Introduction

1. Introduction

2. Supervised Learning

3. Models

4. Big Data and Numerical Issues

5. Conclusion
Introduction

Data is the new Oil!

Couched in 2005 by Clay Munday, a British data commercialization entrepreneur, its now famous phrase was emphasized by the World Economic Forum in a 2011 report, which considered data to be an economic asset, like oil.

“DATA IS THE NEW OIL.”

From the beginning of recorded time until 2000, we created 5 exabytes of data. In 2011 the same amount was created every two days. By 2013, it’s expected that the time will shorten to 10 minutes.

Every hour, we create as much data as could fill 7 billion DVDs. If the world’s oceans were the world’s brain, then 5.8 days would be enough to fill it.

There are 133 million blogs on the web.

As of August 2012, there were just over 4 million articles in the English Wikipedia.

60% of all homes in the world have a mobile phone.

Out of 5 billion residents, 1 billion are smartphone owners. In Singapore, 80% of citizens are smartphone owners.

80% of all humans get a billion pieces of news text. Out of 5 billion residents, 1 billion are smartphone owners.

English is the dominant language of the web. But by 2014 it will be Chinese, if the current rate of Chinese confidence.

The languages used on the web (May 2011):

Hindi
English
Russian
French
Arabic
German
Portuguese
Spanish
Japanese
Chinese
English

247 billion emails are sent every year (or 67 billion are spam).

10% of all photos ever taken were taken in 2011.

With new fiber-optic cable, the round trip time between New York and London will be 5 milliseconds.

This 5 milliseconds saving is worth millions of dollars to the trading systems on the island (and who will pay millions to do so).

60% of all humans get a billion pieces of news text. Out of 5 billion residents, 1 billion are smartphone owners.

10% of all photos ever taken were taken in 2011.

How they save 5 milliseconds

The depth of the Atlantic Ocean average.

The new cable will be on areas of the ocean floor that are up to 6,500 feet shallower than the current bottom cable. By using this area, the new bottom cable can be as much as 40 miles shorter, meaning that the time it takes to send data along it is shortened.

The new cable takes a shalower, faster oceanFloor.

50% of 3-year-old kids in the U.S. are given access to a smartphone.

High-frequency traders, with the help of computer algorithms, are big data followers to follow trends and to act quickly on their findings.

5 milliseconds from the current 60 milliseconds it takes for traders to determine to travel between New York City and London.

These specialized algorithms make split-second decisions to buy or sell a commodity. New cable being laid under the Atlantic will shave 5 milliseconds off the current travel time.
Major Influences

Four major influences act today:

- The formal theories of statistics
- Accelerating developments in computers and display devices
- The challenge, in many fields, of more and ever larger bodies of data
- The emphasis on quantification in an ever wider variety of disciplines
Major Influences - Tukey (1962)

Four major influences act today:

- The formal theories of statistics
- Accelerating developments in computers and display devices
- The challenge, in many fields, of more and ever larger bodies of data
- The emphasis on quantification in an ever wider variety of disciplines

- He was talking of Data Analysis.
- Data mining, Machine learning, Big Data...
A new Context

Data everywhere
- Huge volume,
- Huge variety...

Affordable computation units
- Cloud computing
- Graphical Processor Units (GPU)...

- Growing academic and industrial interest!
Introduction

Figure 2-2. The data science process

- Doing Data Science: Straight talk from the frontline.
  - Rachel Schutt, Cathy O’Neil
  - O’Reilly
Big Data, Data Science and Machine Learning

- **Big Data**: buzzword to raise money (or data sets too large or too complex to be handled by the current system).
- **Data Science**: art (or science) of the generalizable extraction of knowledge from data.
- **Machine Learning**: construction and study of algorithms that can learn from and make predictions on data.

- Exciting challenges in the industrial and the academic worlds.

