Probabilistic and Optimization Approaches in Classification

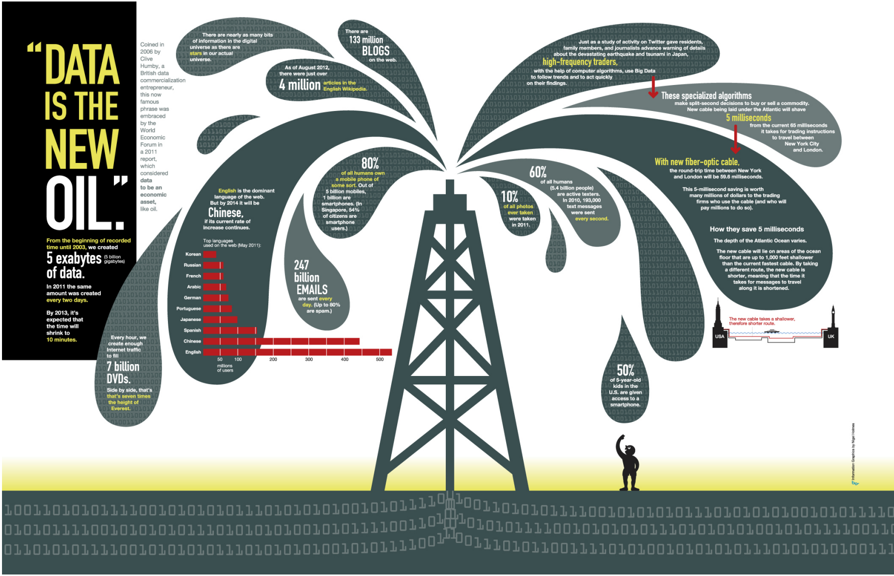
E. Le Pennec - CMAP, École polytechnique

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

JSTAR - AgroCampus, Rennes - 20/10/2016

Outline

Data is the new Oil!



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!

Doing Data Science

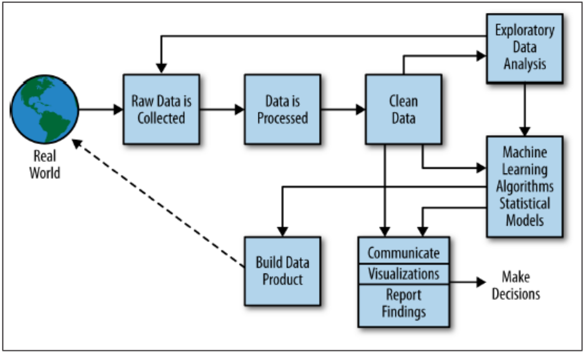


Figure 2-2. The data science process

- 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.

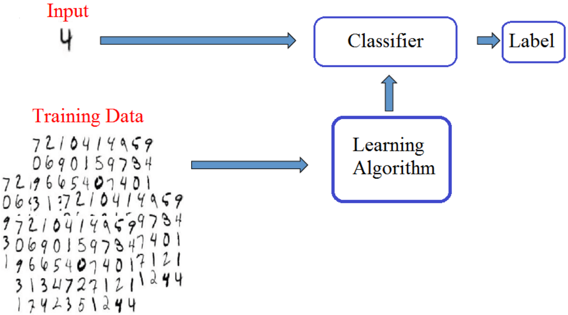
Machine Learning

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

Outline

Outline

Supervised Learning



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** : $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- **Predictor**: $f : \mathcal{X} \to \mathcal{Y}$ measurable
- **Cost/Loss function** : $\ell(f(\mathbf{X}), Y)$ measure how well $f(\mathbf{X})$ "predicts" Y
- **Risk**:

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{X}))\right]\right]$$

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

Goal

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

Best Solution

- The best solution f^* (which is independent of $\mathcal{D}_n$) is

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

Bayes Classifier (explicit solution)

- In binary classification with $0-1$ loss:

$$f^*(\mathbf{X}) = \begin{cases} +1 & \text{if} \quad \mathbb{P}\left\{Y=+1|\mathbf{X}\right\} \geq \mathbb{P}\left\{Y=-1|\mathbf{X}\right\} \\ & \quad \Leftrightarrow \mathbb{P}\left\{Y=+1|\mathbf{X}\right\} \geq 1/2 \\ -1 & \text{otherwise} \end{cases}$$

