Changing the distribution to **better** learn in an I.I.D. setting

Semi-supervised learning

Imbalanced data sets

Learning from **positive examples** only

Active Learning

LUPI (**L**earning **U**sing **P**rivileged **I**nformation)

Antoine Cornuéjols

AgroParisTech – INRAE MIA Paris-Saclay

EKINOCS research group

AgroParisTech

INRAE
la science pour la vie, l'humain, la terre

- The learning task is I.I.D. (In Distribution learning)
- But **we make it** O.O.D. to **help** solve it!

P_X(train) = P_X(test) but we treat it as **OOD learning pb**

- In which scenarios?

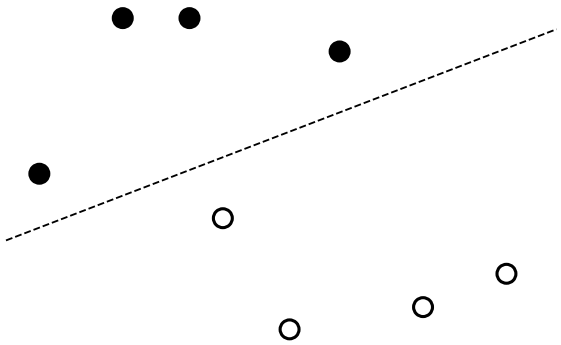
$P_X(\text{train}) \neq P_X(\text{test})$

In which scenarios?

Outline

1. Semi-supervised learning
2. Classes severely unbalanced
3. Learning from positive examples only
4. Active learning
5. LUPI (Learning Using Privileged Information)

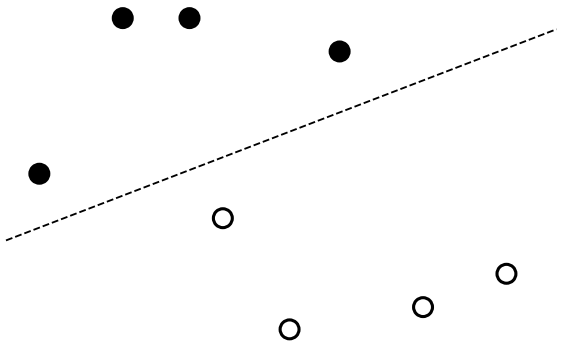
Illustration



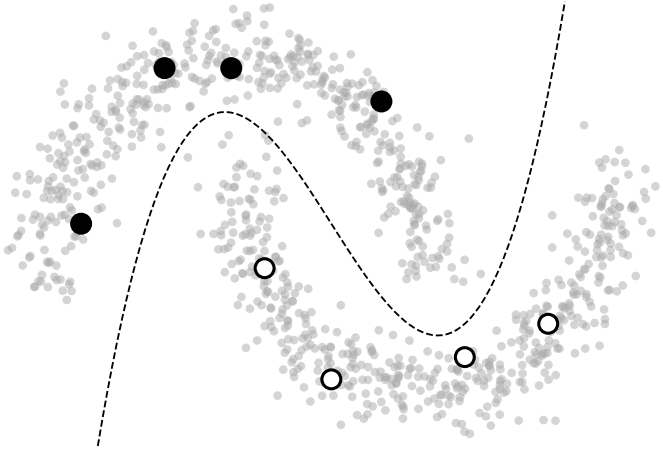
Labeled examples only

From [Kevin Murphy. **Probabilistic Machine Learning: An introduction**. MIT Press, 2022, p.634]

Illustration



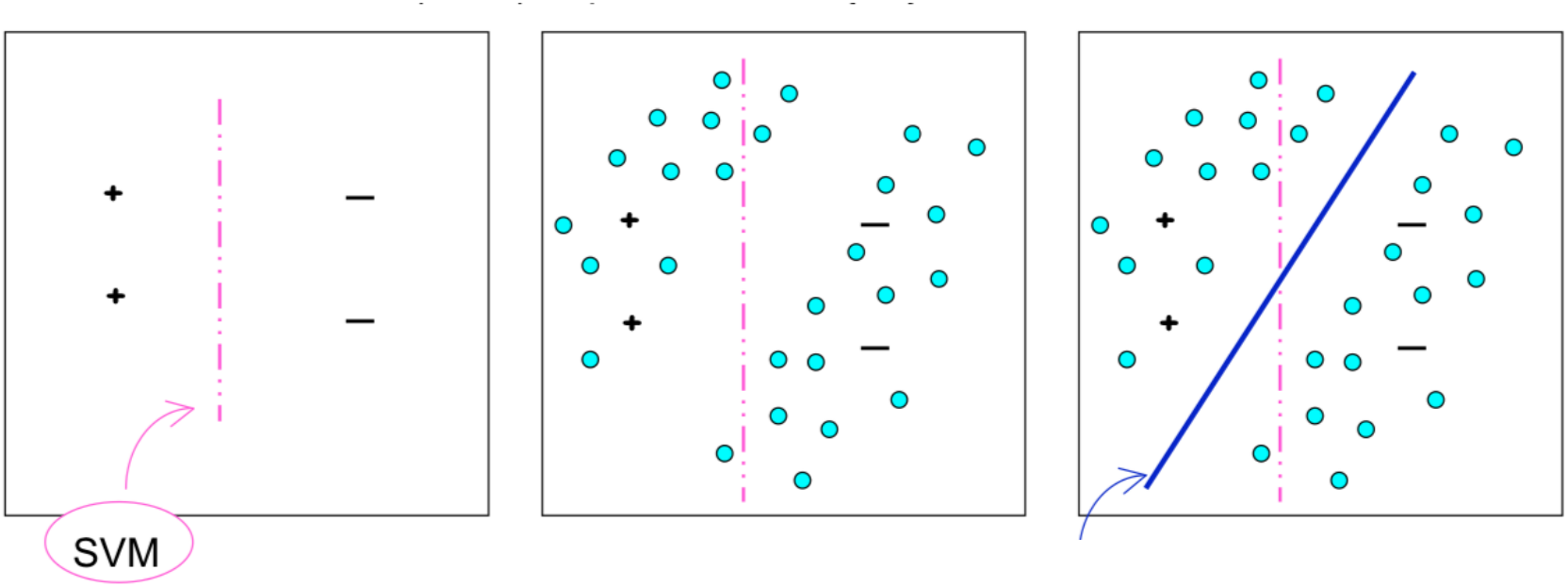
Labeled examples only



Labeled examples
+ **unlabeled** ones

From [Kevin Murphy. **Probabilistic Machine Learning: An introduction**. MIT Press, 2022, p.634]

The idea



...

Semi-supervised learning

- **Unsupervised** learning $\mathbf{P}_{x}$
- **Supervised** learning $\mathbf{P}_{y|x}$

Semi-supervised learning

- **Unsupervised** learning $\mathbf{P}_{x}$
- **Supervised** learning $\mathbf{P}_{y|x}$

When can **unsupervised** learning help **supervised** learning?

Semi-supervised learning

- **Unsupervised** learning $\mathbf{P}_{\mathcal{X}}$
- **Supervised** learning $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$

When can **unsupervised** learning help **supervised** learning?

→ $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$ and $\mathbf{P}_{\mathcal{X}}$ should be related

Semi-supervised learning

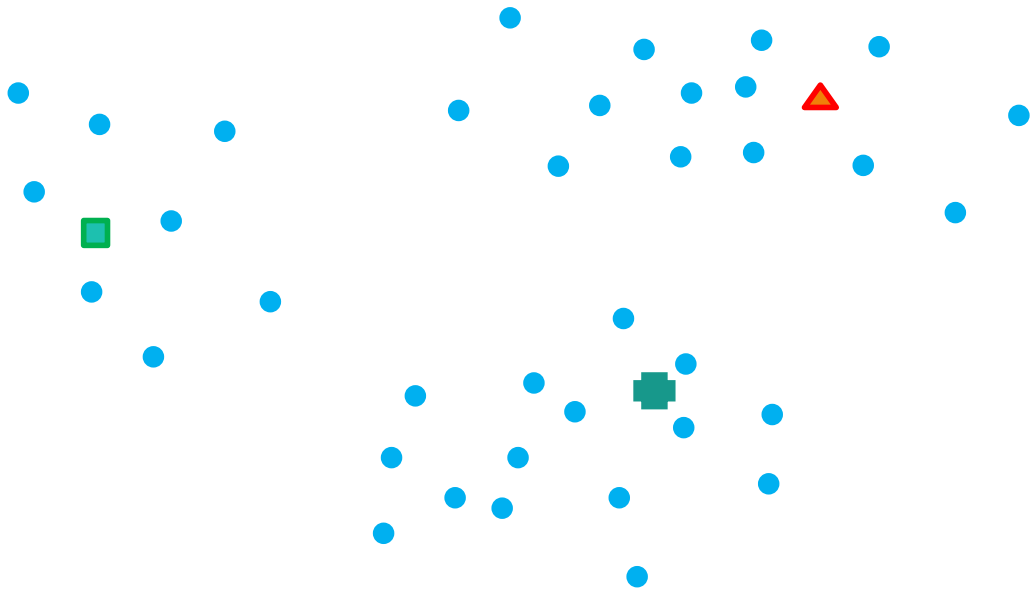
The common underlying assumption:

The decision function (hypothesis *h*) **should not cut** through **high density** regions

