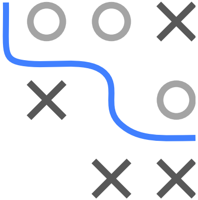
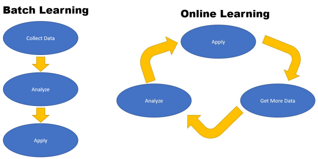
Advanced Machine Learning

Introduction to Online Learning

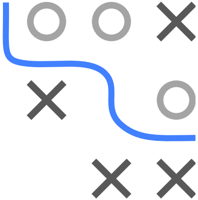




Learning goals

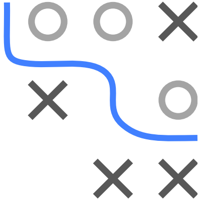
- Understand the difference between batch and online learning
- Know the basic and the extended learning protocol in online learning
- Know how performance is measured in online learning

BATCH LEARNING



- The conventional machine learning is rooted in the *statistical learning theory* and is sometimes referred to as the *batch learning scenario*:
  - A data set $\mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n}$ is given beforehand in form of a random sample (iid observations).

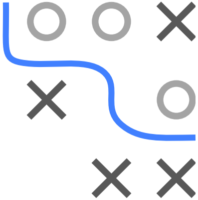
BATCH LEARNING



- The conventional machine learning is rooted in the *statistical learning theory* and is sometimes referred to as the *batch learning scenario*:
  - A data set $\mathcal{D} = \left\{(\mathbf{x}^{(i)}, y^{(i)})\right\}_{i=1}^{n}$ is given beforehand in form of a random sample (iid observations).

  $\rightsquigarrow$ a *batch* of data
  - The goal is to learn a single predictor (model), i.e., a mapping $f : \mathcal{X} \to \mathcal{Y}$ that will have a good prediction accuracy (low risk) on future, unseen data in $\mathcal{X} \times \mathcal{Y}$.

BATCH LEARNING

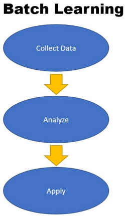


- The conventional machine learning is rooted in the *statistical learning theory* and is sometimes referred to as the *batch learning scenario*:

  - A data set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$ is given beforehand in form of a random sample (iid observations).

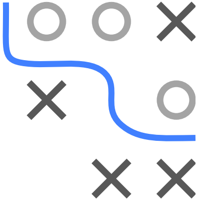
  $\rightsquigarrow$ a *batch* of data

  - The goal is to learn a single predictor (model), i.e., a mapping $f : \mathcal{X} \to \mathcal{Y}$ that will have a good prediction accuracy (low risk) on future, unseen data in $\mathcal{X} \times \mathcal{Y}$.

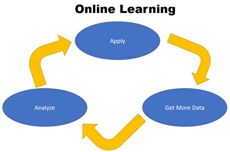


- The learning task on the available data beforehand is called the *training phase* and the prediction on the unseen data is called the *testing phase.* Both phases are **separated**.

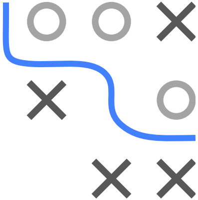
ONLINE LEARNING



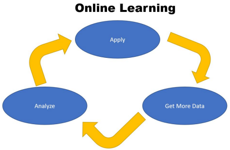
- However, many real-world problems are *dynamic* with the following aspects:
  - *Sequential order* — data is generated only bit by bit;
  - *On-the-fly decisions* — decisions or predictions have to be made during the data generating process;
  - *Unforeseeable consequences* — decisions can have a drastic influence on the data generating process;
  - *Constraints* — there is a specific time limit or computational limit for the decision.



ONLINE LEARNING

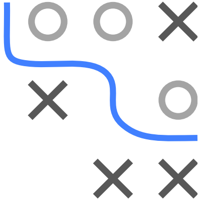


- However, many real-world problems are *dynamic* with the following aspects:
  - *Sequential order* — data is generated only bit by bit;
  - *On-the-fly decisions* — decisions or predictions have to be made during the data generating process;
  - *Unforeseeable consequences* — decisions can have a drastic influence on the data generating process;
  - *Constraints* — there is a specific time limit or computational limit for the decision.



