On-line learning
Where are we so far?

Antoine Cornuéjols

MMIP, AgroParisTech, Paris

May 14th, 2008

"Incremental learning": a new topic?

The first learning algorithms were all incremental:

- Perceptron [Rosenblatt, 1957-1962]
- CHECKER [Samuel, 1959]
- ARCH [Winston, 1970]
- Version Space [Mitchell, 1978, 1982], ...

"Incremental learning": a new topic?

The first learning algorithms were all incremental:

- Perceptron [Rosenblatt, 1957-1962]
- CHECKER [Samuel, 1959]
- ARCH [Winston, 1970]
- Version Space [Mitchell, 1978, 1982], ...

However, **most existing learning algorithms are not!**

- C4.5 / Regression trees / ...
- SVM / Neural Networks / ...
- ILP systems / Grammatical inference / ...
- ...

Outline

Outline

The standard setting

Learning algorithms geared to the analysis of large data bases

- **Stationary and identical distribution** for learning and test
- **i.i.d. assumption** (independently and identically distributed)

$$\boxed{\mathbf{P}_{\mathcal{Y}}} \xrightarrow{\;y\;} \boxed{\mathbf{P}_{\mathcal{X}|\mathcal{Y}}} \xrightarrow{\;< x, y >\;}$$

Figure: Generative process for the examples.

The standard setting

Learning algorithms geared to the analysis of large data bases

- **Stationary and identical distribution** for learning and test
- **i.i.d. assumption** (independently and identically distributed)

$$\boxed{\mathbf{P}_{\mathcal{Y}}} \xrightarrow{\;y\;} \boxed{\mathbf{P}_{\mathcal{X}|\mathcal{Y}}} \xrightarrow{<\boldsymbol{x}, y>}$$

Figure: Generative process for the examples.

Almost correct prediction (most of the time) (PAC)

$$L(h) \;=\; \mathbf{P}_{\mathcal{XY}}\{h(\boldsymbol{x}) \neq y\}$$

The standard setting

Optimizing the expected risk

Real risk: expected loss

$$R(h) \;=\; \mathbb{E}[\ell(h(\boldsymbol{x}), y)] \;=\; \int_{\boldsymbol{x}\in\mathcal{X}, y\in\mathcal{Y}} \ell(h(\boldsymbol{x}), y)\; \mathbf{P}_{\mathcal{X}\mathcal{Y}}\; d(\boldsymbol{x}, y)$$

The standard setting

Optimizing the expected risk

Real risk: expected loss

$$R(h) \;=\; \mathbb{E}[\ell(h(\boldsymbol{x}), y)] \;=\; \int_{\boldsymbol{x}\in\mathcal{X}, y\in\mathcal{Y}} \ell(h(\boldsymbol{x}), y)\; \mathbf{P}_{\mathcal{X}\mathcal{Y}}\; d(\boldsymbol{x}, y)$$

But $\mathbf{P}_{\mathcal{X}\mathcal{Y}}$ is unknown, then use: $\mathcal{S}_m = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$

Empirical risk Minimization

$$\hat{h} \;=\; \underset{h\in\mathcal{H}}{\text{ArgMin}}\left[R_m(h)\right] \qquad = \underset{h\in\mathcal{H}}{\text{ArgMin}}\left[\frac{1}{m}\sum_{i=1}^{m} \ell(h(\boldsymbol{x}_i), y_i)\right]$$

The standard setting

Optimizing the expected risk

Real risk: expected loss

$$R(h) \;=\; \mathbb{E}[\ell(h(\boldsymbol{x}), y)] \;=\; \int_{\boldsymbol{x}\in\mathcal{X}, y\in\mathcal{Y}} \ell(h(\boldsymbol{x}), y)\; \mathbf{P}_{\mathcal{X}\mathcal{Y}}\; d(\boldsymbol{x}, y)$$

But $\mathbf{P}_{\mathcal{X}\mathcal{Y}}$ is unknown, then use: $\mathcal{S}_m = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$

Empirical risk Minimization

$$\hat{h} \;=\; \underset{h\in\mathcal{H}}{\mathrm{ArgMin}}\big[R_m(h)\big] + \text{Reg}\big] \;=\; \underset{h\in\mathcal{H}}{\mathrm{ArgMin}}\left[\frac{1}{m}\sum_{i=1}^{m}\ell(h(\boldsymbol{x}_i), y_i)\right] + \lambda\,\text{Capacity}(\mathcal{H})\big]$$

The standard setting

Optimizing the expected risk

Real risk: expected loss

$$R(h) \;=\; \mathbb{E}[\ell(h(\boldsymbol{x}), y)] \;=\; \int_{\boldsymbol{x}\in\mathcal{X}, y\in\mathcal{Y}} \ell(h(\boldsymbol{x}), y)\; \mathbf{P}_{\mathcal{X}\mathcal{Y}}\; d(\boldsymbol{x}, y)$$

But $\mathbf{P}_{\mathcal{X}\mathcal{Y}}$ is unknown, then use: $\mathcal{S}_m = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$

Empirical risk Minimization

$$\hat{h} \;=\; \underset{h\in\mathcal{H}}{\text{ArgMin}}\big[R_m(h)\big] + \text{Reg}\big] \;=\; \underset{h\in\mathcal{H}}{\text{ArgMin}}\left[\frac{1}{m}\sum_{i=1}^{m} \ell(h(\boldsymbol{x}_i), y_i)\right] + \lambda\,\text{Capacity}(\mathcal{H})\big]$$

1. All examples are equal: **no forgetting**
2. Commutative criterion: **no information from the sequence**

On-line learning: why bother?