Semi-supervised learning

Simplest approach

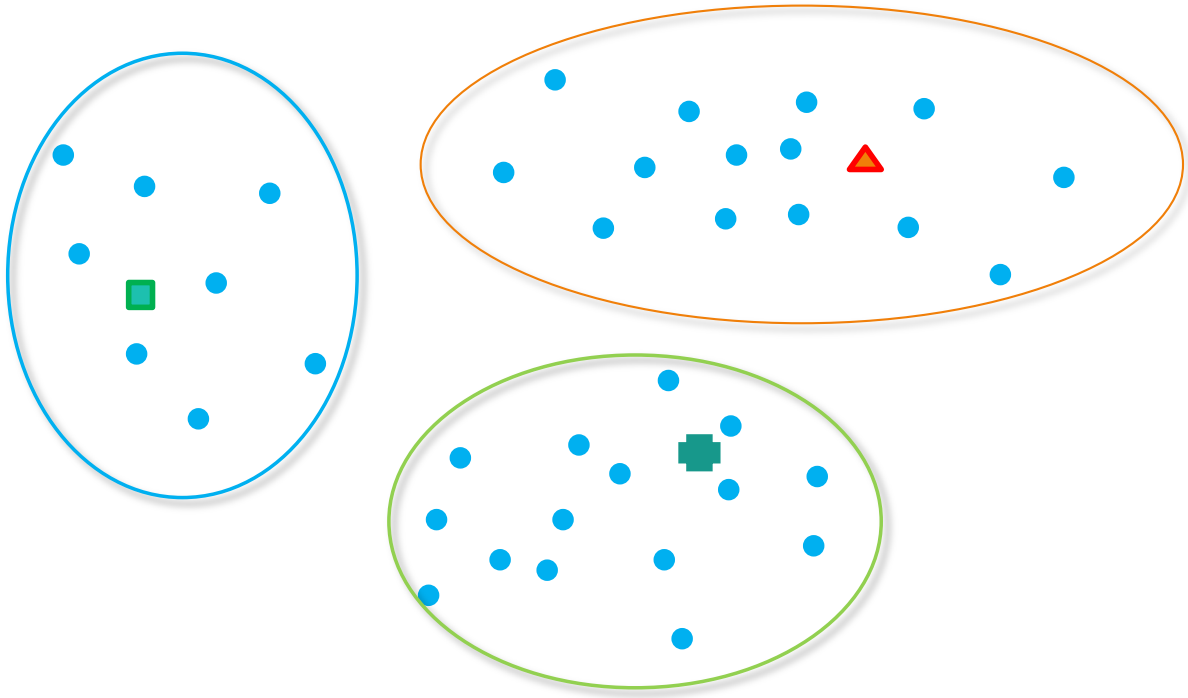
1. Compute a **clustering** of the all data (labeled and unlabeled)
2. For each cluster, **assign its class** to the majority vote of the labeled examples that belong to it



Semi-supervised learning

Simplest approach

1. Compute a **clustering** of the all data (labeled and unlabeled)
2. For each cluster, **assign its class** to the majority vote of the labeled examples that belong to it



Self-training approach

1. Given $\mathcal{S}_L = \{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq l}$ and $\mathcal{S}_U = \{(\mathbf{x}_j)\}_{1 \leq j \leq u}$
2. Train on S_L to obtain h_1
3. Apply h_1 to S_U
4. Remove a set of unlabeled data from S_U and add them to S_L (the ones where $h(\mathbf{x})$ is the more confident) with the label $h(\mathbf{x})$
5. Go to 2 and **repeat** until **convergence**

Semi-supervised learning

- Idea: endow unlabeled data with pseudo-labels
  (the likeliest class at time *t*)

$$y_i = \begin{cases} 1 & \text{if } i = \text{argmax}_{i \in \{1,\ldots,C\}} \, h_i^t(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases}$$

Output of the i[th] output neuron

- Train with the empirical risk:

$$R_{\text{emp}}(h) = \frac{1}{m_l} \sum_{i=1}^{m_l} \sum_{j=1}^{C} \ell(h_j(\mathbf{x}_i), y_j^i) + \alpha(t) \frac{1}{m_u} \sum_{i=1}^{m_u} \sum_{j=1}^{C} \ell(h_j(\mathbf{x}_i), \underbrace{y_j^i}_{\text{pseudo-label}})$$

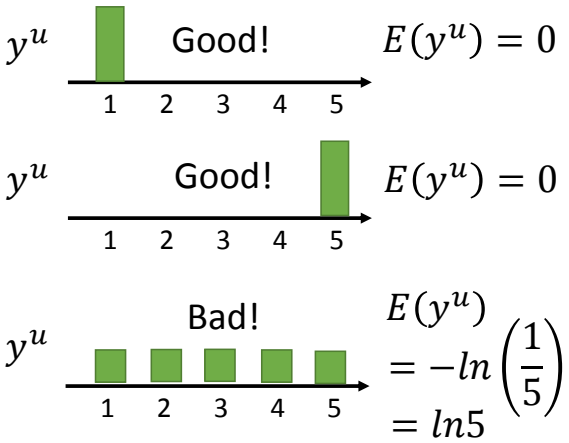
Weights the tentatively labeled data

Crucial to set $\alpha(t)$ with great care

[Dong-Hyun Lee (2013) "*Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks*", ICML-2013]

Entropy regularization approach

$$\hat{h} = \underset{h \in \mathcal{H}}{\text{ArgMin}} \Bigg[\underbrace{\frac{1}{l} \sum_{i=1}^{l} \ell\big(h(\mathbf{x}_i), y_i\big)}_{\text{Empirical risk on labeled data}} + \lambda \underbrace{\sum_{j=1}^{u} -h(\mathbf{x}_j) \log h(\mathbf{x}_j)}_{\text{Entropy of the predictions}} \Bigg]$$



- You have to make assumptions about what you think is reasonable as a bias
  - E.g. that classes are separated by low density regions

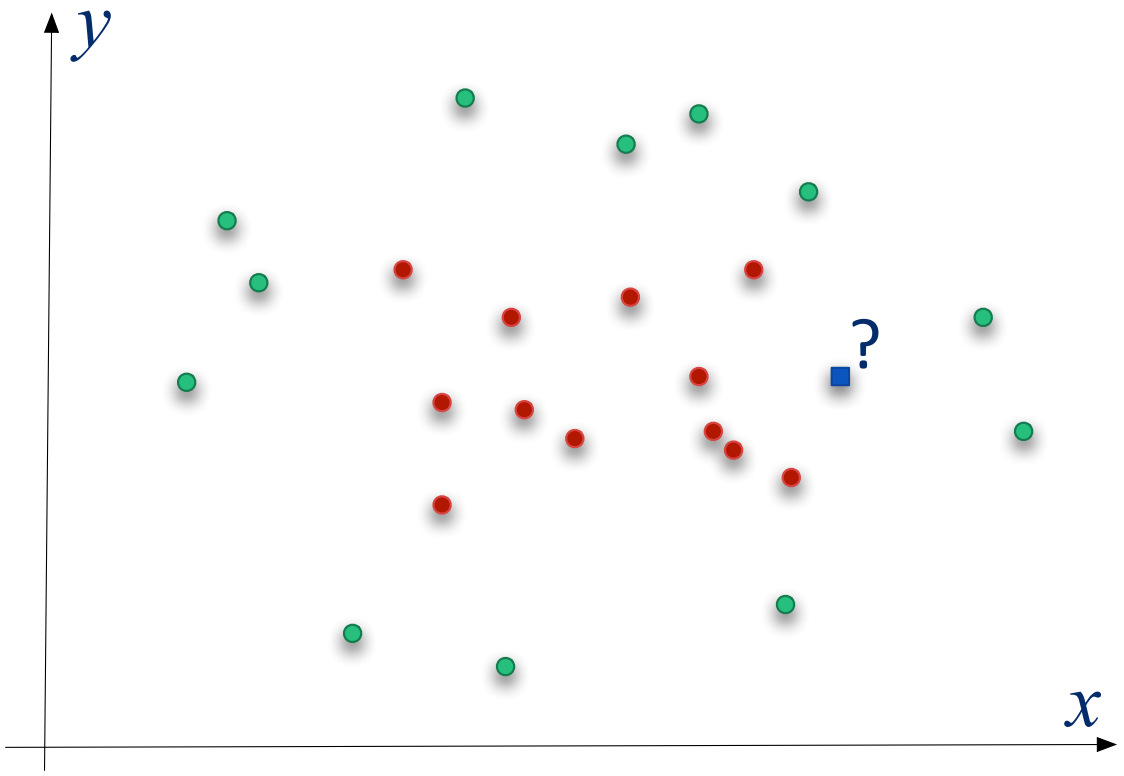
- Then, you show that if the assumption is met by Nature, then you find a correct hypothesis

A **remark** on semi-supervised learning

- Could be regarded as **transductive learning** where one wants to label unlabeled training instances

Transductive learning

- I know in advance where I will be queried



Semi supervised learning: how to approach it theoretically?

The **ability of unlabeled data to help** depends on two quantities:

1. The extent to which

   the target function indeed **satisfies the given assumptions**

   i.e. the decision function does not cut high density regions

2. The extent to which

   the distribution allows this assumption to **rule out alternative hypotheses**

   i.e. the unsupervised training sample help in limiting the possible decision functions

What kind of theory for semi-supervised learning?

A PAC learning approach

How to derive guarantees for semi-supervised learning?

[Balcan & Blum (2006). *"An augmented PAC model for semi-supervised learning"*]

- Let's assume that **it is reasonable that** the frontier between two classes **does not cut** through **high density regions** of the input space *X*
  - **Then** the unlabeled data points bring constraints on the possible decision functions -> gain of information

- Formally: let's define a compatibility function $\chi : \mathcal{H} \times X \rightarrow [0,1]$
  - E.g. $\chi(h, x)$ could be an increasing function of the distance of *x* to the decision function (separator) *h*

$$\chi(h, \mathcal{D}) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}\left[\chi(h, \mathbf{x})\right]$$ Compatibility between *h* and *D*

$$\chi(h, S) = \frac{1}{m} \sum_{i=1}^{m} \chi(h, \mathbf{x}_i)$$ Empirical compatibility measured on *S*

How to derive guarantees for semi-supervised learning?

