Advanced Topics in Machine Learning

Part I: Elements of Statistical Learning Theory

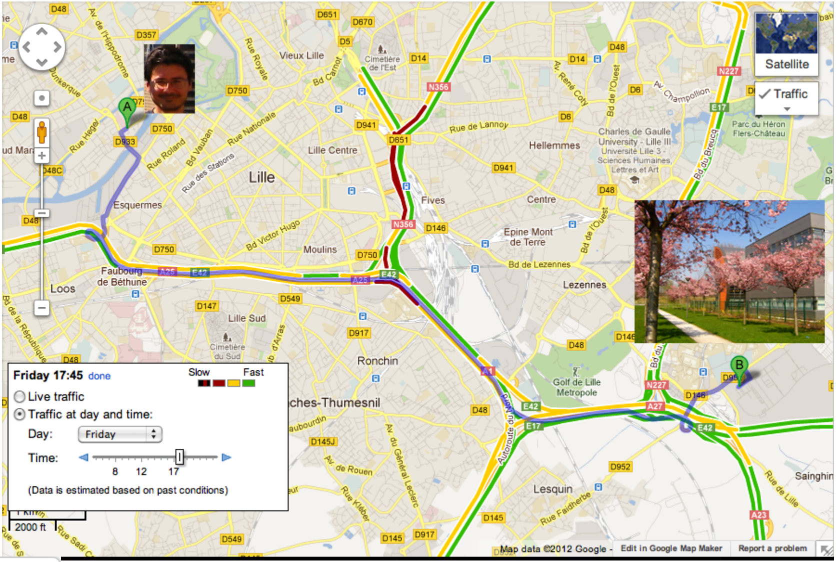
A. LAZARIC (*INRIA-Lille*)

DEI, Politecnico di Milano

SequeL – INRIA Lille

April 2-15, 2012

A Motivating Example



A Motivating Example

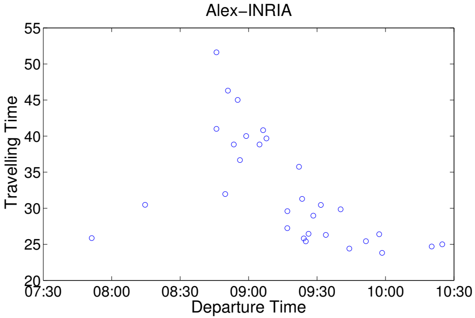
Problem: estimate the travelling time from home to INRIA depending on the departure time.

Data available: a database of 30 (working) days in the form

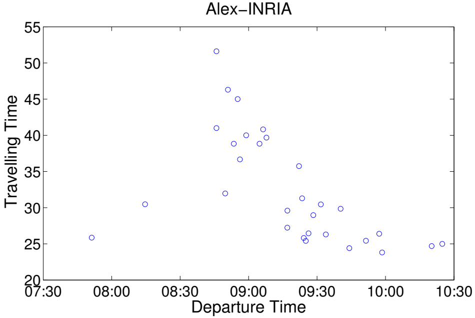
Dep. Time	*Time*
9:06	23 min
8:26	27 min
9:43	19 min
9:30	25 min
8:58	40 min
10:03	15 min
...	...

***n**: number of training samples*

A Motivating Example

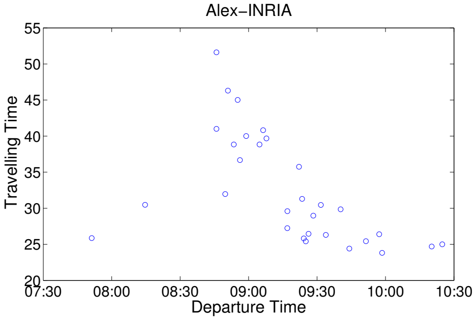


A Motivating Example



Data are sampled from a ***sampling distribution***

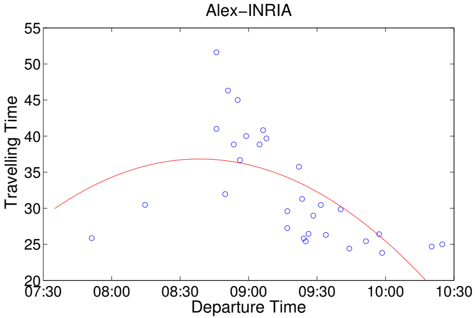
A Motivating Example



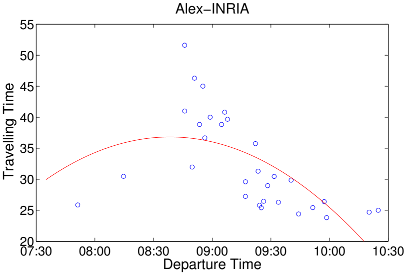
Solution: fit the data with a polynomial of degree 2

$$f(x) = ax^2 + bx + c$$

A Motivating Example



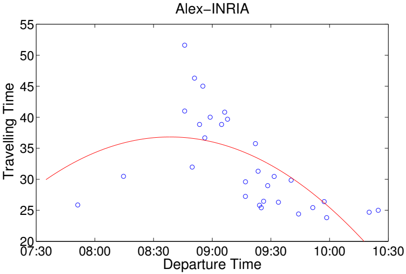
A Motivating Example



Result: mean–squared error after testing for one year

$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 24.5600$$

A Motivating Example

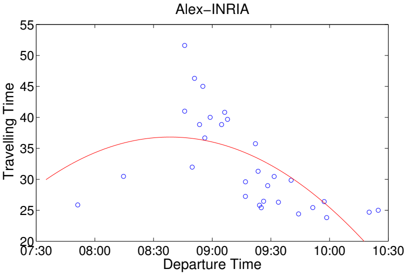


Result: mean–squared error after testing for one year

$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 24.5600$$

The performance is measured with a ***loss function***

A Motivating Example



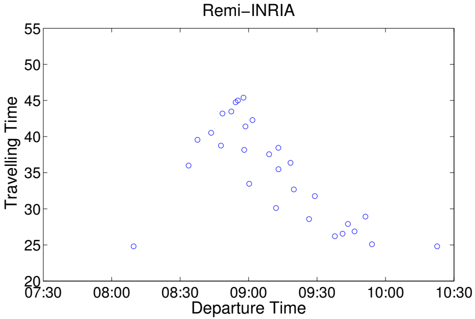
Testing error $\neq$ *Training* error

A Motivating Example

Question: What if we use data collected from Rémi (30 days)?

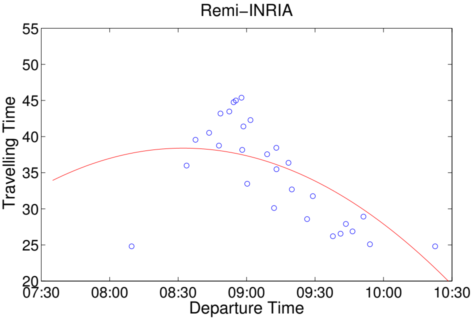
A Motivating Example

Question: What if we use data collected from Rémi (30 days)?



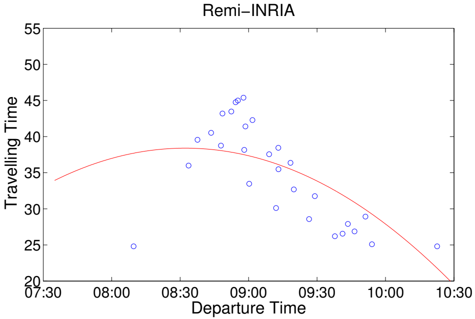
A Motivating Example

Question: What if we use data collected from Rémi (30 days)?



A Motivating Example

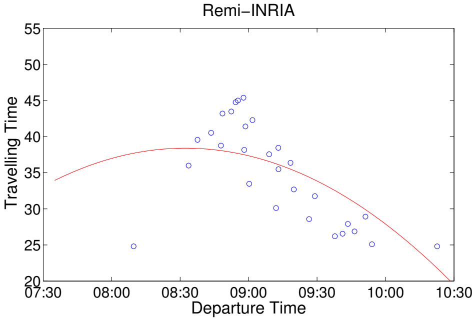
Question: What if we use data collected from Rémi (30 days)?



