Online Machine Learning

Seminar : Computational Aspects of Machine Learning WS17/18

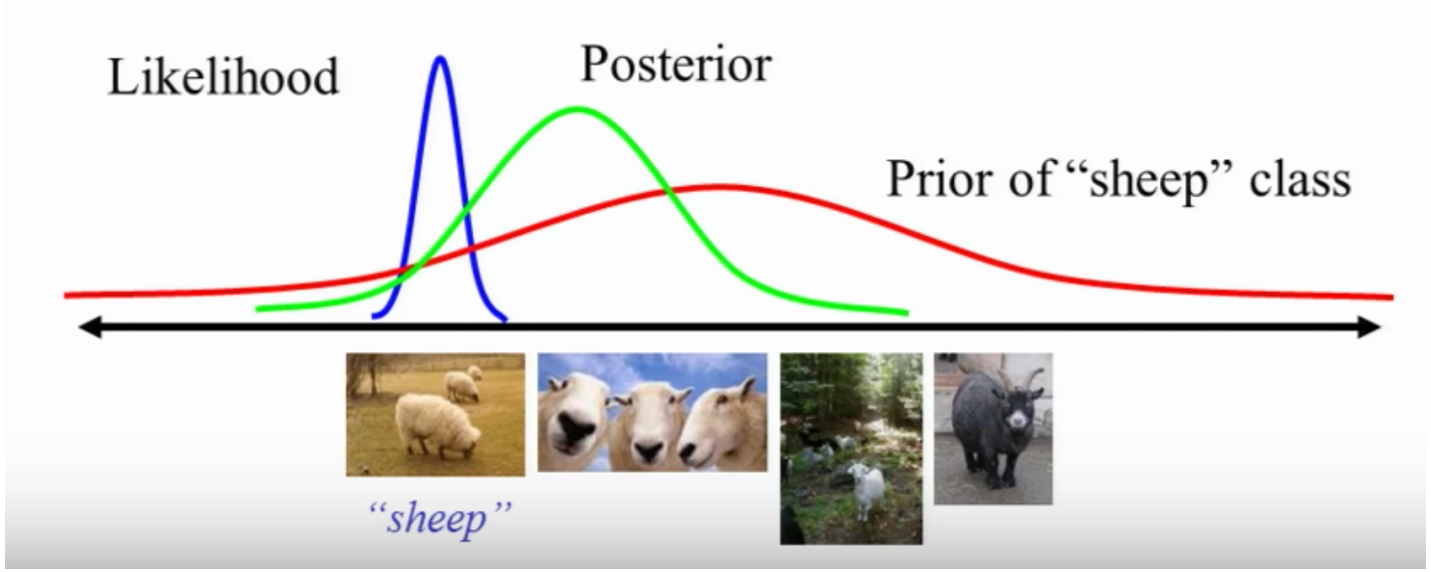
Abhilasha Mohania

Outline

1. Introduction to Online Learning
2. Use Cases
3. Online Algorithms
4. Sequential K-means Algorithm
5. Convexity
6. Current Trends/ Related Areas
7. References

Learning

- Learning is the ability to improve performance by observing data.



- Ted Talk : https://www.youtube.com/watch?v=cplaWsiu7Yg

Learning

- Learning can be broadly described in two ways:
  - Batch learning: Given a batch of data, want to extract rules.
  - Online learning: Data comes in one at a time.

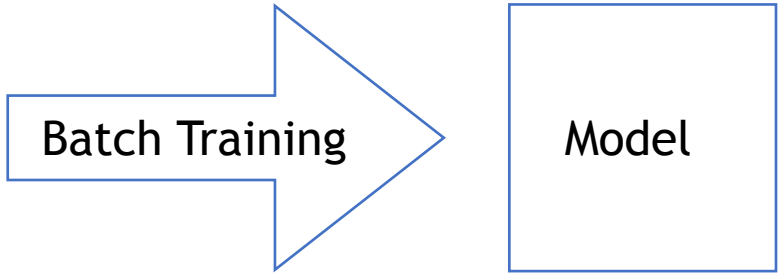
Batch learning

- access to whole data

d features

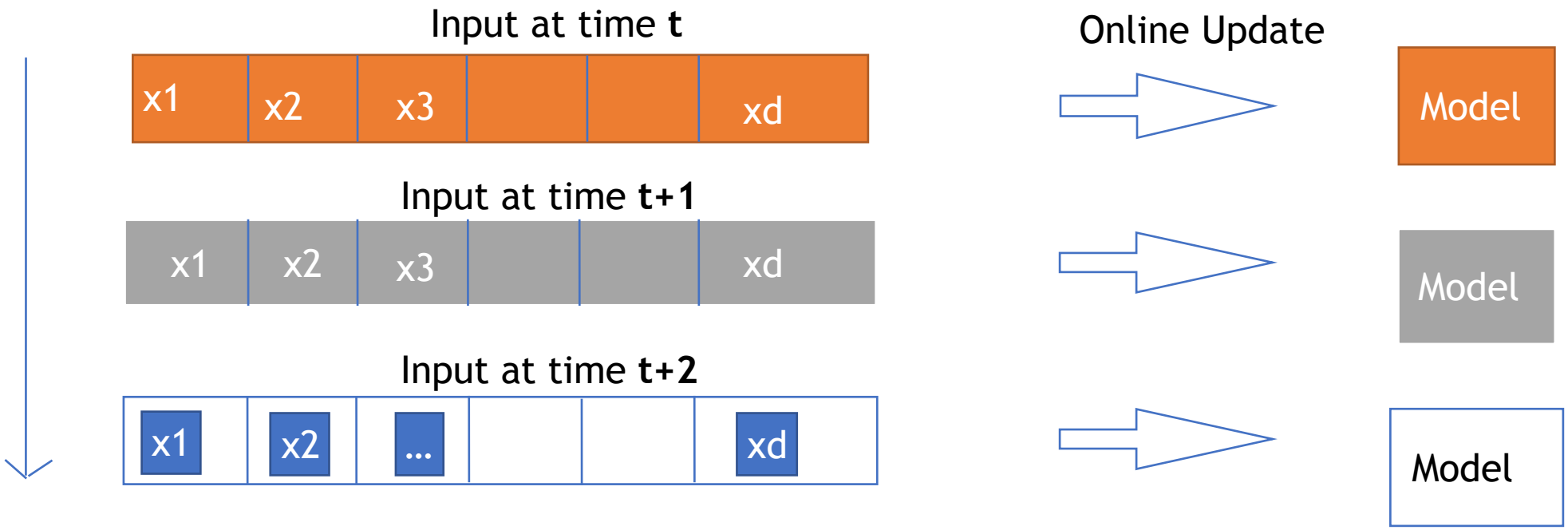
N samples

X1	X2	X3	…	Xd	Y
X1	X2	X3	…	Xd	Y
X1	X2	X3	…	Xd	Y
X1	X2	X3	…	Xd	Y
X1	X2	X3	…	Xd	Y



Online learning

- In Online Learning, your model evolves as it sees the new data, one example at a time



Online Learning Framework

Online Learning

for $t = 1, 2, \ldots$

receive question $\mathbf{x}_t \in \mathcal{X}$

predict $p_t \in D$

receive true answer $y_t \in \mathcal{Y}$

suffer loss $l(p_t, y_t)$

each t represents a trail.