- These dynamic aspects outline the framework where **online learning** is settled.
- Characteristically: In the online learning scenario the training phase and the testing phase are **interleaved**.

ONLINE LEARNING: EXAMPLES



There are many real-world applications which fit into the online learning scenario:

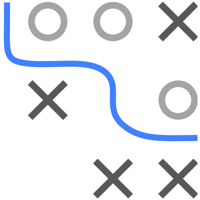
- *Weather Forecasting* — Observe meteorological data as data streams by satellites for instance and keep the current weather prediction up to date.

ONLINE LEARNING: EXAMPLES

There are many real-world applications which fit into the online learning scenario:

- *Weather Forecasting* — Observe meteorological data as data streams by satellites for instance and keep the current weather prediction up to date.
- *Sequential Investment* — A bank has to allocate its total capital on the financial market, where asset prices are varying over time.

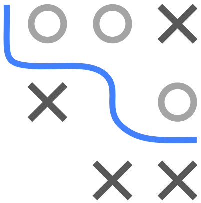
ONLINE LEARNING: EXAMPLES



There are many real-world applications which fit into the online learning scenario:

- *Weather Forecasting* — Observe meteorological data as data streams by satellites for instance and keep the current weather prediction up to date.
- *Sequential Investment* — A bank has to allocate its total capital on the financial market, where asset prices are varying over time.
- *Navigation systems* — Find the best path from A to B given the current traffic situation.

ONLINE LEARNING: EXAMPLES

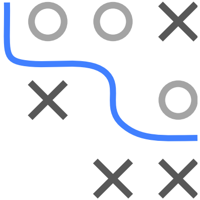


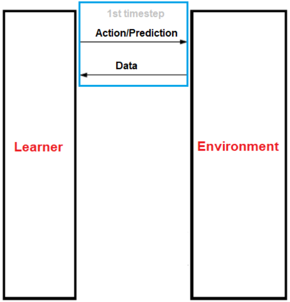
There are many real-world applications which fit into the online learning scenario:

- *Weather Forecasting* — Observe meteorological data as data streams by satellites for instance and keep the current weather prediction up to date.
- *Sequential Investment* — A bank has to allocate its total capital on the financial market, where asset prices are varying over time.
- *Navigation systems* — Find the best path from A to B given the current traffic situation.
- *Autonomous driving systems* — Steer the automotive, while constantly monitoring the environment and react appropriately to any changes.
- . . .

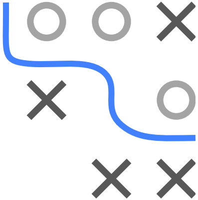
ONLINE LEARNING: ILLUSTRATION

The data is available only in a **sequential order** generated by the environment and the learner's actions/predictions have to be made **on-the-fly.**

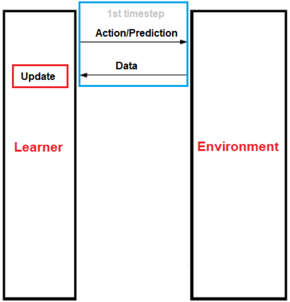




ONLINE LEARNING: ILLUSTRATION

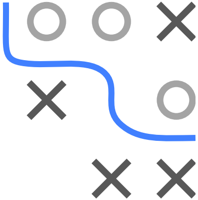


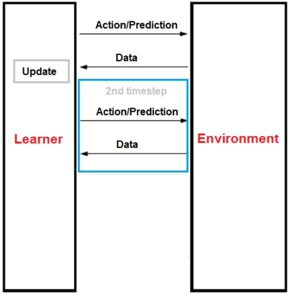
The data is available only in a **sequential order** generated by the environment and the learner's actions/predictions have to be made **on-the-fly.**
$\Rightarrow$ Learning algorithms have to be dynamically adapted (Update of internal model).



