Computational Oracle Inequalities for Large Scale Model Selection Problems

Peter Bartlett

University of California at Berkeley

Queensland University of Technology

SLDM, June 2012

Joint work with Alekh Agarwal, John Duchi and Clément Levrard.

Large Scale Data Analysis

Observation:

For many prediction problems, the amount of data available is *effectively unlimited*.

Large Scale Data Analysis

Observation:

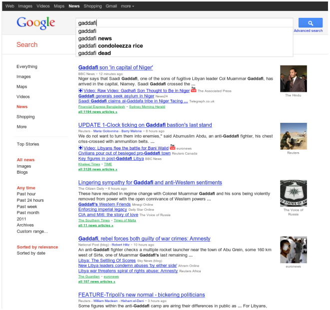
For many prediction problems, the amount of data available is *effectively unlimited*.

Information retrieval: Web search

10^8 websites.

10^{10} pages.

10^9 queries/day.



Large Scale Data Analysis

Observation:

For many prediction problems, the amount of data available is *effectively unlimited*.

Natural language processing:
Spelling correction
Google Linguistics Data Consortium n-gram corpus: 10^{11} sentences.

Large Scale Data Analysis

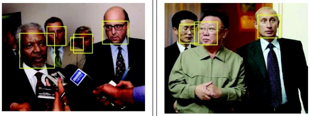
Observation:

For many prediction problems, the amount of data available is *effectively unlimited*.

Computer vision: Captions

Facebook:

10^{11} photos.



United Nations Secretary General **Kofi Annan** stands with U.N. Security Council President and U.S. Ambassador to the U.N. **John D. Negroponte** as Annan . . .

North Korean leader **Kim Jong Il**, and Russian President **Vladimir Putin** walk after talks in Vladivostok, Friday, Aug. 23, 2002. North Korean leader Kim . . .

Large Scale Data Analysis

Observation:

For many prediction problems, the amount of data available is *effectively unlimited*.

- Information retrieval: Web search
- Natural language processing: Spelling correction
- Computer vision: Captions

Large Scale Data Analysis

Observation:

For many prediction problems, performance is limited by *computational resources*, not sample size.

- Information retrieval: Web search
- Natural language processing: Spelling correction
- Computer vision: Captions

Large Scale Data Analysis

Example:

- Peter Norvig, "Internet-Scale Data Analysis":
  On a spelling correction problem, trivial prediction rules, estimated with a massive dataset perform much better than complex prediction rules (which allow only a dataset of modest size).
- Given a limited computational budget,
  what is the best trade-off?
  That is, should we spend our computation on gathering more data, or on estimating richer prediction rules?

Prediction Problem

- i.i.d. $Z_1, Z_2, \ldots, Z_n, Z$ from $\mathcal{Z}$.
- Use data $Z_1, \ldots, Z_n$ to choose $\hat{f}$ from a class F.
- Aim to ensure $\hat{f}$ has small *risk*:

$$L(f) = \mathbb{E}\ell(f, Z),$$

where $\ell : F \times \mathcal{Z}$ is a loss function.

Prediction Problem: Examples

- Aim to ensure $\hat{f}$ has small *risk*: $L(f) = \mathbb{E}\ell(f, Z)$.

Regression

$$\begin{aligned} Z &= (X, Y) \qquad Y \in \mathbb{R}, \\ \ell(f, Z) &= (f(X) - Y)^2 . \end{aligned}$$

Pattern Classification

$$\begin{aligned} Z &= (X, Y) \qquad Y \in \{1, \ldots, m\}, \\ \ell(f, Z) &= 1\,[f(X) \neq Y] . \end{aligned}$$

Density Estimation

$$\ell(f, Z) = -\log f(Z).$$

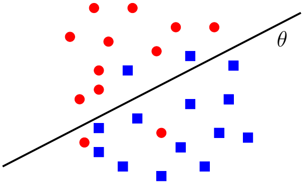
Approximation-Estimation Trade-Off

- Define the *Bayes risk*, $L^* = \inf_f L(f)$, where the infimum is over measurable f.
- We can decompose the excess risk as

$$L(\hat{f}) - L^* = \underbrace{\left(L(\hat{f}) - \inf_{f \in F} L(f) \right)}_{\text{estimation error}} + \underbrace{\left(\inf_{f \in F} L(f) - L^* \right)}_{\text{approximation error}}.$$

- Model selection: automatically choose F to optimize this trade-off.