$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 26.6078$$

A Motivating Example

Question: What if we use data collected from Rémi (30 days)?



$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 26.6078$$

*The performance **changes** at each training set.*

A Motivating Example

Question: What if we use all the data together (60 days)?

A Motivating Example

Question: What if we use all the data together (60 days)?

$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 23.1641$$

A Motivating Example

Question: What if we use all the data together (60 days)?

$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 23.1641$$

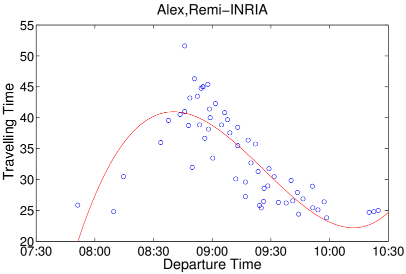
*The performance **improves** as the number of samples increases.*

A Motivating Example

Question: What if we used a polynomial of degree 4?

A Motivating Example

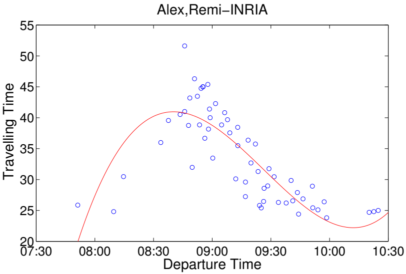
Question: What if we used a polynomial of degree 4?



$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 12.0554$$

A Motivating Example

Question: What if we used a polynomial of degree 4?



$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 12.0554$$

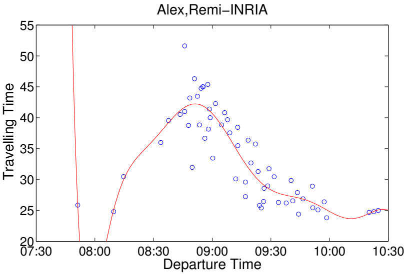
The performance ***improves*** *with the complexity of the polynomial.*

A Motivating Example

Question: Let's try a polynomial of degree 10!

A Motivating Example

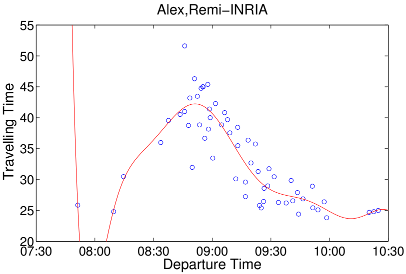
Question: Let's try a polynomial of degree 10!



$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 2.2488x10^5$$

A Motivating Example

Question: Let's try a polynomial of degree 10!

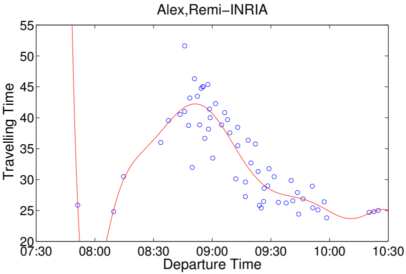


$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t)-y_t)^2 = 2.2488x10^5$$

The performance ~~improves~~ with the complexity of the polynomial

A Motivating Example

Question: Let's try a polynomial of degree 10!



$$\frac{1}{T}\sum_{t=1}^{T}(f(x_t) - y_t)^2 = 2.2488x10^5$$

The performance ***changes*** *with the complexity of the polynomial*

A Motivating Example

Lessons learned from the example

A Motivating Example

Lessons learned from the example

- The samples are distributed according to a ***sampling distribution***

A Motivating Example

Lessons learned from the example

- The samples are distributed according to a ***sampling distribution***
- The ***performance changes*** with the specific training set used to train the polynomial

A Motivating Example

Lessons learned from the example

- The samples are distributed according to a ***sampling distribution***
- The ***performance changes*** with the specific training set used to train the polynomial
- The performance ***improves with the number of samples*** in the training set

A Motivating Example

Lessons learned from the example

- The samples are distributed according to a ***sampling distribution***
- The ***performance changes*** with the specific training set used to train the polynomial
- The performance ***improves with the number of samples*** in the training set
- The performance ***changes with the complexity*** of the polynomial

A Motivating Example

Questions we will try to answer to

A Motivating Example

Questions we will try to answer to

- How much the performance changes with the training set?

A Motivating Example

Questions we will try to answer to

- How much the performance changes with the training set?
- How many samples do we need to guarantee a sufficient accuracy?

A Motivating Example

Questions we will try to answer to

- How much the performance changes with the training set?
- How many samples do we need to guarantee a sufficient accuracy?
- How should we choose the complexity of the polynomial?
- ...

Outline

Outline

The Binary Classification Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} = \{0, 1\}$

The Binary Classification Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} = \{0, 1\}$

The learner

- Hypothesis space $\mathcal{H} = \{h : \mathcal{X} \rightarrow \mathcal{Y}\}$

The Binary Classification Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} = \{0, 1\}$

The learner

- Hypothesis space $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$

The performance

- Loss function $\ell(y, \hat{y}) = \mathbb{I}\{y \neq \hat{y}\}$

The Binary Classification Problem: Examples

- Computer vision (e.g., medical imagining, character recognition, video tracking)

The Binary Classification Problem: Examples

- Computer vision (e.g., medical imagining, character recognition, video tracking)
- Natural language processing (e.g., document classification, spam filtering)

The Binary Classification Problem: Examples

- Computer vision (e.g., medical imagining, character recognition, video tracking)
- Natural language processing (e.g., document classification, spam filtering)
- Geostatistics (e.g, petroleum geology, meteorology, pollution monitoring)

The Binary Classification Problem: Examples

- Computer vision (e.g., medical imagining, character recognition, video tracking)
- Natural language processing (e.g., document classification, spam filtering)
- Geostatistics (e.g, petroleum geology, meteorology, pollution monitoring)
- Biostatistics (e.g, protein folding, sequence analysis)

The Binary Classification Problem: Examples

- Computer vision (e.g., medical imagining, character recognition, video tracking)
- Natural language processing (e.g., document classification, spam filtering)
- Geostatistics (e.g, petroleum geology, meteorology, pollution monitoring)
- Biostatistics (e.g, protein folding, sequence analysis)
- Economics (e.g., fraud detection, market trends)
- ...

The Empirical Risk Minimizer

The training set

- Samples of the form input–output $Z_n = \{z_t = (x_t, y_t)\}_{t=1}^n$

The Empirical Risk Minimizer

The training set

- Samples of the form input–output $Z_n = \{z_t = (x_t, y_t)\}_{t=1}^n$

The empirical risk minimizer

- Empirical risk of a hypothesis $h \in \mathcal{H}$ for the training set Z_n

$$\widehat{R}(h; Z_n) = \frac{1}{n} \sum_{t=1}^{n} \ell(y_t, h(x_t))$$

The Empirical Risk Minimizer

The training set

- Samples of the form input–output $Z_n = \{z_t = (x_t, y_t)\}_{t=1}^n$

The empirical risk minimizer

- Empirical risk of a hypothesis $h \in \mathcal{H}$ for the training set Z_n

$$\widehat{R}(h; Z_n) = \frac{1}{n} \sum_{t=1}^{n} \ell(y_t, h(x_t))$$

- The ERM

$$\hat{h}(\cdot; Z_n) = \arg\min_{h \in \mathcal{H}} \widehat{R}(h; Z_n)$$

A Stochastic Generative Model

Assumption (*Stochastic generative model*)

- *There exist a distribution $\mathcal{P}$ on the input–output space $\mathcal{X} \times \mathcal{Y}$*
- *All the pairs (x, y) are i.i.d. samples drawn from $\mathcal{P}$*

A Stochastic Generative Model

Assumption (*Stochastic generative model*)

- *There exist a distribution $\mathcal{P}$ on the input–output space $\mathcal{X} \times \mathcal{Y}$*
- *All the pairs (x, y) are i.i.d. samples drawn from $\mathcal{P}$*

- Expected risk of a hypothesis $h \in \mathcal{H}$ for distribution $\mathcal{P}$

$$R(h; \mathcal{P}) = \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h(x))\big]$$

A Stochastic Generative Model

Assumption (*Stochastic generative model*)

- *There exist a distribution $\mathcal{P}$ on the input–output space $\mathcal{X} \times \mathcal{Y}$*
- *All the pairs (x, y) are i.i.d. samples drawn from $\mathcal{P}$*

- Expected risk of a hypothesis $h \in \mathcal{H}$ for distribution $\mathcal{P}$

$$R(h; \mathcal{P}) = \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h(x))\big]$$

- Expected risk minimizer

$$h^*(\cdot; \mathcal{P}) = \arg\min_{h \in \mathcal{H}} R(h; \mathcal{P})$$

The Risk Bound Problem

Question: can we ***predict*** how well the ERM $\hat{h}$ will perform w.r.t. the best hypothesis h^*?