A wealth of new applications

1. **Limited resources:**
   - Learning from very large data bases (e.g. Telecoms: millions of examples ; EGEE: billions of examples, ...)
2. **"Anytime" constraints**: Data streaming

On-line learning: why bother?

A wealth of new applications

1. **Limited resources:**
   - Learning from very large data bases (e.g. Telecoms: millions of examples ; EGEE: billions of examples, ...)
2. **“Anytime” constraints**: Data streaming
3. **Covariate shift**: stationary target concept but changing distribution
4. **Active learning**

On-line learning: why bother?

A wealth of new applications

1. **Limited resources:**
   - Learning from very large data bases (e.g. Telecoms: millions of examples ; EGEE: billions of examples, ...)
2. **“Anytime” constraints**: Data streaming
3. **Covariate shift**: stationary target concept but changing distribution
4. **Active learning**
5. **Concept drift**
6. **Transfer learning** from one task to another
7. **Tutored learning** with a professor

On-line learning: the issues

Computational constraints

Possible in principle with standard (one-shot and i.i.d.) approach, but too costly computationally

→ **Reduce time and space complexity**

- ***Stochastic gradient / incremental learning***

On-line learning: the issues

Computational constraints

Possible in principle with standard (one-shot and i.i.d.) approach, but too costly computationally

→ **Reduce time and space complexity**

- ***Stochastic gradient / incremental learning***

Stationary environment but changing distribution + anytime constraints

Not i.i.d.

→ **Anticipate and take advantage of sequence information**

- ***covariate shift / transductive learning / tracking***

On-line learning: the issues

Computational constraints

Possible in principle with standard (one-shot and i.i.d.) approach, but too costly computationally

→ **Reduce time and space complexity**

- ***Stochastic gradient / incremental learning***

Stationary environment but changing distribution + anytime constraints

Not i.i.d.

→ **Anticipate and take advantage of sequence information**

- ***covariate shift / transductive learning / tracking***

Changing environment

Not i.i.d. + Non stationary

→ **Anticipate and take advantage of sequence information**

- ***concept drift***
- ***transfer learning / tutored learning***

Outline

Stochastic gradient vs. total gradient

Total gradient

Stochastic gradient vs. total gradient

Total gradient

$$\hat{h} = \operatorname*{ArgMin}_{h \in \mathcal{H}} R_m(h) = \operatorname*{ArgMin}_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \ell(h(\boldsymbol{x}_i), y_i)$$

Total gradient

$$\begin{aligned} h_t &= h_{t-1} - \Phi_t \frac{\partial R_m(h_{t-1})}{\partial h} \\ &= h_{t-1} - \Phi_t \frac{1}{m} \frac{\partial}{\partial h} \sum_{i=1}^{m} (h_{t-1}(\boldsymbol{x}_i), y_i) \end{aligned}$$

Linear convergence towards the optimum $\hat{h}$ de $R_m(h)$: $(h_t - \hat{h})^2$ converges as e^{-t}.

BLC05 L. Bottou et Y. Le Cun (2005) *"On-line learning for very large datasets"* Applied Stochastic Models in Business and Industry, 21(2):137-151, 2005.

Stochastic gradient vs. total gradient

Stochastic gradient

$$\hat{h} \; = \; \underset{h \in \mathcal{H}}{\text{ArgMin}} \, R_m(h) \; = \; \underset{h \in \mathcal{H}}{\text{ArgMin}} \, \frac{1}{m} \sum_{i=1}^{m} \ell(h(\boldsymbol{x}_i), y_i)$$

Stochastic gradient

$$h_t \; = \; h_{t-1} - \frac{1}{t} \, \Phi_t \, \frac{\partial L}{\partial h}(h_{t-1}(\boldsymbol{x}_t), y_t)$$

Stochastic gradient vs. total gradient

Stochastic gradient

$$\hat{h} \;=\; \underset{h \in \mathcal{H}}{\text{ArgMin}}\, R_m(h) \;=\; \underset{h \in \mathcal{H}}{\text{ArgMin}}\, \frac{1}{m} \sum_{i=1}^{m} \ell(h(\boldsymbol{x}_i), y_i)$$

Stochastic gradient

$$h_t \;=\; h_{t-1} - \frac{1}{t}\, \Phi_t\, \frac{\partial L}{\partial h}(h_{t-1}(\boldsymbol{x}_t), y_t)$$

- Converges slowly towards a local optimum of $R_m(h)$: $(h_t - \hat{h})^2$ converges as $\frac{1}{t}$.
- In fact, converges quickly towards the region of the optimum but slowly then because of the noisy (stochastic) gradient.