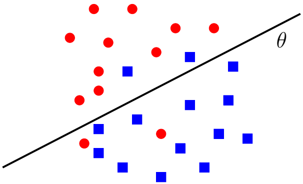
Example 1: Norm of a linear predictor



- Many linear classification algorithms minimize:

$$\min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} \ell\left(y_i, \langle \theta, x_i \rangle\right) \quad \text{subject to} \quad \|\theta\|_2 \leq r.$$

Example 1: Norm of a linear predictor



- Many linear classification algorithms minimize:

$$\min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} \ell\left(y_i, \langle \theta, x_i \rangle\right) \quad \text{subject to} \quad \|\theta\|_2 \leq r.$$

- Statistical and computational complexities depend on the bound r
- Often select from a grid $r_1 \leq r_2 \leq r_3 \leq \ldots$

Example 2: Feature selection from an ordered set

- $\theta \in \mathbb{R}^d$, select subset of $\{1, 2, \ldots, d\}$ where $\theta_i \neq 0$

Example 2: Feature selection from an ordered set

- $\theta \in \mathbb{R}^d$, select subset of $\{1, 2, \ldots, d\}$ where $\theta_i \neq 0$
- Natural ordering amongst feature complexity in many problems
  - Natural language: Unigrams $\prec$ Bigrams $\prec \cdots \prec$ n-grams
  - Function fitting: polynomial degree, Fourier basis dim, ...
  - Computer vision: hierarchy of wavelet filters
- Include features in order of complexity

Example 2: Feature selection from an ordered set

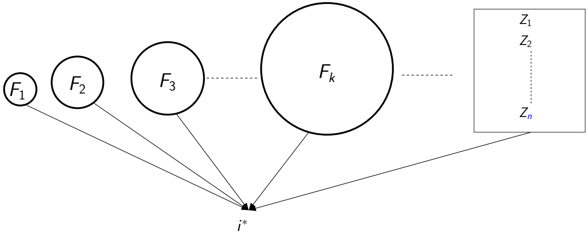
- $\theta \in \mathbb{R}^d$, select subset of $\{1, 2, \ldots, d\}$ where $\theta_i \neq 0$
- Natural ordering amongst feature complexity in many problems
    - Natural language: Unigrams $\prec$ Bigrams $\prec \cdots \prec$ n-grams
    - Function fitting: polynomial degree, Fourier basis dim, ...
    - Computer vision: hierarchy of wavelet filters
- Include features in order of complexity
- Statistical and computational complexities depend on dimensionality
- Want the right number of features: $d_1 \leq d_2 \leq d_3 \leq \ldots$

Model selection over nested hierarchies

- Nested hierarchy of model classes, $F_1 \subseteq F_2 \subseteq F_3 \subseteq \dots$
- Examples:
    - $F_i = \{\theta \in \mathbb{R}^d \ : \ \|\theta\| \le r_i\}$, $r_1 \le r_2 \le r_3 \le \dots$
    - $F_i = \{\theta \in \mathbb{R}^{d_i} \ : \ \|\theta\| \le 1\}$, $d_1 \le d_2 \le d_3 \le \dots$

Model selection over nested hierarchies

- Nested hierarchy of model classes, $F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots$
- Data $Z_1, Z_2, \ldots, Z_n$



Want i^* that optimizes estimation-approximation trade-off

$$L(\hat{f}_i) - L(f^*) = \underbrace{\left(L(\hat{f}_i) - \inf_{f \in F_i} L(f)\right)}_{\text{Estimation error}} + \underbrace{\left(\inf_{f \in F_i} L(f) - L(f^*)\right)}_{\text{Approximation error}}$$

The Model Selection Problem

Given function classes $F_1, F_2, \ldots,$, use the data $Z_1, \ldots, Z_n$ to choose $\hat{f} \in \bigcup_i F_i$ that gives a good trade-off between the approximation error and the estimation error.

