Statistical (Supervised) Learning Theory

FoPPS Logic and Learning School

Lady Margaret Hall   University of Oxford

Varun Kanade

July 1, 2018

Previous Mini-Course

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

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

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

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 $\mathcal{X} \times \mathcal{Y}$

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: $D(\mathbf{x}, y) = D(\mathbf{x})D(y|\mathbf{x})$

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: $D(\mathbf{x}, y) = D(\mathbf{x})D(y|\mathbf{x})$

Make no assumptions about a specific functional relationship between $\mathbf{x}$ 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)

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

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

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

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

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 $D(\mathbf{x})$ and/or $D(y|\mathbf{x})$

Aside : Connections to classical Statistics/ML

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

Generative Models: Model the full joint distribution $D(\mathbf{x}, y)$

- Gaussian Discriminant Analysis, Naïve Bayes

Aside : Connections to classical Statistics/ML

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

Generative Models: Model the full joint distribution $D(\mathbf{x}, y)$

- Gaussian Discriminant Analysis, Naïve Bayes

Discriminative Models: Model only the conditional distribution $D(y|\mathbf{x})$

- Linear Regression: $y|w_0, \mathbf{w}, \mathbf{x} \sim w_0 + \mathbf{w} \cdot \mathbf{x} + \mathcal{N}(0, \sigma^2)$
- Classification: $y|w_0, \mathbf{w}, \mathbf{x} \sim 2 \cdot \mathrm{Bernoulli}(\mathrm{sigmoid}(w_0 + \mathbf{w} \cdot \mathbf{x})) - 1$

Aside : Connections to classical Statistics/ML

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

Generative Models: Model the full joint distribution $D(\mathbf{x}, y)$

- Gaussian Discriminant Analysis, Naïve Bayes

Discriminative Models: Model only the conditional distribution $D(y|\mathbf{x})$

- Linear Regression: $y|w_0, \mathbf{w}, \mathbf{x} \sim w_0 + \mathbf{w} \cdot \mathbf{x} + \mathcal{N}(0, \sigma^2)$
- Classification: $y|w_0, \mathbf{w}, \mathbf{x} \sim 2 \cdot \mathrm{Bernoulli}(\mathrm{sigmoid}(w_0 + \mathbf{w} \cdot \mathbf{x})) - 1$

The (basic) PAC model in CLT assumes a functional form, $y = c(\mathbf{x})$, for some concept c in class C, and the VC dimension of C controls learnability.

Statistical (Supervised) Learning Theory Framework

$\mathcal{X}$ instance space; $\mathcal{Y}$ target values

Distribution D over $\mathcal{X} \times \mathcal{Y}$

Statistical (Supervised) Learning Theory Framework

$\mathcal{X}$ instance space; $\mathcal{Y}$ target values

Distribution D over $\mathcal{X} \times \mathcal{Y}$

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

Statistical (Supervised) Learning Theory Framework

$\mathcal{X}$ instance space; $\mathcal{Y}$ target values

Distribution D over $\mathcal{X} \times \mathcal{Y}$

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 $p \geq 1$

Statistical (Supervised) Learning Theory Framework

$\mathcal{X}$ instance space; $\mathcal{Y}$ target values

Distribution D over $\mathcal{X} \times \mathcal{Y}$

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 $(\mathbf{x}, y)$ is given by

$$\ell(f; \mathbf{x}, y) = \gamma(f(\mathbf{x}), y)$$

Statistical (Supervised) Learning Theory Framework

$\mathcal{X}$ instance space; $\mathcal{Y}$ target values

Distribution D over $\mathcal{X} \times \mathcal{Y}$

Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ 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. $\mathcal{Y} = \mathbb{R}$, $\gamma(y', y) = \left|y' - y\right|^p$ for $p \geq 1$

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

$$\ell(f; \mathbf{x}, y) = \gamma(f(\mathbf{x}), y)$$

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

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

Statistical (Supervised) Learning Theory Framework

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

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

Statistical (Supervised) Learning Theory Framework

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

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

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

Statistical (Supervised) Learning Theory Framework

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

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

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

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

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

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

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, $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ a sample of size m drawn i.i.d. (independent and identically distributed) from D

Statistical (Supervised) Learning Theory Framework

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

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

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, $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ 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:

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

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, $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$ 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 $\widehat{f} = A(S)$, then for some small $\epsilon > 0$:

$$R(\widehat{f}) \leq \inf_{f\in\mathcal{F}} R(f) + \epsilon$$

Empirical Risk Minimisation

Training sample $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$

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

Empirical Risk Minimisation

Training sample $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$

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

Define the empirical risk on a sample S as:

$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

Empirical Risk Minimisation

Training sample $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$

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

Define the empirical risk on a sample S as:

$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

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

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

Empirical Risk Minimisation

Training sample $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$

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

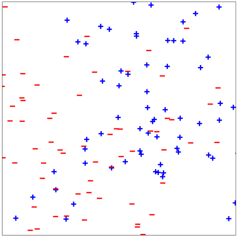
Define the empirical risk on a sample S as:

$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

ERM (Empirical Risk Minimisation) principle suggests that we find $f \in \mathcal{F}$ 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



Empirical Risk Minimisation

Training sample $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$

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

Define the empirical risk on a sample S as:

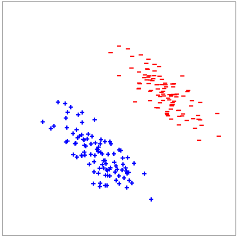
$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

ERM (Empirical Risk Minimisation) principle suggests that we find $f \in \mathcal{F}$ 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

$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

Empirical Risk Minimisation

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

$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

- 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

$$\widehat{R}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

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

$$\widehat{R}_S(f) = \frac{1}{m}\sum_{i=1}^{m} \gamma(f(\mathbf{x}_i), y_i)$$

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

Theorem (Vapnik, Chervonenkis) [14,16]

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

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

Empirical Risk Minimisation

Theorem (Vapnik, Chervonenkis) [14,16]

Let $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$ with $\mathrm{VC}(\mathcal{F}) = d < \infty$. 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 $f \in \mathcal{F}$,

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

Empirical Risk Minimisation

Theorem (Vapnik, Chervonenkis) [14,16]

Let $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$ with $\mathrm{VC}(\mathcal{F}) = d < \infty$. 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 $f \in \mathcal{F}$,

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

Suppose f^* is the "minimiser" of the true risk R and $\widehat{f}$ is the minimiser of the empirical risk $\widehat{R}_S$

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 $f \in \mathcal{F}$,

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

Suppose f^* is the "minimiser" of the true risk R and $\widehat{f}$ is the minimiser of the empirical risk $\widehat{R}_S$

Then, we have,

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

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 $f \in \mathcal{F}$,

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

Suppose f^* is the "minimiser" of the true risk R and $\widehat{f}$ is the minimiser of the empirical risk $\widehat{R}_S$

Then, we have,

$$\begin{aligned} R(\widehat{f}) &\leq \widehat{R}_S(\widehat{f}) + \epsilon/2 && \text{Using Theorem} \\ &\leq \widehat{R}_S(f^*) + \epsilon/2 && \text{As } \widehat{f} \text{ minimises } \widehat{R}_S \end{aligned}$$

Empirical Risk Minimisation

Theorem (Vapnik, Chervonenkis) [14,16]

Let $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$ with $\mathrm{VC}(\mathcal{F}) = d < \infty$. 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 $f \in \mathcal{F}$,