ONLINE LEARNING: ILLUSTRATION

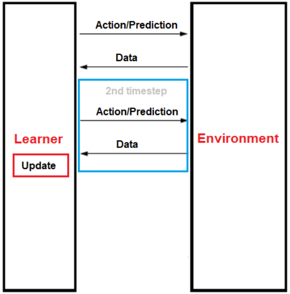
The data is available only in a **sequential order** generated by the environment and the learner's actions/predictions have to be made **on-the-fly.**
$\Rightarrow$ Learning algorithms have to be dynamically adapted (Update of internal model).



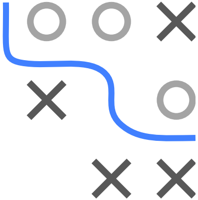


ONLINE LEARNING: ILLUSTRATION

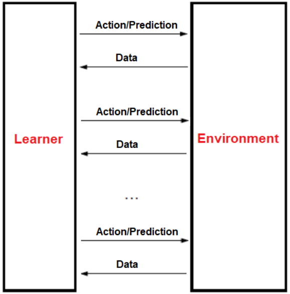
The data is available only in a **sequential order** generated by the environment and the learner's actions/predictions have to be made **on-the-fly.**
$\Rightarrow$ Learning algorithms have to be dynamically adapted (Update of internal model).



ONLINE LEARNING: ILLUSTRATION

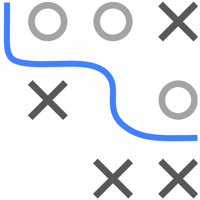


The data is available only in a **sequential order** generated by the environment and the learner's actions/predictions have to be made **on-the-fly.**
$\Rightarrow$ Learning algorithms have to be dynamically adapted (Update of internal model).



$\Rightarrow$ The learner and the environment are alternately performing their actions.

THE BASIC ONLINE LEARNING PROTOCOL



Formally, an online learning problem consists of:

- a learner (forecaster, agent resp. decision maker), an environment (user resp. adversary, system resp. nature,),
- time steps $1, 2, \dots, T$ (may be infinite),
- available actions $\mathcal{A}$ for the learner (may be infinite),
- environmental data space $\mathcal{Z}$,
- a loss function $L : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$.

THE BASIC ONLINE LEARNING PROTOCOL

Formally, an online learning problem consists of:

- a learner (forecaster, agent resp. decision maker), an environment (user resp. adversary, system resp. nature,),
- time steps $1, 2, \ldots, T$ (may be infinite),
- available actions $\mathcal{A}$ for the learner (may be infinite),
- environmental data space $\mathcal{Z}$,
- a loss function $L : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$.

Mechanism: In each time step t

- learner chooses an action $a_t \in \mathcal{A}$,
- environment generates data $z_t \in \mathcal{Z}$,
- learner observes the environmental data and suffers loss $L(a_t, z_t)$,
- learner update its model/ knowledge basis.

THE BASIC ONLINE LEARNING PROTOCOL

Formally, an online learning problem consists of:

- a learner (forecaster, agent resp. decision maker), an environment (user resp. adversary, system resp. nature,),
- time steps $1, 2, \ldots, T$ (may be infinite),
- available actions $\mathcal{A}$ for the learner (may be infinite),
- environmental data space $\mathcal{Z}$,
- a loss function $L : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$.

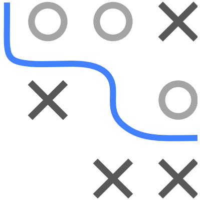
Mechanism: In each time step t

- learner chooses an action $a_t \in \mathcal{A}$,
- environment generates data $z_t \in \mathcal{Z}$,
- learner observes the environmental data and suffers loss $L(a_t, z_t)$,
- learner update its model/ knowledge basis.

Typically $\mathcal{A} = \mathcal{Z} = \mathcal{Y}$, so that

- the learner's chosen action $a_t = \hat{y}_t$ corresponds to a prediction,
- the generated data point $z_t = y_t$ is the revealed outcome.