- In regression with the quadratic loss

$$f^*(\mathbf{X}) = \mathbb{E}\left[Y|\mathbf{X}\right]$$

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

Goal

Machine Learning

- Learn a rule to construct a **classifier** $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. **the risk** $\mathcal{R}(\widehat{f})$ 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

$$\widehat{f} = f_{\widehat{\theta}} = \operatorname*{argmin}_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(\mathbf{X}_i))$$

- Examples:
    - Linear regression
    - Linear discrimination with

$$\mathcal{S} = \{\mathbf{x} \mapsto \mathtt{sign}\{\beta^T \mathbf{x} + \beta_0\} \,/\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}\}$$

Probality vs Optimization?

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

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

Canonical approach: $\widehat{f}_{\mathcal{S}} = \operatorname{argmin}_{f \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f(\mathbf{X}_i))$

Problems

- How to choose $\mathcal{S}$?
- How to compute the minimization?

A Probabilistic Point of View

Solution: For $\mathbf{X}$, estimate $Y|\mathbf{X}$ 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 ℓ by an upper bound ℓ' and minimize the empirical loss: **SVR, SVM, Neural Network,Tree, Boosting**

Probabilistic Approach

- If $Y|\mathbf{X}$ is known, one can compute the best solution f^*

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

Bayes Plugin

- **Learning:** Estimation of $Y|x$ and pluging of this estimate in the Bayes classifier
- **Plugin**: a classifier $\widehat{f} : \mathcal{X} \to \mathcal{Y}$
  - $\ell^{0/1}$ loss:

$$\hat{f}(\mathbf{x}) = \begin{cases} +1 & \text{if } \widehat{p}_{+1}(\mathbf{x}) \geq \widehat{p}_{-1}(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$

  - Quadratic loss:

$$\hat{f}(\mathbf{x}) = \mathbb{E}\left[Y|\mathbf{x}\right]$$

- **Instantiations**:
  - Generative Modeling and Bayesian Methods
  - Parametric Conditional Models
  - Kernel Conditional Density Methods
- Importance of a corresponding efficient **numerical scheme**!

Optimization Approach

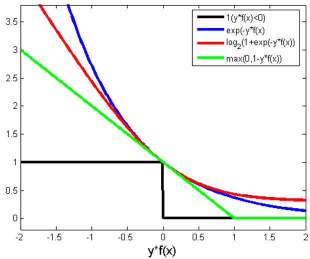
- The best solution f^* is the one minimizing
$$f^* = \arg\min R(f) = \arg\min \mathbb{E}\left[\ell(Y, f(X))\right]$$

Empirical Risk Minimization

- 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
$$\widehat{f} = f_{\widehat{\theta}} = \operatorname*{argmin}_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

- **Issue:** Minimization may be impossible in practice.
- **Solution:** Replace ℓ by ℓ' 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: $\ell^{0/1}(y, f(x)) = \mathbf{1}_{y \neq f(x)}$
- 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

Probabilistic Approach

- **Principle:** estimate the **conditional law** $Y|X$ and use it to take an **informed** decision.
- **Motto:** If you know the world, everything is easy!
- Emphasis on **Interpretation**
- **Pro:**
  - Interpretable models.
  - Lots of flexibility in the generative model.
  - Simultaneous decision optimization.
- **Cons:**
  - Computational issue.
  - No need to know the law to take a decision.

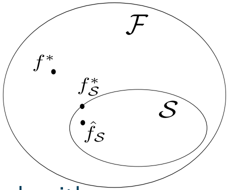
Optimization Approach

- **Principle:** construct a **surrogate decision** criterion and use it to take an **optimized** decision.
- **Motto:** You should focus on your goal!
- Emphasis on **Prediction**
- **Pro:**
  - Focus on the true goal!
  - Can use very clever optimization algorithm.
  - No need to obtain the best solution.
- **Cons:**
  - Black box model.
  - Not robust to a change of decision zone.