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

Suppose f^* is the "minimiser" of the true risk R and $\widehat{f}$ is the minimiser of the empirical risk $\widehat{R}_S$

Then, we have,

$$\begin{aligned} R(\widehat{f}) &\leq \widehat{R}_S(\widehat{f}) + \epsilon/2 && \text{Using Theorem} \\ &\leq \widehat{R}_S(f^*) + \epsilon/2 && \text{As } \widehat{f} \text{ minimises } \widehat{R}_S \\ &\leq R(f^*) + \epsilon && \text{Using Theorem (flipped)} \end{aligned}$$

Empirical Risk Minimisation

Theorem (Vapnik, Chervonenkis) [14,16]

Let $\mathcal{F} \subset \{0,1\}^{\mathcal{X}}$ with $\mathrm{VC}(\mathcal{F}) = d < \infty$. 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 $f \in \mathcal{F}$,

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

Suppose f^* is the "minimiser" of the true risk R and $\widehat{f}$ is the minimiser of the empirical risk $\widehat{R}_S$

Then, we have,

$$\begin{aligned} R(\widehat{f}) &\leq \widehat{R}_S(\widehat{f}) + \epsilon/2 && \text{Using Theorem} \\ &\leq \widehat{R}_S(f^*) + \epsilon/2 && \text{As } \widehat{f} \text{ minimises } \widehat{R}_S \\ &\leq R(f^*) + \epsilon && \text{Using Theorem (flipped)} \end{aligned}$$

Where ϵ is chosen to be a suitable function of δ and m

Structural Risk Minimisation

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

Structural Risk Minimisation

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

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

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

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:

$$\widehat{f} = \operatorname*{argmin}_{f \in \mathcal{F}_d, d \in \mathbb{N}} \widehat{R}_S(f) + \kappa(d, m)$$

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

Structural Risk Minimisation

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

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 $\{\mathcal{F}_d : d = 1, 2, \dots\}$ and find the minimiser:

$$\widehat{f} = \underset{f \in \mathcal{F}_d, d \in \mathbb{N}}{\text{argmin}} \ \widehat{R}_S(f) + \kappa(d, m)$$

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:

$$\widehat{f} = \underset{f \in \mathcal{F}}{\text{argmin}} \ \widehat{R}_S(f) + \lambda \cdot \text{regulariser}(f)$$

Outline

Linear Regression

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

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

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:

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

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 $g(x) = \mathbb{E}[y \mid \mathbf{x}]$

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 $g(x) = \mathbb{E}[y \mid \mathbf{x}]$

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

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

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 $g(x) = \mathbb{E}[y \mid \mathbf{x}]$

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

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

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 $g(x) = \mathbb{E}[y \mid \mathbf{x}]$

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

$$\begin{aligned} R(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - y)^2\right] &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}) + g(\mathbf{x}) - y)^2\right] \\ &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))^2\right] + \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(g(\mathbf{x}) - y)^2\right] \\ &\quad \underbrace{+ 2 \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))(g(\mathbf{x}) - y)\right]}_{=0 \text{ as } E[y \mid \mathbf{x}] = g(\mathbf{x})} \end{aligned}$$

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 $g(x) = \mathbb{E}[y \mid \mathbf{x}]$

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

$$\begin{aligned}
R(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - y)^2\right] &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}) + g(\mathbf{x}) - y)^2\right] \\
&= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))^2\right] + \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(g(\mathbf{x}) - y)^2\right] \\
&\quad \underbrace{+ 2 \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))(g(\mathbf{x}) - y)\right]}_{=0 \text{ as } E[y \mid \mathbf{x}] = g(\mathbf{x})} \\
&= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))^2\right] + R(g)
\end{aligned}$$

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 $g(x) = \mathbb{E}[y \mid \mathbf{x}]$

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

$$\begin{aligned} R(h) = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - y)^2\right] &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}) + g(\mathbf{x}) - y)^2\right] \\ &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))^2\right] + \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(g(\mathbf{x}) - y)^2\right] \\ &\quad \underbrace{+ 2 \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))(g(\mathbf{x}) - y)\right]}_{=0 \text{ as } E[y \mid \mathbf{x}] = g(\mathbf{x})} \\ &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim D}\left[(h(\mathbf{x}) - g(\mathbf{x}))^2\right] + R(g) \end{aligned}$$

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

Aside: Maximum Likelihood Principle

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

Aside: Maximum Likelihood Principle

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

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

$$p(y_1, \ldots, y_m \mid \mathbf{w}, \mathbf{x}_1, \ldots, \mathbf{x}_m) = \frac{1}{(2\pi\sigma^2)^{m/2}} \prod_{i=1}^{m} \exp\left(-\frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}{2\sigma^2}\right)$$

Aside: Maximum Likelihood Principle

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

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

$$p(y_1, \ldots, y_m \mid \mathbf{w}, \mathbf{x}_1, \ldots, \mathbf{x}_m) = \frac{1}{(2\pi\sigma^2)^{m/2}} \prod_{i=1}^{m} \exp\left(-\frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}{2\sigma^2}\right)$$

Looking at the log likelihood is slightly simpler

$$\mathsf{LL}(y_1, \ldots, y_m \mid \mathbf{w}, \mathbf{x}_1, \ldots, \mathbf{x}_m) = -\frac{m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2$$

Aside: Maximum Likelihood Principle

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

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

$$p(y_1, \ldots, y_m \mid \mathbf{w}, \mathbf{x}_1, \ldots, \mathbf{x}_m) = \frac{1}{(2\pi\sigma^2)^{m/2}} \prod_{i=1}^{m} \exp\left(-\frac{(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2}{2\sigma^2}\right)$$

Looking at the log likelihood is slightly simpler

$$\mathsf{LL}(y_1, \ldots, y_m \mid \mathbf{w}, \mathbf{x}_1, \ldots, \mathbf{x}_m) = -\frac{m}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \mathbf{w} \cdot \mathbf{x}_i)^2$$

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

The method of least squares goes back at least 200 years to Gauss, Laplace

Linear Regression

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

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

Linear Regression

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

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

ERM for Linear Regression

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in K} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2$$

Linear Regression

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

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

ERM for Linear Regression

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in K} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2$$

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

Linear Regression

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

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

ERM for Linear Regression

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in K} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2$$

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, ...

Linear Regression

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

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

ERM for Linear Regression

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in K} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2$$

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

Empirical Rademacher Complexity

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

Empirical Rademacher Complexity

Let $\mathcal{G}$ 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 $\mathcal{G}$ 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 $\mathcal{G}$ with respect to S is defined as:

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim_u \{-1,1\}^m} \left[\sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right]$$

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

Rademacher Complexity

Empirical Rademacher Complexity

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim_u \{-1,1\}^m} \left[\sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right]$$

Rademacher Complexity

Empirical Rademacher Complexity

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim_u \{-1,1\}^m} \left[\sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right]$$

Rademacher Complexity

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

$$\mathfrak{R}_m(\mathcal{G}) = \mathop{\mathbb{E}}_{S \sim D^m} \left[\widehat{\mathfrak{R}}_S(\mathcal{G}) \right]$$

Rademacher Complexity

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim_u \{-1,1\}^m} \left[\sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right] ; \quad \mathfrak{R}_m(\mathcal{G}) = \mathop{\mathbb{E}}_{S \sim D^m} \left[\widehat{\mathfrak{R}}_S(\mathcal{G}) \right]$$

Rademacher Complexity

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim_u \{-1,1\}^m} \left[\sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i) \right] ; \quad \mathfrak{R}_m(\mathcal{G}) = \mathop{\mathbb{E}}_{S \sim D^m} \left[\widehat{\mathfrak{R}}_S(\mathcal{G}) \right]$$

Theorem [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 $g \in \mathcal{G}$:

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

Rademacher Complexity

$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}\sim_u\{-1,1\}^m}\left[\sup_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i)\right]; \quad \mathfrak{R}_m(\mathcal{G}) = \mathop{\mathbb{E}}_{S\sim D^m}\left[\widehat{\mathfrak{R}}_S(\mathcal{G})\right]$$

Theorem[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 $g \in \mathcal{G}$:

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

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

$$\mathop{\widehat{\mathbb{E}}}_{z\sim_u S}[g(z)] = \frac{1}{m}\sum_{i=1}^{m} g(z_i)$$

We will see a full proof of this theorem. First, let's apply this to linear regression.

