new}}\), only need to compute \(\kappa(x_i, x_{\text{new}})\) for support vectors \(x_i\) (for which \(\alpha_i > 0\)
Polynomial Kernels

Rather than perform basis expansion,

\[ \kappa(x, x') = (1 + x \cdot x')^d \]

This gives all terms of degree up to \( d \)

If we use \( \kappa(x, x') = (x \cdot x')^d \), we get only degree \( d \) terms

**Linear Kernel:** \( \kappa(x, x') = x \cdot x' \)

All of these satisfy the Mercer or positive-definite condition
Gaussian or RBF Kernel

Radial Basis Function (RBF) or Gaussian Kernel

$$\kappa(x, x') = \exp \left( -\frac{\|x - x'\|^2}{2\sigma^2} \right)$$

$\sigma^2$ is known as the bandwidth

We used this with $\gamma = \frac{1}{2\sigma^2}$ when we studied kernel basis expansion for regression

Can generalise to more general covariance matrices

Results in a Mercer kernel
Outline

Statistical (Supervised) Learning Theory Framework

Linear Regression

Rademacher Complexity

Support Vector Machines

Kernels

Neural Networks

Algorithmic Stability
A unit in a neural network computes an affine function of its input and is then composed with a non-linear activation function.

For example, the activation function could be the logistic sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$
Neural Networks: Unit

A unit in a neural network computes an affine function of its input and is then composed with a non-linear activation function.

For example, the activation function could be the logistic sigmoid:

\[ \sigma(z) = \frac{1}{1 + e^{-z}} \]

where \( z = b + w \cdot x \) and \( x \) is the input, \( w \) is the weight, and \( b \) is the bias.
A unit in a neural network computes an affine function of its input and is then composed with a non-linear activation function. For example, the activation function could be the logistic sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$
A unit in a neural network computes an affine function of its input and is then composed with a non-linear activation function $a$.

For example, the activation function could be the logistic sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$
Feedforward Neural Networks

Layer 1  (Input)  Layer 2  (Hidden)  Layer 3  (Hidden)  Layer 4  (Output)

Fully Connected Layer
Neural Networks

Only consider fully-connected, feed-forward neural networks, with non-linear activation functions applied element-wise to units.
Neural Networks

Only consider fully-connected, feed-forward neural networks, with non-linear activation functions applied element-wise to units.

A layer $l : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ consists of an element-wise composition of a non-linear activation $a$, e.g. rectifier or logistic sigmoid, and an affine map

\[ l(z) = a(Wz + b) \]
Neural Networks

Only consider fully-connected, feed-forward neural networks, with non-linear activation functions applied element-wise to units

A layer $l : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}$ consists of an element-wise composition of a non-linear activation $a$, e.g. rectifier or logistic sigmoid, and an affine map

$$l(z) = a(Wz + b)$$

An $L$-hidden layer network represents a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$f(x) = w \cdot l_L(l_{L-1}(\cdots(l_1(x)\cdots) + b$$
Neural Networks

Only consider fully-connected, feed-forward neural networks, with non-linear activation functions applied element-wise to units.

A layer \( l : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2} \) consists of an element-wise composition of a non-linear activation \( a \), e.g. rectifier or logistic sigmoid, and an affine map

\[
l(z) = a(Wz + b)
\]

An \( L \)-hidden layer network represents a function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \)

\[
f(x) = w \cdot l_L(l_{L-1}( \cdots (l_1(x) \cdots ) + b)
\]

Typically, the output layer is simply an affine map of the penultimate layer (without any non-linear activation)
## Capacity of Neural Networks

<table>
<thead>
<tr>
<th>VC dim. of Neural Nets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informally, if $a$ is the $\text{sgn}$ function, and $C$ is the class of all neural networks with at most $\omega$ parameters then, $\text{VC}(C) \leq 2\omega \log_2(e\omega)$</td>
</tr>
</tbody>
</table>
## Capacity of Neural Networks

### VC dim. of Neural Nets
Informally, if \( a \) is the \( \text{sgn} \) function, and \( C \) is the class of all neural networks with at most \( \omega \) parameters then, \( \text{VC}(C) \leq 2\omega \log_2(e\omega) \)

### Rademacher Complexity of Neural Nets
Suppose \( F \) is the class of feed-forward neural nets with \( L - 1 \) hidden layers
- every row of \( w \) of any \( W \) in the net satisfying \( \|w\|_1 \leq W \)
### Capacity of Neural Networks