- Incompatibility

$$er_{\text{unl}}(h) \;=\; 1 - \chi(h, \mathcal{D})$$

$$\widehat{err}_{\text{unl}}(h) \;=\; 1 - \chi(h, S)$$

- Let's define the **set of hypotheses whose incompatibility is** at most some given value τ

$$\mathcal{H}_{\mathcal{D},\mathcal{X}}(\tau) \;=\; \left\{h \in \mathcal{H} \;:\; err_{\text{un}}(h) \leq \tau\right\}$$

$$\mathcal{H}_{S,\mathcal{X}}(\tau) \;=\; \left\{h \in \mathcal{H} \;:\; \widehat{err}_{\text{un}}(h) \leq \tau\right\}$$

How to derive guarantees for semi-supervised learning?

- Theorem (realizable case and $\mathcal{H}$ finite)

If we see m_l labeled examples and m_u unlabeled examples, where

$$m_l \geq \frac{1}{\varepsilon}\left[\ln|\mathcal{H}| + \ln\frac{2}{\delta}\right] \quad \text{and} \quad m_u \geq \frac{1}{\varepsilon}\left[\ln|\mathcal{H}_{\mathcal{D},\mathcal{X}}(\varepsilon)| + \ln\frac{2}{\delta}\right]$$

then, with probability $\geq 1-\delta$, any $h \in \mathcal{H}$ with $\widehat{err}(h) = 0$

and $\widehat{err}_{\text{unl}}(h) = 0$ has $err(h) \leq \varepsilon$

How to derive guarantees for semi-supervised learning?

- Proof:

The probability that **a given hypothesis** h with $err_{\text{unl}}(h) > \varepsilon$ has $\widehat{err}_{\text{unl}}(h) = 0$

is at most $(1-\varepsilon)^{m_u} < \frac{\delta}{2}$ when $m_u \geq \frac{1}{\varepsilon}\left[\ln\frac{2}{\delta}\right]$

Therefore, by the union bound, if the number of **unlabeled** examples is $m_u \geq \frac{1}{\varepsilon}\left[\ln|\mathcal{H}_{\mathcal{D},\mathcal{X}}(\varepsilon)| + \ln\frac{2}{\delta}\right]$

then, with probability $1-\delta/2$, the hypotheses in $\mathcal{H}_{\mathcal{D},\mathcal{X}}(\varepsilon)$ have $\widehat{err}_{\text{unl}}(h) = 0$.

Similarly, the number of **labeled** examples $m_l \geq \frac{1}{\varepsilon}\left[\ln|\mathcal{H}| + \ln\frac{2}{\delta}\right]$ ensures that with probability $1-\delta/2$,

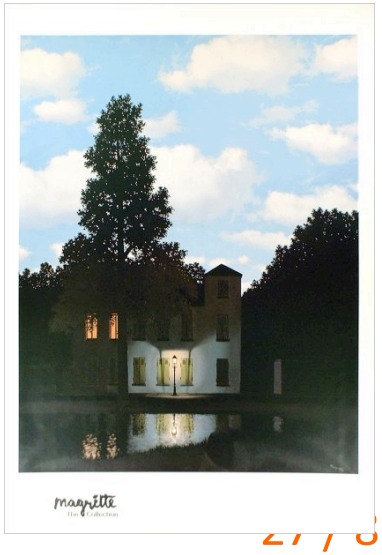
none of those hypotheses whose true error is $\geq \varepsilon$ have an empirical error of 0,

yielding the theorem.

Lesson

- You have to make assumptions about the world, and therefore what you think is reasonable as a bias
  - E.g. that classes are **separated** by **low density** regions

- Then, you show that **if** the assumption is met by Nature, **then** your algorithm finds a correct hypothesis



How to derive guarantees for semi-supervised learning?

- The theorem **assumes**
  - The data is **i.i.d.** (standard PAC learning)
    1. Probability of **each hypothesis** to obey the criteria and still be in error
    2. **Union bound**
  - The true target functions **obey the compatibility criterion**

How to derive guarantees for semi-supervised learning?

- The theorem **assumes**
  - The data is **i.i.d.** (standard PAC learning)
    1. Probability of **each hypothesis** to obey the criteria and still be in error
    2. **Union bound**
  - The true target functions **obey the compatibility criterion**

What if Nature **does not obey** these assumptions?

E.g. the interesting decision functions **cut through high density regions** of *X*

Can you think of a case when this may happen?

Outline

1. Semi-supervised learning
2. Classes severely unbalanced
3. Learning from positive examples only
4. Active learning
5. LUPI (Learning Using Privileged Information)

Illustrations

- Rare pathologies
- Anomaly detection
- Fraud
- Rare species
  - E.g. Pl@ntNet: **46,000** species, but only ~**1000** well represented

Remedies

Remedies

- If **enough** data
  - **undersample** the over-represented classes

Remedies

- If **enough** data
  - **undersample** the over-represented classes
- If **not enough** data

Remedies

- If **enough** data
  - **undersample** the over-represented classes
- If **not enough** data
  - **oversample** the under-represented classes
    - Create **noisy** clones of the data points
    - Create **new** data points generated by **well chosen transformations**
      - E.g. respecting **invariances** (E.g. translations, rotations, change of luminosity, …)

Remedies

- If **not enough** data
  - **oversample** the under-represented classes
    - Create **noisy** clones of the data points
    - Create **new** data points generated by **well chosen transformations**
      - E.g. respecting **invariances** (E.g. translations, rotations, change of luminosity, …)

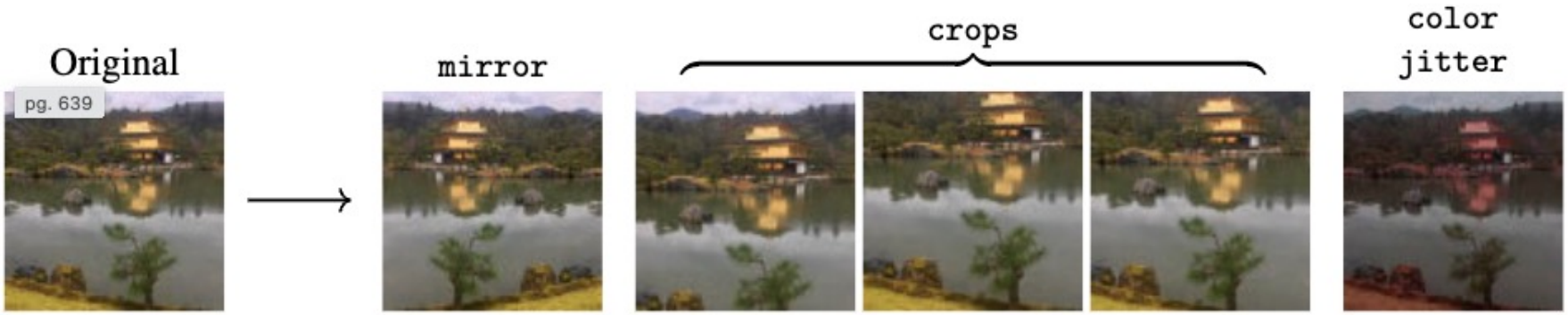


Figure 36.1: A few common types of data augmentation.