Much **simpler** than batch

Stochastic gradient vs. total gradient

Computational complexity

- Batch
  - Store N examples
  - Gradient in $\mathcal{O}(N)$ operations

Stochastic gradient vs. total gradient

Computational complexity

- Batch
  - Store N examples
  - Gradient in $\mathcal{O}(N)$ operations
- On-line
  - Must memorize the “sufficient past” (in h_t)
  - Gradient in $\mathcal{O}(1)$ operations

Stochastic gradient vs. total gradient

Computational complexity

- Batch
  - Store N examples
  - Gradient in $\mathcal{O}(N)$ operations
- On-line
  - Must memorize the "sufficient past" (in h_t)
  - Gradient in $\mathcal{O}(1)$ operations

Approximation

- Batch
  - Converges towards $h* = \mathrm{ArgMin}_{h \in \mathcal{H}} R(h)$ with approximation $\mathcal{O}(1/t)$
- On-line
  - Converges towards $h* = \mathrm{ArgMin}_{h \in \mathcal{H}} R(h)$ with approximation $\mathcal{O}(1/t)$

Stochastic gradient vs. total gradient

Computational complexity

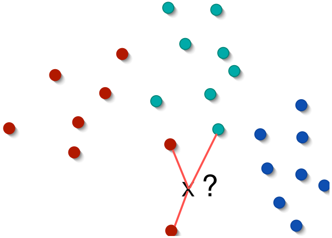
- Batch
  - Store N examples
  - Gradient in $\mathcal{O}(N)$ operations
- On-line
  - Must memorize the “sufficient past” (in h_t)
  - Gradient in $\mathcal{O}(1)$ operations

Approximation

- Batch
  - Converges towards $h* = \mathrm{ArgMin}_{h \in \mathcal{H}} R(h)$ with approximation $\mathcal{O}(1/t)$
- On-line
  - Converges towards $h* = \mathrm{ArgMin}_{h \in \mathcal{H}} R(h)$ with approximation $\mathcal{O}(1/t)$
  - But **can consider more examples !!**
    ($\mathcal{O}(N \log N)$ instead of N for batch)

Incremental learning
Illustration

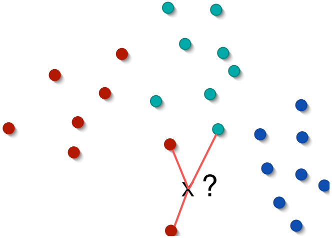
Nearest neighbors



- Simple algorithm ("*lazy learning*")

Incremental learning
Illustration

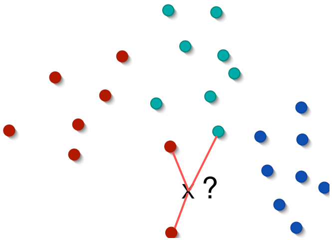
Nearest neighbors



- Simple algorithm ("*lazy learning*")
- Order independent

Incremental learning
Illustration

Nearest neighbors

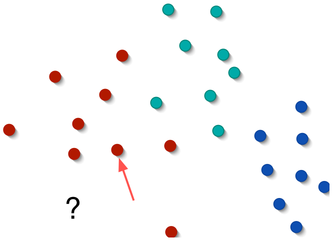


- Simple algorithm ("*lazy learning*")
- Order independent
- But growing computational cost (time and space): $\mathcal{O}(m)$

Incremental learning

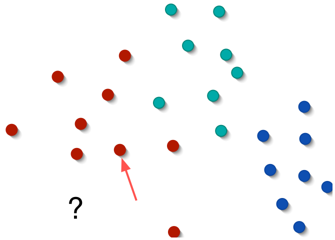
Illustration

Nearest neighbors (2)
with limited memory



Incremental learning
Illustration

Nearest neighbors (2)
with limited memory



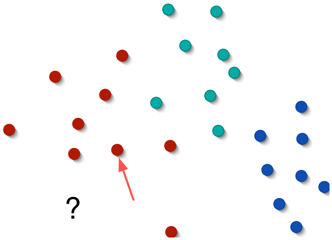
- Selection of “prototypes”
  - Eliminate the outlier
  - Eliminate the most ancien
  - Compute and keep center of gravity with the closest point
  - ...

Incremental learning

Illustration

Nearest neighbors (2)

with limited memory

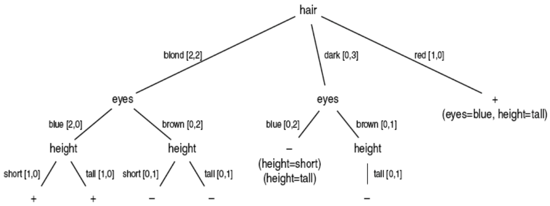


- Selection of “prototypes”
  - Eliminate the outlier
  - Eliminate the most ancien
  - Compute and keep center of gravity with the closest point
  - ...
- Order dependent

Incremental learning
Illustration

Incremental induction of decision trees (ID5R)

class	height	hair	eyes
–	short	blond	brown
–	tall	dark	brown
+	tall	blond	blue
–	tall	dark	blue
–	short	dark	blue
+	tall	red	blue
–	tall	blond	brown
+	short	blond	blue