Online Learning

for $t = 1, 2, \ldots$

receive question $\mathbf{x}_t \in \mathcal{X}$

predict $p_t \in D$

receive true answer $y_t \in \mathcal{Y}$

suffer loss $l(p_t, y_t)$

each t represents
a trail.

input data for trail t

Online Learning

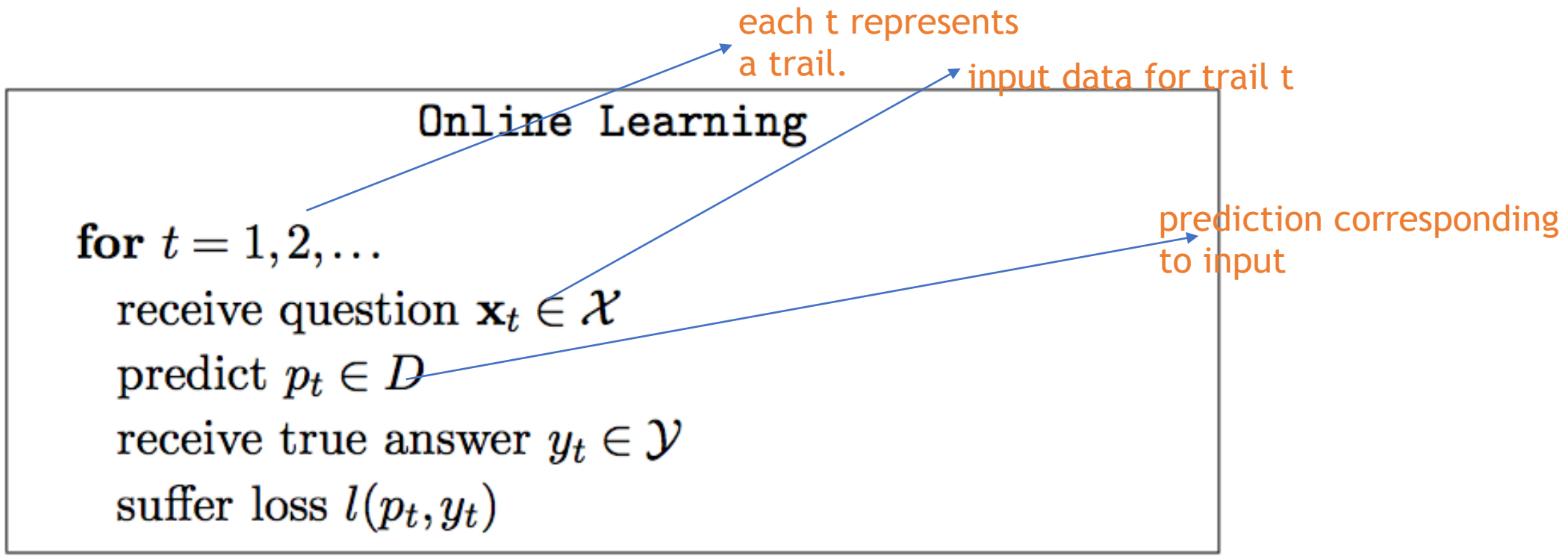
for $t = 1, 2, \ldots$

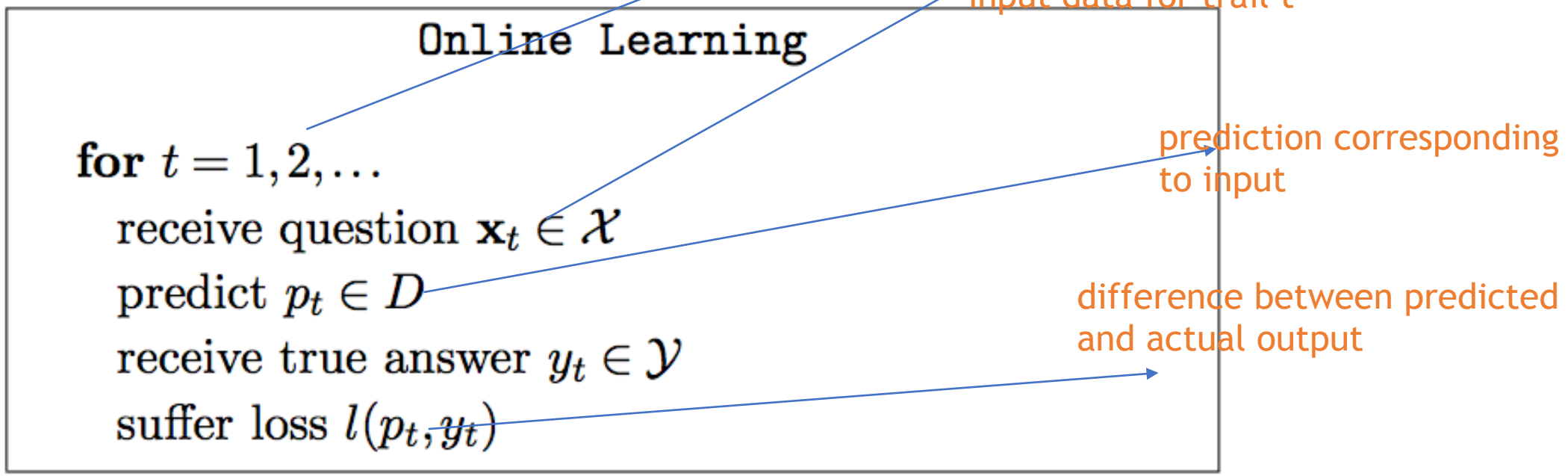
receive question $\mathbf{x}_t \in \mathcal{X}$