Example: *Complexity-penalized model selection.*

$$f_n^i = \arg\min_{f \in F_i} L_n(f),$$

$$\hat{f} = \text{minimizer of } L_n(f_n^i) + \gamma_i(n),$$

where $\gamma_i(n)$ is a *complexity penalty* and L_n is the empirical risk:

$$L_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f, Z_i).$$

A Simple Oracle Inequality

Theorem

Suppose that we have risk bounds for each F_i: w.p. $1-\delta$,

$$\sup_{f \in F_i} |L(f) - L_n(f)| \leq \gamma_i(n) + c\sqrt{\frac{\log 1/\delta}{n}}.$$

If $\hat{f}$ is chosen via complexity regularization:

$$f_n^i = \arg\min_{f \in F_i} L_n(f), \qquad \hat{f} = \text{minimizer of } L_n(f_n^i) + \gamma_i(n),$$

then with probability $1-\delta$,

$$L(\hat{f}) \leq \min_i \left(\inf_{f \in F_i} L(f) + 2\gamma_i(n) + c\sqrt{\frac{\log 1/\delta + \log K}{n}} \right).$$

A Simple Oracle Inequality

- Notice that, for each F_i satisfying

$$\sup_{f \in F_i} |L(f) - L_n(f)| \leq \gamma_i(n) + c\sqrt{\frac{\log 1/\delta}{n}},$$

we have $$L(f_n^i) \leq \inf_{f \in F_i} L(f) + 2\gamma_i(n) + c\sqrt{\frac{\log 1/\delta}{n}}.$$

- But complexity regularization gives $\hat{f}$ satisfying

$$L(\hat{f}) \leq \min_i \left(\inf_{f \in F_i} L(f) + 2\gamma_i(n) + c\sqrt{\frac{\log 1/\delta + \log K}{n}} \right).$$

- Thus, $\hat{f}$ gives a near-optimal trade-off between the approximation error and the (bound on) estimation error, with only a log K penalty.

Computation versus sample size

- Complexity regularization involves computation of the empirical risk minimizer for each F_i:

$$f_n^i = \arg\min_{f \in F_i} L_n(f), \qquad \hat{f} = \text{minimizer of } L_n(f_n^i) + \gamma_i(n),$$

  So computation typically grows *linearly with* K.

- The oracle inequality gives the best trade-off *for a given sample size*:

$$L(\hat{f}) \leq \min_i \left(\inf_{f \in F_i} L(f) + 2\gamma_i(n) + c\sqrt{\frac{\log 1/\delta + \log K}{n}} \right).$$

1. Computation is precious, not sample size
   - Model selection
   - Oracle inequalities
2. Computational oracle inequalities for nested hierarchies
   - Problem formulation
   - Algorithm
   - Oracle Inequality
3. Fast rates
   - Complexity regularization
   - Algorithms
   - Computational Oracle Inequalities
4. Removing the nesting assumption
   - Algorithm
   - Oracle Inequality
5. Summary and open problems

Scaling of penalties with computation

Recall

$\gamma_i(n)$ is the complexity penalty for the class F_i with sample size n.

Scaling of penalties with computation

Recall

$\gamma_i(n)$ is the complexity penalty for the class F_i with sample size n.

Define

$p_i(T)$ as the complexity penalty for the class F_i with computational budget T.

$$\text{computation } T \implies \text{sample size } n_i(T) \text{ for } F_i$$

We set $p_i(T) = \gamma_i(n_i(T))$.

Scaling of penalties with computation

Define

$p_i(T)$ as the complexity penalty for the class F_i with computational budget T.