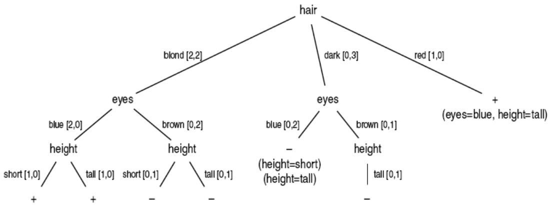


UTG89 Paul Utgoff (1989) *"Incremental Induction of Decision Trees"* Machine Learning Journal, vol.4, No.2, 161-186

Incremental learning
Illustration

Incremental induction of decision trees (ID5R)

class	height	hair	eyes
–	short	blond	brown
–	tall	dark	brown
+	tall	blond	blue
–	tall	dark	blue
–	short	dark	blue
+	tall	red	blue
–	tall	blond	brown
+	short	blond	blue



- Actually memorizes all examples
- Order independent
- But computational time at each step : $\mathcal{O}(m \cdot d \cdot b^d)$

UTG89 Paul Utgoff (1989) *"Incremental Induction of Decision Trees"* Machine Learning Journal, vol.4, No.2, 161-186

Incremental learning

Assessment

Numerous heuristic algorithms

Motivations

- Computational constraints (e.g. constant time)
- Time series

New questions

- What to keep in memory?
- Sequence effects
  - How to reduce them?
  - (How to use the information in the sequence?)

On-line learning: the issues

Computational constraints

Possible in principle with standard (one-shot and i.i.d.) approach, but too costly computationally

→ **Reduce time and space complexity**

- ***Stochastic gradient / incremental learning***

Stationary environment but changing distribution + anytime constraints

Not i.i.d.

→ **Anticipate and take advantage of sequence information**

- ***covariate shift / transductive learning / tracking***

Changing environment

Not i.i.d. + Non stationary

→ **Anticipate and take advantage of sequence information**

- ***concept drift***
- ***transfer learning / tutored learning***

Covariate shift

Definition

Changing $\mathbf{P}_{\mathcal{X}}$

Covariate shift
Definition

Changing $\mathbf{P}_{\mathcal{X}}$

Examples:

- **Non stationary input data** ($\mathbf{P}_{\mathcal{X}}$ changes but not $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$)
  - Medicine: seasonal variations
  - Spam filtering (adaptation to a new user)

Covariate shift

Definition

Changing $\mathbf{P}_{\mathcal{X}}$

Examples:

- **Non stationary input data** ($\mathbf{P}_{\mathcal{X}}$ changes but not $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$)
  - Medicine: seasonal variations
  - Spam filtering (adaptation to a new user)
- **Bias in the selection process** in learning
  - Artificially balanced training data (but not in test)
  - Active learning
  - Interpolation vs. extrapolation (in regression)

Covariate shift

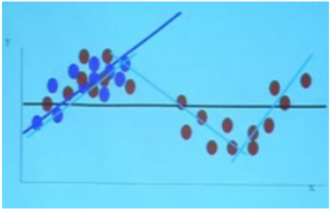
Why is it a problem?

No longer a “direct” link between empirical risk and real risk

Covariate shift

Why is it a problem?

No longer a “direct” link between empirical risk and real risk



Covariate shift

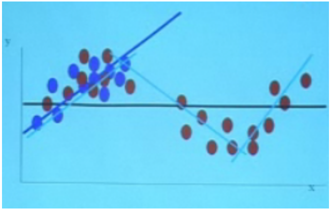
Why is it a problem?

No longer a “direct” link between empirical risk and real risk

Modify the inductive criterion

The performance for the target distribution $\mathbf{P}'_{\mathcal{X}}$ (*generalization*) depends on :

- The performance for $\mathbf{P}_{\mathcal{X}}$ (*learning*)
- The similarity between $\mathbf{P}_{\mathcal{X}}$ and $\mathbf{P}'_{\mathcal{X}}$



Covariate shift

Approaches

"Importance weighted" inductive criterion

Principle : weighting the classical ERM

$$R_{Cov}(h) \;=\; \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\mathbf{P}_{\mathcal{X}'}(\boldsymbol{x}_i)}{\mathbf{P}_{\mathcal{X}}(\boldsymbol{x}_i)}\right)^{\lambda}(h(\boldsymbol{x}_i - y_i)^2$$

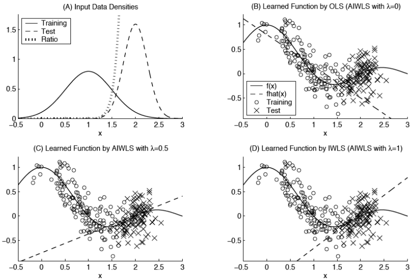
Covariate shift
Approaches

"Importance weighted" inductive criterion

Principle : weighting the classical ERM

$$R_{Cov}(h) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\mathbf{P}_{\mathcal{X}'}(\boldsymbol{x}_i)}{\mathbf{P}_{\mathcal{X}}(\boldsymbol{x}_i)}\right)^{\lambda}(h(\boldsymbol{x}_i - y_i))^2$$

λ controls the stability / consistency (absence of bias)



SKM07 M. Sugiyama and M. Kraudelat and K.-R. Müller (2007) *"Covariate Shift Adaptation by Importance Weighted Cross Validation"* Journal of Machine Learning Research, vol.8: 985-1005.