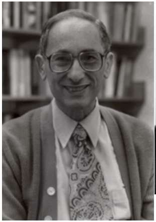
$$R(\hat{h}; \mathcal{P}) - R(h^*; \mathcal{P}) = ???$$

Outline

An Estimation Problem

Toss a (biased) coin n times.

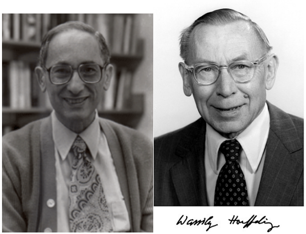
What is the probability of observing more than $n/2$ heads?



An Estimation Problem

Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with $p > 1/2$.

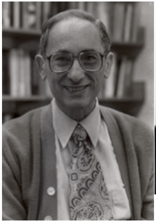
What is the probability of observing more than $n/2$ times the event $\{X_t = 1\}$?



An Estimation Problem

Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with $p > 1/2$.

What is the probability of observing more than $n/2$ times the event $\{X_t = 1\}$?

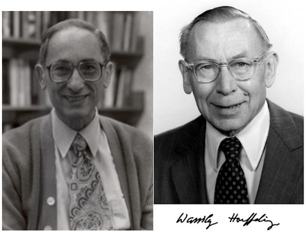


$$\mathbb{P}\Big[\sum_{t=1}^{n} X_t > \frac{n}{2}\Big] = \sum_{i=n/2+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

An Estimation Problem

Let $X_1, \ldots, X_n$ be independent Bernoulli random variables with $p > 1/2$.

What is the probability of observing more than $n/2$ times the event $\{X_t = 1\}$?



$$\mathbb{P}\Big[\sum_{t=1}^{n} X_t > \frac{n}{2}\Big] = \sum_{i=n/2+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

$$\mathbb{P}\Big[\sum_{t=1}^{n} X_t > \frac{n}{2}\Big] \geq 1 - \exp\left(-2n\left(p - \frac{1}{2}\right)^2\right)$$

The Chernoff–Hoeffding Bound

Theorem

Let $X_1, \ldots, X_n$ *be i.i.d. samples from a distribution bounded in* $[a, b]$*, then for any* $\varepsilon > 0$

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\right| > \varepsilon\right] \leq 2\exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right)$$

The Chernoff–Hoeffding Bound

Theorem

Let $X_1, \dots, X_n$ be i.i.d. samples from a distribution bounded in $[a, b]$, then for any $\varepsilon > 0$

$$\mathbb{P}\Big[\underbrace{\Big|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\Big|}_{\text{deviation}} > \underbrace{\varepsilon}_{\text{accuracy}}\Big] \leq \underbrace{2\exp\Big(-\frac{2n\varepsilon^2}{(b-a)^2}\Big)}_{\text{confidence}}$$

The Chernoff–Hoeffding Bound (Cont.d)

Theorem

Let $X_1, \ldots, X_n$ be i.i.d. samples from a distribution bounded in $[a, b]$, then for any $\delta \in (0, 1)$

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\right| > (b-a)\sqrt{\frac{\log 2/\delta}{2n}}\right] \leq \delta$$

The Chernoff–Hoeffding Bound (Cont.d)

Theorem

Let $X_1, X_2, \ldots$ be i.i.d. samples from a distribution bounded in $[a, b]$, then for any $\delta \in (0, 1)$ and $\varepsilon > 0$

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} X_t - \mathbb{E}[X_1]\right| > \varepsilon\right] \leq \delta$$

if $n \geq \frac{(b-a)^2 \log 2/\delta}{2\varepsilon^2}$.

Back to the Binary Classification Problem (1)

Recall that

$$\hat{h}(\cdot; Z_n) = \arg\min_{h \in \mathcal{H}} \widehat{R}(h; Z_n) \quad \text{and} \quad h^*(\cdot; \mathcal{P}) = \arg\min_{h \in \mathcal{H}} R(h; \mathcal{P})$$

Back to the Binary Classification Problem (1)

Recall that

$$\hat{h}(\cdot; Z_n) = \arg\min_{h \in \mathcal{H}} \widehat{R}(h; Z_n) \quad \text{and} \quad h^*(\cdot; \mathcal{P}) = \arg\min_{h \in \mathcal{H}} R(h; \mathcal{P})$$

so we should first understand what is the difference between

$$\widehat{R}(h; Z_n) = \frac{1}{n} \sum_{t=1}^{n} \ell(y_t, h(x_t)) \quad \text{and} \quad R(h; \mathcal{P}) = \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h(x))\big]$$

Back to the Binary Classification Problem (1)

Notice that for any fixed $h \in \mathcal{H}$ and training set Z_n

$$\left| \widehat{R}(h; Z_n) - R(h; \mathcal{P}) \right|$$

Back to the Binary Classification Problem (1)

Notice that for any fixed $h \in \mathcal{H}$ and training set Z_n

$$\left|\widehat{R}(h; Z_n) - R(h; \mathcal{P})\right|$$

$$\left|\frac{1}{n}\sum_{t=1}^{n} \ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h(x))\big]\right|$$

Back to the Binary Classification Problem (1)

Notice that for any fixed $h \in \mathcal{H}$ and training set Z_n

$$\left| \widehat{R}(h; Z_n) - R(h; \mathcal{P}) \right|$$

$$\left| \frac{1}{n} \sum_{t=1}^{n} \ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}} \big[\ell(y, h(x)) \big] \right|$$

$$\left| \frac{1}{n} \sum_{t=1}^{n} X_t - \mathbb{E}[X_1] \right|$$

Back to the Binary Classification Problem (1)

Lemma

Let Z_n be a training set of n i.i.d. samples drawn from a distribution $\mathcal{P}$, then for any fixed $h \in \mathcal{H}$ and $\delta \in (0,1)$

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(y, h(x))\right]\right| > \sqrt{\frac{\log 2/\delta}{2n}}\right] \leq \delta$$

Back to the Binary Classification Problem (1)

Lemma

Let Z_n be a training set of n i.i.d. samples drawn from a distribution $\mathcal{P}$, then for any ***fixed*** $h \in \mathcal{H}$ *and* $\delta \in (0,1)$

$$\mathbb{P}\Big[\Big|\underbrace{\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h(x_t))}_{\text{empirical risk}} - \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h(x))\big]}_{\text{expected risk}}\Big| > \sqrt{\frac{\log 2/\delta}{2n}}\Big] \leq \delta$$

Back to the Binary Classification Problem (1)

Problem: we want to study the performance of the ***random*** ERM

$$\hat{h}(\cdot; Z_n) = \arg\min_{h \in \mathcal{H}} \widehat{R}(h; Z_n)$$

The Union Bound

Also known as: Boole's inequality, Bonferroni inequality, etc.

Theorem

Let $A_1, A_2, \ldots$ be a countable set of events, then

$$\mathbb{P}\Big[\bigcup_i A_i\Big] \leq \sum_i \mathbb{P}[A_i].$$

The Union Bound

Also known as: Boole's inequality, Bonferroni inequality, etc.