Generalisation Bounds for Linear Regression

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

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

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

Generalisation Bounds for Linear Regression

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

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

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

Let $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. Then we have:

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{w}, \|\mathbf{w}\|_2 \leq W} \sum_{i=1}^{m} \sigma_i (\mathbf{w} \cdot \mathbf{x}_i) \right]$$

Generalisation Bounds for Linear Regression

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

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

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

Let $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. Then we have:

$$\begin{aligned}\widehat{\mathfrak{R}}_S(\mathcal{F}) &= \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{w}, \|\mathbf{w}\|_2 \le W} \sum_{i=1}^{m} \sigma_i(\mathbf{w} \cdot \mathbf{x}_i)\right] \\ &= \frac{1}{m}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{w}, \|\mathbf{w}\|_2 \le W} \mathbf{w} \cdot \sum_{i=1}^{m} \sigma_i \mathbf{x}_i\right]\end{aligned}$$

Generalisation Bounds for Linear Regression

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

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

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

Let $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. Then we have:

$$
\begin{aligned}
\widehat{\mathfrak{R}}_S(\mathcal{F}) &= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{w}, \|\mathbf{w}\|_2 \leq W} \sum_{i=1}^{m} \sigma_i (\mathbf{w} \cdot \mathbf{x}_i) \right] \\
&= \frac{1}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{w}, \|\mathbf{w}\|_2 \leq W} \mathbf{w} \cdot \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right] \\
&= \frac{W}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2 \right]
\end{aligned}
$$

The last step follows from (the equality condition of) the Cauchy-Schwartz Inequality

Generalisation Bounds for Linear Regression

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

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

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

Generalisation Bounds for Linear Regression

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

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

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

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \frac{W}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2 \right]$$

Generalisation Bounds for Linear Regression

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

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

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

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \frac{W}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2 \right] \leq \frac{W}{m} \left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2^2 \right] \right)^{\frac{1}{2}}$$

Generalisation Bounds for Linear Regression

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

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

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

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \frac{W}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2 \right] \leq \frac{W}{m} \left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2^2 \right] \right)^{\frac{1}{2}}$$

$$= \frac{W}{m} \left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sum_{i=1}^{m} \sigma_i^2 \|\mathbf{x}_i\|_2^2 + \underbrace{2 \sum_{i<j} \sigma_i \sigma_j \mathbf{x}_i \cdot \mathbf{x}_j}_{=0 \text{ as } \sigma_i \text{ are independent}} \right] \right)^{\frac{1}{2}}$$

Generalisation Bounds for Linear Regression

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

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

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

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \frac{W}{m} \mathbb{E}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2 \right] \leq \frac{W}{m} \left(\mathbb{E}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|_2^2 \right] \right)^{\frac{1}{2}}$$

$$= \frac{W}{m} \left(\mathbb{E}_{\boldsymbol{\sigma}} \left[\sum_{i=1}^{m} \sigma_i^2 \|\mathbf{x}_i\|_2^2 + \underbrace{2 \sum_{i<j} \sigma_i \sigma_j \mathbf{x}_i \cdot \mathbf{x}_j}_{=0 \text{ as } \sigma_i \text{ are independent}} \right] \right)^{\frac{1}{2}}$$

$$= \frac{W}{m} \left(\sum_{i=1}^{m} \|\mathbf{x}_i\|_2^2 \right)^{\frac{1}{2}} = \frac{WX}{\sqrt{m}}$$

Talagrand's Lemma

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

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

Then Talagrand's Lemma tells us that:

$$\widehat{\mathfrak{R}}_S(\phi \circ \mathcal{G}) \leq L \cdot \widehat{\mathfrak{R}}_S(\mathcal{G})$$
$$\mathfrak{R}_m(\phi \circ \mathcal{G}) \leq L \cdot \mathfrak{R}_m(\mathcal{G})$$

Generalisation of Linear Regression

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

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

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

Generalisation of Linear Regression

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

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

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

Consider the following:

$$\mathcal{H} = \{(\mathbf{x}, y) \mapsto (f(\mathbf{x}) - y)^2 \mid \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F}\}$$
$$\phi : [-(M + WX), (M + WX)] \to \mathbb{R},\ \phi(z) = z^2$$

Generalisation of Linear Regression

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

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

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

Consider the following:

$$\mathcal{H} = \{(\mathbf{x}, y) \mapsto (f(\mathbf{x}) - y)^2 \mid \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F}\}$$
$$\phi : [-(M + WX), (M + WX)] \to \mathbb{R},\ \phi(z) = z^2$$

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

Generalisation of Linear Regression

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

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

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

Consider the following:

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

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

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

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

Generalisation of Linear Regression

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

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

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

Consider the following:

$$\mathcal{H} = \{(\mathbf{x}, y) \mapsto (f(\mathbf{x}) - y)^2 \mid \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F}\}$$
$$\phi : [-(M + WX), (M + WX)] \to \mathbb{R},\ \phi(z) = z^2$$

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

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

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

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

Generalisation of Linear Regression

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

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

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

Consider the following:

$$\mathcal{H} = \{(\mathbf{x}, y) \mapsto (f(\mathbf{x}) - y)^2 \mid \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{F}\}$$
$$\phi : [-(M + WX), (M + WX)] \to \mathbb{R}, \ \phi(z) = z^2$$

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

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

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

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

Note that $\mathfrak{R}_m(\mathcal{H}) = \mathop{\mathbb{E}}_{S \sim D^m}\left[\widehat{\mathfrak{R}}_S(\mathcal{H})\right] \leq \sup_{S, |S| = m} \widehat{\mathfrak{R}}_S(\mathcal{H})$

Aside: Algorithms for the Linear Regression Model

ERM for Linear Regression

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

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}, \|\mathbf{w}\|_2 \leq W}{\operatorname{argmin}} J(\mathbf{w})$$

Aside: Algorithms for the Linear Regression Model

ERM for Linear Regression

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

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}, \|\mathbf{w}\|_2 \leq W}{\operatorname{argmin}} \; J(\mathbf{w})$$

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

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

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}, \|\mathbf{w}\|_2 \leq W}{\operatorname{argmin}} \; J(\mathbf{w})$$

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

Aside: Algorithms for the Linear Regression Model

ERM for Linear Regression

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

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}, \|\mathbf{w}\|_2 \leq W} J(\mathbf{w})$$