In more detail:
with computation T, we can ensure that, with high probability,

$$\sup_{f \in F_i} \left| L(f) - L_{n_i(T)}(f) \right| \leq \gamma_i(n_i(T)),$$

hence

$$L(f^i_{n_i(T)}) \leq \inf_{f \in F_i} L(f) + O(p_i(T)).$$

Scaling of penalties with computation

Define

$p_i(T)$ as the complexity penalty for the class F_i with computational budget T.

Our goal: A computational oracle inequality:
$\hat{f}$ compares favorably with each model, estimated using the entire computational budget.

$$L(\hat{f}) \leq \min_i \left(\underbrace{\inf_{f \in F_i} L(f) + O(p_i(T))}_{\text{c.f. estimate } f \text{ using the entire budget}} \right).$$

Scaling of penalties with computation

Define

$p_i(T)$ as the complexity penalty for the class F_i with computational budget T.

Our goal: A computational oracle inequality:
$\hat{f}$ compares favorably with each model, estimated using the entire computational budget.

$$L(\hat{f}) \leq \min_i \left(\underbrace{\inf_{f \in F_i} L(f) + O\left(p_i\left(\frac{T}{\log T}\right)\right)}_{\text{c.f. estimate } f \text{ using almost the entire budget}} \right).$$

Naïve solution: grid search

- Allocate budget T/K to each model.
- Use a sample of size $n_i(T/K)$ for F_i.
- Choose

$$
\begin{aligned}
f^i_{n_i} &= \arg\min_{f \in F_i} L_{n_i}(f), \\
\hat{f} &= \text{minimizer of } L_{n_i}(f^i_{n_i}) + \gamma_i\,(n_i)\,.
\end{aligned}
$$

- Satisfies oracle inequality

$$
L(\hat{f}) \leq \min_i \left(\inf_{f \in F_i} L(f) + p_i\left(\frac{T}{K}\right)\right).
$$

Model selection from nested classes

- Suppose that the models are ordered by inclusion:

$$F_1 \subseteq F_2 \subseteq \cdots \subseteq F_K.$$

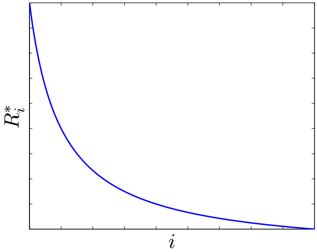
- Examples:
  - $F_i = \left\{ f_\theta : \theta \in \mathbb{R}^d, \ \|\theta\| \leq r_i \right\}, r_1 \leq r_2 \leq \cdots \leq r_K.$
  - $F_i = \left\{ f_\theta : \theta \in \mathbb{R}^{d_i}, \ \|\theta\| \leq 1 \right\}, d_1 \leq d_2 \leq \cdots \leq d_K.$
- Suppose that we have risk bounds for each F_i: w.p. $1 - \delta$,

$$\sup_{f \in F_i} |L(f) - L_n(f)| \leq \gamma_i(n) + c\sqrt{\frac{\log 1/\delta}{n}}.$$

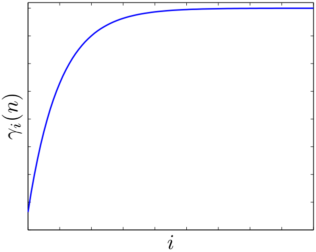
Exploiting structure of nested classes

Want to exploit monotonicity of risks and penalties

Excess risk, $R_i^* = \inf_{f \in F_i} L(f) - L^*$:

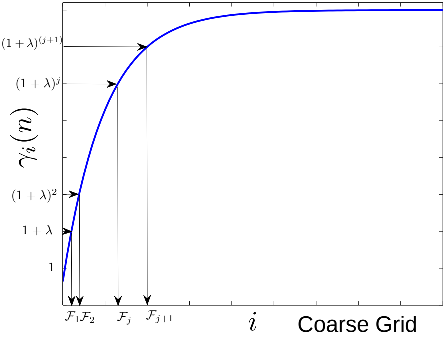


Penalty, $\gamma_i(n)$:



Coarse grid sets

- Want to spend computation on only few classes.
- Use monotonicity to interpolate for the rest.
- Partition based on penalty values.