Theorem

Let $A_1, A_2, \dots$ be a countable set of events, then

$$\mathbb{P}\Big[\bigcup_i A_i\Big] \leq \sum_i \mathbb{P}[A_i].$$

Back to the Binary Classification Problem (2)

Problem: we want to study the performance of the ***random*** ERM

$$\hat{h}(\cdot; Z_n) = \arg\min_{h \in \mathcal{H}} \widehat{R}(h; Z_n)$$

Back to the Binary Classification Problem (2)

Problem: we want to study the performance of the ***random*** ERM

$$\hat{h}(\cdot; Z_n) = \arg\min_{h \in \mathcal{H}} \widehat{R}(h; Z_n)$$

$$\mathbb{P}\left[\exists h \in \mathcal{H} : \left| \frac{1}{n} \sum_{t=1}^{n} \ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h(x))\big] \right| > \varepsilon \right]$$

Back to the Binary Classification Problem (2)

Problem: we want to study the performance of the ***random*** ERM

$$\hat{h}(\cdot; Z_n) = \arg\min_{h\in\mathcal{H}} \widehat{R}(h; Z_n)$$

$$\mathbb{P}\Big[\Big\{\Big|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h_1(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h_1(x))\big]\Big| > \varepsilon\Big\}\bigcup$$
$$\Big\{\Big|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h_2(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h_2(x))\big]\Big| > \varepsilon\Big\}\bigcup$$
$$\dots$$
$$\Big\{\Big|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h_N(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h_N(x))\big]\Big| > \varepsilon\Big\}\bigcup$$
$$\dots\Big]$$

Back to the Binary Classification Problem (2)

Lemma

Let Z_n be a training set of n i.i.d. samples drawn from a distribution $\mathcal{P}$ and $\mathcal{H}$ a finite hypothesis set with $|\mathcal{H}| = N$, then for any $\delta \in (0,1)$

$$\mathbb{P}\left[\exists h \in \mathcal{H} : \left|\frac{1}{n}\sum_{t=1}^{n} \ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(y, h(x))\right]\right| > \sqrt{\frac{\log 2/\delta}{2n}}\right] \leq$$

$$N\mathbb{P}\left[\left|\frac{1}{n}\sum_{t=1}^{n} \ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(y, h(x))\right]\right| > \sqrt{\frac{\log 2/\delta}{2n}}\right] \leq N\delta$$

Back to the Binary Classification Problem (2)

Lemma

Let Z_n be a training set of n i.i.d. samples drawn from a distribution $\mathcal{P}$ and $\mathcal{H}$ a finite hypothesis set with $|\mathcal{H}| = N$, then for any $\delta \in (0, 1)$

$$\mathbb{P}\left[\exists h \in \mathcal{H}: \left|\frac{1}{n}\sum_{t=1}^{n} \ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(y, h(x))\right]\right| > \sqrt{\frac{\log 2N/\delta}{2n}}\right] \leq \delta$$

Back to the Binary Classification Problem (2)

Problem: In general $\mathcal{H}$ contains an infinite number of hypotheses (e.g., a linear classifier)

The Symmetrization Trick

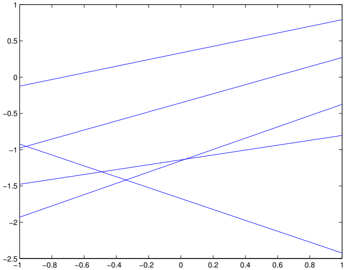
$$\mathbb{P}\Big[\exists h \in \mathcal{H} : \Big|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\big[\ell(y, h(x))\big]\Big| > \varepsilon\Big]$$

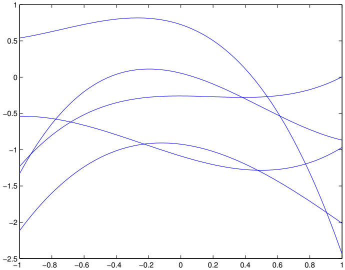
$$\leq 2\mathbb{P}\Big[\exists h \in \mathcal{H} : \Big|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h(x_t)) - \frac{1}{n}\sum_{t=1}^{n}\ell(y'_t, h(x'_t))\Big| > \frac{\varepsilon}{2}\Big]$$

with the ghost samples $\{(x'_t, y'_t)\}_{t=1}^{n}$ independently drawn from $\mathcal{P}$.

The VC dimension

Not all the *infinities* are the same...

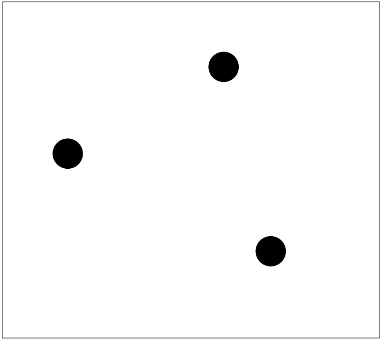




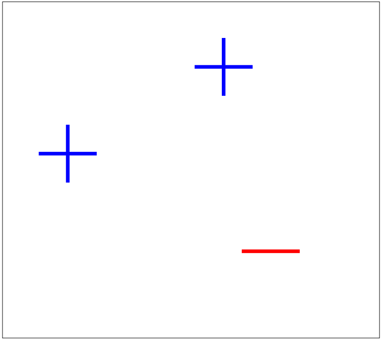
The VC dimension (cont'd)

How many *different predictions* can a space $\mathcal{H}$ produce over n distinct inputs?

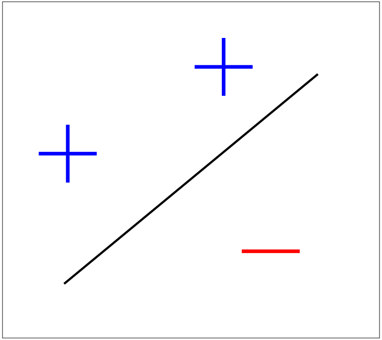
The VC dimension (cont'd)



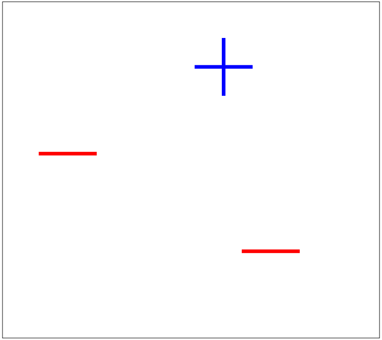
The VC dimension (cont'd)



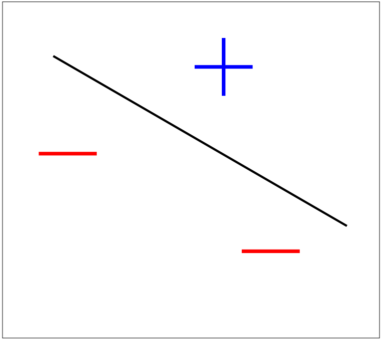
The VC dimension (cont'd)



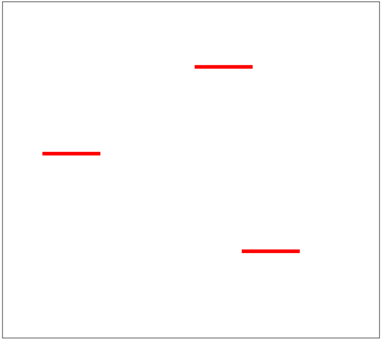
The VC dimension (cont'd)



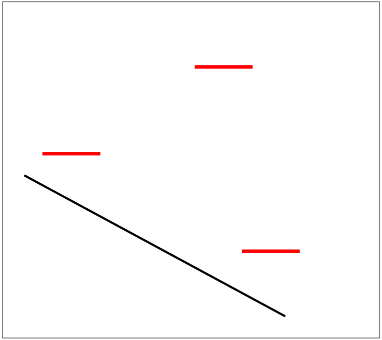
The VC dimension (cont'd)



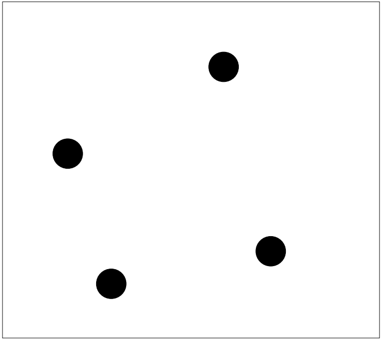
The VC dimension (cont'd)



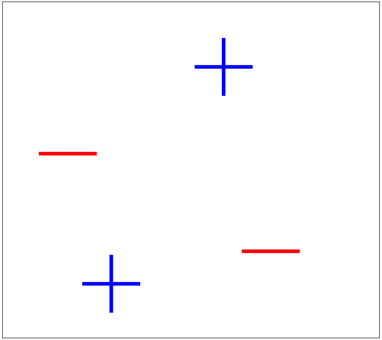
The VC dimension (cont'd)



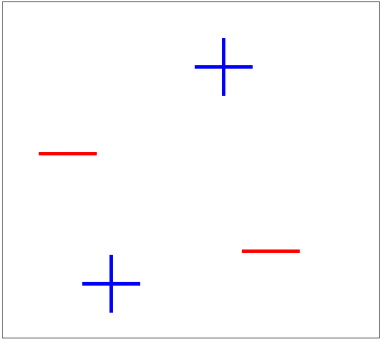
The VC dimension (cont'd)



The VC dimension (cont'd)



The VC dimension (cont'd)



The *VC dimension* of a linear classifier in dim. 2 is $VC(\mathcal{H}) = 3$.

