Robust Learning from Multiple Sources

Christoph H. Lampert
IST Austria (Institute of Science and Technology Austria), Vienna

IST AUSTRIA
Institute of Science and Technology

IST Austria (Institute of Science and Technology Austria)

- institute for **basic research**, opened in 2009
- located in outskirts of Vienna

Research at IST Austria

- curiosity-driven
- focus on interdisciplinarity
  - Computer Science, Mathematics, Physics, Chemistry, Neuroscience, Biology
- ELLIS unit since 2019

We're hiring! (on all levels)

- interns, PhD students, postdocs,
- faculty (tenure-track or tenured), ...

More information: `chl@ist.ac.at`, **or** `https://cvml.ist.ac.at`

Topics in Our Research Group

Machine Learning Theory

- Transfer Learning
- Lifelong Learning/ Meta-learning
- Robust Learning
- Theory of Deep Learning

Models/Algorithms

- Zero-shot Learning
- Continual Learning
- Weakly-supervised Learning
- Trustworthy/Robust Learning

Learning for Computer Vision

- Scene Understanding
- Generative Models
- Abstract Reasoning
- Semantic Representations

Topics in Our Research Group

Machine Learning Theory

- Transfer Learning
- Lifelong Learning/ Meta-learning
- Robust Learning
- Theory of Deep Learning

Models/Algorithms

- Zero-shot Learning
- Continual Learning
- Weakly-supervised Learning
- Trustworthy/Robust Learning

Learning for Computer Vision

- Scene Understanding
- Generative Models
- Abstract Reasoning
- Semantic Representations

Schedule

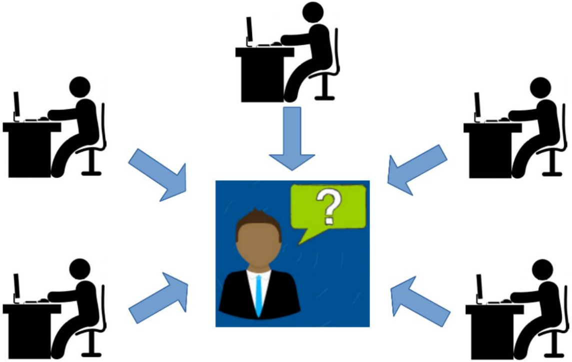
Overview

Refresher of PAC Learning

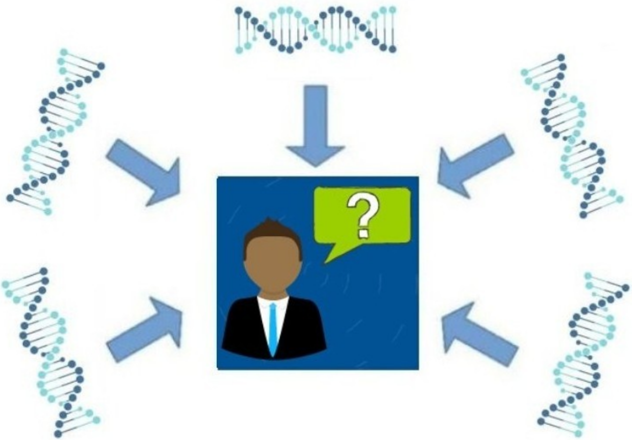
Learning From Untrusted Sources

Slides available at: `http://cvml.ist.ac.at`

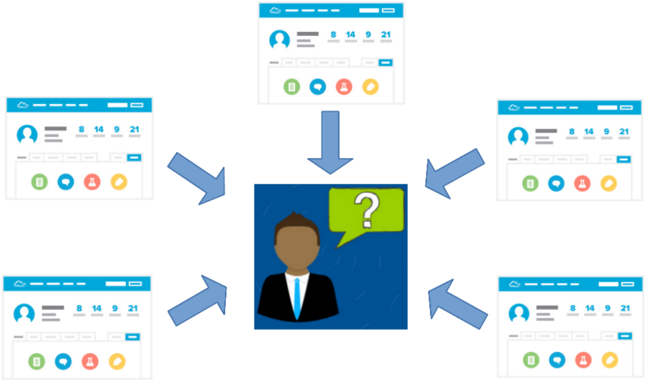
Crowdsourcing



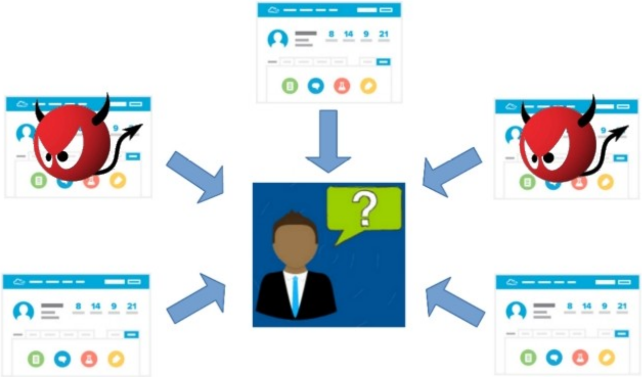
Using data from multiple labs



Collecting data from online sources



Collecting data from online sources



How much can be learned even if some data is corrupted or manipulated?