Data augmentation

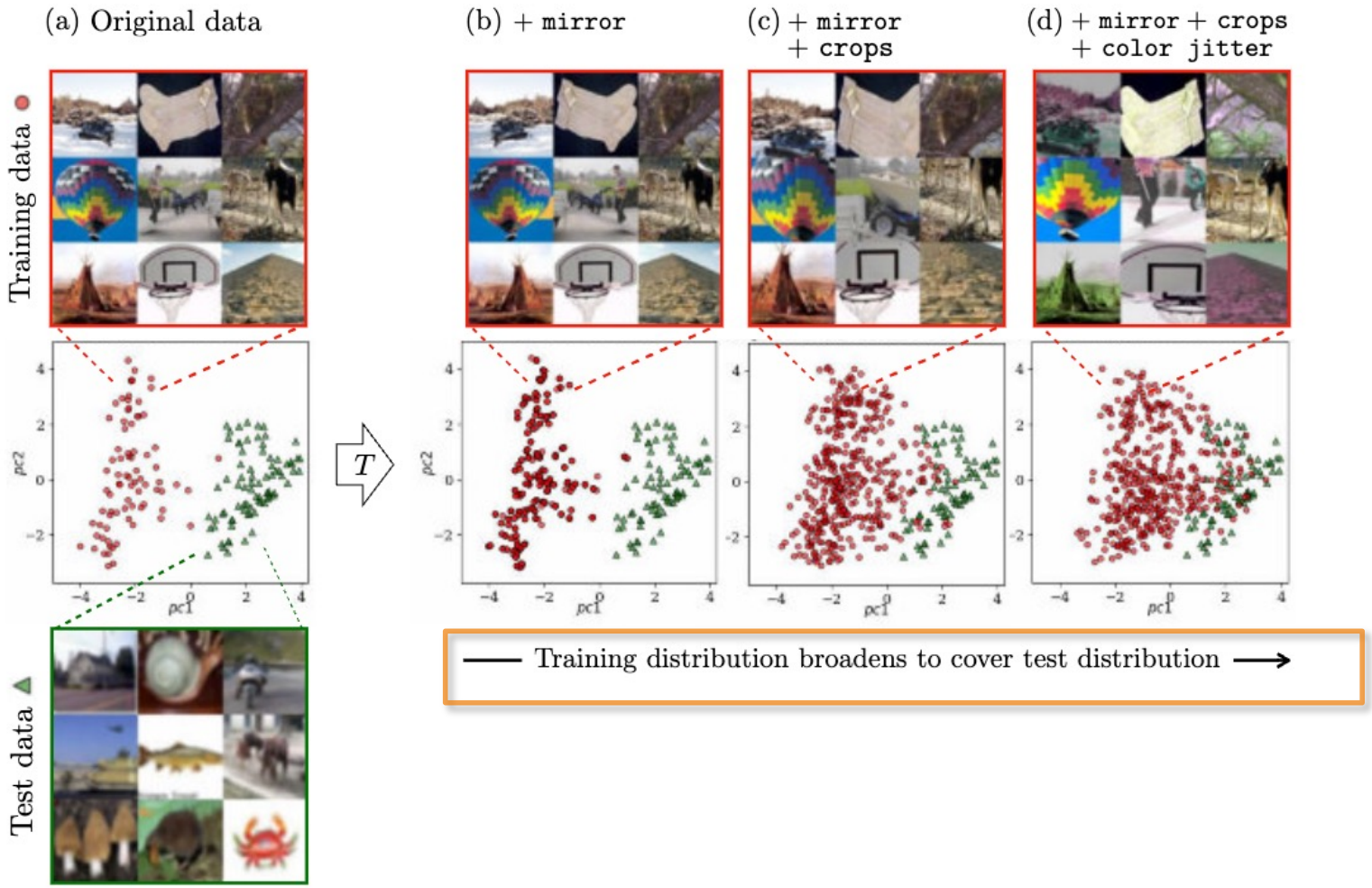


Figure 36.2: Data augmentation broadens the training distribution so that it might better cover the test cases. (a) Training data are from Caltech256 [175] and test data are from CIFAR100 [278]. The scatter plots show these data in a 2D feature space [the first two principle components (pcs) of CLIP [394]]. (b) Training data after `mirror` augmentations (random horizontal flips). (c) The same plus `crops` (crop then rescale to the original size). (d) The same plus `color jitter` (random shifts in color and contrast).

Remedies

- If **enough** data
  - **undersample** the over-represented classes
- If **not enough** data
  - **oversample** the under-represented classes
    - Create **noisy** clones of the data points
    - Create **new** data points generated by **well chosen transformations**
      - E.g. respecting **invariances** (E.g. translations, rotations, change of luminosity, …)
- Modify the **loss function**
  - **Penalize** more the errors on the under-represented class

$$\ell_{\hat{\mathrm{M}},\mathrm{m}} P_{\hat{\mathrm{M}},\mathrm{m}} + \ell_{\hat{\mathrm{m}},\mathrm{M}} P_{\hat{\mathrm{m}},\mathrm{M}} \qquad \text{with} \qquad \ell_{\hat{\mathrm{M}},\mathrm{m}} >> \ell_{\hat{\mathrm{m}},\mathrm{M}}$$

Proportion of all points where points of the minority class are misclassified as from the Majority one

Outline

1. Semi-supervised learning
2. Classes severely unbalanced
3. Learning from positive examples only
4. Active learning
5. LUPI (Learning Using Privileged Information)

Scenarios for learning from positive examples only

- ???

Scenarios for learning from **positive examples only**

- Collaborative science
  - Biodiversity
  - E.g. Pl@ntNet
    - The users take pictures of plants: positive examples
    - That does not say: “these other plants were not present”
- Medicine
  - Reports of subjects with some disease does not say how many and which ones do not have the disease
- Adds on web pages
  - Pages that have **not been visited** are not necessarily uninteresting

Scenarios for learning from **positive examples only**

- In general
  - Detecting absence can be more difficult

    than detecting presence

Possibly **lots** of

false negative

E.g. The absence of observation of a bird in a region does not mean it does not exist

The fully observable case

- We look for a **hypothesis** where *L* is the number of possible classes (labels)

$$h : \mathcal{X} \to [0,1]^L$$

A **vector** of predictions (probabilities)

- We want to **minimize the risk** with a *loss function* (e.g. binary cross-entropy)

$$R(h) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\ \ell(h(\mathbf{x}),\mathbf{y})$$

$$\ell : [0,1]^L \times \mathcal{Y} \to \mathbb{R}$$

$$\ell_{\mathrm{BCE}}(h(\mathbf{x}_n),\mathbf{y}_n) = -\frac{1}{L}\sum_{i=1}^{L} P(\mathbf{y}_n^i = 1|\mathbf{x}_n)\log\big(h(\mathbf{x}_n^i)\big) + P(\mathbf{y}_n^i = 0|\mathbf{x}_n)\log\big(1 - h(\mathbf{x}_n^i)\big)$$

- Given a dataset with *N* examples we want to find a hypothesis that minimizes the **empirical risk**

$$\mathcal{S} = \{(\mathbf{x}_n,\mathbf{y}_n)\}_{1\leq n\leq N}$$

$$\hat{h}_{\mathrm{fully}} = \operatorname*{ArgMin}_{h\in\mathcal{H}} \frac{1}{N}\sum_{n=1}^{N} \ell\big(h(\mathbf{x}_n),\mathbf{y}_n\big)$$

The partially observable case

- We look for a **hypothesis** $h_{\text{partial}} : \mathcal{X} \to [0,1]^L$

- During training, we observe $\mathbf{z}_n \in \mathcal{Z} = \{0, 1, \oslash\}^L$

  where $\mathbf{z}_n^i = \oslash$ ← indicates that the i^{th} label is **unobserved**

  or $\mathbf{z}_n^i = 1$

  never $\mathbf{z}_n^i = 0$

- Given a dataset $\mathcal{S} = \{(\mathbf{x}_n, \mathbf{z}_n)\}_{1 \leq n \leq N}$

  we want to find a hypothesis that

  minimizes the **empirical risk** $\hat{h}_{\text{partial}} = \underset{h \in \mathcal{H}}{\text{ArgMin}} \frac{1}{N} \sum_{n=1}^{N} \ell\big(h(\mathbf{x}_n), \mathbf{z}_n\big)$

- Which **loss function** to use?

- Which **loss function** to use?

1. Assume **unobserved** examples are negative

2. Assume **unobserved** examples are negative

   but **smooth** the loss function

Approach "assume **unobserved** are **negative**"

- Assume that all **unobserved** labels are **negative**

$$P(\mathbf{y}_n^i = 1 | \mathbf{x}_n) = 0 \quad \text{if} \quad \mathbf{z}_n^i = \oslash$$

- The resulting loss is

$$\ell_{\text{AN}}(h(\mathbf{x}_n), \mathbf{y}_n) = -\frac{1}{L} \sum_{i=1}^{L} \mathbb{1}_{[\mathbf{z}_n^i = 1]} \log\big(h(\mathbf{x}_n^i)\big) + \mathbb{1}_{[\mathbf{z}_n^i \neq 1]} \log\big(1 - h(\mathbf{x}_n^i)\big)$$

$\mathbb{1}_{[\mathbf{z}_n^i = 1]} = 1$ if $\mathbf{z}_n^i = 1$ and 0, otherwise

- We expect **false negatives** (considered as 0 when $\mathbf{z}_n^i = \oslash$)

Approach "assume **unobserved** are **negative**" + **smoothing**

- Assume that all **unobserved** labels are **negative**

$$P(\mathbf{y}_n^i = 1 | \mathbf{x}_n) = 0 \quad \text{if} \quad \mathbf{z}_n^i = \oslash$$

- And give **more weight to the observed examples**. The resulting loss is

$$\ell_{\text{AN-LS}}(h(\mathbf{x}_n), \mathbf{y}_n) = -\frac{1}{L} \sum_{i=1}^{L} \mathbb{1}^{0.95}_{[\mathbf{z}_n^i = 1]} \log\big(h(\mathbf{x}_n^i)\big) + \mathbb{1}^{0.05}_{[\mathbf{z}_n^i \neq 1]} \log\big(1 - h(\mathbf{x}_n^i)\big)$$

Observed as **positive**

No observation reported

Hence assumed as **negative**

Intuitively $R(\hat{h}_{\text{fully}}) \leq R(\hat{h}_{\text{partial}})$

Risk for the **fully observable** case

Risk for the **partially observable** case

- But **by how much**?

- In the case of “assume unobserved = negative”

Intuitively

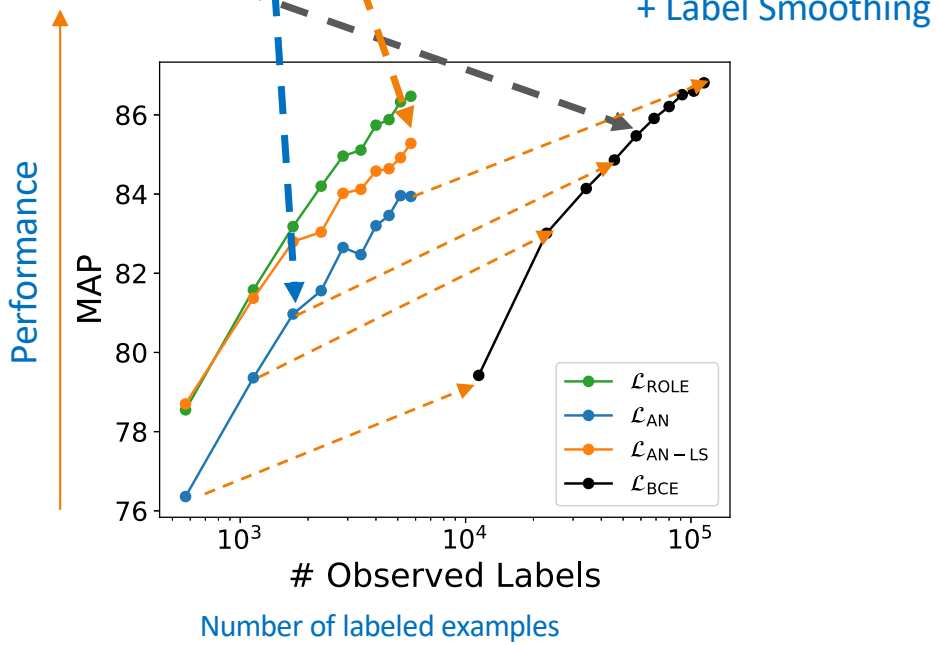
$$R(\hat{h}_{\text{fully}}) \leq R(\hat{h}_{\text{partial}})$$

AN: Assume Negative

AN-LS: Assume Negative + Label Smoothing

- But **by how much**?