The VC dimension (cont'd)

Let $S = (x_1, \ldots, x_d)$ be an arbitrary sequence of points, then

$$\Pi_S(\mathcal{H}) = \{(h(x_1), \ldots, h(x_d)), h \in \mathcal{H}\}$$

is the set of all the possible ways the d points can be classified by hypothesis in $\mathcal{H}$.

The VC dimension (cont'd)

Let $S = (x_1, \ldots, x_d)$ be an arbitrary sequence of points, then

$$\Pi_S(\mathcal{H}) = \{(h(x_1), \ldots, h(x_d)), h \in \mathcal{H}\}$$

is the set of all the possible ways the d points can be classified by hypothesis in $\mathcal{H}$.

Definition

A set S is shattered by a hypothesis space $\mathcal{H}$ if $|\Pi_S(\mathcal{H})| = 2^d$.

The VC dimension (cont'd)

Definition (VC Dimension)

The VC dimension of a hypothesis space $\mathcal{H}$ is

$$VC(\mathcal{H}) = \max\{d \mid \exists |S| = d, |\Pi_S(\mathcal{H})| = 2^d\}$$

The VC dimension (cont'd)

Definition (VC Dimension)

The VC dimension of a hypothesis space $\mathcal{H}$ is

$$\mathrm{VC}(\mathcal{H}) = \max\{d \mid \exists |S| = d, |\Pi_S(\mathcal{H})| = 2^d\}$$

Lemma (Sauer's Lemma)

Let $\mathcal{H}$ be a hypothesis space with VC dimension d, then for any sequence of n points $S = (x_1, \ldots, x_n)$ with $n > d$

$$|\Pi_S(\mathcal{H})| \leq \sum_{i=0}^{d} \binom{n}{i} \leq n^d$$

Back to the Binary Classification Problem (3)

Question: how many values can $\ell(\cdot,\cdot)$ take on $2n$ samples?

$$2\mathbb{P}\left[\exists h \in \mathcal{H} : \left|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h(x_t)) - \frac{1}{n}\sum_{t=1}^{n}\ell(y'_t, h(x'_t))\right| > \frac{\varepsilon}{2}\right]$$

Back to the Binary Classification Problem (3)

Question: how many values can $\ell(\cdot,\cdot)$ take on $2n$ samples?

$$2\mathbb{P}\left[\exists h \in \mathcal{H} : \left|\frac{1}{n}\sum_{t=1}^{n}\ell(y_t, h(x_t)) - \frac{1}{n}\sum_{t=1}^{n}\ell(y'_t, h(x'_t))\right| > \frac{\varepsilon}{2}\right]$$

If $\mathrm{VC}(\mathcal{H}) = d$ and $2n > d$, then the answer is ***at most*** $(2n)^d$!

Back to the Binary Classification Problem (3)

Lemma

Let Z_n be a training set of n i.i.d. samples drawn from a distribution $\mathcal{P}$ and $\mathcal{H}$ a hypothesis space with $VC(\mathcal{H}) = d$, then for any $\delta \in (0,1)$

$$\mathbb{P}\left[\exists h \colon \left| \frac{1}{n}\sum_{t=1}^{n} \ell(y_t, h(x_t)) - \mathbb{E}_{(x,y)\sim\mathcal{P}}\left[\ell(y, h(x))\right] \right| > 2\sqrt{\frac{\log 2N/\delta}{2n}}\right] \leq 2\delta$$

with $N = (2n)^d$.

Back to the Binary Classification Problem (3)

A *simplified reading* of the previous lemma.

For any training set Z_n and any hypothesis $h \in \mathcal{H}$ the error of using the empirical risk instead of the expected risk is

$$\left|\widehat{R}(h; Z_n) - R(h; \mathcal{P})\right| \leq O\Big(\sqrt{\frac{d \log n/\delta}{n}}\Big)$$

with at least $1 - \delta$ probability.

The Final Proof

Putting all the pieces together...

$$R(\hat{h}; \mathcal{P}) - R(h^*; \mathcal{P}) =$$

The Final Proof

Putting all the pieces together...

$$R(\hat{h}; \mathcal{P}) - R(h^*; \mathcal{P}) =$$
$$= R(\hat{h}; \mathcal{P}) - \widehat{R}(\hat{h}; Z_n) + \widehat{R}(\hat{h}; Z_n) - \widehat{R}(h^*; Z_n) + \widehat{R}(h^*; Z_n) - R(h^*; \mathcal{P})$$

The Final Proof

Putting all the pieces together...

$$R(\hat{h};\mathcal{P}) - R(h^*;\mathcal{P}) =$$
$$= \underbrace{R(\hat{h};\mathcal{P}) - \widehat{R}(\hat{h};Z_n)}_{\text{diff empirical/expected}} + \underbrace{\widehat{R}(\hat{h};Z_n) - \widehat{R}(h^*;Z_n)}_{\hat{h}\text{ is the ERM}} + \underbrace{\widehat{R}(h^*;Z_n) - R(h^*;\mathcal{P})}_{\text{diff empirical/expected}}$$

The Final Proof

Putting all the pieces together...

$$R(\hat{h};\mathcal{P}) - R(h^*;\mathcal{P}) =$$

$$= \underbrace{R(\hat{h};\mathcal{P}) - \widehat{R}(\hat{h};Z_n)}_{\text{diff empirical/expected}} + \underbrace{\widehat{R}(\hat{h};Z_n) - \widehat{R}(h^*;Z_n)}_{\hat{h} \text{ is the ERM}} + \underbrace{\widehat{R}(h^*;Z_n) - R(h^*;\mathcal{P})}_{\text{diff empirical/expected}}$$

$$\leq O\Big(\sqrt{\frac{d \log n/\delta}{n}}\Big)$$

The Final Proof

Putting all the pieces together...

$$R(\hat{h};\mathcal{P}) - R(h^*;\mathcal{P}) =$$

$$= \underbrace{R(\hat{h};\mathcal{P}) - \widehat{R}(\hat{h};Z_n)}_{\text{diff empirical/expected}} + \underbrace{\widehat{R}(\hat{h};Z_n) - \widehat{R}(h^*;Z_n)}_{\hat{h}\text{ is the ERM}} + \underbrace{\widehat{R}(h^*;Z_n) - R(h^*;\mathcal{P})}_{\text{diff empirical/expected}}$$

$$\leq O\Big(\sqrt{\frac{d \log n/\delta}{n}}\Big) + 0$$

The Final Proof

Putting all the pieces together...

$$R(\hat{h};\mathcal{P}) - R(h^*;\mathcal{P}) =$$

$$= \underbrace{R(\hat{h};\mathcal{P}) - \widehat{R}(\hat{h};Z_n)}_{\text{diff empirical/expected}} + \underbrace{\widehat{R}(\hat{h};Z_n) - \widehat{R}(h^*;Z_n)}_{\hat{h}\text{ is the ERM}} + \underbrace{\widehat{R}(h^*;Z_n) - R(h^*;\mathcal{P})}_{\text{diff empirical/expected}}$$

$$\leq O\Big(\sqrt{\frac{d \log n/\delta}{n}}\Big) + 0 + O\Big(\sqrt{\frac{d \log n/\delta}{n}}\Big)$$