THE EXTENDED ONLINE LEARNING PROTOCOL



- In some applications, the environmental data consists of two parts: $z_t = (z_t^{(1)}, z_t^{(2)})$, where the first part of the data, $z_t^{(1)}$, is revealed to the learner **before** the action is made. After the learner carries out its action, the remaining part of the environmental data is revealed, that is, $z_t^{(2)}$.

THE EXTENDED ONLINE LEARNING PROTOCOL

- In some applications, the environmental data consists of two parts: $z_t = (z_t^{(1)}, z_t^{(2)})$, where the first part of the data, $z_t^{(1)}$, is revealed to the learner **before** the action is made. After the learner carries out its action, the remaining part of the environmental data is revealed, that is, $z_t^{(2)}$.
- The **mechanism** in such an online learning problem is then as follows: In each time step t
  - the environment generates data $z_t = (z_t^{(1)}, z_t^{(2)}) \in \mathcal{Z}$,
  - the learner observes the first part of the environmental data $z_t^{(1)}$,
  - the learner chooses an action $a_t \in \mathcal{A}$,
  - the learner observes the remaining part of the environmental data $z_t^{(2)}$ and suffers loss $L(a_t, z_t)$,
  - the learner updates its knowledge base.

THE EXTENDED ONLINE LEARNING PROTOCOL

- In some applications, the environmental data consists of two parts: $z_t = (z_t^{(1)}, z_t^{(2)})$, where the first part of the data, $z_t^{(1)}$, is revealed to the learner **before** the action is made. After the learner carries out its action, the remaining part of the environmental data is revealed, that is, $z_t^{(2)}$.
- The **mechanism** in such an online learning problem is then as follows: In each time step t
  - the environment generates data $z_t = (z_t^{(1)}, z_t^{(2)}) \in \mathcal{Z}$,
  - the learner observes the first part of the environmental data $z_t^{(1)}$,
  - the learner chooses an action $a_t \in \mathcal{A}$,
  - the learner observes the remaining part of the environmental data $z_t^{(2)}$ and suffers loss $L(a_t, z_t)$,
  - the learner updates its knowledge base.
- Apparently, the learner can take the a priori information in form of $z_t^{(1)}$ at each time step t into account when choosing its action.

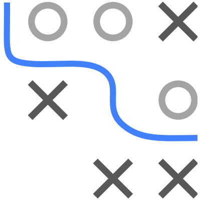
THE EXTENDED ONLINE LEARNING PROTOCOL

- In some applications, the environmental data consists of two parts: $z_t = (z_t^{(1)}, z_t^{(2)})$, where the first part of the data, $z_t^{(1)}$, is revealed to the learner **before** the action is made. After the learner carries out its action, the remaining part of the environmental data is revealed, that is, $z_t^{(2)}$.
- The **mechanism** in such an online learning problem is then as follows: In each time step t
  - the environment generates data $z_t = (z_t^{(1)}, z_t^{(2)}) \in \mathcal{Z}$,
  - the learner observes the first part of the environmental data $z_t^{(1)}$,
  - the learner chooses an action $a_t \in \mathcal{A}$,
  - the learner observes the remaining part of the environmental data $z_t^{(2)}$ and suffers loss $L(a_t, z_t)$,
  - the learner updates its knowledge base.
- Apparently, the learner can take the a priori information in form of $z_t^{(1)}$ at each time step t into account when choosing its action.
- We call this setting the *extended online learning protocol.*