Refresher: Supervised Learning

Setting:

- Inputs: $x \in \mathcal{X}$, e.g. strings, images, vectors, ...
- Outputs: $y \in \mathcal{Y}$. For simplicity, we use $\mathcal{Y} = \{\pm 1\}$. (binary classification)
- Probability distribution: $p(x, y)$ over $\mathcal{X} \times \mathcal{Y}$, unknown to the learner
- Loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. For simplicty, we use 0/1-loss: $\ell(y, \bar{y}) = [\![y \neq \bar{y}]\!]$

Abstract Goal:

- find a prediction function, $f : \mathcal{X} \to \mathcal{Y}$, such that the expected number of errors

$$\mathrm{er}(h) = \mathbb{E}_{(x,y)\sim p}\big([\![f(x) \neq y]\!]\big) = \mathrm{Pr}_{(x,y)\sim p}\{f(x) \neq y\}$$

on *future data* is small.

Learning from data:

- training data: $S = \{(x_1, y_1), \ldots, (x_m, y_m)\} \overset{i.i.d.}{\sim} p$
- model class: $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$, e.g.
  - $\mathcal{H} =$ *"all linear classifiers"*, $\mathcal{H} =$ *"all neural networks of a fixed architecture"*, ...
- learning algorithm $\mathcal{L} : \mathbb{P}(\mathcal{X} \times \mathcal{Y}) \to \mathcal{H}$, $\qquad \mathbb{P}(\cdot) =$ power set
  - input: a training set, $S \subset \mathcal{X} \times \mathcal{Y}$,
  - output: a trained model $\mathcal{L}(S) \in \mathcal{H}$ (= prediction function).

Learning from data:

- training data: $S = \{(x_1, y_1), \dots, (x_m, y_m)\} \overset{i.i.d.}{\sim} p$
- model class: $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$, e.g.
  - $\mathcal{H} =$ *"all linear classifiers"*, $\mathcal{H} =$ *"all neural networks of a fixed architecture"*, ...
- learning algorithm $\mathcal{L} : \mathbb{P}(\mathcal{X} \times \mathcal{Y}) \to \mathcal{H}$, $\mathbb{P}(\cdot) =$ power set
  - input: a training set, $S \subset \mathcal{X} \times \mathcal{Y}$,
  - output: a trained model $\mathcal{L}(S) \in \mathcal{H}$ (= prediction function).

Central question in statistical learning theory:

Is there a universal learning algorithm, such that: $\mathrm{er}(\mathcal{L}(S)) \overset{|S|\to\infty}{\to} \min_{h\in\mathcal{H}} \mathrm{er}(h)$?

Learning from data:

- training data: $S = \{(x_1, y_1), \dots, (x_m, y_m)\} \overset{i.i.d.}{\sim} p$
- model class: $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$, e.g.
  - $\mathcal{H} =$ *"all linear classifiers"*, $\mathcal{H} =$ *"all neural networks of a fixed architecture"*, ...
- learning algorithm $\mathcal{L} : \mathbb{P}(\mathcal{X} \times \mathcal{Y}) \to \mathcal{H}$, $\qquad \mathbb{P}(\cdot) =$ power set
  - input: a training set, $S \subset \mathcal{X} \times \mathcal{Y}$,
  - output: a trained model $\mathcal{L}(S) \in \mathcal{H}$ (= prediction function).

Central question in statistical learning theory:

Is there a universal learning algorithm, such that: $\mathrm{er}(\mathcal{L}(S)) \overset{|S|\to\infty}{\to} \min_{h\in\mathcal{H}} \mathrm{er}(h)$?

Classic result: If and only if $\mathbf{VC}(\mathcal{H}) < \infty$: empirical risk minimization (ERM) works

$$\mathcal{L}(S) \leftarrow \operatorname*{argmin}_{h\in\mathcal{H}} \widehat{\mathrm{er}}(h) \qquad \text{for } \widehat{\mathrm{er}}(h) := \frac{1}{m}\sum_{(x,y)\in S} [\![f(x) \neq y]\!].$$

Learning from unreliable/malicious data:

- training set: $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- but: data has issues: some of the data points might not really be samples from p
  - e.g. sensor problems, transmission errors, numeric problems, sloppy annotators, online trolls, annotator bias, translation issues, adversarial examples, ...

Learning from unreliable/malicious data:

- training set: $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- but: data has issues: some of the data points might not really be samples from p
  - e.g. sensor problems, transmission errors, numeric problems, sloppy annotators, online trolls, annotator bias, translation issues, adversarial examples, ...
- formally: adversary $\mathcal{A}$ that can manipulate a fraction α of the dataset
  - input: dataset S
  - output: dataset S' with $\lceil (1-\alpha)m \rceil$ points are unchanged and $\lfloor \alpha m \rfloor$ are arbitrary

Question: Is ERM still be a universally good learning strategy?