The Final Proof

Putting all the pieces together...

$$
\begin{aligned}
&R(\hat{h};\mathcal{P}) - R(h^*;\mathcal{P}) = \\
&= \underbrace{R(\hat{h};\mathcal{P}) - \widehat{R}(\hat{h};Z_n)}_{\text{diff empirical/expected}} + \underbrace{\widehat{R}(\hat{h};Z_n) - \widehat{R}(h^*;Z_n)}_{\hat{h}\text{ is the ERM}} + \underbrace{\widehat{R}(h^*;Z_n) - R(h^*;\mathcal{P})}_{\text{diff empirical/expected}} \\
&\leq O\Big(\sqrt{\frac{d \log n/\delta}{n}}\Big) + 0 + O\Big(\sqrt{\frac{d \log n/\delta}{n}}\Big) \qquad \text{w.p. } 1 - 2\delta
\end{aligned}
$$

The Final Bound

Theorem (VC–Bound)

Let Z_n be a training set of n i.i.d. samples from a distribution $\mathcal{P}$ and $\mathcal{H}$ be a hypothesis space with $VC(\mathcal{H}) = d$. If

$$\hat{h}(\cdot; Z_n) = \arg\min_{h \in \mathcal{H}} \widehat{R}(h; Z_n)$$

and

$$h^*(\cdot; \mathcal{P}) = \arg\min_{h \in \mathcal{H}} R(h; \mathcal{P})$$

then

$$R(\hat{h}; \mathcal{P}) \leq R(h^*; \mathcal{P}) + O\left(\sqrt{\frac{d \log n/\delta}{n}}\right)$$

with probability at least $1 - \delta$ (w.r.t. the randomness in the training set).

Reading the Bound

$$\underbrace{R(\hat{h};\mathcal{P})}_{\textit{risk}} \leq \underbrace{R(h^*;\mathcal{P})}_{\textit{approximation error}} + \underbrace{O\Big(\sqrt{\frac{d\log n/\delta}{n}}\Big)}_{\textit{estimation error}}$$

Reading the Bound (cont'd)

Question: If we have n samples and we use a linear classifier in a d-dim space, we want to predict how much error we make with a confidence $1 - \delta$.

Reading the Bound (cont'd)

Question: If we have n samples and we use a linear classifier in a d-dim space, we want to predict how much error we make with a confidence $1-\delta$.

Answer:

$$R(\hat{h};\mathcal{P}) \leq R(h^*;\mathcal{P}) + O\Big(\sqrt{\frac{(d+1)\log n/\delta}{n}}\Big)$$

Reading the Bound (cont'd)

Question: What happens if we keep increasing the number of samples?

Reading the Bound (cont'd)

Question: What happens if we keep increasing the number of samples?

Answer:

$$\lim_{n\to\infty} R(\hat{h};\mathcal{P}) \leq R(h^*;\mathcal{P})$$

We converge to the same performance as the best hypothesis h^* in our space.

Reading the Bound (cont'd)

Question: We can accept at most an error ε over $(1-\delta)\%$ of times, how many samples should we use?

Reading the Bound (cont'd)

Question: We can accept at most an error ε over $(1-\delta)$% of times, how many samples should we use?

Answer:

$$n \geq O\Big(\frac{d \log 1/\delta}{\varepsilon^2}\Big)$$

Reading the Bound (cont'd)

Question: We are using polynomials, what is the right degree d to use?

Reading the Bound (cont'd)

Question: We are using polynomials, what is the right degree d to use?

partial **Answer**: it depends on how good your space $\mathcal{H}$ is and how many samples you have.

Reading the Bound (cont'd)

Question: We are using polynomials, what is the right degree d to use?

Remark 1: if $d > n$ then $O\left(\sqrt{d \log(n/\delta)/n}\right) \approx 1$... not very useful...

Reading the Bound (cont'd)

Question: We are using polynomials, what is the right degree d to use?

Remark 2: let $R(h^*; \mathcal{P})$ be a decreasing function of d (say $f(d)$), then there exist an optimal d^* such that

$$d^* = \arg\min_d \left(f(d) + O\Big(\sqrt{\frac{(d+1)\log n/\delta}{n}}\Big)\right)$$

Outline

The Regression Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} \subseteq \mathbb{R}$

The Regression Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} \subseteq \mathbb{R}$

The learner

- Function space $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$

The Regression Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} \subseteq \mathbb{R}$

The learner

- Function space $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$

The performance

- Loss function $\ell(y, \hat{y})$

The Least–squares Regression Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} \subseteq \mathbb{R}$

The Least–squares Regression Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} \subseteq \mathbb{R}$

The learner

- Basis functions $\varphi_i : \mathcal{X} \rightarrow \mathcal{Y}$, $i = 1, \ldots, d$
- Linear d-dim function space
  $\mathcal{F} = \{f_\alpha(\cdot) = \sum_{i=1}^d \alpha_i \varphi_i(\cdot);\ \alpha \in \mathbb{R}^d\}$

The Least–squares Regression Problem

The environment

- Input space $\mathcal{X} \subseteq \mathbb{R}^s$
- Output space $\mathcal{Y} \subseteq \mathbb{R}$

The learner

- Basis functions $\varphi_i : \mathcal{X} \to \mathcal{Y}$, $i = 1, \ldots, d$
- Linear d-dim function space
  $\mathcal{F} = \{f_\alpha(\cdot) = \sum_{i=1}^d \alpha_i \varphi_i(\cdot);\ \alpha \in \mathbb{R}^d\}$

The performance

- Loss function $\ell(y, \hat{y}) = (y - \hat{y})^2$

The Least–squares Regression Problem (cont'd)

In the polynomial regression example (e.g., order 2):

- Basis functions: $\varphi_1(x) = x^2, \varphi_2(x) = x, \varphi_3(x) = 1$
- Function space

$$\mathcal{F} = \{f_\alpha(x) = \alpha_1 x^2 + \alpha_2 x + \alpha_3\}$$

The Empirical Risk Minimizer

The training set

- Samples of the form input–output $Z_n = \{z_t = (x_t, y_t)\}_{t=1}^n$

The Empirical Risk Minimizer

The training set

- Samples of the form input–output $Z_n = \{z_t = (x_t, y_t)\}_{t=1}^n$

The empirical risk minimizer

- Empirical risk of a function $f_\alpha \in \mathcal{F}$ for the training set Z_n

$$\widehat{R}(f_\alpha; Z_n) = \frac{1}{n}\sum_{t=1}^{n}(y_t - f_\alpha(x_t))^2$$

The Empirical Risk Minimizer

The training set

- Samples of the form input–output $Z_n = \{z_t = (x_t, y_t)\}_{t=1}^n$

The empirical risk minimizer

- Empirical risk of a function $f_\alpha \in \mathcal{F}$ for the training set Z_n

$$\widehat{R}(f_\alpha; Z_n) = \frac{1}{n}\sum_{t=1}^{n}(y_t - f_\alpha(x_t))^2$$

- The ERM

$$f_{\hat{\alpha}}(\cdot; Z_n) = \arg\min_{f_\alpha \in \mathcal{F}} \widehat{R}(f; Z_n)$$

A Stochastic Generative Model

Assumption (*Stochastic generative model*)

- *There exists a distribution ρ on the input space $\mathcal{X}$*
- *There exists a target function $f^* : \mathcal{X} \rightarrow \mathcal{Y}$*
- *There exists a (zero-mean bounded) noise ξ, such that $\mathbb{E}[\xi] = 0$ and $|\xi| < C$*
- *All the pairs (x, y) are i.i.d. samples generated as*

$$y = f^*(x) + \xi, \quad x \sim \mathcal{P}_{\mathcal{X}}$$

A Stochastic Generative Model

Assumption (*Stochastic generative model*)