THE EXTENDED ONLINE LEARNING PROTOCOL

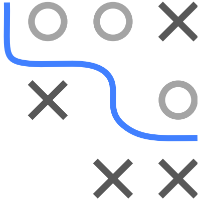
- In some applications, the environmental data consists of two parts: $z_t = (z_t^{(1)}, z_t^{(2)})$, where the first part of the data, $z_t^{(1)}$, is revealed to the learner **before** the action is made. After the learner carries out its action, the remaining part of the environmental data is revealed, that is, $z_t^{(2)}$.
- The **mechanism** in such an online learning problem is then as follows: In each time step t
  - the environment generates data $z_t = (z_t^{(1)}, z_t^{(2)}) \in \mathcal{Z}$,
  - the learner observes the first part of the environmental data $z_t^{(1)}$,
  - the learner chooses an action $a_t \in \mathcal{A}$,
  - the learner observes the remaining part of the environmental data $z_t^{(2)}$ and suffers loss $L(a_t, z_t)$,
  - the learner updates its knowledge base.
- Apparently, the learner can take the a priori information in form of $z_t^{(1)}$ at each time step t into account when choosing its action.
- We call this setting the *extended online learning protocol.*
- Typically $\mathcal{A} = \mathcal{Y}$ and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, so that
  - the first part $z_t^{(1)} = \mathbf{x}_t$ is some feature information,
  - the learner's chosen action $a_t = \hat{y}_t$ corresponds to a prediction (dep. on $\mathbf{x}_t$),
  - the second part $z_t^{(2)} = y_t$ is the corresponding outcome.

DATA GENERATION IN ONLINE LEARNING



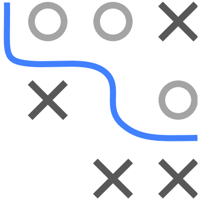
- Typically for the online learning setting is that **no** statistical assumptions is made on how the sequence of environmental data is generated.
- In particular, the environmental data are not necessarily generated by a probability distribution!

DATA GENERATION IN ONLINE LEARNING



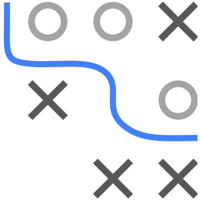
- Typically for the online learning setting is that **no** statistical assumptions is made on how the sequence of environmental data is generated.
- In particular, the environmental data are not necessarily generated by a probability distribution!
- This also covers the area of *adversarial learning*: the data can even be generated by an adversary trying to fool the learner.
- However, the online learning setting can of course also be considered in a statistical setting.

ONLINE LEARNING: REQUIREMENTS



- The dynamical aspects have to be incorporated for the design of efficient learning algorithms.
- The online learner has to cope with the sequential availability of the data and to cope with time as well as computational constraints.
- Roughly speaking, one seeks to construct an online learning algorithm which is adaptive to the environment and allows incremental as well as preferably cheap updates over time.

ONLINE LEARNING: REQUIREMENTS



- The dynamical aspects have to be incorporated for the design of efficient learning algorithms.
- The online learner has to cope with the sequential availability of the data and to cope with time as well as computational constraints.
- Roughly speaking, one seeks to construct an online learning algorithm which is adaptive to the environment and allows incremental as well as preferably cheap updates over time.
- Although consideration of time and memory constraints is important for practical purposes, we will only implicitly consider these constraints in this lecture.
- We will mainly focus our theoretical analysis on the performance of the learner in terms of its (cumulative) loss, which, however, will usually ignore computational aspects of the learner.

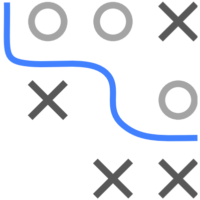
MEASURE OF QUALITY IN ONLINE LEARNING

- In order to measure the quality of an online learner one can compute the difference between the cumulative loss of the learner and the cumulative loss by taking some competing action $a \in \mathcal{A}$:

$$R_T(a) = \sum_{t=1}^{T} L(a_t, z_t) - \sum_{t=1}^{T} L(a, z_t).$$

- This value is called the *(cumulative) regret of a learner* with respect to an action $a \in \mathcal{A}$.