- In the case of "assume unobserved = negative"



Number of labeled examples

With 20 times **fewer labeled** examples (10, 20, …, 7500), the performance is not that bad *on this dataset* compared to the fully observable case with (200, 400, …, 150 000) labeled examples.

COLE, Elijah, MAC AODHA, Oisin, LORIEUL, Titouan, *et al.* **Multi-label learning from single positive labels**. In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021. p. 933-942.

Learning from positive examples only: lots of approaches

- Approaches
  - Assume that *the missing labels are negative*
  - *Ignore* the missing labels
  - Perform *label matrix reconstruction*
  - Learn *label correlations*
  - Learn *generative probabilistic models*
  - Train *label cleaning networks*

  - Related to **learning with label noise**
    - Here, some **unobserved labels** are incorrectly treated as being **absent**
  - Related to learning from a set of **positive examples** and a set of **unlabeled** ones (**PU** learning)

Lessons

1. **Fomalize** the assumptions about your problem
   - The labelling process
   - The type of target (and hypothesis) function

2. Design a **loss function** appropriate for the problem
   - Able to **explore efficiently** the hypothesis space
     and to find a good minimum of the empirical risk

3. Design a good **evaluation scheme**

Outline

1. Semi-supervised learning
2. Classes severely unbalanced
3. Learning from positive examples only
4. Active learning
5. LUPI (Learning Using Privileged Information)

Active learning

- When the learner can **actively ask** for pieces of information
  - Labels of selected **examples**
  - Values of some selected **descriptors**
    - E.g. ask for a medical examination

- Examples
  - MasterMind
  - Scientific activity

Active learning

- When the learner can **actively ask** for pieces of information
  - Labels of selected **examples**
  - Values of some selected **descriptors**
    - E.g. ask for a medical examination

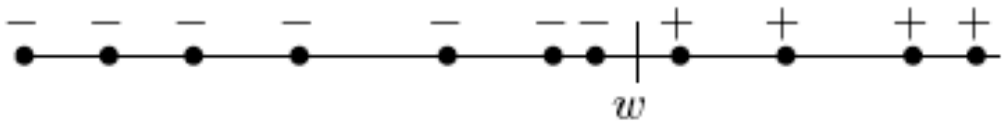
- The **hope**
  - Need of **less** (costly) examples
  - Having a **faster** convergence rate

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R_{\text{Réel}}(h) \leq R_{\text{Emp}}(h) + \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m} \right] > 1 - \delta$$

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R_{\text{Réel}}(h) \leq R_{\text{Emp}}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{2\,m}} \right] > 1 - \delta$$

Active learning

$$h_w(x) = \begin{cases} 1 & \text{if } x \geq w \\ 0 & \text{if } x < w \end{cases}$$



How to find the **best** threshold from querying points?

- By **random** selection of points $m = \mathcal{O}(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$
- By **active** selection $m = \mathcal{O}(\log \frac{1}{\varepsilon})$

Much faster!

Active learning

- Two main approaches
  - "**Constructive**" approach
    - The learner **constructs** queries
  - "**Selective**" (pool-based) approach
    - The learner **selects** points among the **unsupervised** ones

Why is the **constructive** approach sometimes **not** applicable?

How to **select** the examples? (some ideas)

- The more **informative** examples

1. The ones where the **confidence** of the current hypothesis is the **lowest**

   - Measured by a **probability**

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathcal{S}_U}{\text{ArgMax}}\ \text{Uncertain}(\mathbf{x})$$

$$\text{Uncertain}(\mathbf{x}) = \frac{1}{\text{ArgMax}_{y\in\mathcal{Y}}\, p\big(h_t(\mathbf{x})=y\big)}$$

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathcal{S}_U}{\text{ArgMax}} \left\{ -\sum_i p\big(h_t(\mathbf{x})=y_i\big)\ \log p\big(h_t(\mathbf{x})=y_i\big) \right\}$$

Entropy criteria

   - Measured by a **distance** to the decision function

How to **select** the examples? (some ideas)

- The more **informative** examples

1. The ones where the **confidence** of the current hypothesis is the **lowest**

   - Measured by a **probability**

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathcal{S}_U}{\text{ArgMax}}\ \text{Uncertain}(\mathbf{x})$$

$$\text{Uncertain}(\mathbf{x}) = \frac{1}{\text{ArgMax}_{y\in\mathcal{Y}}\, p\big(h_t(\mathbf{x})=y\big)}$$

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathcal{S}_U}{\text{ArgMax}} \left\{ -\sum_i p\big(h_t(\mathbf{x})=y_i\big)\ \log p\big(h_t(\mathbf{x})=y_i\big) \right\}$$

Entropy criteria

   - Measured by **distance** to the decision function

2. Learn an **ensemble** of hypotheses and select the examples where they **disagree** the most

Illustration

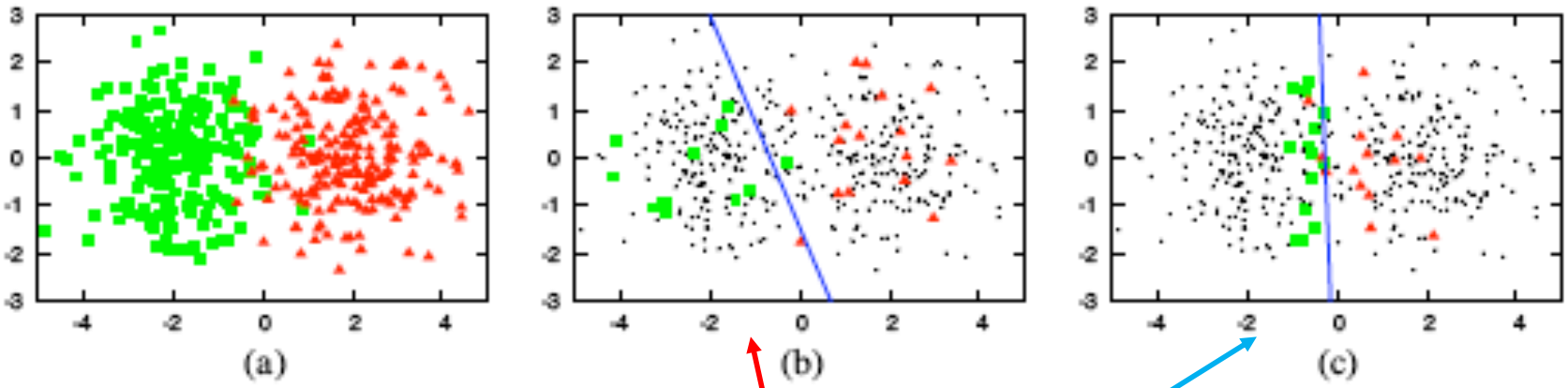


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem domain. The line represents the decision boundary of the classifier (accuracy = 0.7). (c) A logistic regression model trained with 30 actively queried instances using uncertainty sampling (accuracy = 0.9).

Active Learning

- What is the danger?

Active Learning

- What is the **danger**?
  - **No** more **theoretical** guarantees

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R_{\text{Réel}}(h) \leq R_{\text{Emp}}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{2\,m}} \right] > 1 - \delta$$

Does not make sense anymore!!

  - Why?

Active learning: **lessons**

- Active learning is **not much used** in practice
  1. **Costly** to identify informative examples
  2. **Risk** of ignoring important regions of *X*

- Interesting: learning under **budget constraints**
  - **What measurements** should I made under some budget **constraints**?

Outline

1. Semi-supervised learning
2. Classes severely unbalanced
3. Learning from positive examples only
4. Active learning
5. LUPI (Learning Using Privileged Information)

Learning Using Privileged Information

Inspired by learning at school

- The goal is to learn a function $h : \mathbf{x} \in \mathcal{X} \rightarrow y \in \{-1, +1\}$

- Suppose that at learning time there is more available information than at test time

$$\mathcal{S}^* = \{(\underbrace{\mathbf{x}_i, \mathbf{x}_i^*}_{\mathcal{X}'}, y_i)\}_{1 \leq i \leq m}$$

- **Can we then improve the generalization performance** wrt. the one obtained with *S* only?