- *There exists a distribution ρ on the input space $\mathcal{X}$*
- *There exists a target function $f^* : \mathcal{X} \to \mathcal{Y}$*
- *There exists a (zero-mean bounded) noise ξ, such that $\mathbb{E}[\xi] = 0$ and $|\xi| < C$*
- *All the pairs (x, y) are i.i.d. samples generated as*

$$y = f^*(x) + \xi, \quad x \sim \mathcal{P}_{\mathcal{X}}$$

- Expected risk of $f \in \mathcal{F}$ w.r.t. the target function f^* and a distribution ρ

$$R(f_\alpha; f^*, \rho) = \mathbb{E}_{x\sim\rho}\big[(f_\alpha(x) - f^*(x))^2\big]$$

A Stochastic Generative Model

Assumption (*Stochastic generative model*)

- *There exists a distribution ρ on the input space $\mathcal{X}$*
- *There exists a target function $f^* : \mathcal{X} \to \mathcal{Y}$*
- *There exists a (zero-mean bounded) noise ξ, such that $\mathbb{E}[\xi] = 0$ and $|\xi| < C$*
- *All the pairs (x, y) are i.i.d. samples generated as*

$$y = f^*(x) + \xi, \quad x \sim \mathcal{P}_{\mathcal{X}}$$

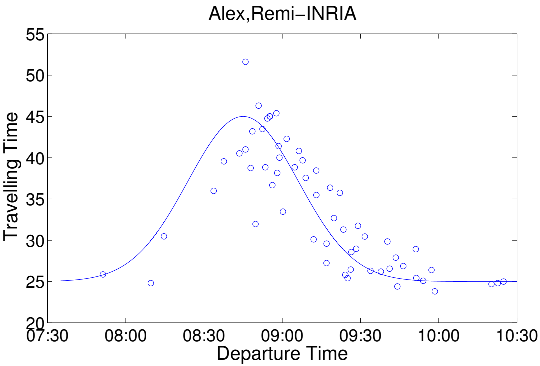
- Expected risk of $f \in \mathcal{F}$ w.r.t. the target function f^* and a distribution ρ

$$R(f_\alpha; f^*, \rho) = \mathbb{E}_{x\sim\rho}\big[(f_\alpha(x) - f^*(x))^2\big]$$

- Expected risk minimizer

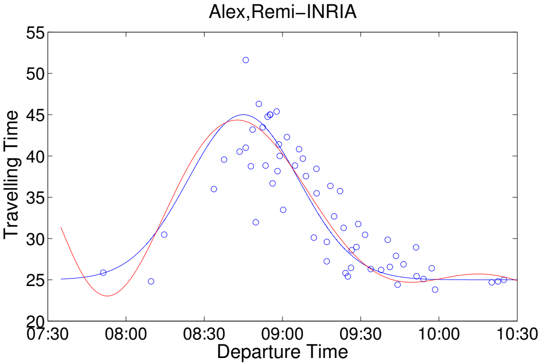
$$f_{\alpha^*}(\cdot; f^*, \rho) = \arg\min_{f_\alpha \in \mathcal{F}} R(f_\alpha; f^*, \rho)$$

Back to the Motivating Example



$$f^*(x) = a + b\exp\left(-\frac{(x-c)^2}{d^2}\right)$$

Back to the Motivating Example



$$f^*(x) = a + b \exp\left(-\frac{(x-c)^2}{d^2}\right)$$

A Bit More of Notation

Norms

- L2–weighted norm of f w.r.t. a distribution ρ

$$||f||_{2,\rho}^2 = \mathbb{E}_{x\sim\rho}[f(x)^2]$$

- L2–weighted empirical norm of f w.r.t. a sequence $(x_1, \ldots, x_n)$

$$||f||_{2,n}^2 = \frac{1}{n}\sum_{t=1}^{n} f(x_t)^2$$

- L2–weighted empirical norm of a vector $v \in \mathbb{R}^n$

$$||v||_{2,n}^2 = \frac{1}{n}\sum_{t=1}^{n} v_t^2$$

A Bit More of Notation (cont'd)

Vector space (from $\mathcal{F}$ on $(x_1, \ldots, x_n)$)

$$\mathcal{F}_n = \{(f_\alpha(x_1), \ldots, f_\alpha(x_n));\ f_\alpha \in \mathcal{F}\}$$

Projection operator

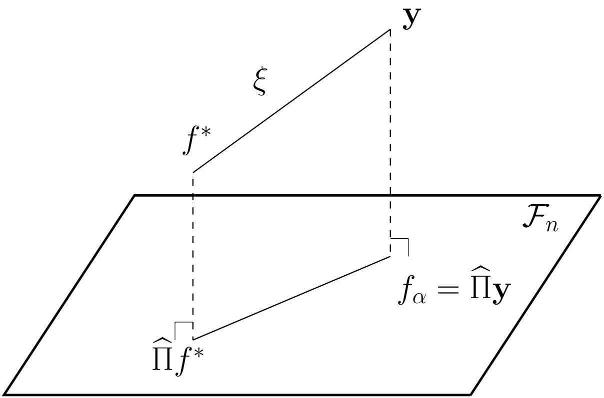
- Projection operator Π of a function f^* onto a function space $\mathcal{F}$

$$\Pi f^* = \arg\min_{f \in \mathcal{F}} ||f - f^*||_{2,\rho}$$

- Empirical projection operator $\widehat{\Pi}_n$ of a vector $\mathbf{y}$ onto a vector space $\mathcal{F}_n$

$$\widehat{\Pi}_n \mathbf{y} = \arg\min_{\mathbf{f} \in \mathcal{F}_n} ||\mathbf{f} - \mathbf{y}||_{2,n}$$

A Geometric View



Least–squares Solution

Recalling the definition of risk above we have ($\mathbf{y} = (y_1, \ldots, y_n)$)

$$f_{\alpha^*} = \Pi f^*$$

$$f_{\hat{\alpha}} = \hat{\Pi}_n \mathbf{y}$$

Least–squares Solution

Recalling the definition of risk above we have ($\mathbf{y} = (y_1, \ldots, y_n)$)

$$f_{\alpha^*} = \Pi f^*$$

$$f_{\hat{\alpha}} = \hat{\Pi}_n \mathbf{y}$$

Given feature matrix $\Phi \in \mathbb{R}^{n \times d}$

$$\Phi_{t,i} = \varphi_i(X_t)$$

the least–squares solution is

$$\hat{\alpha} = (\Phi^\top \Phi)^{-1} \mathbf{y}$$

A Prediction Error Bound

Theorem

Let the training set Z_n be generated according to the generative model above with f^* the target function and a bounded noise $|\xi| \leq C$. If $\mathcal{F}$ is a d-dimensional linear function space, then the least–squares solution satisfies:

$$||f_{\hat{\alpha}} - f^*||^2_{2,\rho} \leq 8||f_{\alpha^*} - f^*||^2_{2,\rho} + O\Big(\frac{d \log n/\delta}{n}\Big)$$

with probability $1 - \delta$.

A Prediction Error Bound (cont'd)

$$\underbrace{||f_{\hat{\alpha}} - f^*||^2_{2,\rho}}_{\text{prediction error}} \leq \underbrace{8||f_{\alpha^*} - f^*||^2_{2,\rho}}_{\text{approximation error}} + \underbrace{O\Big(\frac{d \log n/\delta}{n}\Big)}_{\text{estimation error}}$$

A Prediction Error Bound (cont'd)

Least–squares regression vs binary classification

$$\underbrace{O\Big(\frac{d\log n/\delta}{n}\Big)}_{\text{LS regression}} \ll \underbrace{O\Big(\sqrt{\frac{d\log n/\delta}{n}}\Big)}_{\text{classification}}$$

$$\underbrace{8\|f_{\alpha^*}-f^*\|_{2,\rho}^2}_{\text{LS regression}} \gg \underbrace{R(h^*;\mathcal{P})}_{\text{classification}}$$

Least–squares Solution in High-Dimensions

Question: How should we design the basis functions so as to have a small approximation error?

Least–squares Solution in High-Dimensions

Question: How should we design the basis functions so as to have a small approximation error?