Machine Learning

- **Fundamental** ingredient in data science.
- **Probability** and **Optimization** play a central role.
- Model **Competition/Collaboration**
- New **computational constraints** in Big Data setting
Outline

1. Introduction
2. Supervised Learning
3. Models
4. Big Data and Numerical Issues
5. Conclusion
Outline

1. Introduction
2. Supervised Learning
3. Models
4. Big Data and Numerical Issues
5. Conclusion
A definition by Tom Mitchell
(http://www.cs.cmu.edu/~tom/)

A computer program is said to learn from experience \( E \) with respect to some class of tasks \( T \) and performance measure \( P \), if its performance at tasks in \( T \), as measured by \( P \), improves with experience \( E \).
Supervised Learning

Experience, Task and Performance measure

- **Training data**: \( D = \{(X_1, Y_1), \ldots, (X_n, Y_n)\} \) (i.i.d. \( \sim P \))
- **Predictor**: \( f : \mathcal{X} \rightarrow \mathcal{Y} \) measurable
- **Cost/Loss function**: \( \ell(f(X), Y) \) measure how well \( f(X) \) “predicts” \( Y \)
- **Risk**:
  \[
  R(f) = \mathbb{E}[\ell(Y, f(X))] = \mathbb{E}_X \left[ \mathbb{E}_{Y|X} [\ell(Y, f(X))] \right]
  \]

- Often \( \ell(f(X), Y) = 1_{Y \neq f(X)} \) or \( \ell(f(X), Y) = |f(X) - Y|^2 \)

Goal

- Learn a rule to construct a classifier \( \hat{f} \in \mathcal{F} \) from the training data \( \mathcal{D}_n \) s.t. the risk \( R(\hat{f}) \) is small on average or with high probability with respect to \( \mathcal{D}_n \).
The best solution $f^*$ (which is independent of $D_n$) is

$$f^* = \arg \min_{f \in \mathcal{F}} R(f) = \arg \min_{f \in \mathcal{F}} \mathbb{E} [\ell(Y, f(X))] = \arg \min_{f \in \mathcal{F}} \mathbb{E}_X \left[ \mathbb{E}_{Y|x} [\ell(Y, f(x))] \right]$$

**Bayes Classifier (explicit solution)**

- In binary classification with $0-1$ loss:
  $$f^*(X) = \begin{cases} +1 & \text{if } \mathbb{P} \{ Y = +1 | X \} \geq \mathbb{P} \{ Y = -1 | X \} \\ -1 & \text{otherwise} \end{cases}$$
  $$\iff \mathbb{P} \{ Y = +1 | X \} \geq 1/2$$

- In regression with the quadratic loss
  $$f^*(X) = \mathbb{E} [Y|X]$$

**Issue:** Explicit solution requires to **know** $\mathbb{E} [Y|X]$ for all values of $X$!
Machine Learning

- Learn a rule to construct a classifier \( \hat{f} \in \mathcal{F} \) from the training data \( \mathcal{D}_n \) s.t. the risk \( \mathcal{R}(\hat{f}) \) is small on average or with high probability with respect to \( \mathcal{D}_n \).

Canonical example: Empirical Risk Minimizer

- Restrict \( f \) to a subset of functions \( \mathcal{S} = \{f_\theta, \theta \in \Theta\} \)
- Replace the minimization of the average loss by the minimization of the empirical loss

\[
\hat{f} = \hat{f}_\theta = \arg\min_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(X_i))
\]

Examples:

- Linear regression
- Linear discrimination with

\[
\mathcal{S} = \{ \mathbf{x} \mapsto \text{sign}\{\beta^T \mathbf{x} + \beta_0\} / \beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R} \} \]
Probabilistic vs Optimization?

How to find a good function $f$ with a small risk

$$R(f) = \mathbb{E} [\ell(Y, f(X))]$$

Canonical approach: $\hat{f}_S = \arg\min_{f \in S} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(X_i))$

Problems

- How to choose $S$?
- How to compute the minimization?