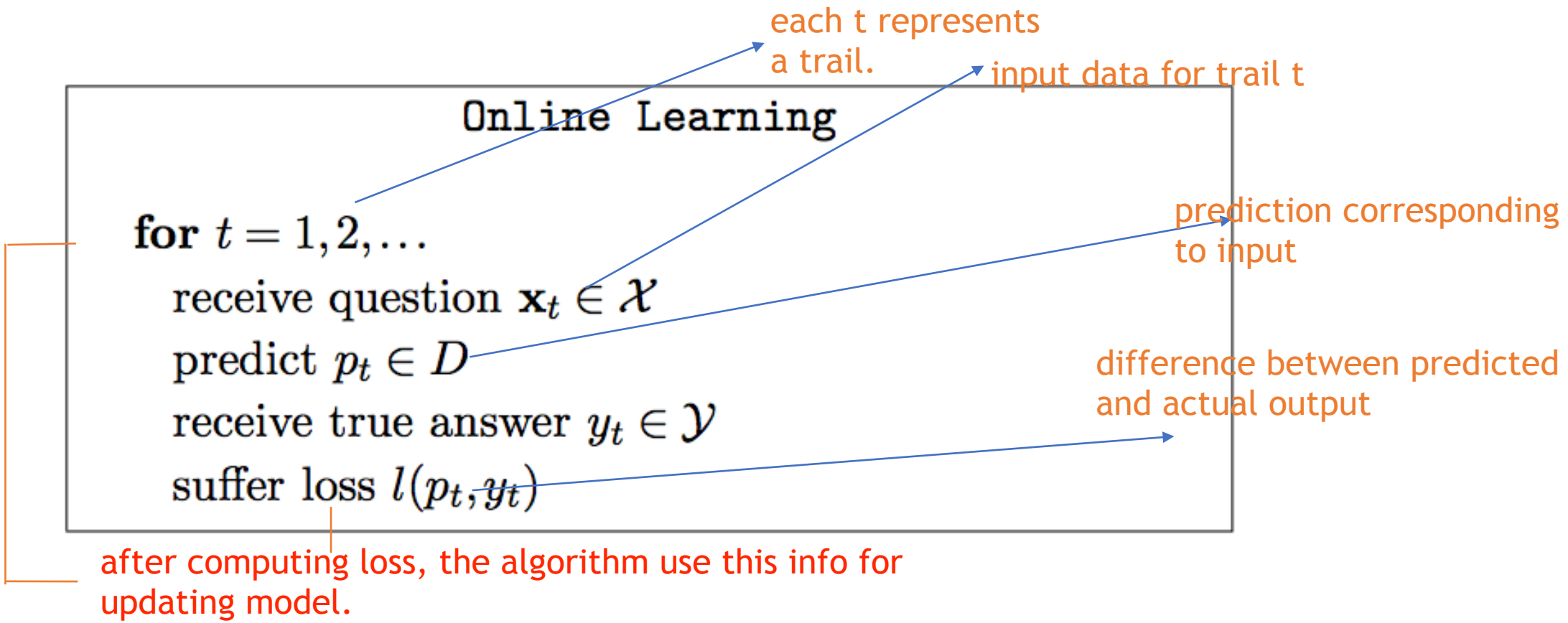
predict $p_t \in D$

receive true answer $y_t \in \mathcal{Y}$

suffer loss $l(p_t, y_t)$


each t represents
a trail.
input data for trail t
prediction corresponding
to input
Online Learning
for $t = 1, 2, \ldots$
receive question $\mathbf{x}_t \in \mathcal{X}$
predict $p_t \in D$
receive true answer $y_t \in \mathcal{Y}$
suffer loss $l(p_t, y_t)$


each t represents
a trail.
input data for trail t
Online Learning
for $t = 1, 2, \ldots$
receive question $\mathbf{x}_t \in \mathcal{X}$
predict $p_t \in D$
receive true answer $y_t \in \mathcal{Y}$
suffer loss $l(p_t, y_t)$
prediction corresponding
to input
difference between predicted
and actual output


each t represents
a trail.
input data for trail t
Online Learning
for $t = 1, 2, \ldots$
receive question $\mathbf{x}_t \in \mathcal{X}$
predict $p_t \in D$
receive true answer $y_t \in \mathcal{Y}$
suffer loss $l(p_t, y_t)$
prediction corresponding
to input
difference between predicted
and actual output
after computing loss, the algorithm use this info for
updating model.

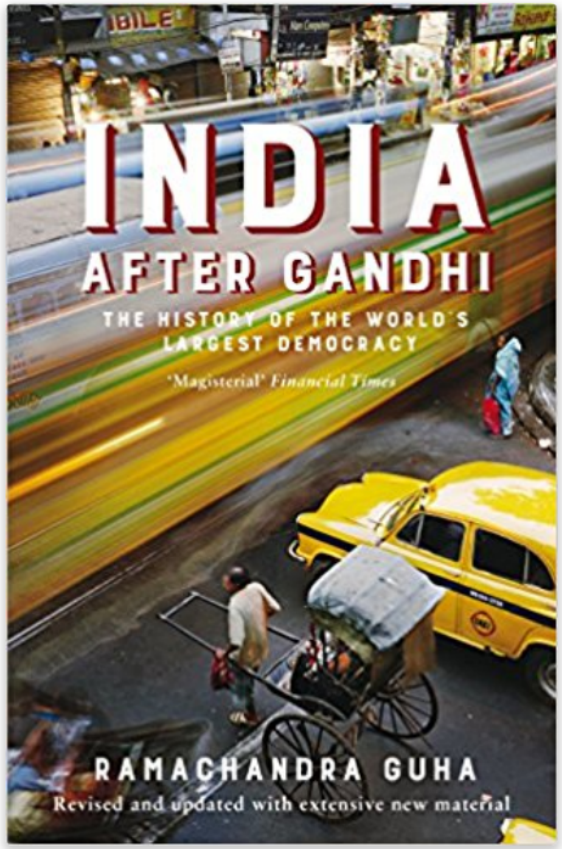
Examples : Spam Detection

- For t = 1,2,…,T
  - Receive an Email $\hat{x}_t$
  - Predict : spam or not $\hat{y}_t \in \{-1, +1\}$
  - sometimes get feedback from user
  - suffer loss : $l(y_t, \hat{y}_t)$
  - Perform better next time

Examples: Recommendation syste

You Looked at

Blick ins Buch



India After Gandhi: The History of the World's Largest Democracy

Paperback – 13. July 2017

by Ramachandra Guha (Autor)

★★★★½ 5 customer reviews

See all 9 formats and editions

Kindle Edition	Hardcover	Paperback
EUR 14.16	from EUR 28.83	EUR 16.99 ✓prime

Read with Our Free App

Want it delivered by Today, 6pm-9pm? Order within 59 mins and choose **Evening-Express** at checkout. Details

38 New from EUR 16.99 | 2 Used from EUR 23.99

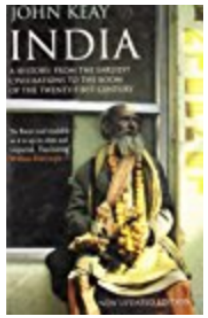
Born against a background of privation and civil war, divided along lines of caste, class, language and religion, independent India emerged, somehow, as a united and democratic country. Ramachandra Guha's hugely acclaimed book tells the full story – the pain and the struggle, the

E.g. Recommendation Systems

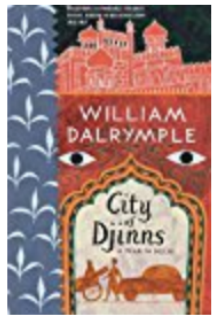
featured recommendations

India: A History
› John Keay
★★★★☆ 6
Kindle Edition
EUR 9.99

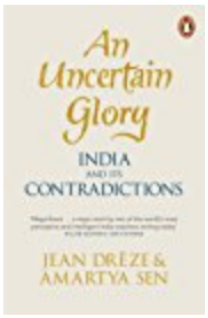
The God of Small Things
› Arundhati Roy
★★★★☆ 408
Kindle Edition



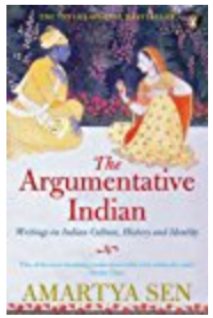
India: A History
› John Keay
★★★★☆ 6
Taschenbuch
EUR 11.99 prime



City of Djinns: A Year in Delhi
› William Dalrymple
★★★★★ 5
Taschenbuch



An Uncertain Glory: India and its...
Jean Dreze
★★★★★ 1
Taschenbuch



The Argumentat
Indian: Writings
Indian History,
and...
› Amartya Sen

Other examples

- Dynamic pricing
- Financial Markets - trading decisions, stock forecasts, gambling
- Weather prediction
- Fraud Detection
- Online ad Placement

What is Online Learning?

- Learn one step at a time.
- Data is being revealed sequentially.
- Predict(and often act based on) the next element of a sequence of outcomes/observations, given knowledge of past element + other info

Why Online learning ?

- It is more natural mode to many applications
- Memory efficient, one pass at a time
- Often distributional assumption free.
- Suitable for changing and adversarial environment

Use Cases

- Online Algorithms are useful in at least two scenarios:
  - when your data is too large to fit into memory
    - therefore, you need to train your model one example at a time
  - when new data is consistently being generated, and/or is dependent upon time.

Approaches

- Approaches that have been discussed in the literature:
  - Online learning from expert advice
  - Online learning from examples
  - General algorithms that may also be used in the online setting.