Answer: If you do not have a specific domain knowledge, just keep adding features! (possibly independent...)

Least–squares Solution in High-Dimensions

Question: How should we design the basis functions so as to have a small approximation error?

Answer: If you do not have a specific domain knowledge, just keep adding features! (possibly independent...)

Problem: the bound scales linearly with d and so the need for samples. So the more the features the more the samples!

Least–squares Solution in High-Dimensions

Question: How should we design the basis functions so as to have a small approximation error?

Answer: If you do not have a specific domain knowledge, just keep adding features! (possibly independent...)

Problem: the bound scales linearly with d and so the need for samples. So the more the features the more the samples!
Actually if $d \geq n$ then the bounds are completely useless!

L1-Regularized Least–squares Regression

Assumption (High–dimensional and Sparsity assumption)

The target function f^* belong to the high–dimensional function space $\mathcal{F}$, that is

$$f_{\alpha^*} = \Pi f^* = f^* \; (||f_{\alpha^*} - f||_{2,\rho} = 0)$$

and it can be represented by a small subset of the d features defining $\mathcal{F}$, that is

$$||\alpha^*||_0 \ll d.$$

L1-Regularized Least–squares Regression

Given the previous assumption we want to force $f_{\hat{\alpha}}$ to be sparse too. Thus,

$$f_{\hat{\alpha}} = \arg\min_{f_\alpha \in \mathcal{F}} \frac{1}{n} \sum_{t=1}^{n} (y_t - f_\alpha(x_t))^2 + \lambda ||\alpha||_0$$

L1-Regularized Least–squares Regression

Given the previous assumption we want to force $f_{\hat{\alpha}}$ to be sparse too. Thus,

$$f_{\hat{\alpha}} = \arg\min_{f_\alpha \in \mathcal{F}} \frac{1}{n} \sum_{t=1}^{n} (y_t - f_\alpha(x_t))^2 + \lambda ||\alpha||_0$$

Problem: this optimization problem is NP-hard...

L1-Regularized Least–squares Regression (cont'd)

The *LASSO* (least absolute shrinkage and selection operator)

$$f_{\hat{\alpha}} = \arg\min_{f_\alpha \in \mathcal{F}} \frac{1}{n} \sum_{t=1}^{n} (y_t - f_\alpha(x_t))^2 + \lambda ||\alpha||_1$$

L1-Regularized Least–squares Regression (cont'd)

The *LASSO* (least absolute shrinkage and selection operator)

$$f_{\hat{\alpha}} = \arg\min_{f_\alpha \in \mathcal{F}} \frac{1}{n} \sum_{t=1}^{n} (y_t - f_\alpha(x_t))^2 + \lambda ||\alpha||_1$$

The L1–norm is known to be a *sparsity–inducing* norm.

L1-Regularized Least–squares Regression (cont'd)

The *LASSO* (least absolute shrinkage and selection operator)

$$f_{\hat{\alpha}} = \arg\min_{f_\alpha \in \mathcal{F}} \frac{1}{n}\sum_{t=1}^{n}(y_t - f_\alpha(x_t))^2 + \lambda||\alpha||_1$$

The L1–norm is known to be a *sparsity–inducing* norm.

Related to: model selection, feature selection, compressed sensing, high–dimensional statistics, etc.

A Prediction Error Bound (1)

Let us first state a bound for an *oracle* which knows in advance the features corresponding to non–zero α^* coefficients.

A Prediction Error Bound (1)

Let us first state a bound for an *oracle* which knows in advance the features corresponding to non–zero α^* coefficients.

Theorem

An oracle running ordinary least–squares on the set of features $S = \{i | \alpha_i^* \neq 0\}$ *with* $|S| = s \ll d$ *would obtain a performance*

$$||f_{\hat{\alpha}}^{ols} - f^*||_{2,n}^2 \leq 8||f_{\alpha^*} - f^*||_{2,n}^2 + O\Big(\frac{s \log n/\delta}{n}\Big)$$

A Prediction Error Bound (1)

Let us first state a bound for an *oracle* which knows in advance the features corresponding to non–zero α^* coefficients.

Theorem

An oracle running ordinary least–squares on the set of features $S = \{i | \alpha_i^* \neq 0\}$ with $|S| = s \ll d$ would obtain a performance

$$||f_{\hat{\alpha}}^{ols} - f^*||_{2,n}^2 \leq O\Big(\frac{s \log n/\delta}{n}\Big)$$

A Prediction Error Bound (1)

Let us first state a bound for an *oracle* which knows in advance the features corresponding to non–zero α^* coefficients.

Theorem

An oracle running ordinary least–squares on the set of features $S = \{i|\alpha_i^* \neq 0\}$ with $|S| = s \ll d$ would obtain a performance

$$||f_{\hat{\alpha}}^{ols} - f^*||_{2,n}^2 \leq O\Big(\frac{s \log n/\delta}{n}\Big)$$

Note: we now consider *fixed* design bounds instead of *random* design bounds.

A Prediction Error Bound (2)

Theorem

Let $f_{\hat{\alpha}}$ be the function returned by LASSO when trained on a training set Z_n and a d-dimensional function space $\mathcal{F}$, then

$$||f_{\hat{\alpha}} - f^*||^2_{2,n} \leq O\Big(||\alpha^*||_1 \sqrt{\frac{\log d/\delta}{n}}\Big)$$

if $\lambda = O(\sqrt{\log(d/\delta)/n})$.

A Prediction Error Bound (2)

Theorem

Let $f_{\hat{\alpha}}$ be the function returned by LASSO when trained on a training set Z_n and a d-dimensional function space $\mathcal{F}$. If a suitable condition on the features holds, then*

$$||f_{\hat{\alpha}} - f^*||_{2,n}^2 \leq O\Big(\frac{s \log d/\delta}{n}\Big)$$

(*) linear independency, restricted isometry property, compatibility condition, ...

Comparison with Least-squares

Recall:

- d number of features
- s level of sparsity of the target function

Method	*Estimation error*
LS	$O\left(\frac{d \log 1/\delta}{n}\right)$
LASSO	$O\left(\frac{s \log(d/\delta)}{n}\right)$
Oracle LS	$O\left(\frac{s \log 1/\delta}{n}\right)$

Outline

Other (Technical) Applications of SLT

- Neural networks
- Margin–based classification
- Regularized least–squares regression
- Reinforcement Learning
- Density estimation
- Matrix completion
- ...

Other (Practical) Applications of SLT

- Computer vision (Kinetc!)
- Spam filtering
- Computer security
- Natural language processing (Watson!)
- Bioinformatics
- Collaborative filtering (Netflix!)
- Brain–computer interface
- ...

Extensions

- Active Learning
- Unsupervised learning
- Semi-supervised learning
- Fixed design learning
- Transductive learning
- Samples from Markov chains
- Samples from weakly–coupled processes
- Learnability for ergodic processes
- ...

Things to Remember

Things to Remember

- Learning algorithms are *stochastic objects* but their behavior can be *predicted* (in probability)

Things to Remember

- Learning algorithms are *stochastic objects* but their behavior can be *predicted* (in probability)
- Theory helps in designing *better algorithms* and good algorithms forces us to develop *smart theory*

Things to Remember

- Learning algorithms are *stochastic objects* but their behavior can be *predicted* (in probability)
- Theory helps in designing *better algorithms* and good algorithms forces us to develop *smart theory*
- Theoretical bounds help in understand the *critical parameters* and their impact on the performance

Things to Remember

- Learning algorithms are *stochastic objects* but their behavior can be *predicted* (in probability)
- Theory helps in designing *better algorithms* and good algorithms forces us to develop *smart theory*
- Theoretical bounds help in understand the *critical parameters* and their impact on the performance
- Theoretical bounds can help in *tuning the parameters*

Things to Remember

"He who loves practice without theory is like the sailor who boards ship without a rudder and compass and never knows where he may cast."

Leonardo da Vinci

Advanced Topics in Machine Learning

Part I: Elements of Statistical Learning Theory

Alessandro Lazaric
alessandro.lazaric@inria.fr
sequel.lille.inria.fr