#### VC dim. of Neural Nets
Informally, if \( a \) is the \( \text{sgn} \) function, and \( C \) is the class of all neural networks with at most \( \omega \) parameters then, 
\[
\text{VC}(C) \leq 2\omega \log_2(e\omega)
\]

#### Rademacher Complexity of Neural Nets
Suppose \( \mathcal{F} \) is the class of feed-forward neural nets with \( L - 1 \) hidden layers
- every row of \( w \) of any \( W \) in the net satisfying \( \|w\|_1 \leq W \)
- every bias vector \( b \) satisfying \( \|b\|_\infty \leq B \)
## Capacity of Neural Networks

### VC dim. of Neural Nets
Informally, if $a$ is the $\text{sgn}$ function, and $C$ is the class of all neural networks with at most $\omega$ parameters then, $VC(C) \leq 2\omega \log_2(e\omega)$

### Rademacher Complexity of Neural Nets
Suppose $\mathcal{F}$ is the class of feed-forward neural nets with $L - 1$ hidden layers
- every row of $w$ of any $W$ in the net satisfying $||w||_1 \leq W$
- every bias vector $b$ satisfying $||b||_\infty \leq B$
- the activations $a$ being 1-Lipschitz
### VC dim. of Neural Nets

Informally, if $a$ is the $\text{sgn}$ function, and $C$ is the class of all neural networks with at most $\omega$ parameters then, $\text{VC}(C) \leq 2\omega \log_2(e\omega)$

### Rademacher Complexity of Neural Nets

Suppose $F$ is the class of feed-forward neural nets with $L - 1$ hidden layers

- every row of $w$ of any $W$ in the net satisfying $\|w\|_1 \leq W$
- every bias vector $b$ satisfying $\|b\|_\infty \leq B$
- the activations $a$ being 1-Lipschitz
- and furthermore, the inputs $x \in \mathcal{X}$ satisfying $\|x\|_\infty \leq 1$
Capacity of Neural Networks

**VC dim. of Neural Nets**
Informally, if $a$ is the $\text{sgn}$ function, and $C$ is the class of all neural networks with at most $\omega$ parameters then, $\text{VC}(C) \leq 2\omega \log_2(e\omega)$

**Rademacher Complexity of Neural Nets**
Suppose $\mathcal{F}$ is the class of feed-forward neural nets with $L-1$ hidden layers
- every row of $w$ of any $W$ in the net satisfying $\|w\|_1 \leq W$
- every bias vector $b$ satisfying $\|b\|_\infty \leq B$
- the activations $a$ being 1-Lipschitz
- and furthermore, the inputs $x \in \mathcal{X}$ satisfying $\|x\|_\infty \leq 1$

Then, $\hat{\mathcal{R}}_m(\mathcal{F}) \leq \frac{1}{\sqrt{m}} \left( (2W)^L + B \sum_{i=0}^{L-1} (2W)^i \right)$
Capacity of Neural Networks

**VC dim. of Neural Nets**

Informally, if \( a \) is the \( \text{sgn} \) function, and \( C \) is the class of all neural networks with at most \( \omega \) parameters then, \( \text{VC}(C) \leq 2\omega \log_2(e\omega) \)

**Rademacher Complexity of Neural Nets**

Suppose \( F \) is the class of feed-forward neural nets with \( L - 1 \) hidden layers

\[
\begin{align*}
\text{▶ } & \text{ every row of } w \text{ of any } W \text{ in the net satisfying } \|w\|_1 \leq W \\
\text{▶ } & \text{ every bias vector } b \text{ satisfying } \|b\|_\infty \leq B \\
\text{▶ } & \text{ the activations } a \text{ being } 1\text{-Lipschitz} \\
\text{▶ } & \text{ and furthermore, the inputs } x \in \mathcal{X} \text{ satisfying } \|x\|_\infty \leq 1
\end{align*}
\]

Then, \( \hat{\mathcal{R}}_m(F) \leq \frac{1}{\sqrt{m}} \left( (2W)^L + B \sum_{i=0}^{L-1} (2W)^i \right) \)

Exercise: Prove this using the fact that if \( \mathcal{G} \) is the function class consisting of all convex combinations of functions in \( G \), then \( \hat{\mathcal{R}}_m(\mathcal{G}) = \hat{\mathcal{R}}(\mathcal{G}) \). (Also prove the latter claim.)
(Simplified) Theorem (Cybenko)\(^6\)

Let \( \sigma \) be the logistic sigmoid activation function. Then the set of functions \( G(x) = \sum_{i=1}^{N} \alpha_j \sigma(w_j \cdot x + b_j) \) are dense in the set of continuous functions on \([0, 1]^n\).
Neural Networks: Universality Results

(Simplified) Theorem (Cybenko)\(^6\)

Let \(\sigma\) be the logistic sigmoid activation function. Then the set of functions
\[ G(x) = \sum_{i=1}^{N} \alpha_j \sigma(w_j \cdot x + b_j) \]
are dense in the set of continuous functions on \([0, 1]^n\).