Approaches : Expert Advice

- In this approach, it is assumed that algorithm has multiple oracles(or experts at), which can use to produce its output, in each trial.
- In other words, the task of this algorithm is simply to learn which of the expert it should use
- The simplest algorithm in this realm is the Weighted Majority Algorithm.

1-Bit prediction

- Forecaster can see advice of N experts {1,2…N} (e.g financial analysts, new media, occult science,..) before making prediction

Day	1	2	3	4
Outcome				
Guess				
Expert_1	f1,1			
Expert_2	f2,1			
Expert_3	f3,1			

1-Bit prediction

- Forecaster can see advice of N experts {1,2…N} (e.g financial analysts, new media, occult science,..) before making prediction

Day	1	2	3	4
Outcome				
Guess	y1_hat			
Expert_1	f1,1			
Expert_2	f2,1			
Expert_3	f3,1			

1-Bit prediction

- Forecaster can see advice of N experts {1,2…N} (e.g financial analysts, new media, occult science,..) before making prediction

Day	1	2	3	4
Outcome	y1			
Guess	y1_hat			
Expert_1	f1,1			
Expert_2	f2,1			
Expert_3	f3,1			

1-Bit prediction

- Forecaster can see advice of N experts {1,2…N} (e.g financial analysts, new media, occult science,..) before making prediction

Day	1	2	3	4
Outcome	y1			
Guess	y1_hat			
Expert_1	f1,1	f1,2		
Expert_2	f2,1	f2,2		
Expert_3	f3,1	f3,2		

1-Bit prediction

- Forecaster can see advice of N experts {1,2…N} (e.g financial analysts, new media, occult science,..) before making prediction

Day	1	2	3	4
Outcome	y1			
Guess	y1_hat	y2_hat		
Expert_1	f1,1	f1,2		
Expert_2	f2,1	f2,2		
Expert_3	f3,1	f3,2		

1-Bit prediction

- Forecaster can see advice of N experts {1,2…N} (e.g financial analysts, new media, occult science,..) before making prediction

Day	1	2	3	4
Outcome	y1	y2		
Guess	y1_hat	y2_hat		
Expert_1	f1,1	f1,2		
Expert_2	f2,1	f2,2		
Expert_3	f3,1	f3,2		

1-Bit prediction

- Forecaster can see advice of N experts {1,2…N} (e.g financial analysts, new media, occult science,..) before making prediction

Day	1	2	3	4
Outcome	y1	y2	—	—
Guess	y1_hat	y2_hat	—-	—
Expert_1	f1,1	f1,2	—-	—
Expert_2	f2,1	f2,2	—-	—
Expert_3	f3,1	f3,2	—-	—

- Goal : Minimize # prediction mistakes

$$M_T(A) = \sum_{t=1}^{T} \mathbb{I}(\hat{y}_t \neq y_t)$$

‘1-Bit prediction’

- Realizability Assumptions (R.A) : suppose a perfect expert (one who never errors)

‘1-Bit prediction’

- Realizability Assumptions (R.A) : suppose a perfect expert (one who never errors)

Ideas ??

‘1-Bit prediction’

- Realizability Assumptions (R.A) : suppose a perfect expert (one who never errors)

The HALVING algorithm [barzdin-freivald’72, Angluin’88]

- Initialize S = {1,2,…N} (Trustworthy experts)
- At each round t
  - Go with majority votes among experts S:

$$\hat{y}_t = 1, if f \left|i \epsilon S : f_{i,t} = 1\right| >= \left|i \epsilon S : f_{i,t} = 0\right|$$

  - After seeing y_t, throw away all wrong experts from S.

- How many mistakes do you think, this algorithm can make ??

‘1-Bit prediction’

- Theorem : if there exists a perfect expert, MAJORITY makes at most $log_2 N$ number of mistakes that is,

$$M_T(MAJ) \leq log_2 N$$

- Proof :
  - Claim: whenever algorithm makes mistakes, >= half the experts are eliminated
  - cannot eliminate all experts since a perfect expert exists

‘1-Bit prediction’

- more realistic setting : what if there isn’t necessary a perfect expert, i.e best expert makes m>= 0 mistakes?

‘1-Bit prediction’

- more realistic setting :what if there isn’t necessary a perfect expert, i.e best expert makes m>= 0 mistakes?
  - straightforward modification of Majority => (m+1)$log_2 N$ mistakes.
  - can we do much better than this ?

‘1-Bit prediction’

- Weighted Majority Algorithm [Littlestone - warmuth’94]
  - parameter : ε ∈ [0,1]
  - Initialize : the weight for expert i, wi,0 = 1 ∀i
  - At each round t>=1
    - Predict $\hat{y_t}$ =1 if, $\sum_{i:f_{i,t}=1} W_{i,t-1} \geq \sum_{i:f_{i,t}=0} W_{i,t-1}$
    - observe and update $w_{i,t} = (w_{i,t-1})(1-\varepsilon)^{\mathbb{I}(f_{i,1} \neq y_t)}$ ∀i

Algorithm 1 Weighted Majority($W - MAJ$)

1: *Parameter*: $\varepsilon \in [0,1]$

2: *Initialize* : the weight for expert i, $w_{i,0} = 1 \quad \forall i$

3: **for** $t = 1,2,3,\ldots$ **do**

4: Predict

$$\hat{p}_t = \begin{cases} 1 & \text{if } \sum_{i:f_{i,t}=1} w_{i,t-1} \geq \sum_{i:f_{i,t}=0} w_{i,t-1} \\ 0 & \text{otherwise} \end{cases}$$

5: Observe y_t

6: $w_{i,t} = w_{i,t-1}(1-\varepsilon)^{\mathbb{I}\{f_{i,t} \neq y_t\}} \quad \forall i$

Mistake Bound for W-MAJ

$$M_T(W-MAJ) \leq 2(1+\varepsilon)M_T(i) + \frac{2}{\varepsilon} logN$$

Note on tuning the parameter ε: It is easy to see that the optimal ε which gives the tighest upper bound is $\sqrt{\frac{logN}{M_T(i)}}$. With this ε, the bound becomes $\mathbf{2}M_T(i) + 4\sqrt{M_T(i)logN}$.

Boosting [schapire, 1990; freund, 1995]

- boosting is highly successful batch learning algorithm.
- idea : combine weak “rules of thumb” to make accurate predictor.
- example : email spam detection
- Given : a set of training examples.
  - (wanna make fast money ? - spam)
  - (how is your thesis going ?, not spam)
- obtain another classifier :
  - e.g “to address” => spam.
- at the end, predict by taking a weighted majority vote

Online Boosting: Motivation

- Boosting is well studied in the batch-setting, but infeasible when the amount of data is huge.
- online learning comes to rescue:
  - memory efficient,
  - works even in an adversarial environment

Approaches : Learning from Examples

- Learning from examples is different from using Expert advice, as we don't need to previously define rebuild experts we will derive our predictions from.

- We need , however, to know what Concept Class we want to search over.
- A concept class is a set of functions (concepts) that subscribe to a particular model.