Learning from unreliable/malicious data:

- training set: $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$
- but: data has issues: some of the data points might not really be samples from p
  - e.g. sensor problems, transmission errors, numeric problems, sloppy annotators, online trolls, annotator bias, translation issues, adversarial examples, ...
- formally: adversary $\mathcal{A}$ that can manipulate a fraction α of the dataset
  - input: dataset S
  - output: dataset S' with $\lceil (1-\alpha)m \rceil$ points are unchanged and $\lfloor \alpha m \rfloor$ are arbitrary

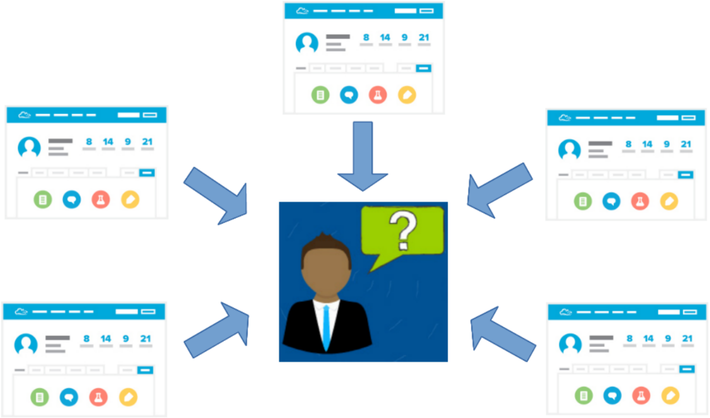
Question: Is ERM still be a universally good learning strategy?

Classic Result: no! [Kerns&Li, 1993]

No learning algorithm can always guarantee an error less than $\frac{\alpha}{1-\alpha}$ on future data!

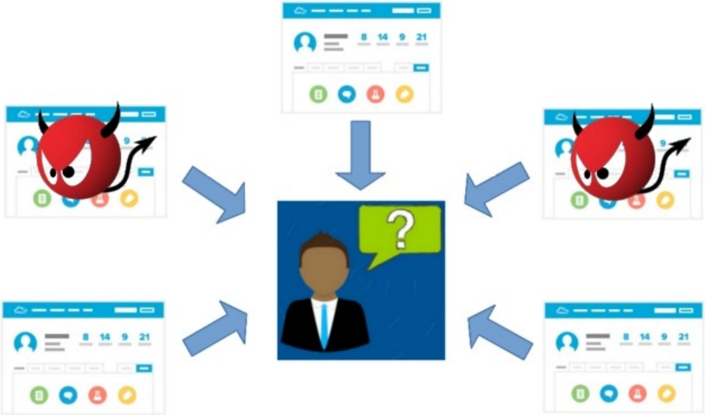
[M. Kearns, M. Li. "Learning in the presence of malicious errors". SIAM Journal on Computing, 1993]

Learning from Multiple Sources

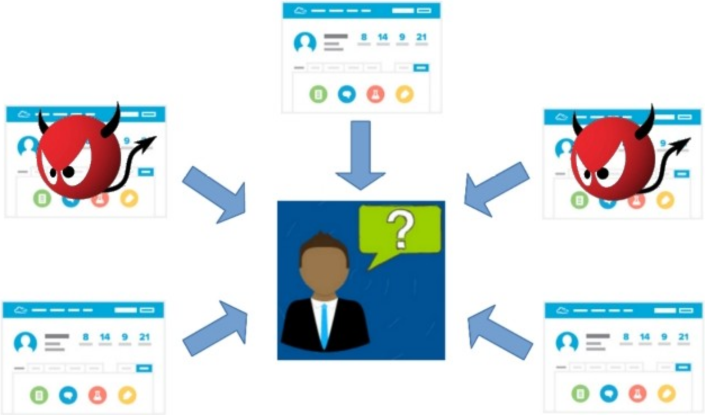


If all sources are i.i.d. samples from the correct data distribution

- naive strategy "merge all datasets and minimize training error" is guaranteed to work.

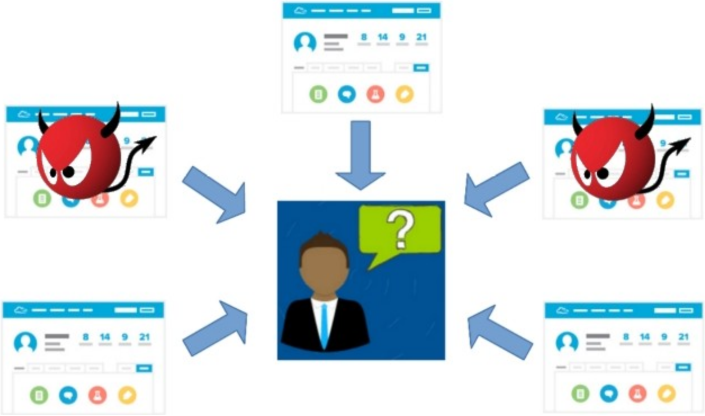


What, if some sources are not reliable?



What, if some sources are not reliable?

- $2/5$ of data malicious: naive strategy can be worse than random guessing! er $\geq 66\%$.
- $1/10$ of data malicious: er $\geq 11\%$. Still pretty bad!



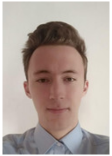
What, if some sources are not reliable?

- $2/5$ of data malicious: naive strategy can be worse than random guessing! er $\geq 66\%$.
- $1/10$ of data malicious: er $\geq 11\%$. Still pretty bad!

Is there a better algorithm? Is there a universal one?

Robust Learning from Unreliable or Malicious Sources

Nikola Konstantinov



Elias Frantar

Dan Alistarh

Disclaimer: "These results have been modified from their original form. They have been edited to fit the screen and the allotted time slot."