Outline

Bias-Variance Dilemna

- General setting:
  - $\mathcal{F} = \{$measurable fonctions $\mathcal{X} \to \mathcal{Y}\}$
  - Best solution: $f^* = \text{argmin}_{f \in \mathcal{F}} \mathcal{R}(f)$
  - Class $\mathcal{S} \subset \mathcal{F}$ of functions
  - Ideal target in $\mathcal{S}$: $f^*_{\mathcal{S}} = \text{argmin}_{f \in \mathcal{S}} \mathcal{R}(f)$
  - Estimate in $\mathcal{S}$: $\widehat{f}_{\mathcal{S}}$ obtained with a numerical algorithm



Approximation error and estimation error (Bias/Variance)

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

- 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**)

Models and Optimization

General Methodology

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

- 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 Θ:
  - $\|\theta\|_p \leq C$,
  - More complex restriction: $\text{comp}(\theta) \leq C$

- Methodology:
  - Choice of the loss function ℓ' (Likelihood / Convex surrogate)
  - Choice of the minimization algorithm...

Models and Optimization

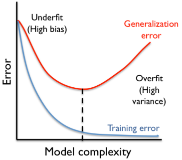
General Penalized Methodology

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

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

- Lots of freedom!
- Example of parametrization:
  - Linear: $f_\theta(x) = \langle \theta, x \rangle$ or $f_\theta(x) = \operatorname{sign}(\langle \theta, x \rangle)$
  - (Deep) Neural Network: much more complex parametrization.
- Restriction on Θ:
  - $\|\theta\|_p \leq C$,
  - More complex restriction: $\operatorname{comp}(\theta) \leq C$
  - Penalization: Lagrangian reformulation
- Methodology:
  - Choice of the loss function ℓ' (Likelihood / Convex surrogate)
  - Choice of the minimization algorithm...

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 $\mathcal{S}$

$$R_n(\widehat{f_{\mathcal{S}}}) \to R_n(\widehat{f_{\mathcal{S}}}) + \mathrm{pen}(\mathcal{S})...$$

- Penalization calibration issue...
- Simultaneous CV control issue...

Penalization and Cross-Validation

Practical Selection Methodology

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

- 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

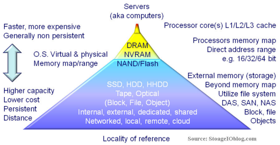
- Combine (randomized) models build in parallel:
  - (Weighted) model averaging,
  - Exponential Weighted Aggregation,
  - Super Learner,
  - Bayesian averaging.
- Less theoretical analysis.

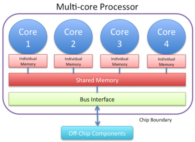
Sequential Combination

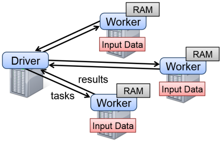
- Boosting / Greedy Gradient Descent Algorithm
- Very efficent in practice / Few convincing analysis...

Outline

Big Data?



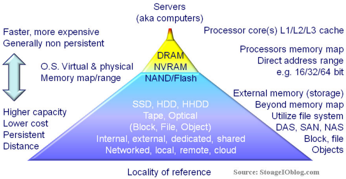


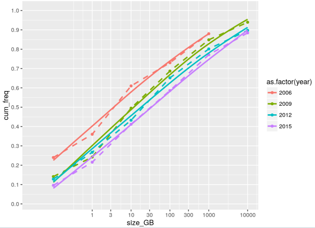


Hardware Constraints

- All the computations are done in a core using data stored somewhere nearby.
- Constrainst:
    - Data access / storage (Locality of Reference).
    - Multiple core architecture (Parallelization).
    - Cluster (Distribution)

Locality of Reference

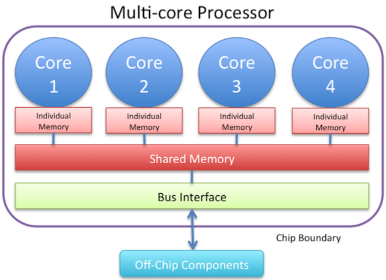




Memory Issue

- 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.)

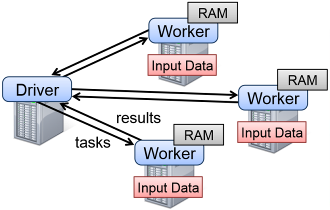
Parallelization

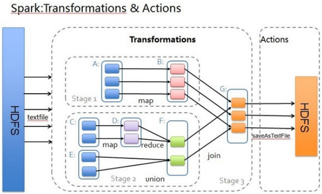


Speed Issue

- **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**!

Distribution





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

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.

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.
- More **deep science** in 2023?