A Probabilistic Point of View

Solution: For $X$, estimate $Y|X$ plug this estimate in the Bayes classifier:

(Generalized) Linear Models, Kernel methods, $k$-nn, Naive Bayes, Tree, Bagging...

An Optimization Point of View

Solution: If necessary replace the loss $\ell$ by an upper bound $\ell'$ and minimize the empirical loss: SVR, SVM, Neural Network, Tree, Boosting
Probabilistic Approach

If $Y|X$ is known, one can compute the best solution $f^*$

$$\arg \min_{f \in \mathcal{F}} \mathbb{E}_X \left[ \mathbb{E}_{Y|X} [\ell(Y, f(x))] \right]$$

Bayes Plugin

- **Learning**: Estimation of $Y|x$ and plugging of this estimate in the Bayes classifier
- **Plugin**: a classifier $\hat{f} : \mathcal{X} \rightarrow \mathcal{Y}$
  - $\ell_{0/1}$ loss:
    $$\hat{f}(x) = \begin{cases} 
    +1 & \text{if } \hat{p}_+ (x) \geq \hat{p}_- (x) \\
    -1 & \text{otherwise}
    \end{cases}$$
  - Quadratic loss:
    $$\hat{f}(x) = \mathbb{E} [Y|x]$$

**Instantiations**:
- Generative Modeling and Bayesian Methods
- Parametric Conditional Models
- Kernel Conditional Density Methods

Importance of a corresponding efficient *numerical scheme*!
Optimization Approach

- The best solution $f^*$ is the one minimizing
  \[ f^* = \arg\min R(f) = \arg\min \mathbb{E} [\ell(Y, f(X))] \]

### Empirical Risk Minimization

- Restrict $f$ to a subset of functions $S = \{f_\theta, \theta \in \Theta\}$
- Replace the minimization of the average loss by the minimization of the empirical loss
  \[ \hat{f} = f_\hat{\theta} = \arg\min_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \]

- **Issue**: Minimization may be impossible in practice.
- **Solution**: Replace $\ell$ by $\ell'$ a simpler (convex) majorant and minimize this upper-bound.
- **Instantiation**: Regression, SVM, Neural Networks...
- Importance of a corresponding efficient **numerical scheme**!
Classification Loss and Convexification

- Classification loss: \( \ell^{0/1}(y, f(x)) = 1_{y \neq f(x)} \)
- Not convex and not smooth!

Classical convexification

- Logistic loss: \( \ell'(y, f(x)) = \log(1 + e^{-yf(x)}) \) (Logistic / NN)
- Hinge loss: \( \ell'(y, f(x)) = (1 - yf(x))_+ \) (SVM)
- Exponential loss: \( \ell'(y, f(x)) = e^{-yf(x)} \) (Boosting...)

- very efficient **numerical scheme**!
# Probabilistic vs Optimization

<table>
<thead>
<tr>
<th>Probabilistic Approach</th>
<th>Optimization Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Principle:</strong> estimate the conditional law $Y</td>
<td>X$ and use it to take an <strong>informed</strong> decision.</td>
</tr>
<tr>
<td><strong>Motto:</strong> If you know the world, everything is easy!</td>
<td><strong>Motto:</strong> You should focus on your goal!</td>
</tr>
<tr>
<td><strong>Emphasis on Interpretation</strong></td>
<td><strong>Emphasis on Prediction</strong></td>
</tr>
<tr>
<td><strong>Pro:</strong></td>
<td><strong>Pro:</strong></td>
</tr>
<tr>
<td>- Interpretable models.</td>
<td>- Focus on the true goal!</td>
</tr>
<tr>
<td>- Lots of flexibility in the generative model.</td>
<td>- Can use very clever optimization algorithm.</td>
</tr>
<tr>
<td>- Simultaneous decision optimization.</td>
<td>- No need to obtain the best solution.</td>
</tr>
<tr>
<td><strong>Cons:</strong></td>
<td><strong>Cons:</strong></td>
</tr>
<tr>
<td>- Computational issue.</td>
<td>- Black box model.</td>
</tr>
<tr>
<td>- No need to know the law to take a decision.</td>
<td>- Not robust to a change of decision zone.</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
2. Supervised Learning
3. Models
4. Big Data and Numerical Issues
5. Conclusion
**Models**