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: η, T
Pick $\mathbf{w}_1 \in K$
for $t = 1, \ldots, T$ **do**
  $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta \nabla J(\mathbf{w}_t)$
  $\mathbf{w}_{t+1} = \Pi_K(\mathbf{w}'_{t+1})$
end for
Output: $\overline{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t$

Aside: Gradient Descent

Algorithm 2 Projected Gradient Descent

Inputs: η, T
Pick $\mathbf{w}_1 \in K$
for $t = 1, \ldots, T$ **do**
  $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta \nabla J(\mathbf{w}_t)$
  $\mathbf{w}_{t+1} = \Pi_K(\mathbf{w}'_{t+1})$
end for
Output: $\overline{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}_t$

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

Aside: Gradient Descent

Algorithm 3 Projected Gradient Descent

Inputs: η, T
Pick $\mathbf{w}_1 \in K$
for $t = 1, \ldots, T$ **do**
  $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta \nabla J(\mathbf{w}_t)$
  $\mathbf{w}_{t+1} = \Pi_K(\mathbf{w}'_{t+1})$
end for
Output: $\overline{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}_t$

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

Informal Theorem [5]

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

$$J(\overline{\mathbf{w}}) \leq \min_{\mathbf{w} \in K} J(\mathbf{w}) + \frac{RL}{\sqrt{T}}$$

Aside: Generalised Linear Models

Can consider more general models called generalised linear models

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

Aside: Generalised Linear Models

Can consider more general models called generalised linear models

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

We can consider the ERM problem:

$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} (u(\mathbf{w} \cdot \mathbf{x}_i) - y_i)^2; \quad \widehat{\mathbf{w}} = \underset{\mathbf{w}, \|\mathbf{w}\|_2 \leq W}{\operatorname{argmin}} J(\mathbf{w})$$

Aside: Generalised Linear Models

Can consider more general models called generalised linear models

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

We can consider the ERM problem:

$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} (u(\mathbf{w} \cdot \mathbf{x}_i) - y_i)^2; \;\; \widehat{\mathbf{w}} = \underset{\mathbf{w}, \|\mathbf{w}\|_2 \leq W}{\operatorname{argmin}} \; J(\mathbf{w})$$

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

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

Aside: Generalised Linear Models

Can consider more general models called generalised linear models

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

Can consider a different cost/loss function:

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

$$\ell(\mathbf{w}; \mathbf{x}, y) \int_0^{\mathbf{w} \cdot \mathbf{x}} (u(z) - y) dz$$

Aside: Generalised Linear Models

Can consider more general models called generalised linear models

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

Can consider a different cost/loss function:

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

$$\ell(\mathbf{w}; \mathbf{x}, y) \int_0^{\mathbf{w}\cdot\mathbf{x}} (u(z) - y)dz$$

The resulting objective function is convex in $\mathbf{w}$

$$\widetilde{J}(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{w}; \mathbf{x}_i, y_i)$$

Aside: Generalised Linear Models

Can consider more general models called generalised linear models

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

Can consider a different cost/loss function:

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

$$\ell(\mathbf{w}; \mathbf{x}, y) \int_0^{\mathbf{w} \cdot \mathbf{x}} (u(z) - y) dz$$

The resulting objective function is convex in $\mathbf{w}$

$$\widetilde{J}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{w}; \mathbf{x}_i, y_i)$$

In the realisable setting, i.e. $\mathbb{E}\left[y \mid \mathbf{x}\right] = u(\mathbf{w} \cdot \mathbf{x})$, the global minimisers of $J(\mathbf{w})$ (squared error) and $\widetilde{J}(\mathbf{w})$ coincide, yielding computationally and statistically efficient algorithms.[12]

Rademacher Complexity : Main Result

Rademacher Complexity : Main Result

Theorem [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 $g \in \mathcal{G}$:

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

Rademacher Complexity : Main Result

Theorem[2,14]

Let $\mathcal{G}$ be a class of functions mapping $\mathcal{Z} \to [0,1]$. 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 $g \in \mathcal{G}$:

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

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

Rademacher Complexity : Main Result

Theorem [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 $g \in \mathcal{G}$:

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

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 $\forall \varepsilon > 0$,

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

Proof of Main Result

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:

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

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:

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

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

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

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:

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

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 $\forall \varepsilon > 0$,

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

Proof of Main Result

McDiarmid's Inequality

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

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

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

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

Applying McDiarmid's inequality with $c_i = 1/m$ for all i,

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

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

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

Applying McDiarmid's inequality with $c_i = 1/m$ for all i,

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

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

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

Proof of Main Result

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

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

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

Proof of Main Result

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

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

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

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

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

Proof of Main Result

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

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

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

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

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

Recall that,

$$\mathop{\widehat{\mathbb{E}}}_{z \sim_u S} [g(z)] = \frac{1}{m} \sum_{i=1}^{m} g(z_i)$$

Proof of Main Result

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

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

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

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

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

Recall that,

$$\mathop{\widehat{\mathbb{E}}}_{z \sim_u S} [g(z)] = \frac{1}{m} \sum_{i=1}^{m} g(z_i)$$

Want to show

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

Proof of Main Result

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

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

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

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

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

Recall that,

$$\mathop{\widehat{\mathbb{E}}}_{z \sim_u S} [g(z)] = \frac{1}{m} \sum_{i=1}^{m} g(z_i)$$

Want to show

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

Wouldn't it be nice if $\mathop{\mathbb{E}}_{S \sim D^m} [\Phi(S)] \leq 2\mathfrak{R}_m(\mathcal{G})$?

Proof of Main Result

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

Proof of Main Result

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

Consider

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

Proof of Main Result

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

Consider

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

Introduce a fresh sample $S' \sim D^m$

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

Proof of Main Result

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

Consider

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

Introduce a fresh sample $S' \sim D^m$

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

Pushing the $\sup$ inside the expectation

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

Proof of Main Result

Pushing the $\sup$ inside the expectation

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

Proof of Main Result

Pushing the $\sup$ inside the expectation

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

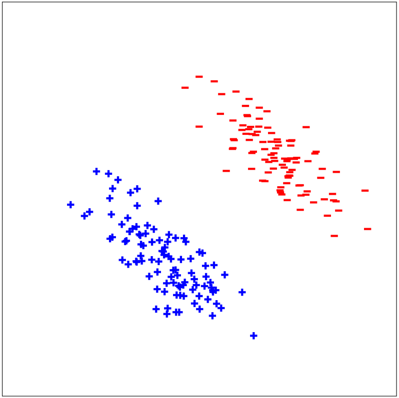
S and S' are identically distributed, so their elements can be swapped by introducing Rademacher random variables $\sigma_i \in \{-1, 1\}$

$$\begin{aligned}\mathop{\mathbb{E}}_{S\sim D^m}\left[\Phi(S)\right] &\leq \mathop{\mathbb{E}}_{S\sim D^m, S'\sim D^m, \boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sigma_i(g(z_i') - g(z_i))\right] \\ &\leq 2\mathop{\mathbb{E}}_{S\sim D^m, \boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sigma_i g(z_i)\right] = 2\mathfrak{R}_m(\mathcal{G})\end{aligned}$$

Outline

- Statistical (Supervised) Learning Theory Framework
- Linear Regression
- Rademacher Complexity
- Support Vector Machines
- Kernels
- Neural Networks
- Algorithmic Stability