MEASURE OF QUALITY IN ONLINE LEARNING



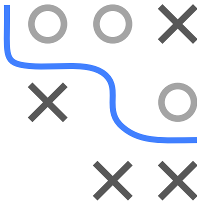
- In order to measure the quality of an online learner one can compute the difference between the cumulative loss of the learner and the cumulative loss by taking some competing action $a \in \mathcal{A}$:

$$R_T(a) = \sum_{t=1}^{T} L(a_t, z_t) - \sum_{t=1}^{T} L(a, z_t).$$

- This value is called the *(cumulative) regret of a learner* with respect to an action $a \in \mathcal{A}$.

- Here,
  - $\sum_{t=1}^{T} L(a_t, z_t)$ is the *cumulative loss of the learner*,
  - $\sum_{t=1}^{T} L(a, z_t)$ is the cumulative loss of the competing action a.

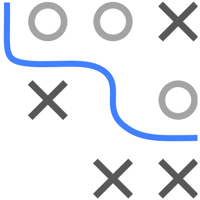
MEASURE OF QUALITY IN ONLINE LEARNING



- It seems natural to compare the incurred cumulative loss of the learner with the *best action(s) in hindsight*:

$$R_T = \sum_{t=1}^{T} L(a_t, z_t) - \inf_{a \in \mathcal{A}} \sum_{t=1}^{T} L(a, z_t).$$

MEASURE OF QUALITY IN ONLINE LEARNING

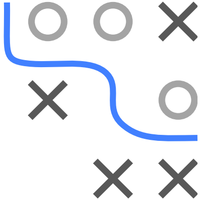


- It seems natural to compare the incurred cumulative loss of the learner with the *best action(s) in hindsight*:

$$R_T = \sum_{t=1}^{T} L(a_t, z_t) - \inf_{a \in \mathcal{A}} \sum_{t=1}^{T} L(a, z_t).$$

- Here,
  - $\sum_{t=1}^{T} L(a_t, z_t)$ is the *cumulative loss of the learner*,
  - $\inf_{a \in \mathcal{A}} \sum_{t=1}^{T} L(a, z_t)$ is the cumulative loss of the *best action(s) in hindsight*.

MEASURE OF QUALITY IN ONLINE LEARNING

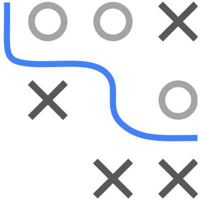


- It seems natural to compare the incurred cumulative loss of the learner with the *best action(s) in hindsight*:

$$R_T = \sum_{t=1}^{T} L(a_t, z_t) - \inf_{a \in \mathcal{A}} \sum_{t=1}^{T} L(a, z_t).$$

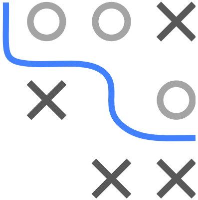
- Here,
  - $\sum_{t=1}^{T} L(a_t, z_t)$ is the *cumulative loss of the learner*,
  - $\inf_{a \in \mathcal{A}} \sum_{t=1}^{T} L(a, z_t)$ is the cumulative loss of the *best action(s) in hindsight*.
- We refer to R_T as the *(cumulative) regret* of the online learner. It is easy to see that $R_T = \sup_{a \in \mathcal{A}} R_T(a)$.

MEASURE OF QUALITY IN ONLINE LEARNING



- The objective of the online learner is to minimize the cumulative regret R_T.

MEASURE OF QUALITY IN ONLINE LEARNING



- The objective of the online learner is to minimize the cumulative regret R_T.
- Note that the cumulative regret can be in principle negative as the action sequence could be such that $L(a_s, z_s) < L(a^*, z_s)$ holds for specific time steps s, where $a^* \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{T} L(a, z_s)$ is one of the best actions in hindsight (may be unique).

MEASURE OF QUALITY IN ONLINE LEARNING

- The objective of the online learner is to minimize the cumulative regret R_T.
- Note that the cumulative regret can be in principle negative as the action sequence could be such that $L(a_s, z_s) < L(a^*, z_s)$ holds for specific time steps s, where $a^* \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{T} L(a, z_s)$ is one of the best actions in hindsight (may be unique).
- If the cumulative regret is always non-negative (which will be usually the case), then the overall goal of an online learner is to have a regret which is *sublinear* in the time horizon T.