Covariate shift

Approaches

"Importance weighted" inductive criterion

How to get $\frac{\mathbf{P}_{\mathcal{X}'}(\boldsymbol{x}_i)}{\mathbf{P}_{\mathcal{X}}(\boldsymbol{x}_i)}$?

Empirical estimation

Semi-supervised learning

Transduction
Definition

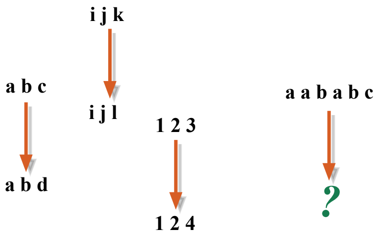
- Given a training set $\mathcal{S}_m = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$,
  and the knowledge of test data points $\boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+k}$
- Identify the best classification vector $y_{m+1}, \ldots, y_{m+k}$ from a given set of possible vectors $Y \in \mathcal{Y}^k$

One is no longer looking for a decision function defined over $\mathcal{X}$!!

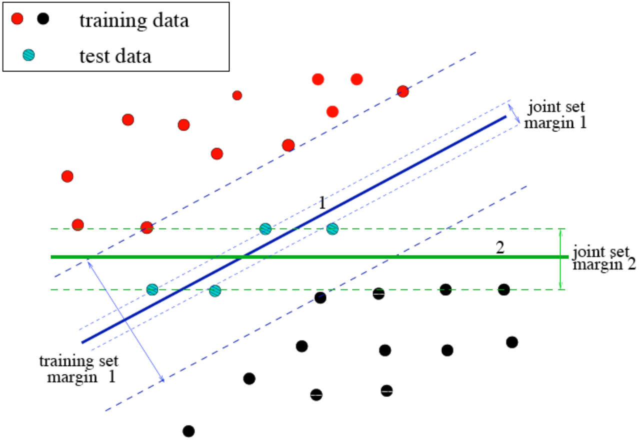
Transduction
Definition

- Given a training set $\mathcal{S}_m = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$, **and** the knowledge of test data points $\boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+k}$
- Identify the best classification vector $y_{m+1}, \ldots, y_{m+k}$ from a given set of possible vectors $Y \in \mathcal{Y}^k$

One is no longer looking for a decision function defined over $\mathcal{X}$!!



Transduction
Methods



On-line learning: the issues

Computational constraints

Possible in principle with standard (one-shot and i.i.d.) approach, but too costly computationally

→ **Reduce time and space complexity**

- ***Stochastic gradient / incremental learning***

Stationary environment but changing distribution + anytime constraints

Not i.i.d.

→ **Anticipate and take advantage of sequence information**

- ***covariate shift / transductive learning / tracking***

Changing environment

Not i.i.d. + Non stationary

→ **Anticipate and take advantage of sequence information**

- ***concept drift***
- ***transfer learning / tutored learning***

Concept drift

Definition

Drift of $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$

- Profiles of customers (purchases function of *income, age, ...*)
- Document filtering function of the interests of the user

Concept drift
Definition

Drift of $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$

- Profiles of customers (purchases function of *income, age, ...*)
- Document filtering function of the interests of the user

Problems:

- Detecting variations but be robust to noise
- Follow the evolutions but stay robust: Control forgetting

- The oldest the data, the more likely they are obsolete

But:

- The larger the training set, the better the generalization

Concept drift

Definition

Drift of $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$

- Profiles of customers (purchases function of *income, age, ...*)
- Document filtering function of the interests of the user

Problems:

- Detecting variations but be robust to noise
- Follow the evolutions but stay robust: Control forgetting

- The oldest the data, the more likely they are obsolete

But:

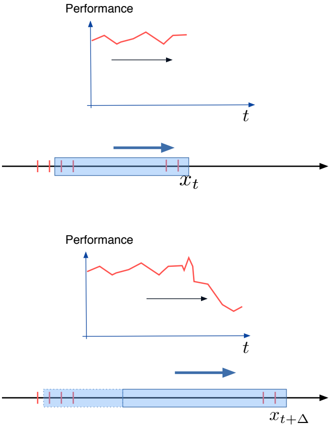
- The larger the training set, the better the generalization

Heuristic approaches

- **Window based approaches**. *Problem: control their size*
- **Weighting the examples** with respect to time. *Problem: control their weights*

Concept drift
Sliding window approach

Principle:

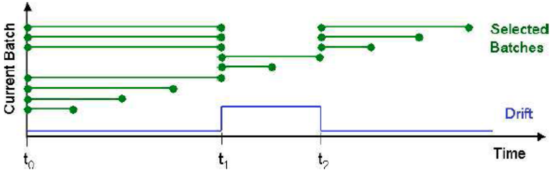


WK96 G. Widmer and M. Kubat (1996) *“Learning in the presence of concept drift ans hidden contexts”* Machine Learning 23: 69–101, 1996.