- Concept classes :
  - The set of all monotone disjunction of N variables
  - the set of non monotone disjunction of N variables
  - Decision list with N variable
  - Linear threshold formulas
  - DNF(Disjunctive normal form) formula

Approaches : Learning from Examples

- The Winnow Algorithm is one simple example that leads monotone disjunctions online.
- it learns any concept(function), provided the concept belongs to concept class of monotone disjunctions.
- it also uses weights

Winnow Algorithm

$\Theta \leftarrow \frac{d}{2}$
$w \leftarrow (1, 1, \ldots, 1)$
for $i = 1, 2, \ldots$ **do**
  if $w^\top x^{(i)} > \Theta$ **then**
    $c_*(x^{(i)}) = 1$
  else
    $c_*(x^{(i)}) = 0$
  end if
  if $c_*(x^{(i)}) \neq c(x^{(i)})$ **then**
    if $c(x^{(i)}) = 1$ **then**
      $\forall j : x_j^{(i)} = 1, w_j \leftarrow 2w_j$
    else
      $\forall j : x_j^{(i)} = 1, w_j \leftarrow 0$
    end if
  end if
end for

- *Move* the hyperplane when a mistake is made
- Θ is often set to $\frac{d}{2}$
- *Promotions* and *eliminations*

- Winnow makes $O(k \log_2 d)$ mistakes
- Optimal mistake bound
- One can use different values for Θ
- Other variants exist
  - *Arbitrary* disjunctions
  - k-DNF (disjunctive normal forms)
    - $(x_1 \wedge x_2) \vee (x_4) \vee (x_7 \wedge \neg x_3)$

Approaches : Other Approaches

- More general algorithms can be used in an online setting, such as :
  - Stochastic Gradient Descent
  - perceptron learning algorithm

Stochastic gradient ascent can be used for online learning!!!

- init $\mathbf{w}^{(1)}=0$, t=1
- Each time step t:
  - Observe input x_t
  - Make a prediction $\hat{y}_t$
  - Observe true output y_t
  - Update coefficients:

for j=0,…,D

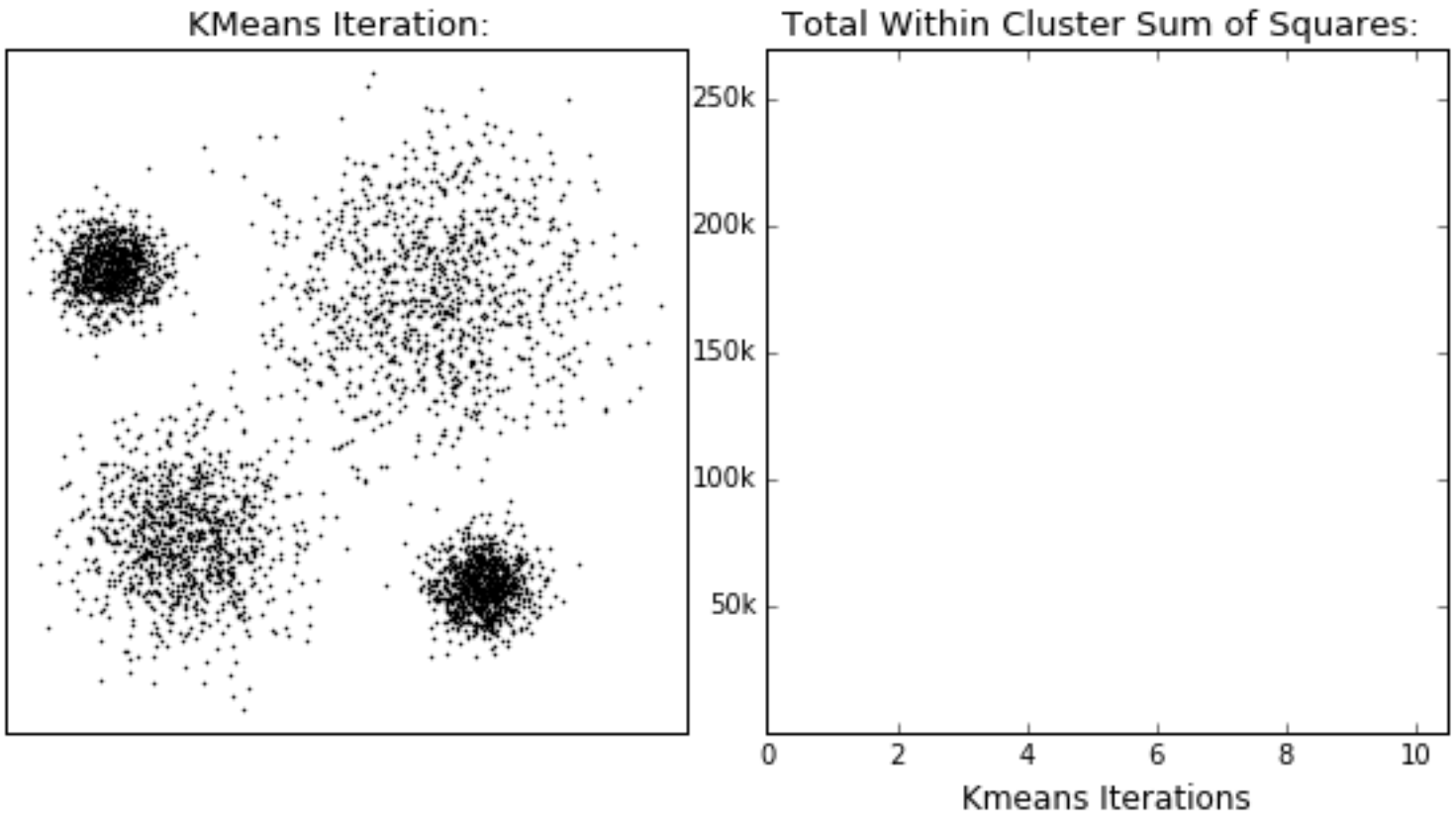
$$w_j^{(t+1)} \leftarrow w_j^{(t)} + \eta \frac{\partial \ell_t(\mathbf{w})}{\partial \mathbf{w}_j}$$

Perceptron Algorithm

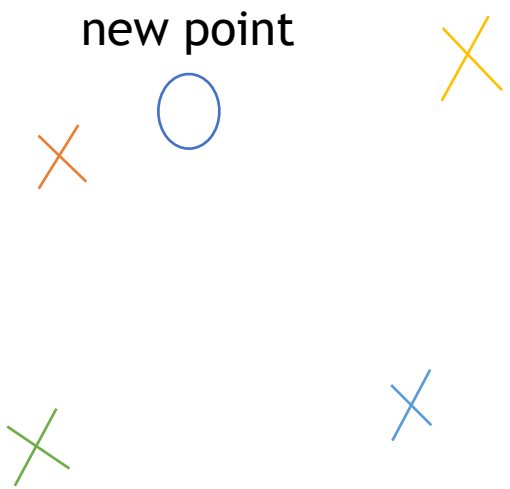
The Perceptron Algorithm:

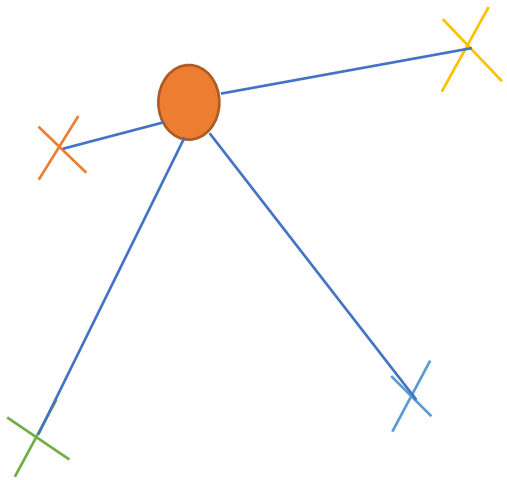
1. Start with the all-zeroes weight vector $\mathbf{w}_1 = \mathbf{0}$, and initialize t to 1.
2. Given example $\mathbf{x}_t$, predict positive iff $\mathbf{w}_t \cdot \mathbf{x}_t > 0$.
3. On a mistake, update as follows:
   - Mistake on positive: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \mathbf{x}_t$.
   - Mistake on negative: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \mathbf{x}_t$.