[N. Konstantinov, E. Frantar, D. Alistarh, CHL. *"On the Sample Complexity of Adversarial Multi-Source PAC Learning"*, ICML 2020]
[N. Konstantinov, CHL. *"Robust Learning from Untrusted Sources"*, ICML 2019]

Learning from Multiple Sources

- multiple training sets $S_1, S_2, \ldots, S_N$
  - each $S_i = \{(x_1^i, y_1^i), \ldots, (x_m^i, y_m^i)\} \overset{i.i.d.}{\sim} p$
- multi-source learning algorithm $\mathcal{L} : (\mathcal{X} \times \mathcal{Y})^{N \times m} \to \mathcal{H}$
  - input: training sets, $S_1, S_2, \ldots, S_N$
  - output: one hypothesis $\mathcal{L}(S_1, \ldots, S_N) \in \mathcal{H}$ (= a trained model).

Learning from Multiple Unreliable/Malicious Sources

- multiple training sets $S_1, S_2, \dots, S_N$
  - each $S_i = \{(x_1^i, y_1^i), \dots, (x_m^i, y_m^i)\} \overset{i.i.d.}{\sim} p$
- multi-source learning algorithm $\mathcal{L} : (\mathcal{X} \times \mathcal{Y})^{N \times m} \to \mathcal{H}$
  - input: training sets, $S'_1, S'_2, \dots, S'_N = \mathcal{A}(S_1, \dots, S_N)$
  - output: one hypothesis $\mathcal{L}(S'_1, S'_2, \dots, S'_N) \in \mathcal{H}$ (= a trained model).
- adversary $\mathcal{A}$
  - input: data sets $S_1, \dots, S_N$
  - output: data sets $S'_1, \dots, S'_N$,
    of which $\lceil (1-\alpha)N \rceil$ are identical to before and $\lfloor \alpha N \rfloor$ are arbitrary
  - the adversary knows the training algorithm
  - two variants: fixed subset of datasets that can be perturbed, or adversary can chose

Learning from Multiple Unreliable/Malicious Sources

- multiple training sets $S_1, S_2, \dots, S_N$
  - each $S_i = \{(x_1^i, y_1^i), \dots, (x_m^i, y_m^i)\} \overset{i.i.d.}{\sim} p$
- multi-source learning algorithm $\mathcal{L} : (\mathcal{X} \times \mathcal{Y})^{N \times m} \to \mathcal{H}$
  - input: training sets, $S'_1, S'_2, \dots, S'_N = \mathcal{A}(S_1, \dots, S_N)$
  - output: one hypothesis $\mathcal{L}(S'_1, S'_2, \dots, S'_N) \in \mathcal{H}$ (= a trained model).
- adversary $\mathcal{A}$
  - input: data sets $S_1, \dots, S_N$
  - output: data sets $S'_1, \dots, S'_N$,
    of which $\lceil (1-\alpha)N \rceil$ are identical to before and $\lfloor \alpha N \rfloor$ are arbitrary
  - the adversary knows the training algorithm
  - two variants: fixed subset of datasets that can be perturbed, or adversary can chose

Is there a universal learning algorithm, such that: $\mathrm{er}(\mathcal{L}(S'_1, \dots, S'_N)) \overset{m \to \infty}{\to} \min_{h \in \mathcal{H}} \mathrm{er}(h)$?

Robust Multi-Source Learning: Related Work

Robust learning from a single dataset

- no universal algorithm: minimum guaranteeable error is $\frac{\alpha}{1-\alpha}$ [Kearns and Li, 1993]
- identical to our situation when each dataset consists of a single point, $m = 1$

$\Rightarrow$ only $N \to \infty$ will probably not be enough to learn arbitrarily well

Robust learning from a single dataset

- no universal algorithm: minimum guaranteeable error is $\frac{\alpha}{1-\alpha}$ [Kearns and Li, 1993]
- identical to our situation when each dataset consists of a single point, $m = 1$

  $\Rightarrow$ only $N \to \infty$ will probably not be enough to learn arbitrarily well

Collaborative learning (multiple parties learn *one model each*)

- universal learning algorithm exists [Blum et al., 2017], [Qiao, 2018]

Robust learning from a single dataset

- no universal algorithm: minimum guaranteeable error is $\frac{\alpha}{1-\alpha}$ [Kearns and Li, 1993]
- identical to our situation when each dataset consists of a single point, $m = 1$

  $\Rightarrow$ only $N \to \infty$ will probably not be enough to learn arbitrarily well

Collaborative learning (multiple parties learn *one model each*)

- universal learning algorithm exists [Blum et al., 2017], [Qiao, 2018]

Density estimation from untrusted batches

- possible, but not the same as our setting [Qiao and Valiant, 2018],[Jain and Orlitsky, 2020]

Robust learning from a single dataset

- no universal algorithm: minimum guaranteable error is $\frac{\alpha}{1-\alpha}$ [Kearns and Li, 1993]
- identical to our situation when each dataset consists of a single point, $m = 1$
  $\Rightarrow$ only $N \to \infty$ will probably not be enough to learn arbitrarily well

Collaborative learning (multiple parties learn *one model each*)

- universal learning algorithm exists [Blum et al., 2017], [Qiao, 2018]

Density estimation from untrusted batches

- possible, but not the same as our setting [Qiao and Valiant, 2018],[Jain and Orlitsky, 2020]