Concept drift
Sliding window approach

One (among many) method for the selection of windows

- Learn a classifier on the last batch
- Test it on every preceding windows
- Keep the windows where error $< \varepsilon$



SK M. Scholz and R. Klinkenberg (1996) *"Boosting classifiers for drifting concepts"* Intelligent Data Analysis (IDA) Journal, Volume 11, Number 1, March 2007.

Concept drift
A Boosting approach

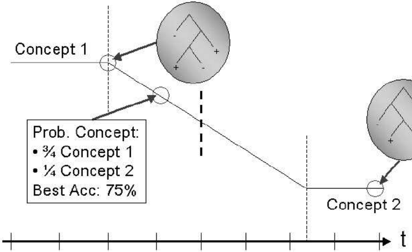


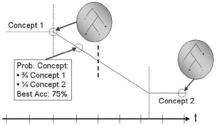
Fig. 4. Continuous concept drift, starting with a pure *Concept 1* and ending with a pure *Concept 2*. In between, the target distribution is a probabilistic mixture. It is optimal to predict *Concept 1* before the dotted line, and *Concept 2*, afterwards.

How to learn concept 2 **before** the end of the transition ?

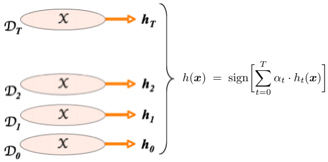
Concept drift
A Boosting approach

Principle :
gradually modify the distribution of the examples

by “substracting” the distribution associated with `concept 1`.



$$\forall \langle \boldsymbol{x}, y\rangle \in \mathcal{X} \times \mathcal{Y} : P_{D'}(\boldsymbol{x}, y) = P_D(\boldsymbol{x}, y) \cdot \frac{P_D[h_{t-1}(\boldsymbol{x}) = \hat{y}] \cdot P_D[y = y*]}{P_D[h_{t-1}(\boldsymbol{x}) = \hat{y}, y = y*]}$$



Concept drift
Assessment

Heuristics

- Efficient in their respective application domains
- Require a fine tuning
- Not easily transferable to other domains
- Lack theoretical fundations

Concept drift
Assessment

Heuristics

- Efficient in their respective application domains
- Require a fine tuning
- Not easily transferable to other domains
- Lack theoretical fundations

Theoretical analyses

What are the conditions for PAC learning with an error of ε ?

- Depends upon $d_{\mathcal{H}}$ and the speed of the drift v
- Possible if $v = \mathcal{O}(\varepsilon^2 / d_{\mathcal{H}}^2 \ln \frac{1}{\varepsilon}))$
- Rk: adversary protocol

HL94 D. P. Helmbold and P. M. Long (1994) *"Tracking drifting concepts by minimizing disagreements"* Machine Learning 14 (1): 27–45, 1994.

A theoretical approach to on-line learning

A weak framework:

- No assumption about the generative process of the examples

A theoretical approach to on-line learning

A weak framework:

- No assumption about the generative process of the examples

Inductive criterion

- No more notion of risk
- Comparison *a posteriori* to the performance of a set of N "experts"

A theoretical approach to on-line learning

A weak framework:

- No assumption about the generative process of the examples

Inductive criterion

- No more notion of risk
- Comparison *a posteriori* to the performance of a set of N "experts"

Learning algorithms

- Maintain a vector of weights on the expert advices

$$\hat{p}_t = \frac{\sum_{i=1}^{N} w_{i,t-1} p_{i,t}}{\sum_{i=1}^{N} w_{i,t-1}}$$

- The weights are function of the *regret* of each expert.

But does not take into account the information in the data sequence

The theoretical viewpoint on on-line learning

Questions ...

Do we have answers to these questions?

- Do we have theoretical guarantees about the performance of usual on-line learning systems (NNs, SVM, ID5, ...)?
- Do we have a satisfactory inductive criterion to replace "Empirical Risk Minimization"?
- Are we able to predict sequence effects?

The theoretical viewpoint on on-line learning

Questions ...

Do we have answers to these questions?

- Do we have theoretical guarantees about the performance of usual on-line learning systems (NNs, SVM, ID5, ...)?
  - NO!
- Do we have a satisfactory inductive criterion to replace “Empirical Risk Minimization”?
  - NO!
- Are we able to predict sequence effects?
  - NO!

Outline

Tracking
Motivation

In a lot of natural settings:

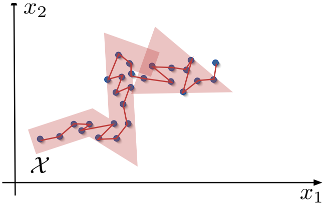
- Data comes *sequentially*
- *Temporal consistency*: consecutive data points come from "similar" distribution: not i.i.d.

SKS:07 R. Sutton and A. Koop and D. Silver (2007) *"On the role of tracking in stationary environments"* (ICML-07) Proceedings of the 24th international conference on Machine learning, ACM, pp.871-878, 2007.