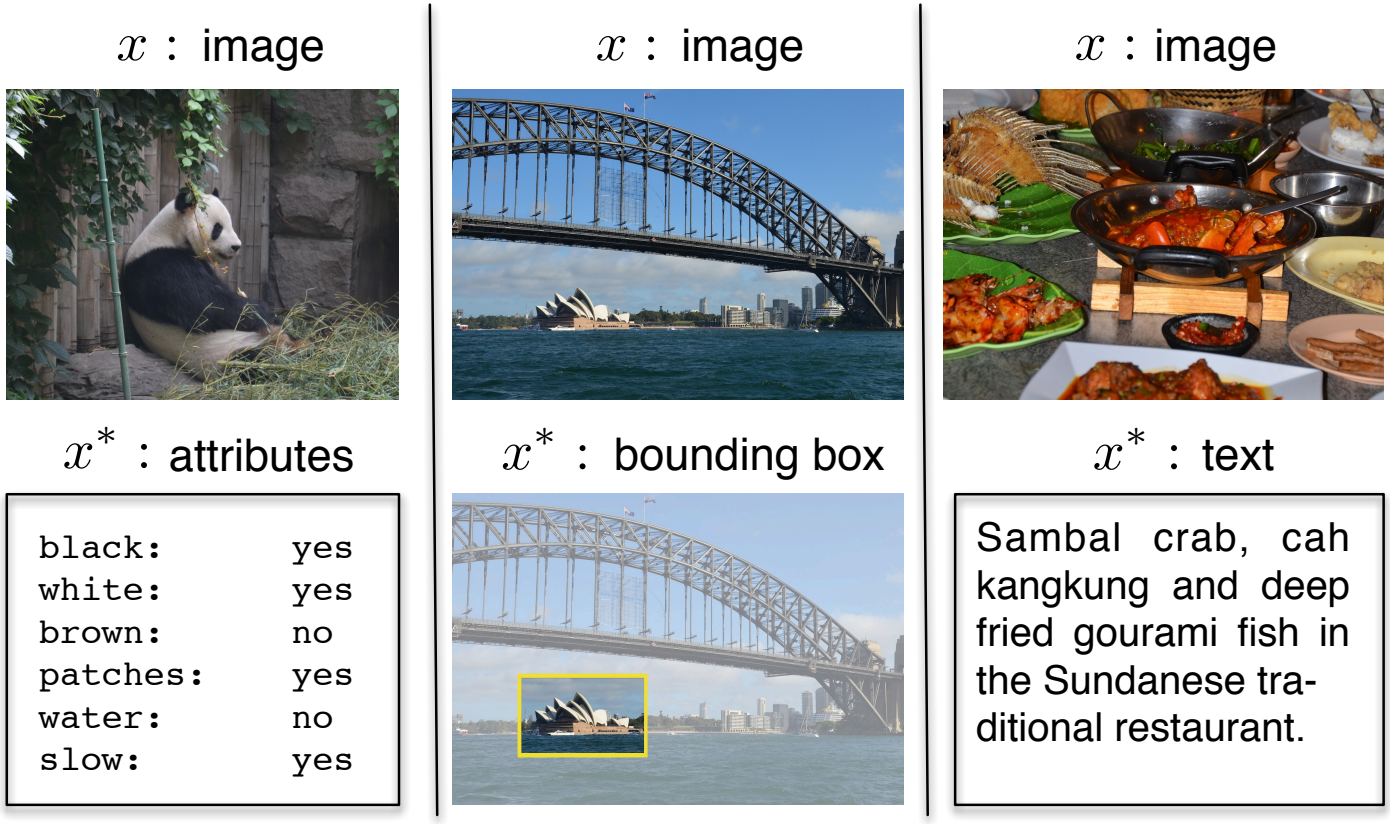
V. Vapnik and A. Vashist (2009) "A new learning paradigm: Learning using privileged information". *Neural Networks*, vol. 22, no. 5, pp. 544–557, 2009

Can you imagine **applications** where privileged information

could be available at *training* time (and not at *testing* time)?

Learning Using Privileged Information

Illustration in computer vision



V. Sharmanska, N. Quadrianto, and Ch. Lamper (2014) "Learning to transfer privileged information". *ArXiv preprint arXiv:1410.0389,* 2014

Two general approaches to LUPI

- **Learning** a hypothesis in the "augmented" input space $h' : \mathcal{X}' \rightarrow y$

$$\mathcal{X}' = \mathcal{X} \cup \mathcal{X}^\star$$

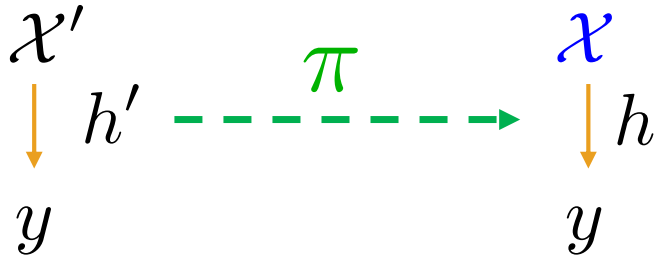
- **Testing**

1. 1st approach: learn to "**complete**" the **description** in $\mathcal{X}$

   then use *h'* $\mathcal{X} \rightarrow \mathcal{X}^\star$

   $h' : \mathcal{X}' \rightarrow y$

2. 2nd approach: **project** back *h'*, the **learnt hypothesis**



Two general approaches to LUPI

- **Learning** a hypothesis in the "augmented" input space $h' : \mathcal{X}' \rightarrow y$

$$\mathcal{X}' = \mathcal{X} \cup \mathcal{X}^\star$$

- **Testing**

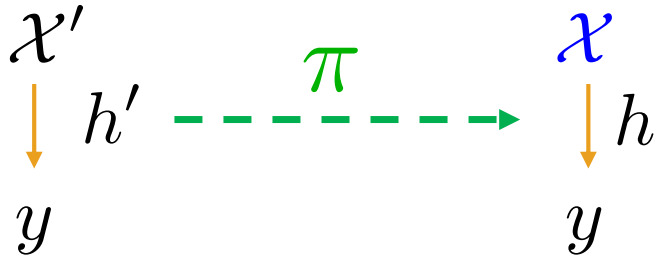
1. 1st approach: learn to "**complete**" the **description** in $\mathcal{X}$

   then use *h'*

$$\mathcal{X} \rightarrow \mathcal{X}^\star$$

$$h' : \mathcal{X}' \rightarrow y$$

2. 2nd approach: **project** back *h'*, the **learnt hypothesis**



Bounds between the **real** risk and the **empirical** risk

- $\mathcal{H}$ finite, realisable case

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R_{\text{Réel}}(h) \leq R_{\text{Emp}}(h) + \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m} \right] > 1 - \delta$$

- $\mathcal{H}$ finite, **non** realisable case

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R_{\text{Réel}}(h) \leq R_{\text{Emp}}(h) + \sqrt{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{2\,m}} \right] > 1 - \delta$$

Example: instead of **600.10^6** training examples, same performance with ~ **775**!!!

First approach to LUPI

- "At the core of our work lies the insight that privileged information allows us to **distinguish between easy and hard examples** in the training set.
- **Assuming** that examples that are easy or hard with respect to the privileged information **will also be easy or hard with respect to the original data**, we enable information transfer from the privileged to the original data modality.
- More specifically, we first define and identify which samples are easy and which are hard for the classification task, and **incorporate the privileged information** into the sample **weights** that encodes its easiness or hardness." (more weight on the easy examples)

One solution: SVM+

- The classical optimization problem

$$\begin{cases} \min \frac{1}{2}\langle \omega, \omega \rangle + C \sum_{i=1}^{m} \xi_i \\ \text{s.t. } y_i[\langle \omega, x_i \rangle + b] \geq 1 - \xi_i, \qquad i = 1, \ldots, m. \end{cases}$$

- is changed into

$$\begin{cases} \min \frac{1}{2}\left[\langle \omega, \omega \rangle + \gamma \langle \omega^*, \omega^* \rangle\right] + C \sum_{i=1}^{m} \left[\langle \omega^*, x^* \rangle + b^*\right] \\ \text{s.t. } y_i[\langle \omega, x_i \rangle + b] \geq 1 - [\langle \omega^*, x_i^* \rangle + b^*], \qquad i = 1, \ldots, m, \\ \quad [\langle \omega^*, x_i^* \rangle + b^*] \geq 0, \qquad i = 1, \ldots, m, \end{cases}$$

C and γ are hyperparameters

- Intuition:
  - Identify the **difficult examples** (outliers)
  - Thus coming back to the **realizable case**

    and obtain **convergence rates** of **1/n** instead of 1/sqrt(n)

Second approach to LUPI

Suppose that in $\mathcal{X}'$, there exists a **good hypothesis space** $\mathcal{H}'$ with very limited capacity (otherwise, why would the teacher be interested?), then the student is expected to identify easily a good hypothesis $h' : \mathcal{X}' \to \mathcal{Y}$. And the whole problem is thus to "project" this hypothesis in $\mathcal{X} \to \mathcal{Y}$

...

Two general approaches to LUPI

- **Learning** a hypothesis in the "augmented" input space $h' : \mathcal{X}' \rightarrow y$