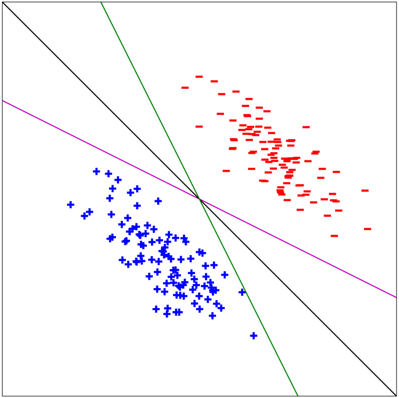
Support Vector Machines: Binary Classification



<ins>Goal</ins>: Find a linear separator

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

Support Vector Machines: Binary Classification

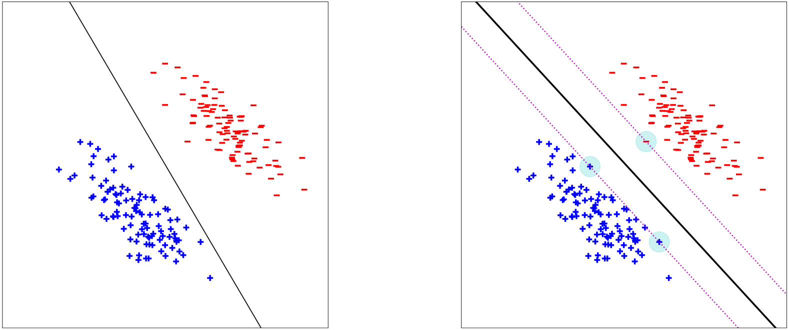


<u>Goal</u>: Find a linear separator

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

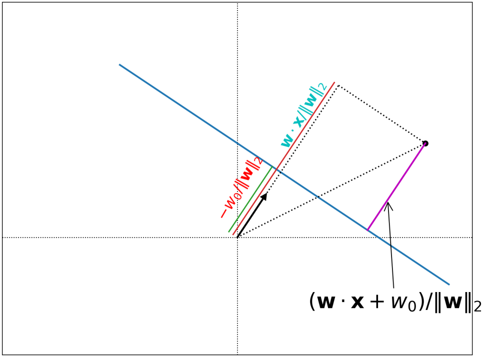


Maximise the distance of the <u>closest</u> point from the decision boundary

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?



Support Vector Machines : Geometric View

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

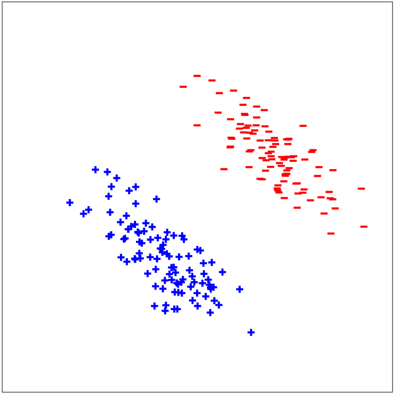
minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2$

subject to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1$$

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 $\frac{1}{\|\mathbf{w}^*\|_2}$ if $\mathbf{w}^*$ is the optimal solution

This is a convex quadratic program and hence can be solved efficiently

SVM Formulation : The Dual

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2$

subject to:

$y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) - 1 \geq 0$

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$

maximise $\displaystyle\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$

subject to:

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

$0 \leq \alpha_i$

for $i = 1, \ldots, m$

Lagrange Function

$$\Lambda(\mathbf{w}, w_0; \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|_2^2 - \sum_{i=1}^{m} \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) - 1)$$

Complementary Slackness

$$\alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) - 1) = 0, \;\; i = 1, \ldots, m$$

SVM Formulation : Non-Separable Case

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

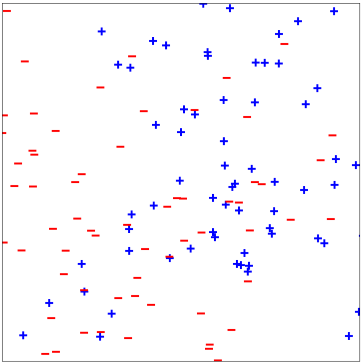
subject to:

$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i$

$\zeta_i \geq 0$

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$



SVM Formulation : Loss Function

minimise: $$\underbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}_{\text{Regulariser}} + \underbrace{C\sum_{i=1}^{m}\zeta_i}_{\text{Loss Function}}$$

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: $$\underbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}_{\text{Regulariser}} + \underbrace{C\sum_{i=1}^{m}\zeta_i}_{\text{Loss Function}}$$

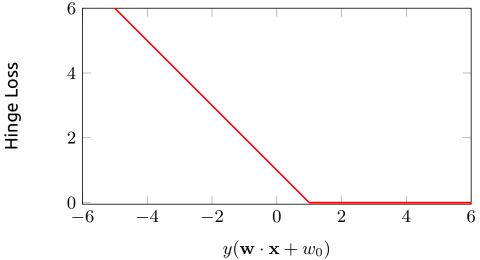
subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0$$

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$



SVM Formulation : Loss Function

minimise: $$\underbrace{\frac{1}{2}\|\mathbf{w}\|_2^2}_{\text{Regulariser}} + \underbrace{C\sum_{i=1}^{m}\zeta_i}_{\text{Loss Function}}$$

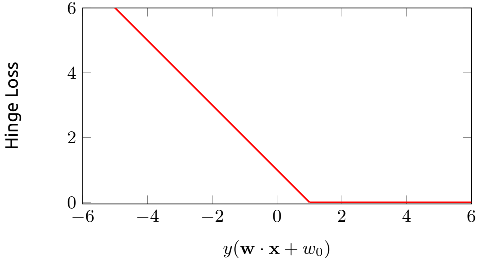
subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0$$

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$



Note that for the optimal solution, $\zeta_i = \max\{0, 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0)\}$

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

$$\Lambda(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\zeta_i - \sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)) - \sum_{i=1}^{m}\mu_i\zeta_i$$

SVM : Deriving the Dual

Lagrange Function

$$\Lambda(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\zeta_i - \sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)) - \sum_{i=1}^{m}\mu_i\zeta_i$$

We write derivatives with respect to $\mathbf{w}$, w_0 and ζ_i,

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

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

$$\nabla_{\mathbf{w}}\Lambda = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i$$

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 Λ (and simplifying), we get a function $g(\boldsymbol{\alpha})$ and some constraints

$$g(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

Constraints

$$0 \leq \alpha_i \leq C \qquad\qquad i = 1, \ldots, m$$
$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

Finding critical points of Λ satisfying the KKT conditions corresponds to finding the maximum of $g(\boldsymbol{\alpha})$ subject to the above constraints

SVM: Primal and Dual Formulations

Primal Form

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 $i = 1,\ldots,m$

Dual Form

maximise $\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j$

subject to:

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

$$0 \leq \alpha_i \leq C$$

for $i = 1,\ldots,m$

KKT Complementary Slackness Conditions

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

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

KKT Complementary Slackness Conditions

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

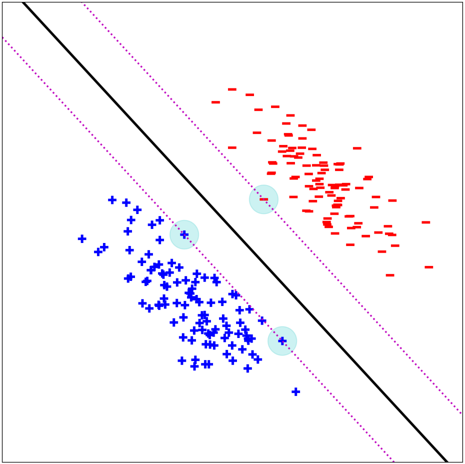
If $\alpha_i > 0$, $y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) = 1 - \zeta_i$