Coarse grids for model selection

Assume

1. Loss is bounded:
$$\ell(f, Z) \in [0, B].$$
2. Computation grows at least linearly with sample size:
$$n_1(T) = O(T).$$
3. Penalty decreases no faster than $1/n$:
$$\gamma_1(n) = \Omega\left(\frac{1}{n}\right).$$

Coarse grids for model selection

Then

- We can ignore F_i with $\gamma_i(n_i(T)) > B$.
- We can cover all smaller classes with a coarse grid of size $s = O(\log(BT))$.

Definition (Coarse grid)

For $S \subseteq \mathbb{N}$, a set $\hat{S} \subseteq S$ is a coarse grid of size s for S if $|\hat{S}| = s$ and for each $i \in S$ there is an index $j \in \hat{S}$ such that

$$\gamma_i\left(n_i\left(\frac{T}{s}\right)\right) \le \gamma_j\left(n_i\left(\frac{T}{s}\right)\right) \le 2\gamma_i\left(n_i\left(\frac{T}{s}\right)\right).$$

Coarse grids for model selection

Then

- We can ignore F_i with $\gamma_i(n_i(T)) > B$.
- We can cover all smaller classes with a coarse grid of size $s = O(\log(BT))$.

- Include a new class only after penalty increases sufficiently.
- $s = \log\left(\frac{B}{\gamma_1(n_1(T))}\right) = O\left(\log BT\right)$ suffices.

Complexity regularization on a coarse grid

Given a coarse grid $\hat{S}$ with cardinality s:

1. Allocate budget T/s to each class in S.
2. Choose

$$f^i = \arg\min_{f \in F_i} L_{n_i(T/s)}(f)$$

$$\hat{f} = \arg\min_{f \in \{f^j : j \in \hat{S}\}} L_{n_j(T/s)}(f) + \gamma_j\left(n_j\left(\frac{T}{s}\right)\right).$$

Complexity regularization on a coarse grid

Theorem

For a nested hierarchy satisfying the uniform convergence bounds, with high probability,

$$L(\hat{f}) \le \min_i \left\{ \inf_{f \in F_i} L(f) + O\left(\gamma_i\left(n_i\left(\frac{T}{s}\right)\right)\right)\right\}$$
$$\le \min_i \left\{ \inf_{f \in F_i} L(f) + O\left(p_i\left(\frac{T}{\log T}\right)\right)\right\}$$

- *Computational cost of model selection* scales logarithmically with T.

Fast Rates

Results so far rely on uniform convergence: bounds on

$$\sup_{f \in F_i} |L(f) - L_n(f)| .$$

Typical fluctuations are of the order

$$|L(f) - L_n(f)| = O\left(\frac{1}{\sqrt{n}}\right) .$$

In some cases, these rates cannot be improved, and additive penalties that scale as

$$\sup_{f \in F_i} |L(f) - L_n(f)| = \Omega\left(\frac{1}{\sqrt{n}}\right)$$

give optimal oracle inequalities.

Fast Rates

However, in many cases, we can obtain faster rates.
e.g., with high probability, for all $f \in F$,

$$L(f) - L(f^*) \leq 2\left(L_n(f) - L_n(f^*)\right) + O\left(\frac{\log n}{n}\right),$$

where $L(f^*) = \min_{f\in F} L(f)$. In these cases, choosing

$$\hat{f} = \arg\min_{f\in F} L_n(f)$$

gives $L(f) \leq L(f^*) + O(\log n/n)$.
Examples: Convex losses [Lee, B., Williamson, 1998; B., Jordan, McAuliffe, 2006], classification with low noise [Mammen and Tsybakov, 2004; Tsybakov, 2004].

Oracle Inequalities with Fast Rates for Complexity Regularization

It turns out that we can use complexity regularization to exploit these faster rates, provided the F_i are ordered by inclusion.