Byzantine-robust distributed optimization

- specific solutions for gradient-based optimization [Yin et al., 2018], [Alistarh et al., 2018]
- results focus on convergence analysis under convexity/smoothness assumptions

Robust Multi-Source Learning: Our Result

Theorem [N. Konstantinov, E. Frantar, D. Alistarh, CHL. ICML 2020]

There exists a learning algorithm, $\mathcal{L}$, such that with high probability:

$$\mathrm{er}(\,\mathcal{L}(S'_1,\dots,S'_N)\,) \leq \min_{h\in\mathcal{H}} \mathrm{er}(h) + \underbrace{\widetilde{\mathcal{O}}\Big(\frac{1}{\sqrt{(1-\alpha)Nm}} + \alpha\frac{1}{\sqrt{m}}\Big)}_{\to 0 \text{ for } m=|S|\to\infty},$$

with $S'_1,\dots,S'_N = \mathcal{A}(S_1,\dots,S_N)$ for any adversary $\mathcal{A}$ with $\alpha < \frac{1}{2}$.

($\widetilde{\mathcal{O}}$-notation hides constant and logarithmic factors)

Question: why is learning easier from multiple sources than from a single source?

Answer: it's not. But the task for the adversary is harder!

- single source: no restrictions how to manipulate the data
- multi-source: manipulation must adhere to the source structure

Algorithm idea: exploit law of large numbers

- majority of datasets are unperturbed
- for $m \to \infty$ these start to look more and more similar
- we can identify (at least) the unperturbed datasets
- we perform ERM only on those

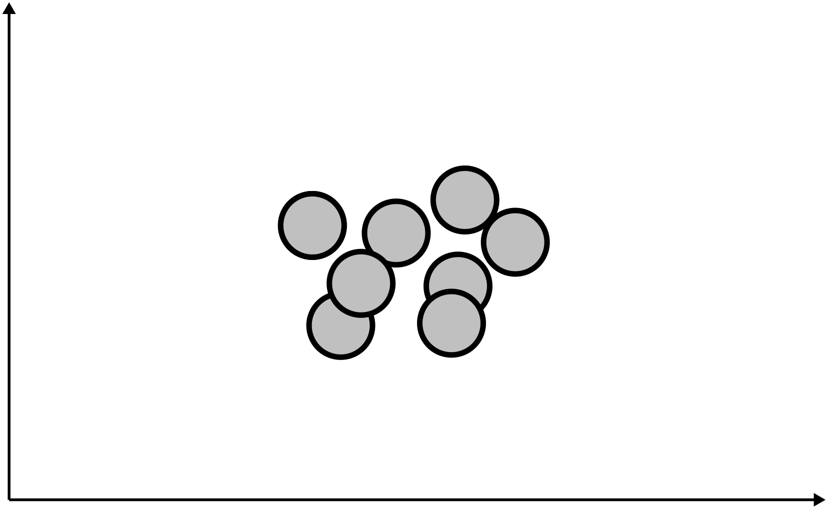
Robust Multi-Source Learning: Algorithm

Robust multi-source learning algorithm:

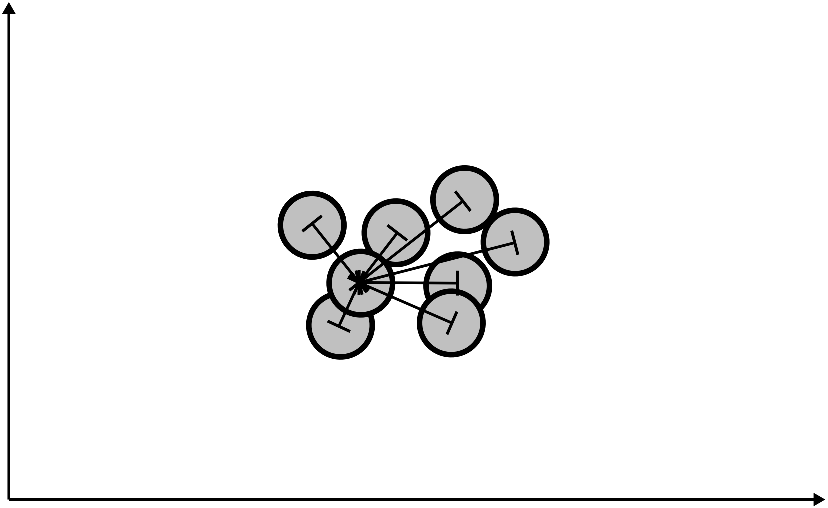
- Step 1) identify which sources to trust
  - compute all pairwise distance d_{ij} between datasets $S'_1, \ldots, S'_N$ (with a suitable distance measure d)
  - for any i: if $d_{ij} < \theta$ for at least $\lfloor \frac{N}{2} \rfloor$ values of $j \neq i$, then $T \leftarrow T \cup \{i\}$ (with a suitable threshold θ)
- Step 2) create a new dataset $\tilde{S}$ by merging data from all sources S_i with $i \in T$
- Step 3) minimize training error on $\tilde{S}$

Open choices:

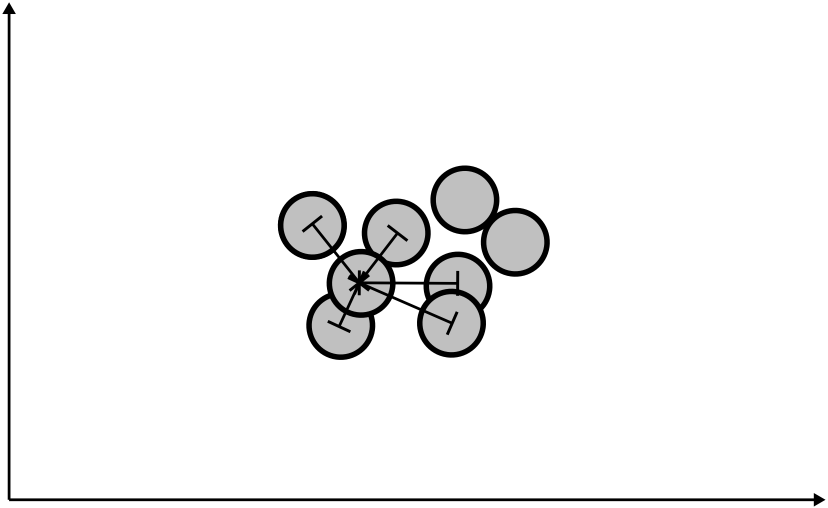
- distance measure d (discussed later)
- threshold θ (not discussed, see paper)