Bias-Variance Dilemna

- General setting:
  - $\mathcal{F} = \{\text{measurable functions } \mathcal{X} \rightarrow \mathcal{Y}\}$
  - Best solution: $f^* = \arg\min_{f \in \mathcal{F}} \mathcal{R}(f)$
  - Class $S \subset \mathcal{F}$ of functions
  - Ideal target in $S$: $f^*_S = \arg\min_{f \in S} \mathcal{R}(f)$
  - Estimate in $S$: $\hat{f}_S$ obtained with a numerical algorithm

**Approximation error and estimation error (Bias/Variance)**

$$\mathcal{R}(\hat{f}_S) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f^*_S) - \mathcal{R}(f^*)}_{\text{Approximation error}} + \underbrace{\mathcal{R}(\hat{f}_S) - \mathcal{R}(f^*_S)}_{\text{Estimation error}}$$

- Different behavior for different model complexity
  - **Low complexity model** are easily learned but the approximation error (“bias”) may be large (**Under-fit**).
  - **High complexity model** may contains a good ideal target but the estimation error (“variance”) can be large (**Over-fit**).
**General Methodology**

- **Modeling**: Chose $S = \{ f_\theta, \theta \in \Theta \}$
- **Methodology**: Minimize over $\theta \in \Theta$

\[
\frac{1}{n} \sum_{i=1}^{n} \ell'(y_i, f_\theta(x_i)) + \lambda \text{comp}(\theta)
\]

- Lots of freedom!
- Example of parametrization:
  - Linear: $f_\theta(x) = \langle \theta, x \rangle$ or $f_\theta(x) = \text{sign}(\langle \theta, x \rangle)$
  - (Deep) Neural Network: much more complex parametrization.
- Restriction on $\Theta$:
  - $\|\theta\|_p \leq C$
  - More complex restriction: $\text{comp}(\theta) \leq C$

- Methodology:
  - Choice of the loss function $\ell'$ (Likelihood / Convex surrogate)
  - Choice of the minimization algorithm...
Models and Optimization

General Penalized Methodology

- **Modeling**: Chose $S = \{ f_\theta, \theta \in \Theta \}$
- **Methodology**: Minimize over $\theta \in \Theta$
  $$\frac{1}{n} \sum_{i=1}^{n} \ell'(y_i, f_\theta(x_i)) + \lambda \text{comp}(\theta)$$

- Lots of freedom!
- Example of parametrization:
  - Linear: $f_\theta(x) = \langle \theta, x \rangle$ or $f_\theta(x) = \text{sign}(\langle \theta, x \rangle)$
  - (Deep) Neural Network: much more complex parametrization.
- Restriction on $\Theta$:
  - $\|\theta\|_p \leq C$
  - More complex restriction: $\text{comp}(\theta) \leq C$
- Penalization: Lagrangian reformulation

- Methodology:
  - Choice of the loss function $\ell'$ (Likelihood / Convex surrogate)
  - Choice of the minimization algorithm…
Models

Competition Between Several Models

- Empirical error biased toward complex models!
- How to select the **best one**?

**Error estimation**

- **Cross validation**: Very efficient (and almost always used in practice!) but slightly biased as it target uses only a fraction of the data.
- **Penalization approach**: use empirical loss criterion but penalize it by a term increasing with the complexity of $S$
  
  $$R_n(\hat{f}_S) \rightarrow R_n(\hat{f}_S) + \text{pen}(S)$$

- Penalization calibration issue...
- Simultaneous CV control issue...
Practical Selection Methodology

- Choose a penalty/complexity shape $\widehat{\text{pen}}(\theta)$.
- Compute the CV error for the minimizer with a penalty $\lambda \widehat{\text{pen}}(\theta)$ for all $\lambda \in \Lambda$.
- Determine $\hat{\lambda}$ the $\lambda$ minimizing the CV error.
- Compute the minimizer with the penalty $\hat{\lambda} \widehat{\text{pen}}(\theta)$.