K-Mean Algorithm

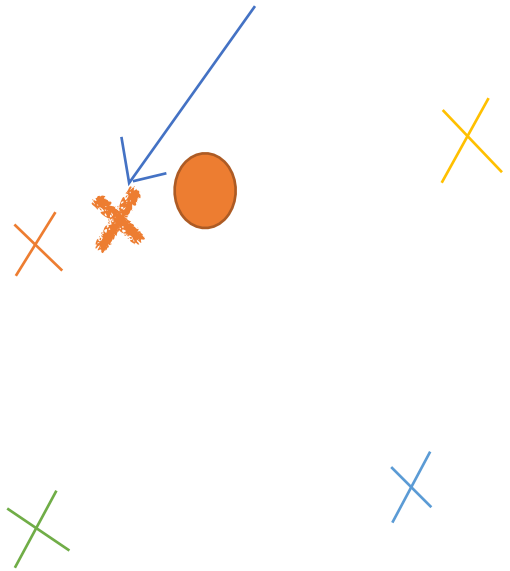


Online K-Mean Algorithm


new point



update orange centre



Online K-Mean

- Make initial guesses for the means $\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_k$
- Set the counts $n_1, n_2, ..., n_k$ to zero
- Until interrupted
  - Acquire the next example, $\mathbf{x}$
  - If $\mathbf{m}_i$ is closest to $\mathbf{x}$
    - Increment n_i
    - Replace $\mathbf{m}_i$ by $\mathbf{m}_i + (1/n_i)*(\mathbf{x} - \mathbf{m}_i)$
  - end_if
- end_until

Regret

- Goal : Low regret
- the main objective of online learning is to minimize the regret.
- We'd like to find expert in an online Manner

$$\sum_{t=1}^{T} \ell(\hat{y}_t, y_t) - \min_i \sum_{t=1}^{T} \ell(x_{t,i}, y_t)$$

- Regret : difference between the performance of :
  - online algorithm,
  - an ideal algorithm that has been able to train on the whole dataset that have been seen so far, in batch fashion

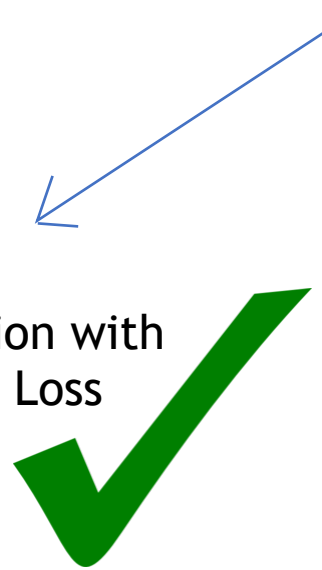
- In other words, the main objective of an online machine learning is to perform as closely to the corresponding offline algorithm as possible.
- this is measured by the regret.