Several other authors proved similar results roughly at the same time\(^1,8,11\)
(Simplified) Theorem (Cybenko)\textsuperscript{6}

Let $\sigma$ be the logistic sigmoid activation function. Then the set of functions $G(x) = \sum_{i=1}^{N} \alpha_j \sigma(w_j \cdot x + b_j)$ are dense in the set of continuous functions on $[0, 1]^n$.

Several other authors proved similar results roughly at the same time\textsuperscript{1,8,11}

Doesn’t give an explicit upper bound on the number of units required
**Neural Networks : Universality Results**

**Simplified Theorem (Cybenko)**

Let $\sigma$ be the logistic sigmoid activation function. Then the set of functions $G(x) = \sum_{i=1}^{N} \alpha_j \sigma(w_j \cdot x + b_j)$ are dense in the set of continuous functions on $[0, 1]^n$.

Several other authors proved similar results roughly at the same time $^{1,8,11}$

Doesn’t give an explicit upper bound on the number of units required

Known that the number of units required can be exponential for arbitrary continuous functions
(Simplified) Theorem (Cybenko)\(^6\)

| Let \(\sigma\) be the logistic sigmoid activation function. Then the set of functions 
| \(G(x) = \sum_{i=1}^{N} \alpha_j \sigma(w_j \cdot x + b_j)\) are dense in the set of continuous functions 
| on \([0, 1]^n\).

Several other authors proved similar results roughly at the same time\(^{1,8,11}\)

Doesn’t give an explicit upper bound on the number of units required

Known that the number of units required can be exponential for arbitrary continuous functions

These kinds of results don’t inform us directly about the success of training algorithms or the possibility of generalisation
Universality results establish that neural nets with one hidden layer are universal approximators.

Establishing the benefits of depth (both for representation and learning) is an active area of research.
Universality results establish that neural nets with one hidden layer are universal approximators.

Establishing the benefits of depth (both for representation and learning) is an active area of research.

Eldan and Shamir\(^7\) established the existence of a function that can be well approximated by a depth-3 (2 hidden layers) neural network using polynomially many units (in dimension), but requires exponentially many units using a depth-2 network.

Telgarsky\(^{15}\) established for each \(k \in \mathbb{N}\), the existence of a function that can be well approximated by a depth-\(k^3\) neural network using polynomially many units (in dimension and \(k\)), but requires exponentially many units using a depth-\(k\) neural network.
Outline

Statistical (Supervised) Learning Theory Framework

Linear Regression

Rademacher Complexity

Support Vector Machines

Kernels

Neural Networks

Algorithmic Stability
Algorithmic Stability

So far we have seen uniform convergence bounds, i.e. bounds of the form that “under suitable conditions” with high probability, \( \forall f \in \mathcal{F} \),

\[
R(f) \leq \hat{R}_S(f) + \epsilon
\]
Algorithmic Stability

So far we have seen uniform convergence bounds, i.e. bounds of the form that “under suitable conditions” with high probability, \( \forall f \in \mathcal{F} \),

\[
R(f) \leq \hat{R}_S(f) + \epsilon
\]

These results only depend on certain complexity/capacity measures of the class of functions \( \mathcal{F} \) used by the learning algorithm.
Algorithmic Stability

So far we have seen uniform convergence bounds, i.e. bounds of the form that “under suitable conditions” with high probability, ∀f ∈ ℱ,

\[ R(f) \leq \hat{R}_S(f) + \epsilon \]

These results only depend on certain complexity/capacity measures of the class of functions ℱ used by the learning algorithm

Q. Can analysing learning algorithms directly yield a (possibly different/better) way to obtain bounds on the true risk?
Algorithmic Stability

Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a sample drawn from $D$ over $\mathcal{X} \times \mathcal{Y}$ and $S'$ be a sample that differs from $S$ on exactly one point, say it has $(x'_m, y'_m)$ instead of $(x_m, y_m)$.
Algorithmic Stability

Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a sample drawn from $D$ over $\mathcal{X} \times \mathcal{Y}$ and $S'$ be a sample that differs from $S$ on exactly one point, say it has $(x'_m, y'_m)$ instead of $(x_m, y_m)$.