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



Generalisation Bounds Based on Margin

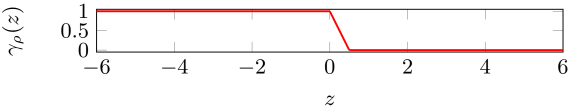
Suppose we solve the SVM objective by constraining $\mathbf{w}$ to be in the set $\{\mathbf{w} \mid \|\mathbf{w}\|_2 \leq W\}$

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:

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

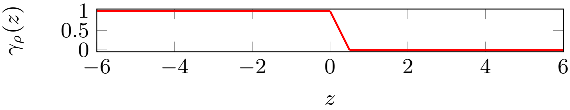


Generalisation Bounds Based on Margin

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

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:

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



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

$$\widehat{\mathfrak{R}}(\varphi_\rho \circ \mathcal{H}) \le \frac{WX}{\rho\sqrt{m}}$$

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 $\gamma(y', y) \leq \gamma_\rho(y', y)$

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

Generalisation Bounds Based on Margin

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

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

Then, we have

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

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

Then, we have

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

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

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

Then, we have

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

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

Gram Matrix

If we put the inputs in matrix $\mathbf{X}$, where the i^{th} row of $\mathbf{X}$ is $\mathbf{x}_i^\top$.

$$\mathbf{K} = \mathbf{X}\mathbf{X}^\top = \begin{bmatrix} \mathbf{x}_1^\top\mathbf{x}_1 & \mathbf{x}_1^\top\mathbf{x}_2 & \cdots & \mathbf{x}_1^\top\mathbf{x}_m \\ \mathbf{x}_2^\top\mathbf{x}_1 & \mathbf{x}_2^\top\mathbf{x}_2 & \cdots & \mathbf{x}_2^\top\mathbf{x}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m^\top\mathbf{x}_1 & \mathbf{x}_m^\top\mathbf{x}_2 & \cdots & \mathbf{x}_m^\top\mathbf{x}_m \end{bmatrix}$$

Gram Matrix

If we put the inputs in matrix $\mathbf{X}$, where the i^{th} row of $\mathbf{X}$ is $\mathbf{x}_i^\top$.

$$\mathbf{K} = \mathbf{X}\mathbf{X}^\top = \begin{bmatrix} \mathbf{x}_1^\top \mathbf{x}_1 & \mathbf{x}_1^\top \mathbf{x}_2 & \cdots & \mathbf{x}_1^\top \mathbf{x}_m \\ \mathbf{x}_2^\top \mathbf{x}_1 & \mathbf{x}_2^\top \mathbf{x}_2 & \cdots & \mathbf{x}_2^\top \mathbf{x}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m^\top \mathbf{x}_1 & \mathbf{x}_m^\top \mathbf{x}_2 & \cdots & \mathbf{x}_m^\top \mathbf{x}_m \end{bmatrix}$$

The matrix $\mathbf{K}$ is positive semi-definite

Gram Matrix

If we put the inputs in matrix $\mathbf{X}$, where the i^{th} row of $\mathbf{X}$ is $\mathbf{x}_i^\intercal$.

$$\mathbf{K} = \mathbf{X}\mathbf{X}^\intercal = \begin{bmatrix} \mathbf{x}_1^\intercal \mathbf{x}_1 & \mathbf{x}_1^\intercal \mathbf{x}_2 & \cdots & \mathbf{x}_1^\intercal \mathbf{x}_m \\ \mathbf{x}_2^\intercal \mathbf{x}_1 & \mathbf{x}_2^\intercal \mathbf{x}_2 & \cdots & \mathbf{x}_2^\intercal \mathbf{x}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m^\intercal \mathbf{x}_1 & \mathbf{x}_m^\intercal \mathbf{x}_2 & \cdots & \mathbf{x}_m^\intercal \mathbf{x}_m \end{bmatrix}$$

The matrix $\mathbf{K}$ is positive semi-definite

If we perform basis expansion

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

then replace entries by $\phi(\mathbf{x}_i)^\intercal \phi(\mathbf{x}_j)$

Gram Matrix

If we put the inputs in matrix $\mathbf{X}$, where the i^{th} row of $\mathbf{X}$ is $\mathbf{x}_i^\mathsf{T}$.

$$\mathbf{K} = \mathbf{X}\mathbf{X}^\mathsf{T} = \begin{bmatrix} \mathbf{x}_1^\mathsf{T}\mathbf{x}_1 & \mathbf{x}_1^\mathsf{T}\mathbf{x}_2 & \cdots & \mathbf{x}_1^\mathsf{T}\mathbf{x}_m \\ \mathbf{x}_2^\mathsf{T}\mathbf{x}_1 & \mathbf{x}_2^\mathsf{T}\mathbf{x}_2 & \cdots & \mathbf{x}_2^\mathsf{T}\mathbf{x}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m^\mathsf{T}\mathbf{x}_1 & \mathbf{x}_m^\mathsf{T}\mathbf{x}_2 & \cdots & \mathbf{x}_m^\mathsf{T}\mathbf{x}_m \end{bmatrix}$$

The matrix $\mathbf{K}$ is positive semi-definite

If we perform basis expansion

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

then replace entries by $\phi(\mathbf{x}_i)^\mathsf{T}\phi(\mathbf{x}_j)$

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

Kernel Trick

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

$$\psi(\mathbf{x}) = \left[1, x_1, x_2, x_1^2, x_2^2, x_1 x_2\right]^{\mathsf{T}}$$

Kernel Trick

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

$$\psi(\mathbf{x}) = \left[1, x_1, x_2, x_1^2, x_2^2, x_1 x_2\right]^{\mathsf{T}}$$

But, we could also use the map:

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

Kernel Trick

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

$$\psi(\mathbf{x}) = \left[1, x_1, x_2, x_1^2, x_2^2, x_1 x_2\right]^{\mathsf{T}}$$

But, we could also use the map:

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

If $\mathbf{x} = [x_1, x_2]^{\mathsf{T}}$ and $\mathbf{x}' = [x_1', x_2']^{\mathsf{T}}$, then

$$\begin{aligned}\phi(\mathbf{x})^{\mathsf{T}}\phi(\mathbf{x}') &= 1 + 2x_1x_1' + 2x_2x_2' + x_1^2(x_1')^2 + x_2^2(x_2')^2 + 2x_1x_2x_1'x_2' \\ &= (1 + x_1x_1' + x_2x_2')^2 = (1 + \mathbf{x} \cdot \mathbf{x}')^2\end{aligned}$$

Kernel Trick

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

$$\psi(\mathbf{x}) = \left[1, x_1, x_2, x_1^2, x_2^2, x_1 x_2\right]^{\mathsf{T}}$$

But, we could also use the map:

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

If $\mathbf{x} = [x_1, x_2]^{\mathsf{T}}$ and $\mathbf{x}' = [x_1', x_2']^{\mathsf{T}}$, then

$$\begin{aligned}\phi(\mathbf{x})^{\mathsf{T}}\phi(\mathbf{x}') &= 1 + 2x_1x_1' + 2x_2x_2' + x_1^2(x_1')^2 + x_2^2(x_2')^2 + 2x_1x_2x_1'x_2' \\ &= (1 + x_1x_1' + x_2x_2')^2 = (1 + \mathbf{x} \cdot \mathbf{x}')^2\end{aligned}$$