  $\mathcal{X}' = \mathcal{X} \cup \mathcal{X}^\star$

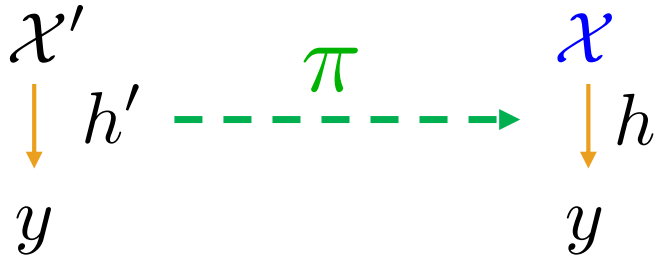
- **Testing**

  1. 1st approach: learn to "**complete**" the **description** in $\mathcal{X}$

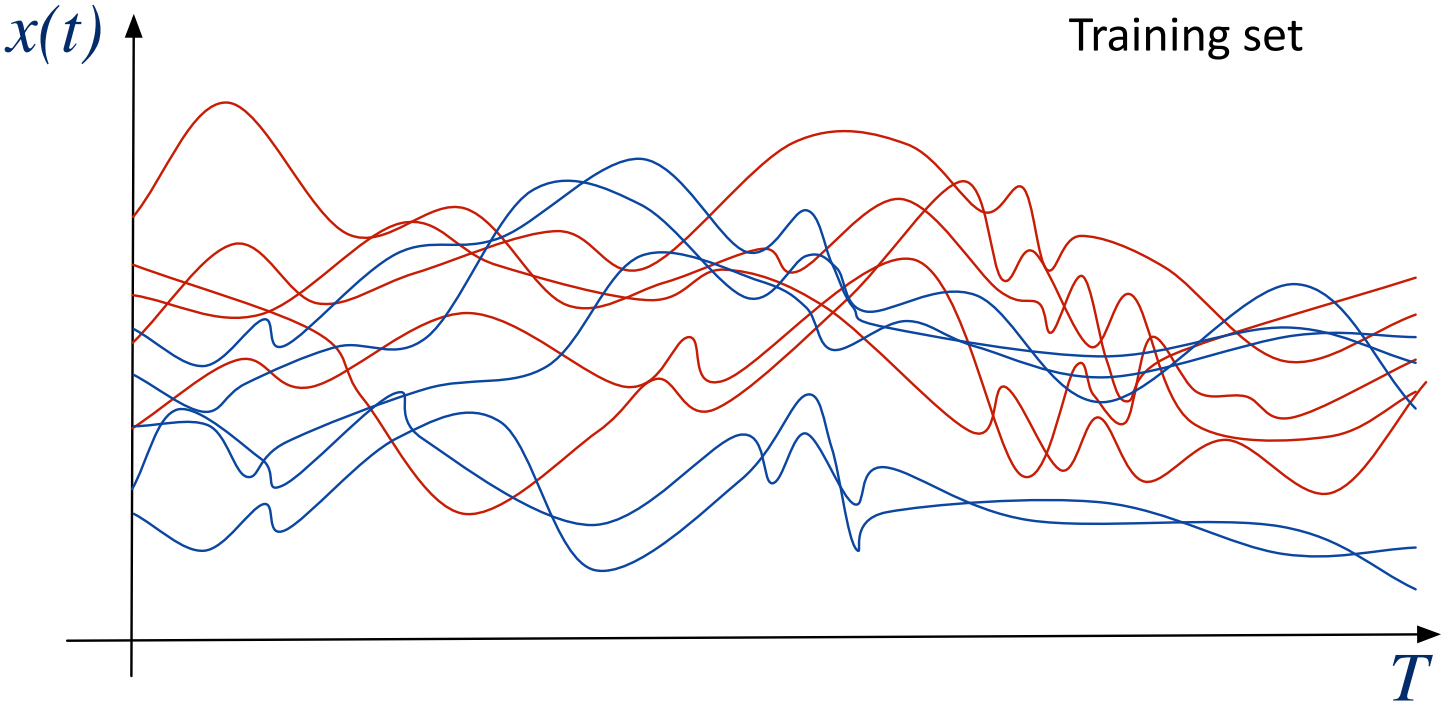
     then use *h'* $\mathcal{X} \rightarrow \mathcal{X}^\star$

     $h' : \mathcal{X}' \rightarrow y$

  2. 2nd approach: **project** back *h'*, the **learnt hypothesis**

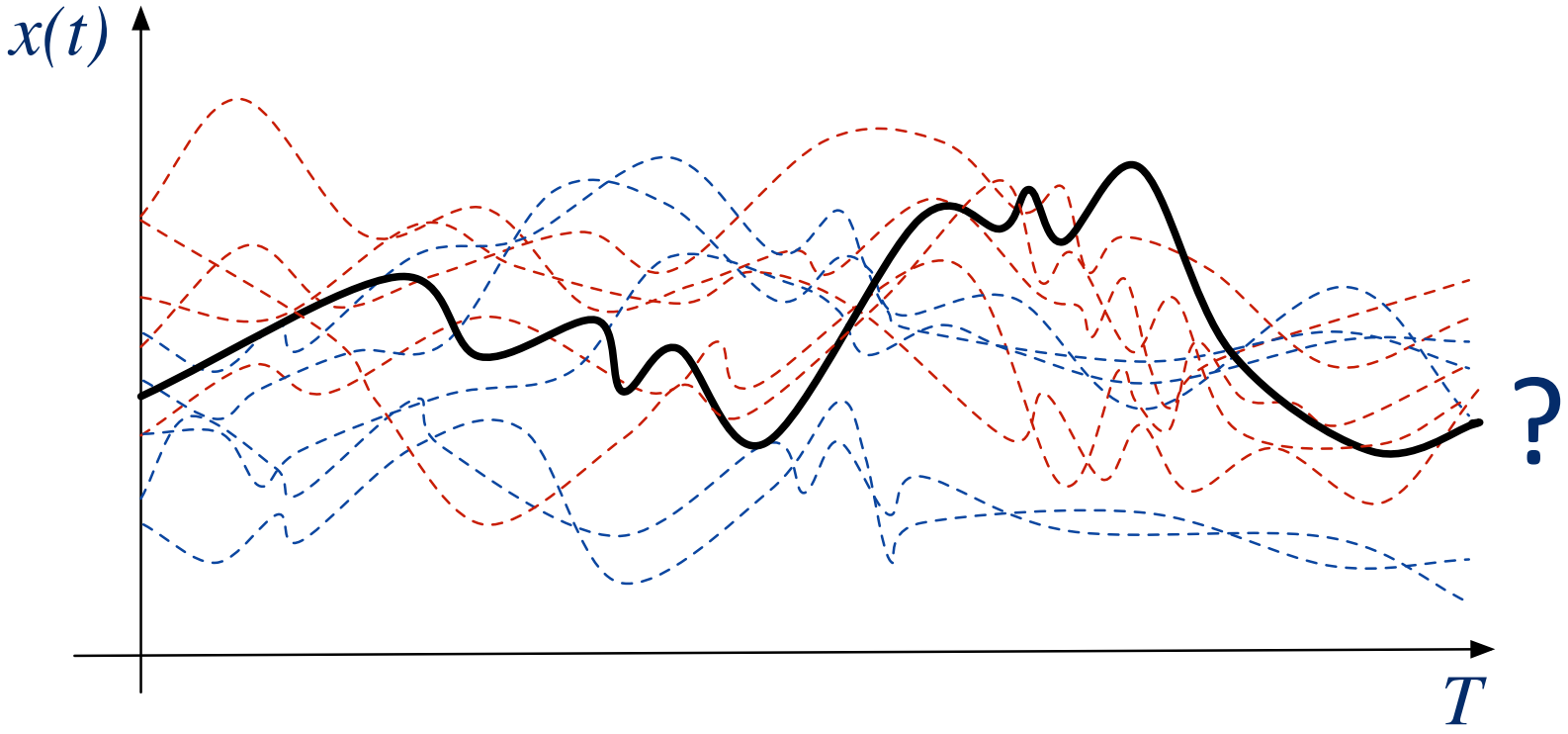


Classification of time series



Standard classification of time series

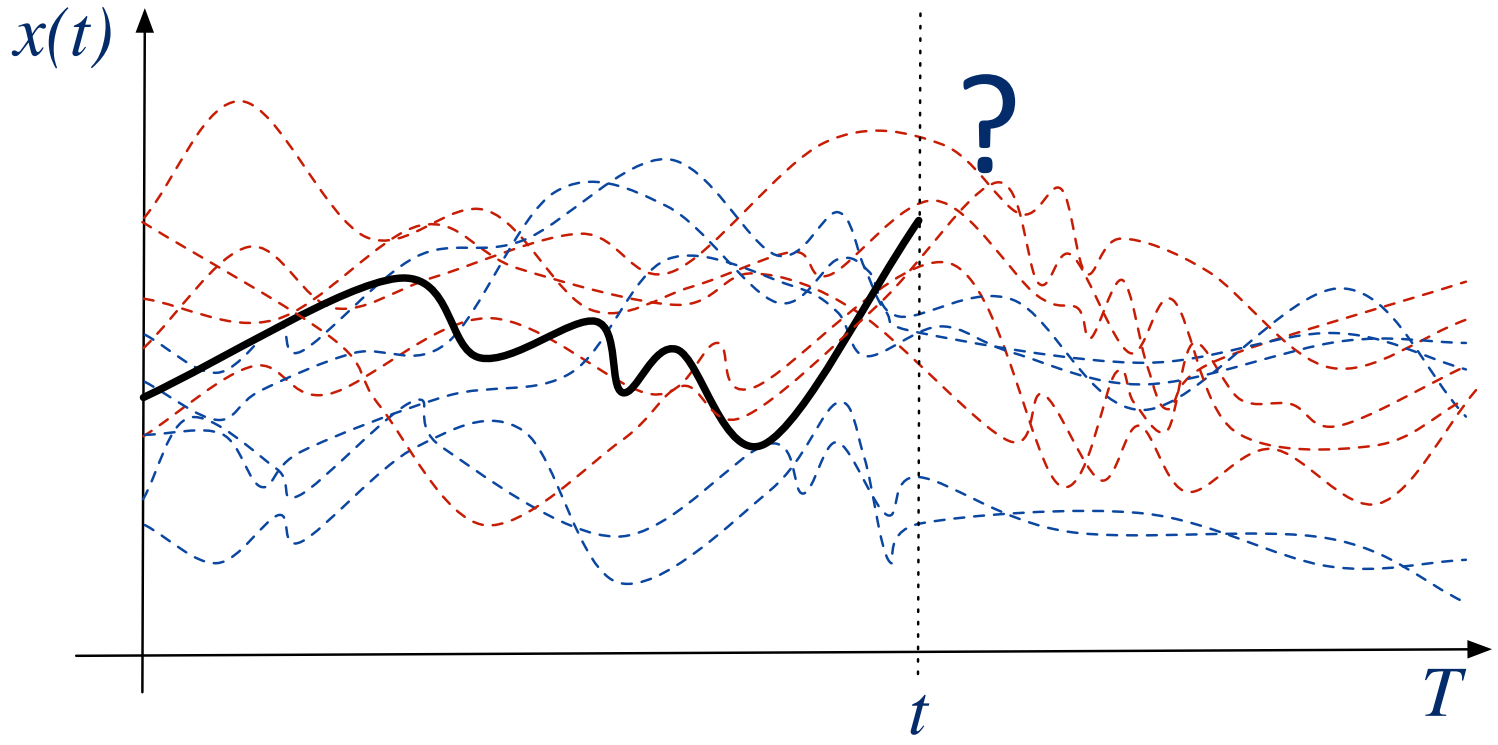
- What is the class of the new time series x_T?