What Prediction Tasks are possible

Regression with
squared Loss

Classification with
0-1 Loss

What Prediction Tasks are possible



Regression with
squared Loss

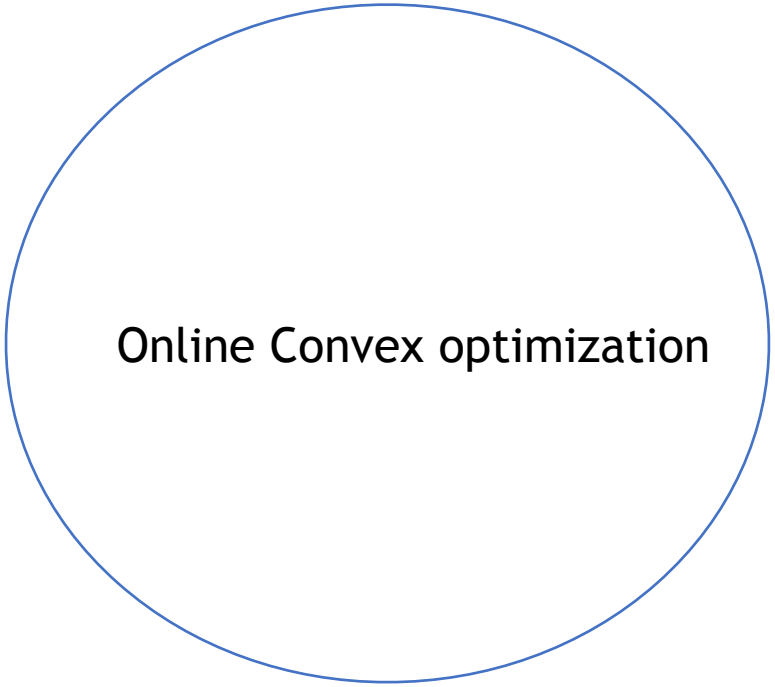
Classification with
0-1 Loss

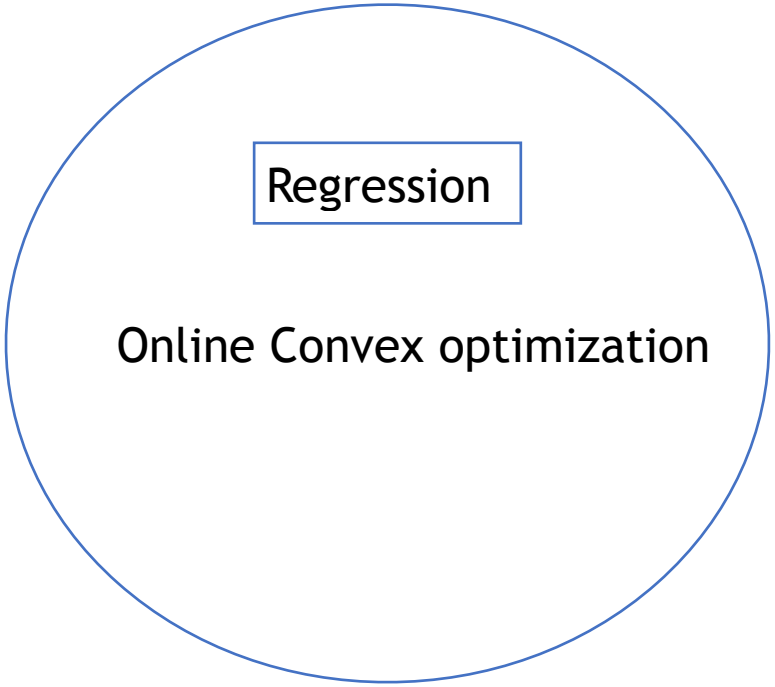
What Prediction Tasks are possible

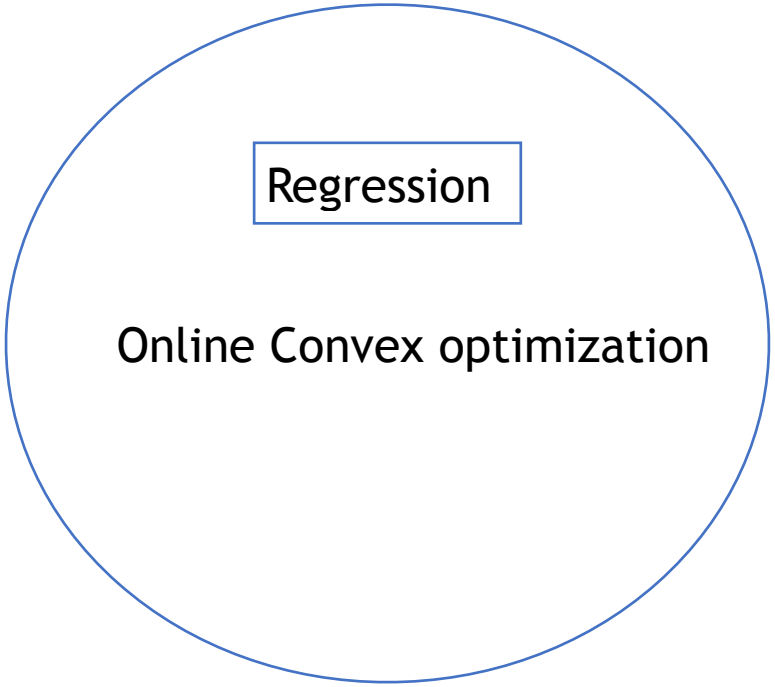
Regression with squared Loss

Classification with 0-1 Loss

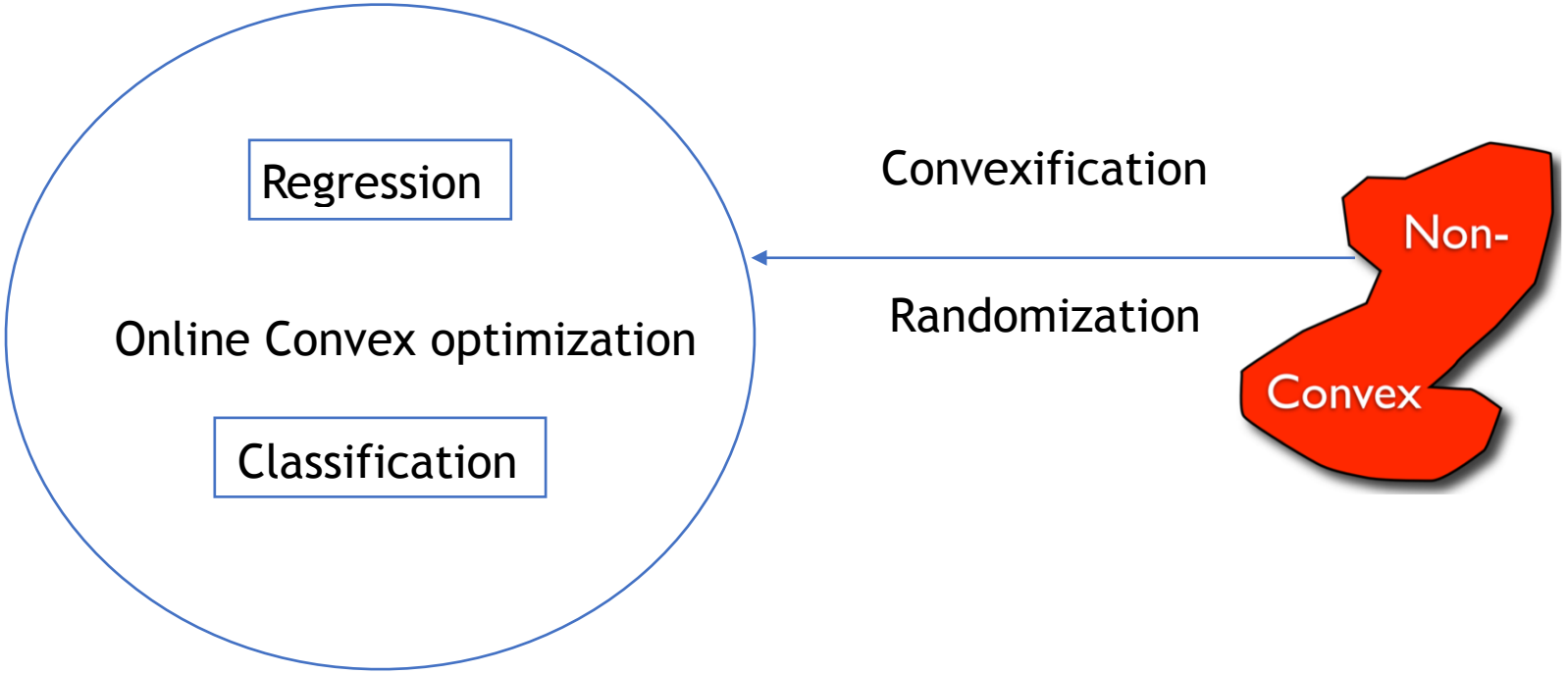
Convexity is the key property


Online Convex optimization


Regression
Online Convex optimization


Regression
Online Convex optimization


Non-
Convex


Regression
Online Convex optimization
Classification
Convexification
Randomization
Non-
Convex

Coping up with Non-Convex Loss function

- Method I : Convexification
  - Find a surrogate loss function

- Method II : Randomization
  - Allow randomized prediction
  - Analyzed expected regret
  - Loss in expectation is convex

Convexification

- Non-Convex loss : 0-1 loss

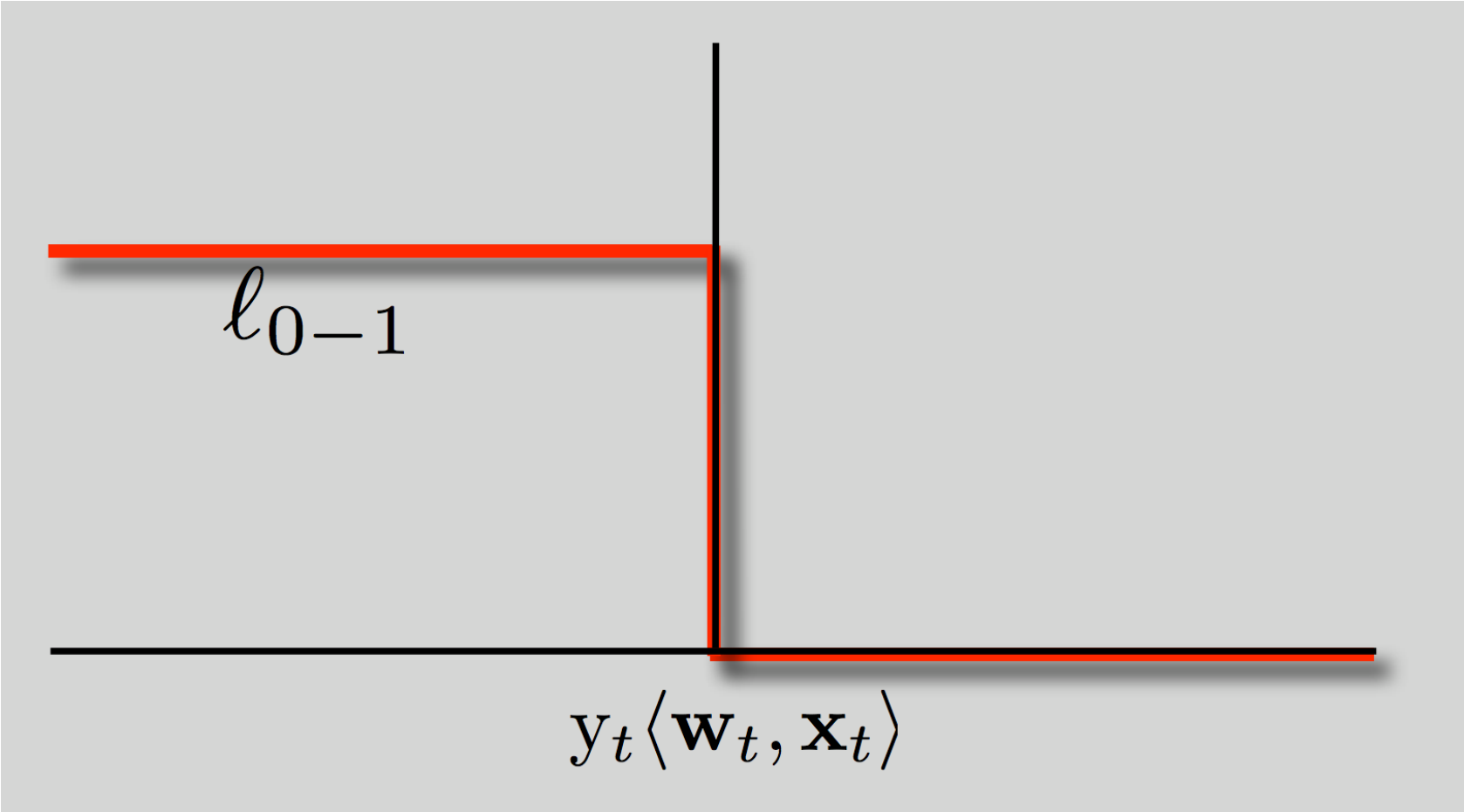
$$\ell_{0-1}(\hat{y}_t, y_t) = \begin{cases} 1 & \text{if } \; y_t \neq \hat{y}_t \\ 0 & \text{otherwise} \end{cases}$$