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)

$$\mathbf{K} = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \kappa(\mathbf{x}_m, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$

Here $\kappa(\mathbf{x}, \mathbf{x}')$ is some measure of similarity between $\mathbf{x}$ and $\mathbf{x}'$

Kernel Trick

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

$$\mathbf{K} = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \kappa(\mathbf{x}_m, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$

Here $\kappa(\mathbf{x}, \mathbf{x}')$ is some measure of similarity between $\mathbf{x}$ and $\mathbf{x}'$

The dual program becomes

maximise $\displaystyle\sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K_{i,j}$

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)

$$\mathbf{K} = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \kappa(\mathbf{x}_m, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$

Here $\kappa(\mathbf{x}, \mathbf{x}')$ is some measure of similarity between $\mathbf{x}$ and $\mathbf{x}'$

The dual program becomes

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 $\mathbf{x}_{\text{new}}$, only need to compute $\kappa(\mathbf{x}_i, \mathbf{x}_{\text{new}})$ for support vectors $\mathbf{x}_i$ (for which $\alpha_i > 0$)

Polynomial Kernels

Rather than perform basis expansion,

$$\kappa(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^d$$

This gives all terms of degree up to d

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

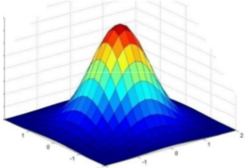
Linear Kernel: $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$

All of these satisfy the Mercer or positive-definite condition

Gaussian or RBF Kernel

Radial Basis Function (RBF) or Gaussian Kernel

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



σ^2 is known as the bandwidth

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

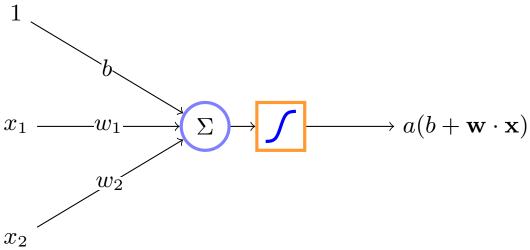
Can generalise to more general covariance matrices

Results in a Mercel kernel

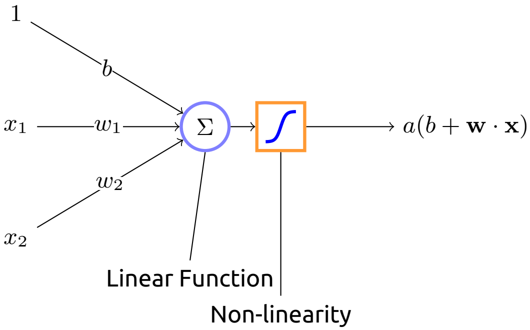
Outline

- Statistical (Supervised) Learning Theory Framework
- Linear Regression
- Rademacher Complexity
- Support Vector Machines
- Kernels
- Neural Networks
- Algorithmic Stability

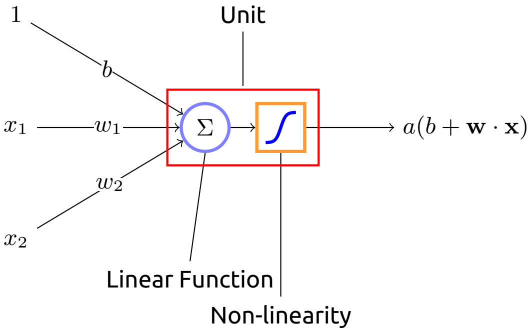
Neural Networks : Unit



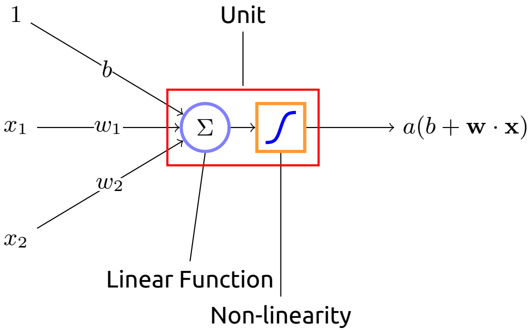
Neural Networks : Unit



Neural Networks : Unit



Neural Networks : Unit

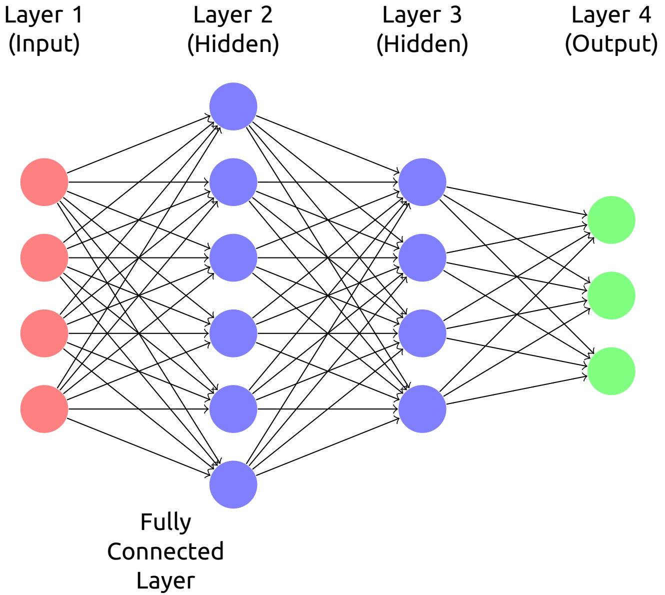


A unit in a neural network computes an affine function of its input and is then composed with a non-linear activation function a

For example the activation function could be the logistic sigmoid

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

Feedforward Neural Networks



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

$$l(\mathbf{z}) = a(W\mathbf{z} + \mathbf{b})$$

Neural Networks

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

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

$$l(\mathbf{z}) = a(W\mathbf{z} + \mathbf{b})$$

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

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

Neural Networks

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

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

$$l(\mathbf{z}) = a(W\mathbf{z} + \mathbf{b})$$

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

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

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

Capacity of Neural Networks

VC dim. of Neural Nets

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

Capacity of Neural Networks

VC dim. of Neural Nets

Informally, if a is the sgn function, and C is the class of all neural networks with at most ω 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 $\mathbf{w}$ of any W in the net satisfying $\|\mathbf{w}\|_1 \leq W$

Capacity of Neural Networks

VC dim. of Neural Nets

Informally, if a is the sgn function, and C is the class of all neural networks with at most ω 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 $\mathbf{w}$ of any W in the net satisfying $\|\mathbf{w}\|_1 \leq W$
- every bias vector $\mathbf{b}$ satisfying $\|\mathbf{b}\|_\infty \leq B$

Capacity of Neural Networks

VC dim. of Neural Nets

Informally, if a is the sgn function, and C is the class of all neural networks with at most ω 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 $\mathbf{w}$ of any W in the net satisfying $\|\mathbf{w}\|_1 \leq W$
- every bias vector $\mathbf{b}$ satisfying $\|\mathbf{b}\|_\infty \leq B$
- the activations a being 1-Lipschitz