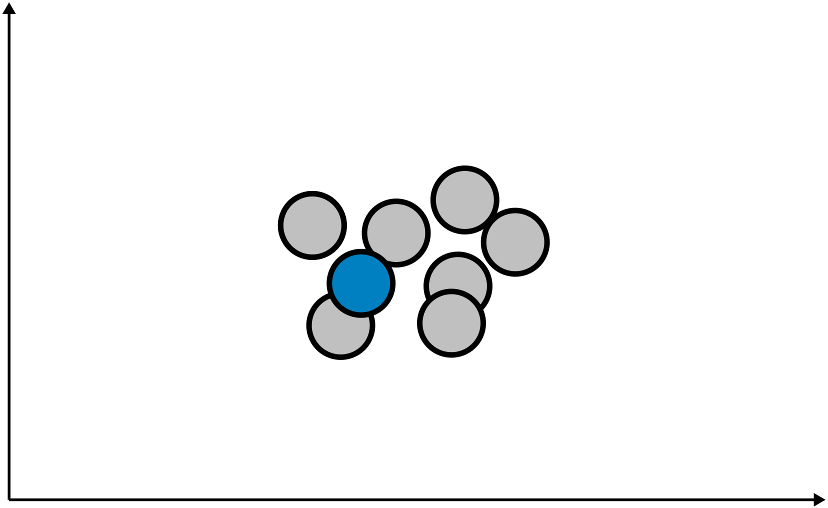
All datasets clean



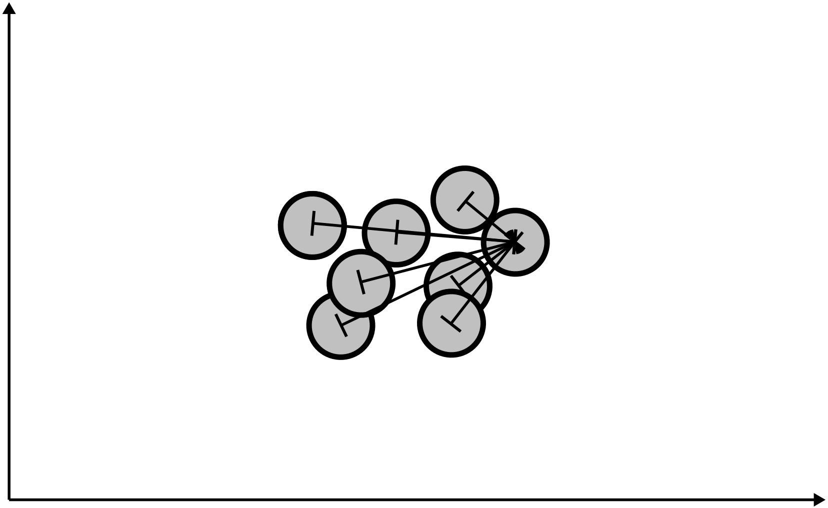
All datasets clean



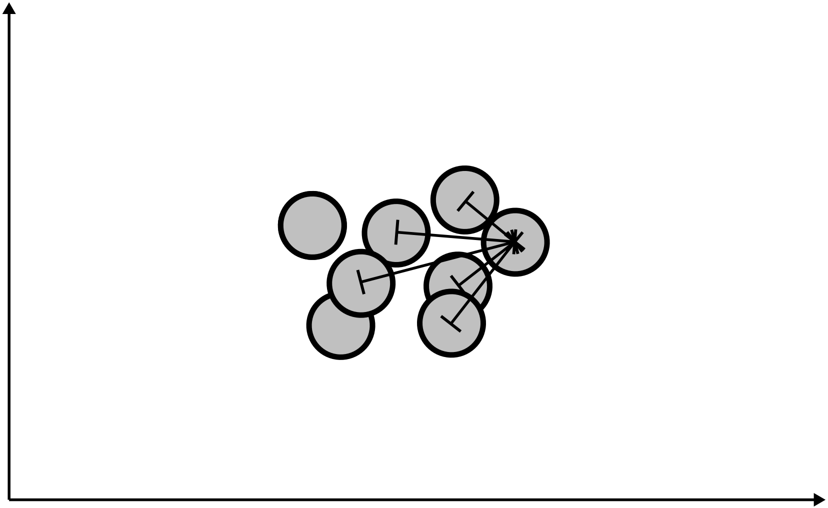
All datasets clean



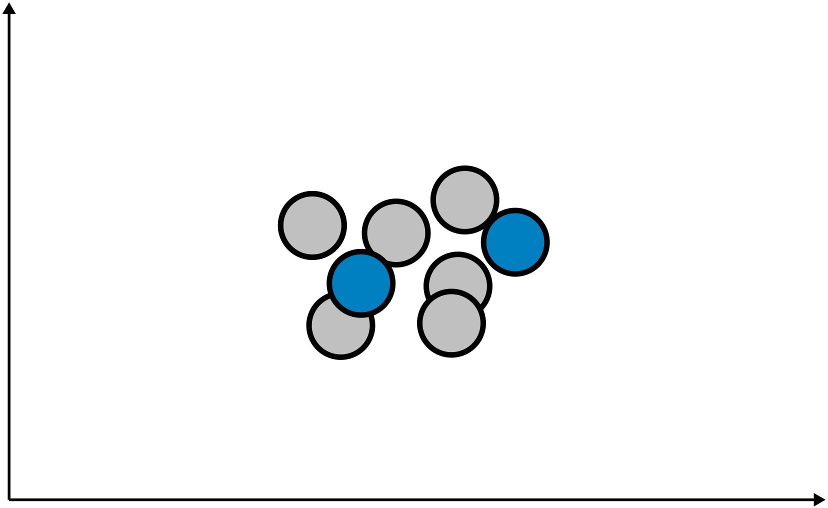
All datasets clean



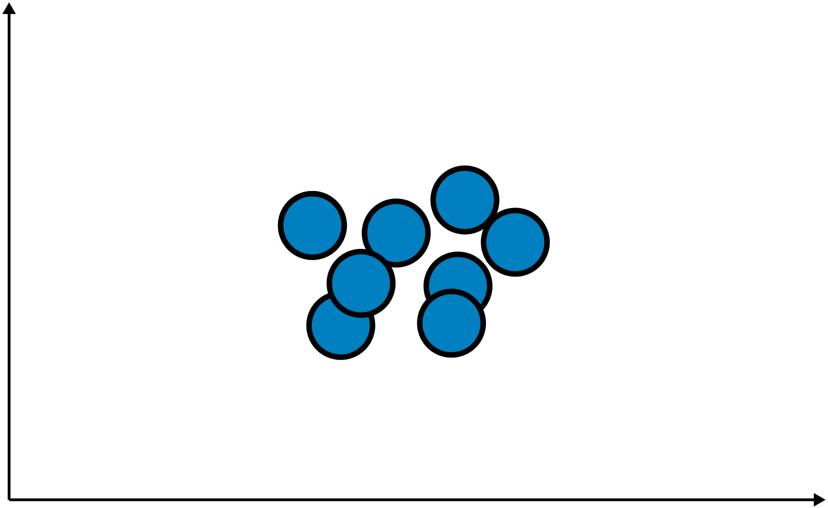
All datasets clean



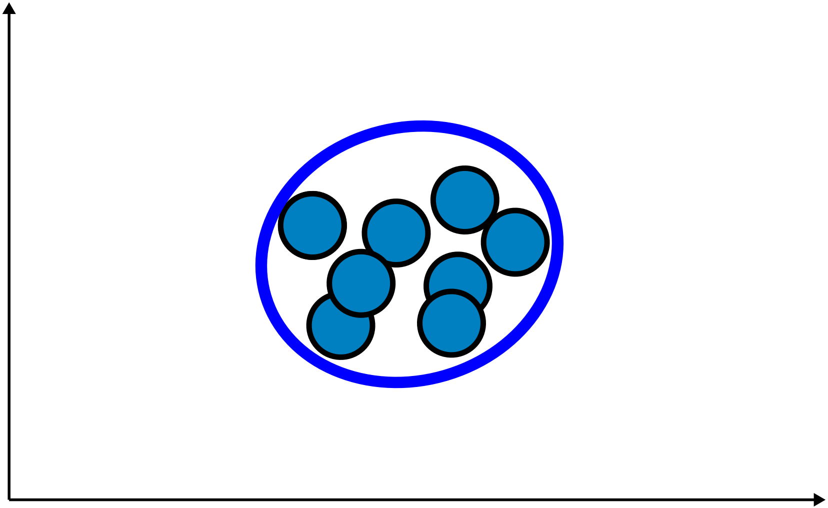