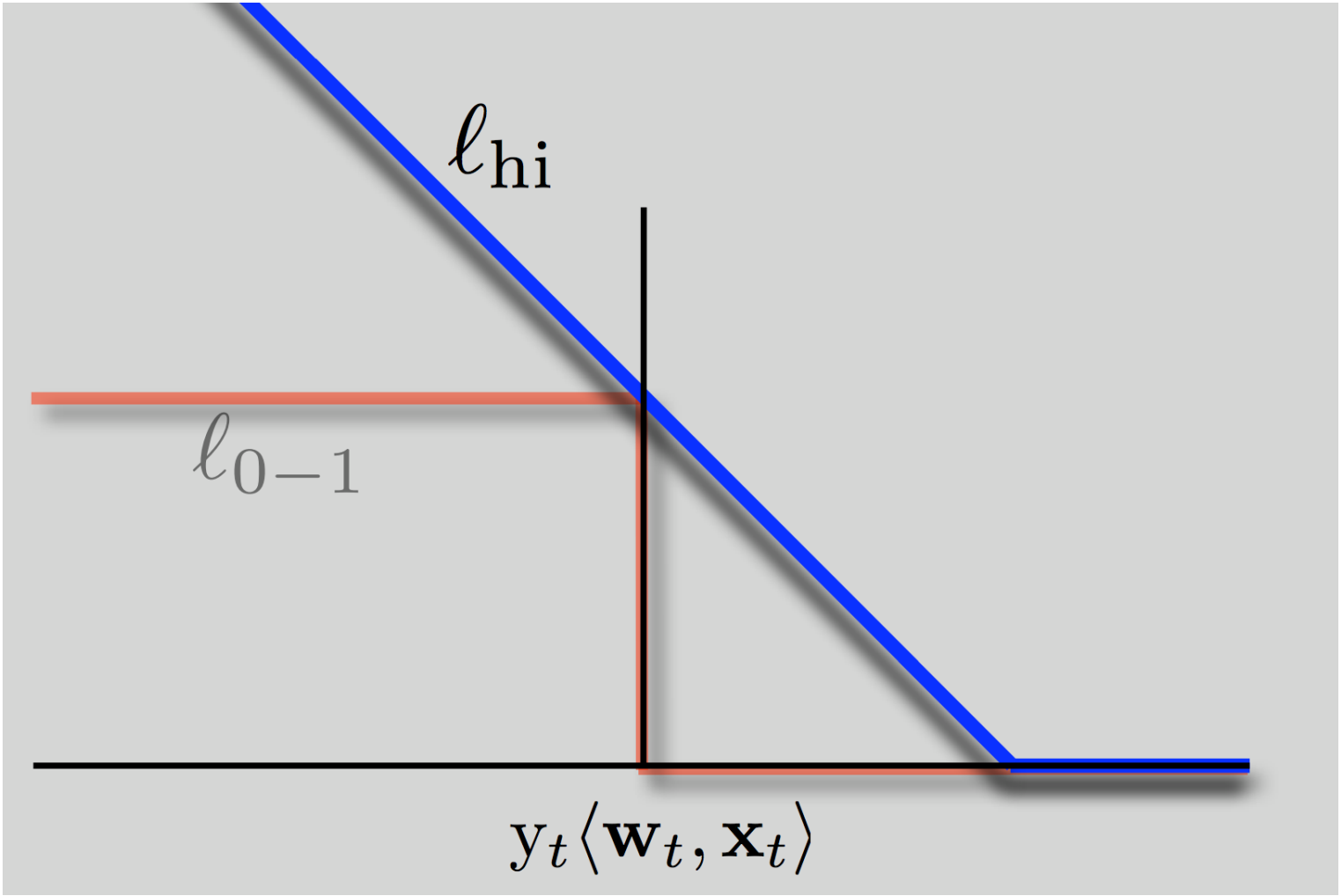
- Surrogate loss function: hinge loss

$$\ell_{\text{hi}}(\mathbf{w}, (\mathbf{x}_t y_t)) = [1 - y_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle]_+$$

where

$$[a]_+ = \max\{a, 0\}$$


ℓ_{0-1}
$y_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle$


ℓ_{hi}
ℓ_{0-1}
$y_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle$

Randomization

- d given experts
- at each round t
- $p_t \in \{1, \ldots, d\}$ be the chosen expert
- Learner receives a vector $\mathbf{y}_t \in [0,1]^d$, , where $y_t[i]$ is the cost of following the advice of i th expert.
- The learner pays the loss $y_t[p_t]$.

- Let $S = \{\mathbf{w} \in \mathbb{R}^d : \mathbf{w} \geq 0 \wedge \|\mathbf{w}\|_1 = 1\}$ probability simplex, which is Convex.
- At round t, learner chooses $\mathbf{w}_t \in S$, based on picks an expert at random according to $\mathbb{P}[p_t = i] = w_t[i]$
- Then cost y is revealed and the learner pays for his expected cost

$$\mathbb{E}[y_t[p_t]] = \sum_{i=1}^{d} \mathbb{P}[p_t = i] y_t[i] = \langle \mathbf{w}_t, \mathbf{y}_t \rangle.$$

Current Trends/Related Areas

- Recommendation System
- Finance
- Adversarial Machine Learning
    - Information Security
    - Games
- one shot Learning - Refers to scenarios where you must perform prediction after seeing just few,
    - Computer Vision:
        - automatic object detection and tracking
        - Online and adaptive event detection
        - Applications using online classification methods

Summary

- Why online learning :
  - fast
  - memory efficient
  - simple to implement
  - No statistical assumption
  - Adaptive

References

- Online Learning and Online Convex Optimization By Shai Shalev-Shwartz
- Prediction, learning, and games Book by Nicolò Cesa-Bianchi
- N. Littlestone. Learning Quickly When Irrelevant Attributes Abound: A New Linearthreshold Algorithm. Machine Learning, 2(1):285-318, 1988.
- Lecture notes on Online Learning, Alexander Raklin, 2009
- Wikipedia
- Scikit-Learn

Thanks!