- Monitoring of ***consumer actions on a web site***: will buy or not
- Monitoring of a ***patient state***: critical or not
- Prediction of daily ***electrical consumption***: high or low

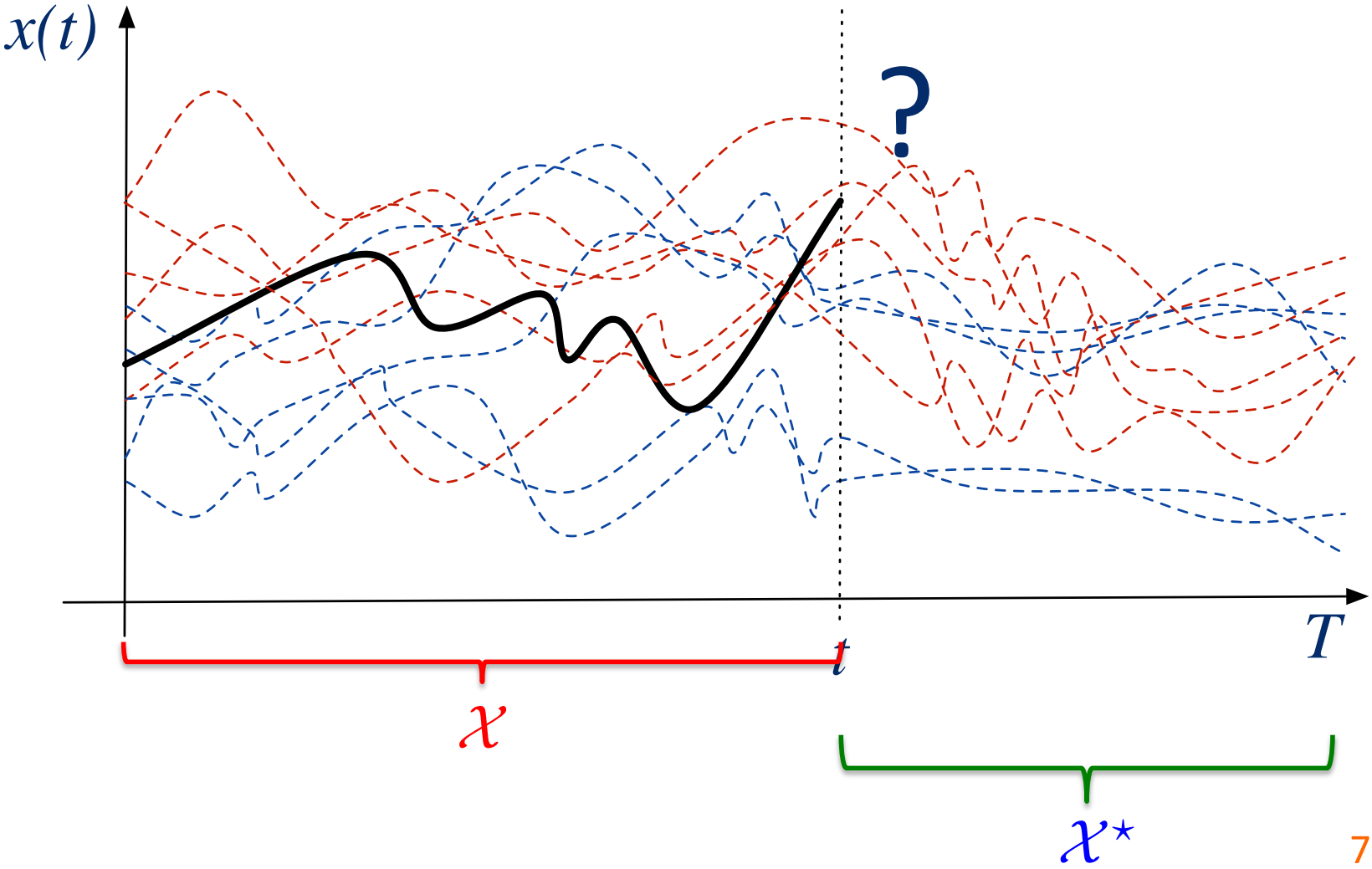
Early classification of time series

- What is the class of the new **incomplete** time series x_t?



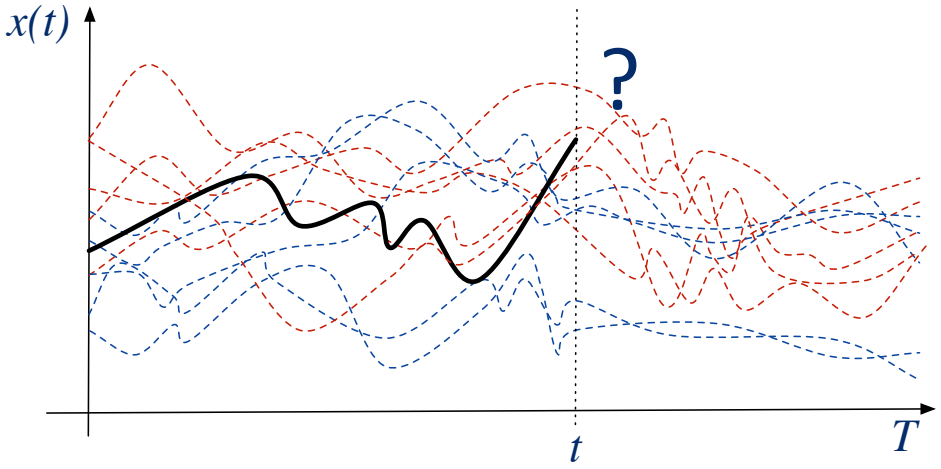
Early classification of time series

- What is the class of the new **incomplete** time series x_t?



Early classification of time series

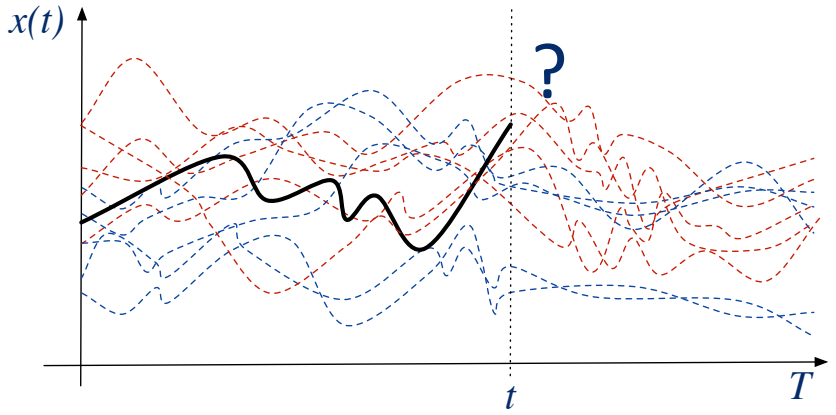
- What is the class of the new **incomplete** time series x_t?



- A LUPI framework

Early classification and LUPI

- This is a LUPI setting



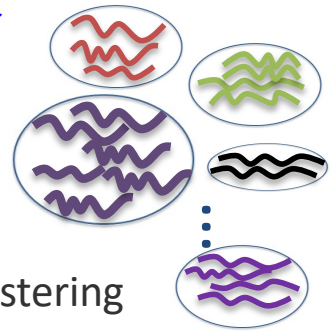
- How to take advantage of this?

The principle

1. During **training**:

→ – identify **meaningful subsets** of time sequences in the training set: c_k

$$P(y|\mathbf{x}_t) \quad \rightarrow \quad P(y|\mathbf{c}_k)$$



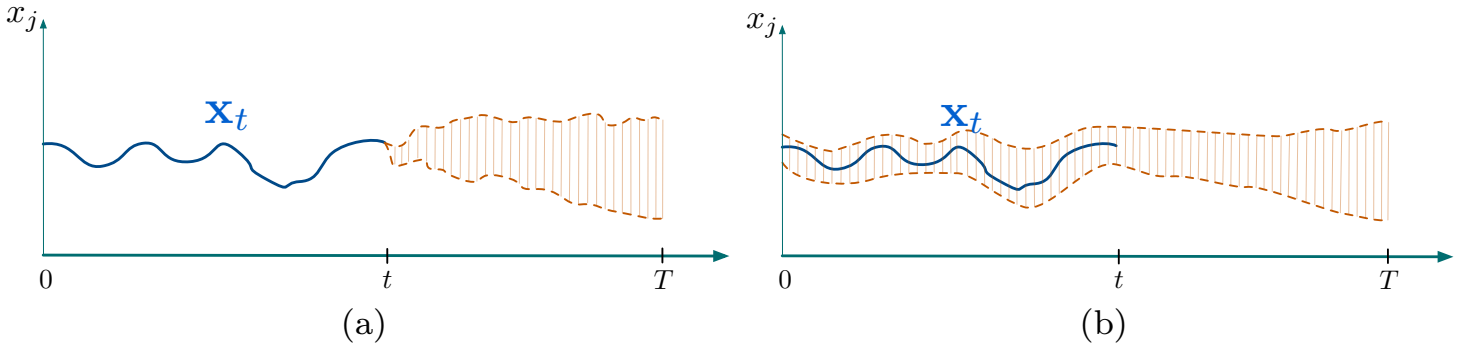


Fig. 1: (a) Given an incomplete time series $\mathbf{x}_t$, the objective is to try to guess the "envelope" of its foreseeable futures. Various methods can be used to do so. (b) The incoming time series $\mathbf{x}_t$ is viewed as a member of or close to some group(s) of times series, and this is used to guess the "envelope" of its foreseeable futures.

Conclusions

- Many in-distribution learning situations benefit from being considered as out-of-distributions learning tasks

- In all cases, assumptions have to be made

Conclusions

- In all cases, assumptions have to be made

  - **Semi-supervised** learning
    - Which **correlation** between P_X and $P_{Y|X}$?
  - **Unbalanced** classes
    - **Data augmentation** techniques **do not change** the decision function
    - Which data augmentation to use?
  - Learning from **positive** examples **only**
    - **How** to **consider** the **unobserved** data in the loss function
  - **Active** learning
    - **Where** to query examples?
  - Learning Using **Privileged** Information (**LUPI**)
    - Which **correlation** to posit between x and x*?
    - Learn to **complete** examples, or to **project** back *h*' (defined overs *X**) to *h*