Theorem (B., 2008)

For $F_1 \subseteq F_2 \subseteq \cdots$ *and* $\gamma_1(n) \leq \gamma_2(n) \leq \cdots$, *if*

$$\sup_i \sup_{f \in F_i} \left(L(f) - L(f_i^*) - 2\left(L_n(f) - L_n(f_i^*)\right) - \gamma_i(n)\right) \leq 0,$$

$$\sup_i \sup_{f \in F_i} \left(L_n(f) - L_n(f_i^*) - 2\left(L(f) - L(f_i^*)\right) - \gamma_i(n)\right) \leq 0,$$

$$\text{then } L(\hat{f}) \leq \inf_i \left(L(f_i^*) + 9\gamma_i(n)\right),$$

where $\hat{f}$ *minimizes* $L_n(f_n^i) + 7\gamma_i(n)/2$ *and* $f_i^* = \arg\min_{f \in F_i} L(f)$.

Oracle Inequalities with Fast Rates for Complexity Regularization

This is *striking:*

- $L_n(f_n^i)$ fluctuates on a scale $1/\sqrt{n}$.
- But adding a tiny penalty $\gamma_i(n) = O(\log n/n)$ gives $L(\hat{f})$ within $O(\log n/n)$ of the best!

The explanation: the fluctuations for different F_i are correlated, because the empirical minimizers are chosen using the *same data*.

Computational Oracle Inequalities?

Can we obtain computational oracle inequalities with these rates?

Computational Oracle Inequalities?

Can we obtain computational oracle inequalities with these rates?

Previous Algorithm

Given a coarse grid $\hat{S}$ with cardinality s:

1. Allocate budget T/s to each class in S.
2. Choose

$$f^i = \arg\min_{f \in F_i} L_{n_i(T/s)}(f)$$

$$\hat{f} = \arg \min_{f \in \{f^j : j \in \hat{S}\}} L_{n_j(T/s)}(f) + \gamma_j \left(n_j \left(\frac{T}{s} \right) \right).$$

Computational Oracle Inequalities?

Previous Algorithm

Given a coarse grid $\hat{S}$ with cardinality s:

1. Allocate budget T/s to each class in S.
2. Choose

$$f^i = \arg\min_{f \in F_i} L_{n_i(T/s)}(f)$$

$$\hat{f} = \arg\min_{f \in \{f^j : j \in \hat{S}\}} L_{n_j(T/s)}(f) + \gamma_j\left(n_j\left(\frac{T}{s}\right)\right).$$

Obstacle: The oracle inequality relies on the use of the *same data*. But to best use our computational budget, we should gather *more* data for simpler classes.

Algorithm for Fast Rates

Given a coarse grid $\hat{S}$ with cardinality s:

1. Allocate budget T/s to each class in S.
2. Choose

$$f^i = \arg\min_{f \in F_i} L_{n_i(T/s^2)}(f)$$

3. Define $\hat{f}$ as the f^i with the largest index i such that for all smaller j,

$$L_{n_i}(f^i) + \gamma_i(n_i) \leq \inf_{f \in F_j} L_{n_i}(f) + \gamma_j(n_i).$$

The *same data* is used in comparing f^i with functions from smaller classes.

Computational Oracle Inequalities

Theorem

For a nested hierarchy exhibiting fast rates, with high probability,

$$L(\hat{f}) \leq \min_i \left\{ \inf_{f \in F_i} L(f) + O\left(p_i\left(\frac{T}{\log^2 T}\right)\right)\right\}.$$

Heterogeneous Models

In general, the F_i can be heterogeneous, not ordered by inclusion.

- Different kernels.
- Graphs in directed graphical models.
- Subsets of features.

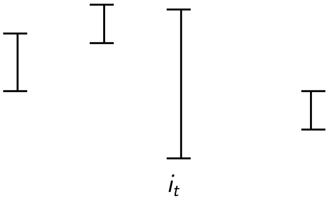
Key idea: Successively allocate computational quanta online.

Multi-Armed Bandits for Model Selection

- Want class i that minimizes

$$\inf_{f \in F_i} L(f) + \gamma_i(n_i(T)).$$

- Use idea of *optimism in the face of uncertainty*:
  neatly trade off exploration and exploitation by choosing the class with the smallest lower bound on the criterion.