- Requires a lot of minimizations! Hence optimization is the bottleneck!

Why not using only CV?

- If the penalized likelihood minimization is easy, much cheaper to compute the CV error for all $\lambda \in \Lambda$ than for all possible estimators...
- CV performs best when the set of candidates is not too big (or is structured...)
# Models: Selection or Combination

## Selection of a Single Model
- Most classical scheme.
- Preserve interpretability of each model.
- Strong theoretical framework!

## Mixture
- Combine (randomized) models built in parallel:
  - (Weighted) model averaging,
  - Exponential Weighted Aggregation,
  - Super Learner,
  - Bayesian averaging.
- Less theoretical analysis.

## Sequential Combination
- Boosting / Greedy Gradient Descent Algorithm
- Very efficient in practice / Few convincing analyses...
Outline

1 Introduction

2 Supervised Learning

3 Models

4 Big Data and Numerical Issues

5 Conclusion
Big Data?

Data access / storage (Locality of Reference).
Multiple core architecture (Parallelization).
Cluster (Distribution)
Locality of Reference

- Data should be as close as possible from the core:
  - **Speed/Price Hierarchy**: Cache > Memory > Disk > Network
  - **Size hierarchy**: Cache < Memory < Disk < Network.

- **In memory**:
  - **Ideal case**: dataset fits in the memory of a single computer.
  - **Useless** if data used only once... (bottleneck = disk)

- **Memory usage**:
  - **Split and Apply**: piecewise computation...
  - Memory growth **faster** than data growth (Death of big data?)
  - Memory req. may be (much) **larger** than data ($O(n^\alpha)$ algo.)
Parallelization

Modern CPU: no more speed increases but more cores.

Parallelization:
- HPC / DS setting: CPU bound tasks / IO bound tasks.
- Data science: Often embarrassingly parallel setting (no interaction between tasks).

Not always acceleration due to IO limitation!
True Big Data Setting?

- Computation in a cluster:
  - Distribution of the data (DS),
  - or/and distribution of the computation (HPC)
- Hadoop/Spark realm.
- Locally parallel in memory computation are faster... if data used more than once.
- Real challenge when not (almost) embarrassingly parallel (interaction, graph...)
Outline

1. Introduction
2. Supervised Learning
3. Models
4. Big Data and Numerical Issues
5. Conclusion
• **Probabilistic vs Optimization** approaches:
  • Related but different,
  • Interpretation vs Prediction,
  • Complementary approaches...

• **Models**: from selection to combination in prediction.

• **Data Science vs Big Data**:
  • Hardware constraints!
  • Lots of algorithmic challenges but few conceptual ones.

• **Next project** (with E. Moulines & E. Scornet, CMAP):
  • Exponentially Weighted Aggregation (L. Montuelle) vs Bayesian averaging.
  • Application to modified random forests.
  • Avoid arbitrary bootstrap and random feature subset sampling.
  • High dimensional MCMC scheme.
Probabilistic vs Optimization approaches:
- Related but different,
- Interpretation vs Prediction,
- Complementary approaches...

Models: from selection to combination in prediction.

Data Science vs Big Data:
- Hardware constraints!
- Lots of algorithmic challenges but few conceptual ones.

Next project (with E. Moulines & E. Scornet, CMAP):
- Exponentially Weighted Aggregation (L. Montuelle) vs Bayesian averaging.
- Application to modified random forests.
- Avoid arbitrary bootstrap and random feature subset sampling.
- High dimensional MCMC scheme.

More deep science in 2023?