A (possibly randomised) learning algorithm $A$ takes a sample $S$ as input and outputs a function $f_S = A(S)$. 
Algorithmic Stability

Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a sample drawn from $D$ over $\mathcal{X} \times \mathcal{Y}$ and $S'$ be a sample that differs from $S$ on exactly one point, say it has $(x'_m, y'_m)$ instead of $(x_m, y_m)$.

A (possibly randomised) learning algorithm $A$ takes a sample $S$ as input and outputs a function $f_S = A(S)$.

Recall that $\gamma : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is the cost function.
Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a sample drawn from $D$ over $\mathcal{X} \times \mathcal{Y}$ and $S'$ be a sample that differs from $S$ on exactly one point, say it has $(x'_m, y'_m)$ instead of $(x_m, y_m)$

A (possibly randomised) learning algorithm $A$ takes a sample $S$ as input and outputs a function $f_S = A(S)$

Recall that $\gamma : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ is the cost function

### Uniform Stability

A learning algorithm $A$ is uniformly $\beta$-stable if for any samples $S, S'$ of size $m$, differing in exactly one point, it holds for every $(x, y)$ that:

$$|\gamma(f_S(x), y) - \gamma(f_{S'}(x), y)| \leq \beta$$
Algorithmic Stability

Uniform Stability

A learning algorithm $A$ is uniformly $\beta$-stable if for any samples $S, S'$ of size $m$, differing in exactly one point, it holds for every $(x, y)$ that:

$$|\gamma(f_S(x), y) - \gamma(f_{S'}(x), y)| \leq \beta$$
Algorithmic Stability

Uniform Stability

A learning algorithm $A$ is uniformly $\beta$-stable if for any samples $S, S'$ of size $m$, differing in exactly one point, it holds for every $(x, y)$ that:

$$|\gamma(f_S(x), y) - \gamma(f_{S'}(x), y)| \leq \beta$$

Theorem (Bousquet & Elisseeff)³

Suppose $\gamma$ is a bounded cost function $|\gamma| \leq M$ and that $A$ is uniformly $\beta$-stable. Let $S \sim D^m$, then for every $\delta > 0$, with probability at least $1 - \delta$, it holds that:

$$R(f_S) \leq \hat{R}_S(f_S) + \beta + (2m\beta + M)\sqrt{\frac{\log(1/\delta)}{2m}}$$

Clearly we need $\beta = o\left(\frac{1}{\sqrt{m}}\right)$ to get a non-trivial bound.

Cannot be used for zero-one classification loss.
Algorithmic Stability

**Uniform Stability**

A learning algorithm $A$ is uniformly $\beta$-stable if for any samples $S, S'$ of size $m$, differing in exactly one point, it holds for every $(x, y)$ that:

$$|\gamma(f_S(x), y) - \gamma(f_{S'}(x), y)| \leq \beta$$

**Theorem (Bousquet & Elisseeff)**

Suppose $\gamma$ is a bounded cost function $|\gamma| \leq M$ and that $A$ is uniformly $\beta$-stable. Let $S \sim D^m$, then for every $\delta > 0$, with probability at least $1 - \delta$, it holds that:

$$R(f_S) \leq \hat{R}_S(f_S) + \beta + (2m\beta + M)\sqrt{\frac{\log(1/\delta)}{2m}}$$

Clearly we need $\beta = o(1/\sqrt{m})$ to get a non-trivial bound.
Algorithmic Stability

Uniform Stability

A learning algorithm $A$ is uniformly $\beta$-stable if for any samples $S, S'$ of size $m$, differing in exactly one point, it holds for every $(x, y)$ that:

$$|\gamma(f_S(x), y) - \gamma(f_{S'}(x), y)| \leq \beta$$

Theorem (Bousquet & Elisseeff)$^3$

Suppose $\gamma$ is a bounded cost function $|\gamma| \leq M$ and that $A$ is uniformly $\beta$-stable. Let $S \sim D^m$, then for every $\delta > 0$, with probability at least $1 - \delta$, it holds that:

$$R(f_S) \leq \hat{R}_S(f_S) + \beta + (2m\beta + M)\sqrt{\frac{\log(1/\delta)}{2m}}$$

Clearly we need $\beta = o(1/\sqrt{m})$ to get a non-trivial bound

Cannot be used for zero-one classification loss
Algorithmic Stability

A cost function $\gamma$ is $\sigma$-admissible with respect to a class of function $\mathcal{F}$, if for every $f, f' \in \mathcal{F}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, it is the case that

$$\gamma(f'(x), y) - f(x, y) \leq \sigma |f'(x) - f(x)|$$
Algorithmic Stability

A cost function $\gamma$ is $\sigma$-admissible with respect to a class of function $\mathcal{F}$, if for every $f, f' \in \mathcal{F}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, it is the case that

$$\gamma(f'(x), y) - f(x, y) \leq \sigma |f'(x) - f(x)|$$

Example of Ridge Regression

The ridge regression method finds

$$\hat{w} = \arg\min_{w \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2 + \lambda \|w\|_2^2$$
Algorithmic Stability