Multi-Armed Bandits for Model Selection

- Want class i that minimizes

$$\inf_{f \in F_i} L(f) + \gamma_i(n_i(T)).$$

- We know it suffices to choose a class i to minimize

$$L_{Tn_i}(f^i_{Tn_i}) + \gamma_i(n_i(T)).$$

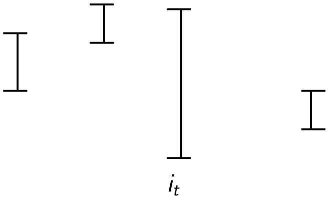
- Use the lower confidence bound:

$$L_n(f^i_n) - \gamma_i(n) - \sqrt{\frac{\log K}{n}} + \gamma_i(n_i(T)),$$

where n is the size of the sample that we have allocated already to class i.

Multi-Armed Bandits for Model Selection

- Assume $n_i(T)$ is linear in T: $n_i(T) = Tn_i$.
- Algorithm picks class i_t with *smallest* lower confidence bound.
- Allocate additional sample of size n_{i_t} to class i_t.
- Regret analysis of upper-confidence-bound algorithm (Auer et al., 2002) extends to give oracle inequalities.



Oracle inequality under separation assumption

$$\text{Define } i^* = \arg\min_i \left(\inf_{f \in F_i} L(f) + \gamma_i(Tn_i) \right),$$

$$\Delta_i = \inf_{f \in F_i} L(f) + \gamma_i(Tn_i) - \left(\inf_{f \in F_{i^*}} L(f) + \gamma_{i^*}(Tn_{i^*}) \right).$$

Assume $\gamma_i(n) = \dfrac{c_i}{\sqrt{n}}$.

Theorem

Let $T_i(T)$ be the number of times class i is queried. There are constants C, κ_1, κ_2 such that with probability at least $1 - \frac{\kappa_1}{TK^4}$,

$$T_i(T) \leq \frac{C}{n_i} \left(\frac{c_i + \kappa_2 \sqrt{\log T}}{\Delta_i} \right)^2.$$

Oracle inequality under separation assumption

$$\text{Define } i^* = \arg\min_i \left(\inf_{f \in F_i} L(f) + \gamma_i(Tn_i) \right),$$
$$\Delta_i = \inf_{f \in F_i} L(f) + \gamma_i(Tn_i) - \inf_{f \in F_{i*}} L(f) + \gamma_{i^*}(Tn_{i^*}).$$

- If we can *incrementally update* the choice f_n^i, then the fraction of budget that is assigned to a suboptimal class i is no more than $\log T/(n_i T \Delta_i^2)$.
- This is essentially optimal (Lai and Robbins, 1985).

Oracle inequality without separation

- Assume that functions in $\{F_i\}$ map to a vector space, and the loss $\ell(\cdot, z)$ is convex.
- Define $\hat{f} = \frac{1}{T}\sum_{t=1}^{T} f_t$, where algorithm produces $f_t \in F_{i_t}$ at time t.

Theorem

There is a constant κ such that with probability at least $1 - \frac{2\kappa}{TK^3}$

$$L(\hat{f}) = \inf_{i \in \{1,\ldots,K\}} \left(\inf_{f \in F_i} L(f) + \gamma_i(Tn_i) \right) + O\left(\sqrt{\frac{K \max\{\log T, \log K\}}{T}} \right).$$

- Linear dependence on K.

Open problems

- For nested hierarchies, the analysis relied on a coarse multiplicative cover of the penalty values. If the penalties are data-dependent, when is this approach possible?
- What other structures on function classes lead to good computational oracle inequalities?

Summary

- For large-scale problems, data is cheap but computation is precious.
- Computational oracle inequalities for model selection: select a near-optimal model without wasting much computation on other models.
- A *nested* complexity hierarchy ensures cost logarithmic in computational budget.
- Faster rates are sometimes possible: More complicated complexity regularization schemes ensure cost polylogarithmic in computational budget.
- If not nested, cost of model selection is linear in size of hierarchy.