Capacity of Neural Networks

VC dim. of Neural Nets

Informally, if a is the sgn function, and C is the class of all neural networks with at most ω 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 $\mathbf{w}$ of any W in the net satisfying $\|\mathbf{w}\|_1 \leq W$
- every bias vector $\mathbf{b}$ satisfying $\|\mathbf{b}\|_\infty \leq B$
- the activations a being 1-Lipschitz
- and furthermore, the inputs $\mathbf{x} \in \mathcal{X}$ satisfying $\|\mathbf{x}\|_\infty \leq 1$

Capacity of Neural Networks

VC dim. of Neural Nets

Informally, if a is the sgn function, and C is the class of all neural networks with at most ω 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 $\mathbf{w}$ of any W in the net satisfying $\|\mathbf{w}\|_1 \leq W$
- every bias vector $\mathbf{b}$ satisfying $\|\mathbf{b}\|_\infty \leq B$
- the activations a being 1-Lipschitz
- and furthermore, the inputs $\mathbf{x} \in \mathcal{X}$ satisfying $\|\mathbf{x}\|_\infty \leq 1$

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

Capacity of Neural Networks

VC dim. of Neural Nets

Informally, if a is the sgn function, and C is the class of all neural networks with at most ω 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 $\mathbf{w}$ of any W in the net satisfying $\|\mathbf{w}\|_1 \leq W$
- every bias vector $\mathbf{b}$ satisfying $\|\mathbf{b}\|_\infty \leq B$
- the activations a being 1-Lipschitz
- and furthermore, the inputs $\mathbf{x} \in \mathcal{X}$ satisfying $\|\mathbf{x}\|_\infty \leq 1$

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

Exercise: Prove this using the fact that if $\overline{\mathcal{G}}$ is the function class consisting of all convex combinations of functions in $\mathcal{G}$, then $\widehat{\mathfrak{R}}_m(\overline{\mathcal{G}}) = \widehat{\mathfrak{R}}(\mathcal{G})$. (Also prove the latter claim.)

Neural Networks : Universality Results

(Simplified) Theorem (Cybenko) [6]

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

Neural Networks : Universality Results

(Simplified) Theorem (Cybenko) [6]

Let σ be the logistic sigmoid activation function. Then the set of functions $G(\mathbf{x}) = \sum_{i=1}^{N} \alpha_j \sigma(\mathbf{w}_j \cdot \mathbf{x} + b_j)$ 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]

Neural Networks : Universality Results

(Simplified) Theorem (Cybenko) [6]

Let σ be the logistic sigmoid activation function. Then the set of functions $G(\mathbf{x}) = \sum_{i=1}^{N} \alpha_j \sigma(\mathbf{w}_j \cdot \mathbf{x} + b_j)$ 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

Neural Networks : Universality Results

(Simplified) Theorem (Cybenko) [6]

Let σ be the logistic sigmoid activation function. Then the set of functions $G(\mathbf{x}) = \sum_{i=1}^{N} \alpha_j \sigma(\mathbf{w}_j \cdot \mathbf{x} + b_j)$ 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

Neural Networks : Universality Results

(Simplified) Theorem (Cybenko)[6]

Let σ be the logistic sigmoid activation function. Then the set of functions $G(\mathbf{x}) = \sum_{i=1}^{N} \alpha_j \sigma(\mathbf{w}_j \cdot \mathbf{x} + b_j)$ 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

Depth Separation Results

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

Depth Separation Results

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

Algorithmic Stability

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

$$R(f) \leq \widehat{R}_S(f) + \epsilon$$

Algorithmic Stability

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

$$R(f) \leq \widehat{R}_S(f) + \epsilon$$

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, $\forall f \in \mathcal{F}$,

$$R(f) \leq \widehat{R}_S(f) + \epsilon$$

These results only depend on certain complexity/capacity measures of the class of functions $\mathcal{F}$ used by the learning algorithm

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

Algorithmic Stability

Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ 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 $(\mathbf{x}'_m, y'_m)$ instead of $(\mathbf{x}_m, y_m)$

Algorithmic Stability

Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ 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 $(\mathbf{x}'_m, y'_m)$ instead of $(\mathbf{x}_m, y_m)$

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

Algorithmic Stability

Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ 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 $(\mathbf{x}'_m, y'_m)$ instead of $(\mathbf{x}_m, y_m)$

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

Algorithmic Stability

Let $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ 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 $(\mathbf{x}'_m, y'_m)$ instead of $(\mathbf{x}_m, y_m)$

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

Uniform Stability

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

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

Algorithmic Stability

Uniform Stability

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

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

Algorithmic Stability

Uniform Stability

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

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

Theorem (Bousquet & Elisseeff) [3]

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

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

Algorithmic Stability

Uniform Stability

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

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

Theorem (Bousquet & Elisseeff) [3]

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

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

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

Algorithmic Stability

Uniform Stability

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

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

Theorem (Bousquet & Elisseeff) [3]

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

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

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

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

Algorithmic Stability

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

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

Example of Ridge Regression

The ridge regression method finds

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

Algorithmic Stability

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

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

Example of Ridge Regression

The ridge regression method finds

$$\widehat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

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

Algorithmic Stability

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

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

Example of Ridge Regression

The ridge regression method finds

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

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

Consequently, $\gamma(y', y) = (y' - y)^2$ is σ-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 γ is σ-admissible with respect to a class of function $\mathcal{F}$, if for every $f, f' \in \mathcal{F}$ and $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, it is the case that

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

Example of Ridge Regression

The ridge regression method finds

$$\widehat{\mathbf{w}} = \underset{\mathbf{w} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

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

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

Theorem. Since γ is convex and σ-admissible, ridge regression is uniformly β-stable with $\beta \leq \frac{\sigma^2 X^2}{m\lambda} = O(1/m)$

Algorithmic Stability

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

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

Example of Ridge Regression

The ridge regression method finds

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_2^2$$

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

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

Theorem. Since γ is convex and σ-admissible, ridge regression is uniformly β-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

[1] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information theory, 39 (3):930--945, 1993.

[2] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 3(Nov):463--482, 2002.

[3] Olivier Bousquet and André Elisseeff. Stability and generalization. Journal of machine learning research, 2(Mar):499--526, 2002.

[4] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In Advanced lectures on machine learning, pages 169--207. Springer, 2004.

[5] Sébastien Bubeck. Convex Optimization: Algorithms and Complexity. Foundations and Trends in Machine Learning. Now, 2015.

[6] George Cybenko. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems, 2(4):303--314, 1989.

[7] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In Conference on Learning Theory, pages 907--940, 2016.

References II

[8] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. Neural networks, 2(3):183--192, 1989.

[9] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: stability of stochastic gradient descent. In Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48, pages 1225--1234, 2016.

[10] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. In Proc. of the 1st Workshop on Algorithmic Learning Theory, pages 21--41, 1990.

[11] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359--366, 1989.

[12] Sham M Kakade, Varun Kanade, Adam Kalai, and Ohad Shamir. Efficient learning of generalized linear and single index models with isotonic regression. In Advances in Neural Information Processing Systems, pages 927--935, 2011.

[13] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. Machine Learning, 17(2-3):115--141, 1994.

References III

[14] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of machine learning. MIT Press, 2012.

[15] Matus Telgarsky. benefits of depth in neural networks. In Conference on Learning Theory, pages 1517--1539, 2016.

[16] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications, XVI(2):264--280, 1971.

[17] Vladimir Vapnik. Statistical learning theory. 1998, volume 3. Wiley, New York, 1998.

[18] Vladimir Vapnik. The nature of statistical learning theory. Springer science & business media, 2013.