A cost function $\gamma$ is $\sigma$-admissible with respect to a class of function $\mathcal{F}$, if for every $f, f' \in \mathcal{F}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, it is the case that

$$\gamma(f'(x), y) - f(x, y) \leq \sigma |f'(x) - f(x)|$$

Example of Ridge Regression

The ridge regression method finds

$$\hat{w} = \arg\min_{w \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2 + \lambda \|w\|_2^2$$

If $\|w\|_2 \leq X$ and if $\mathcal{Y} = [-M, M]$, then it is easy to see that any minimiser $\hat{w}$ has to satisfy $\|w\|_2^2 \leq M^2/\lambda$
Algorithmic Stability

A cost function $\gamma$ is $\sigma$-admissible with respect to a class of function $\mathcal{F}$, if for every $f, f' \in \mathcal{F}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, it is the case that

$$\gamma(f'(x), y) - f(x, y) \leq \sigma |f'(x) - f(x)|$$

Example of Ridge Regression

The ridge regression method finds

$$\hat{w} = \arg\min_{w \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2 + \lambda \|w\|^2$$

If $\|w\|_2 \leq X$ and if $\mathcal{Y} = [-M, M]$, then it is easy to see that any minimiser $\hat{w}$ has to satisfy $\|w\|^2_2 \leq M^2 / \lambda$

Consequently, $\gamma(y', y) = (y' - y)^2$ is $\sigma$-admissible for the class of functions that can be solutions to the ridge regression problem with $\sigma = 2(MX / \sqrt{\lambda} + M)$
Algorithmic Stability

A cost function $\gamma$ is $\sigma$-admissible with respect to a class of function $\mathcal{F}$, if for every $f, f' \in \mathcal{F}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, it is the case that

$$\gamma(f'(x), y) - f(x, y) \leq \sigma |f'(x) - f(x)|$$

Example of Ridge Regression

The ridge regression method finds

$$\hat{w} = \arg\min_{w \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2 + \lambda \|w\|_2^2$$

If $\|w\|_2 \leq X$ and if $\mathcal{Y} = [-M, M]$, then it is easy to see that any minimiser $\hat{w}$ has to satisfy $\|w\|_2^2 \leq M^2/\lambda$

Consequently, $\gamma(y', y) = (y' - y)^2$ is $\sigma$-admissible for the class of functions that can be solutions to the ridge regression problem with $\sigma = 2(MX/\sqrt{\lambda} + M)$

Theorem. Since $\gamma$ is convex and $\sigma$-admissible, ridge regression is uniformly $\beta$-stable with $\beta \leq \frac{\sigma^2 X^2}{m \lambda} = O(1/m)$
A cost function $\gamma$ is $\sigma$-admissible with respect to a class of function $\mathcal{F}$, if for every $f, f' \in \mathcal{F}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, it is the case that

$$\gamma(f'(x), y) - f(x, y)| \leq \sigma |f'(x) - f(x)|$$

**Example of Ridge Regression**

The ridge regression method finds

$$\hat{w} = \arg \min_{w \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2 + \lambda \|w\|_2^2$$

If $\|w\|_2 \leq X$ and if $\mathcal{Y} = [-M, M]$, then it is easy to see that any minimiser $\hat{w}$ has to satisfy $\|w\|_2^2 \leq M^2/\lambda$

Consequently, $\gamma(y', y) = (y' - y)^2$ is $\sigma$-admissible for the class of functions that can be solutions to the ridge regression problem with $\sigma = 2(MX/\sqrt{\lambda} + M)$

**Theorem.** Since $\gamma$ is convex and $\sigma$-admissible, ridge regression is uniformly $\beta$-stable with $\beta \leq \frac{\sigma^2 X^2}{m \lambda} = O(1/m)$

Recent work by Hardt et al.\(^9\) has shown that stochastic gradient descent (with early stopping) is uniformly stable
Summary

Uniform convergence bounds for bounding generalisation error using Rademacher complexity bounds

Application of Rademacher complexity bounds to Linear Regression, GLMs, SVMs

A brief view of some results about neural networks

Algorithmic stability as a means to bound generalisation error
References I


References II