MEASURE OF QUALITY IN ONLINE LEARNING

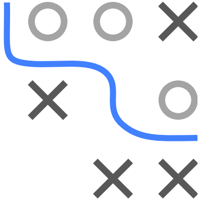
- The objective of the online learner is to minimize the cumulative regret R_T.
- Note that the cumulative regret can be in principle negative as the action sequence could be such that $L(a_s, z_s) < L(a^*, z_s)$ holds for specific time steps s, where $a^* \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{T} L(a, z_s)$ is one of the best actions in hindsight (may be unique).
- If the cumulative regret is always non-negative (which will be usually the case), then the overall goal of an online learner is to have a regret which is *sublinear* in the time horizon T.
- Formally, the following should hold

$$R_T = o(T).$$

Interpretation: The average regret per time step (or per example) goes to zero:

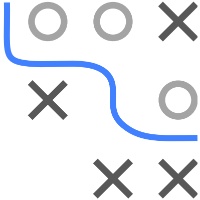
$$\frac{1}{T}\left(\sum_{t=1}^{T} L(a_t, z_t) - \inf_{a \in \mathcal{A}} \sum_{t=1}^{T} L(a, z_t)\right) = \frac{R_T}{T} = o(1).$$

DYNAMIC REGRET



- One might ask why one compares only with a fixed best action in hindsight, say a^*, instead of a sequence of actions $a_1^*, a_2^*, \dots, a_T^*$?

DYNAMIC REGRET



- One might ask why one compares only with a fixed best action in hindsight, say a^*, instead of a sequence of actions $a_1^*, a_2^*, \ldots, a_T^*$?
- The rationale behind this measure of quality is that the best fixed action in hindsight is already reasonably good over all the time steps: it performs almost as well as a batch learner that observes the entire sequence and picks the best action in hindsight.

DYNAMIC REGRET

- One might ask why one compares only with a fixed best action in hindsight, say a^*, instead of a sequence of actions $a_1^*, a_2^*, \ldots, a_T^*$?
- The rationale behind this measure of quality is that the best fixed action in hindsight is already reasonably good over all the time steps: it performs almost as well as a batch learner that observes the entire sequence and picks the best action in hindsight.
- However, this is too optimistic and may not hold in changing environments, where data are evolving and the optimal action is drifting over the time.

DYNAMIC REGRET

- One might ask why one compares only with a fixed best action in hindsight, say a^*, instead of a sequence of actions $a_1^*, a_2^*, \dots, a_T^*$?
- The rationale behind this measure of quality is that the best fixed action in hindsight is already reasonably good over all the time steps: it performs almost as well as a batch learner that observes the entire sequence and picks the best action in hindsight.
- However, this is too optimistic and may not hold in changing environments, where data are evolving and the optimal action is drifting over the time.
- To address this limitation, recent works have also considered the *dynamic regret*:

$$R_T^D(a_1^*, a_2^*, \dots, a_T^*) = \sum\nolimits_{t=1}^{T} L(a_t, z_t) - \sum\nolimits_{t=1}^{T} L(a_t^*, z_t).$$

DYNAMIC REGRET

- One might ask why one compares only with a fixed best action in hindsight, say a^*, instead of a sequence of actions $a_1^*, a_2^*, \ldots, a_T^*$?
- The rationale behind this measure of quality is that the best fixed action in hindsight is already reasonably good over all the time steps: it performs almost as well as a batch learner that observes the entire sequence and picks the best action in hindsight.
- However, this is too optimistic and may not hold in changing environments, where data are evolving and the optimal action is drifting over the time.
- To address this limitation, recent works have also considered the *dynamic regret*:

$$R_T^D(a_1^*, a_2^*, \ldots, a_T^*) = \sum_{t=1}^{T} L(a_t, z_t) - \sum_{t=1}^{T} L(a_t^*, z_t).$$

- We will cover only the static regret in this lecture.