Tracking
Motivation

In a lot of natural settings:

- Data comes *sequentially*
- *Temporal consistency*: consecutive data points come from “similar” distribution: not i.i.d.

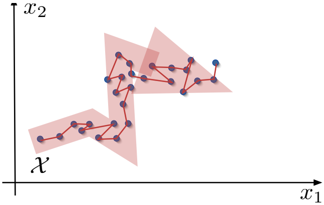


SKS:07 R. Sutton and A. Koop and D. Silver (2007) *“On the role of tracking in stationary environments”* (ICML-07) Proceedings of the 24th international conference on Machine learning, ACM, pp.871-878, 2007.

Tracking
Motivation

In a lot of natural settings:

- Data comes *sequentially*
- *Temporal consistency*: consecutive data points come from "similar" distribution: not i.i.d.



This enables:

- Powerful learning
- with limited resources (time + memory)

SKS:07 R. Sutton and A. Koop and D. Silver (2007) *"On the role of tracking in stationary environments"* (ICML-07) Proceedings of the 24th international conference on Machine learning, ACM, pp.871-878, 2007.

Tracking
Definition

Assumptions:

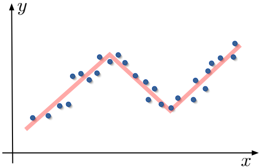
- Data streams
- *Temporal consistency*: consecutive data points come from "similar" distribution: not i.i.d.
- Limited resources: Restricted hypothesis space $\mathcal{H}$

SKS:07 R. Sutton and A. Koop and D. Silver (2007) *"On the role of tracking in stationary environments"* (ICML-07) Proceedings of the 24th international conference on Machine learning, ACM, pp.871-878, 2007.

Tracking
Definition

Assumptions:

- Data streams
- *Temporal consistency*: consecutive data points come from “similar” distribution: not i.i.d.
- Limited resources: Restricted hypothesis space $\mathcal{H}$

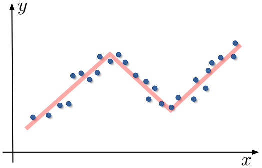


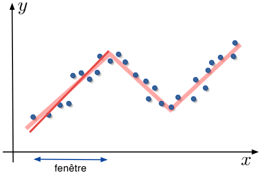
SKS:07 R. Sutton and A. Koop and D. Silver (2007) *“On the role of tracking in stationary environments”* (ICML-07) Proceedings of the 24th international conference on Machine learning, ACM, pp.871-878, 2007.

Tracking
Definition

Assumptions:

- Data streams
- *Temporal consistency*: consecutive data points come from "similar" distribution: not i.i.d.
- Limited resources: Restricted hypothesis space $\mathcal{H}$



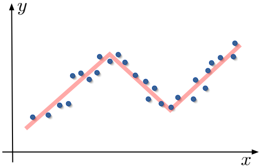


SKS:07 R. Sutton and A. Koop and D. Silver (2007) *"On the role of tracking in stationary environments"* (ICML-07) Proceedings of the 24th international conference on Machine learning, ACM, pp.871-878, 2007.

Tracking
Definition

Assumptions:

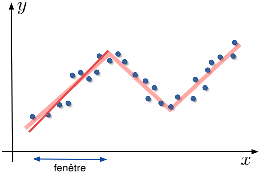
- Data streams
- *Temporal consistency*: consecutive data points come from "similar" distribution: not i.i.d.
- Limited resources: Restricted hypothesis space $\mathcal{H}$



"Local" learning

and local prediction :

$$\begin{aligned} L_t &= \ell(h_t(\boldsymbol{x}_t), y_t) \\ &= \ell(h_t(\boldsymbol{x}_t), f(x_t, \theta_t)) \end{aligned}$$



SKS:07 R. Sutton and A. Koop and D. Silver (2007) *"On the role of tracking in stationary environments"* (ICML-07) Proceedings of the 24th international conference on Machine learning, ACM, pp.871-878, 2007.

Tracking
Characteristics

If “temporal consistency” holds ...

... enormous advantage for learning

1. **Less computational cost**
   - *Time*: take into account fewer examples at each time step
   - *Space*: does not store every past examples
2. Intrinsically **on-line** and adapted to **non stationary environments**

Tracking
Characteristics

If “temporal consistency” holds ...

... enormous advantage for learning

1. **Less computational cost**
   - *Time*: take into account fewer examples at each time step
   - *Space*: does not store every past examples
2. Intrinsically **on-line** and adapted to **non stationary environments**

But can we:

1. **Formalize** this?
2. **Measure** this advantage?
3. Turn this into a **learning strategy**?

Tracking
Analysis

A fundamental tradeoff

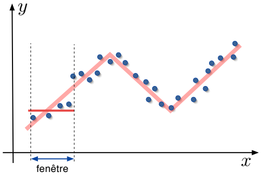
Temporal Consistency		i.i.d. data
Small memory	$\longleftrightarrow$	**Large memory**
Simple $\mathcal{H}$		**"Complex"** $\mathcal{H}$

Tracking

A new inductive problem

Notion of *temporal consistency*

$f(\cdot, \theta_t)$ continuous
and with bounded variation / θ_t



Tracking

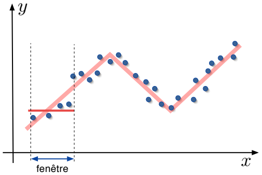
A new inductive problem

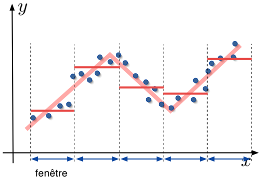
Notion of ***temporal consistency***

$f(\cdot, \theta_t)$ continuous
and with bounded variation / θ_t

New inductive criterion

$$L_{\langle 0,T\rangle}(r) = \sum_{t=0}^{T} \ell(h_t(\boldsymbol{x}_t), y_t) + \lambda \sum ||h_t - h_{t-1}||^2 + \text{Capacity}(\mathcal{R})$$





Tracking

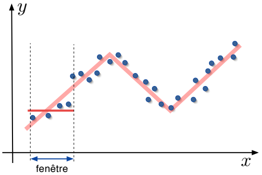
A new inductive problem

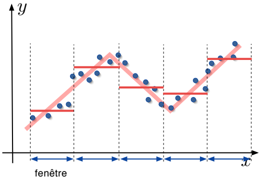
Notion of *temporal consistency*

$f(\cdot, \theta_t)$ continuous
and with bounded variation / θ_t

New inductive criterion

$$L_{\langle 0,T \rangle}(r) = \sum_{t=0}^{T} \ell(h_t(\boldsymbol{x}_t), y_t) + \lambda \sum ||h_t - h_{t-1}||^2 + \text{Capacity}(\mathcal{R})$$





Do not optimize the choice of ONE h any longer!!

but optimize the learning rule ($r \in \mathcal{R}$) instead: $(h_{t-1}, \boldsymbol{x}_t) \xrightarrow{r} h_t$!!

Tracking

Issues in want of answers

New inductive criterion

$$L_{\langle 0,T \rangle} = \sum_{t=0}^{T} \ell(h_t(\boldsymbol{x}_t), y_t) + \underbrace{\lambda \sum ||h_t - h_{t-1}||^2 + \text{Capacity}(\mathcal{R})}_{\text{new criterion}}$$

How to find a good learning rule $r \in \mathcal{R}$?

rule complexity

(memory + complexity)

$\longleftrightarrow$

Local complexity

of target function

Control of bias-variance (overfitting)

Outline

Conclusions

Emerging applications **can not** be solved within the classical setting

- **non i.i.d. data**: the sequence conveys information
- Learning is a **limited rationality activity**

Conclusions

Emerging applications **can not** be solved within the classical setting

- **non i.i.d. data**: the sequence conveys information
- Learning is a **limited rationality activity**

Lots of open questions

Conclusions

Emerging applications **can not** be solved within the classical setting

- **non i.i.d. data**: the sequence conveys information
- Learning is a **limited rationality activity**

Lots of open questions

- ***How to deal with non i.i.d. data***
  - **What to memorize?** / What to forget?
  - How to cope with or take advantage of **ordering effects**?
  - How to **facilitate future learning**: change representations, ...?

Conclusions

Emerging applications **can not** be solved within the classical setting

- **non i.i.d. data**: the sequence conveys information
- Learning is a **limited rationality activity**

Lots of open questions

- ***How to deal with non i.i.d. data***
  - **What to memorize?** / What to forget?
  - How to cope with or take advantage of **ordering effects**?
  - How to **facilitate future learning**: change representations, ...?
- ***What should the inductive criterion be?***
  - How to take the **computational resources** into the inductive criterion?
  - Optimize $h \in \mathcal{H}$ or $r \in \mathcal{R}$?

Conclusions

Emerging applications **can not** be solved within the classical setting

- **non i.i.d. data**: the sequence conveys information
- Learning is a **limited rationality activity**

Lots of open questions

- ***How to deal with non i.i.d. data***
  - **What to memorize?** / What to forget?
  - How to cope with or take advantage of **ordering effects**?
  - How to **facilitate future learning**: change representations, ...?
- ***What should the inductive criterion be?***
  - How to take the **computational resources** into the inductive criterion?
  - Optimize $h \in \mathcal{H}$ or $r \in \mathcal{R}$?

Already **a growing body of works**

- Covariate shift, transduction, concept drift, tracking ...
- ***Transfer*** *between tasks* / ***Teachability***

Conclusions

Emerging applications **can not** be solved within the classical setting

- **non i.i.d. data**: the sequence conveys information
- Learning is a **limited rationality activity**

Lots of open questions

- ***How to deal with non i.i.d. data***
  - **What to memorize?** / What to forget?
  - How to cope with or take advantage of **ordering effects**?
  - How to **facilitate future learning**: change representations, ...?
- ***What should the inductive criterion be?***
  - How to take the **computational resources** into the inductive criterion?
  - Optimize $h \in \mathcal{H}$ or $r \in \mathcal{R}$?

Already **a growing body of works**

- Covariate shift, transduction, concept drift, tracking ...
- ***Transfer*** *between tasks* / ***Teachability***

A mature science of on-line learning is in demand

The future ...

... starts here!

THANK YOU!