All datasets clean



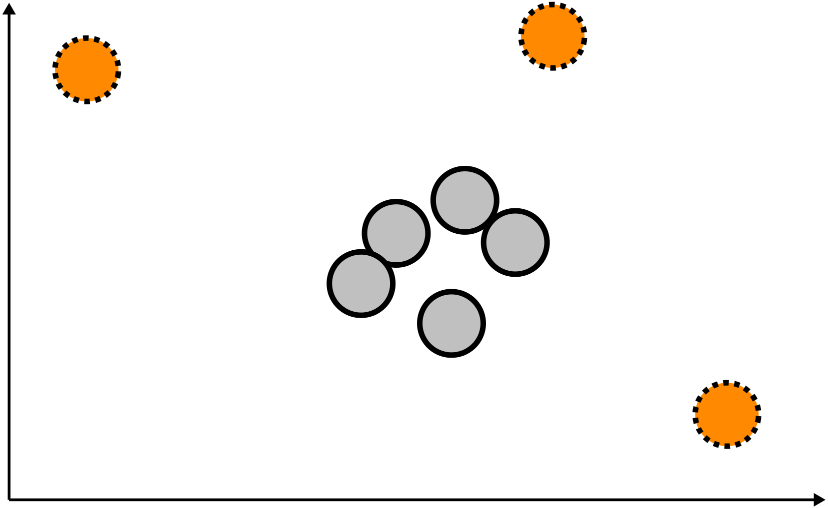
All datasets clean



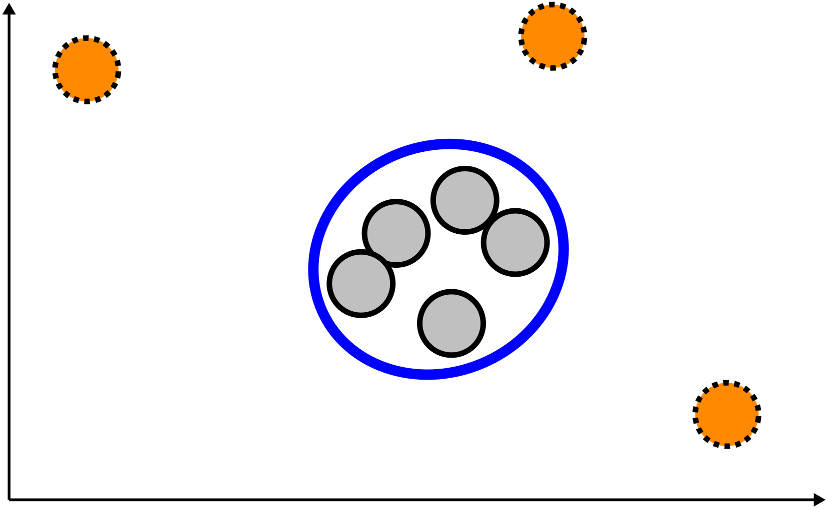
All datasets clean



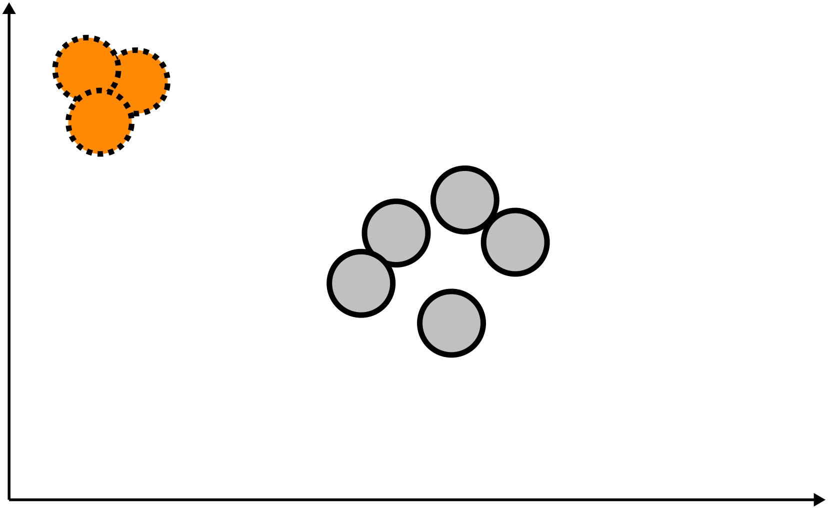
All datasets clean $\rightarrow$ all datasets included $\rightarrow$ same as (optimal) naive algorithm



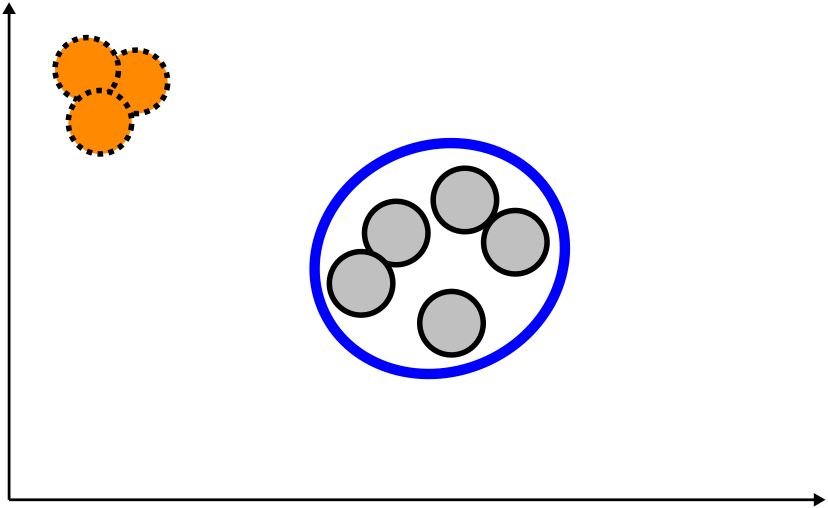
Some datasets manipulated



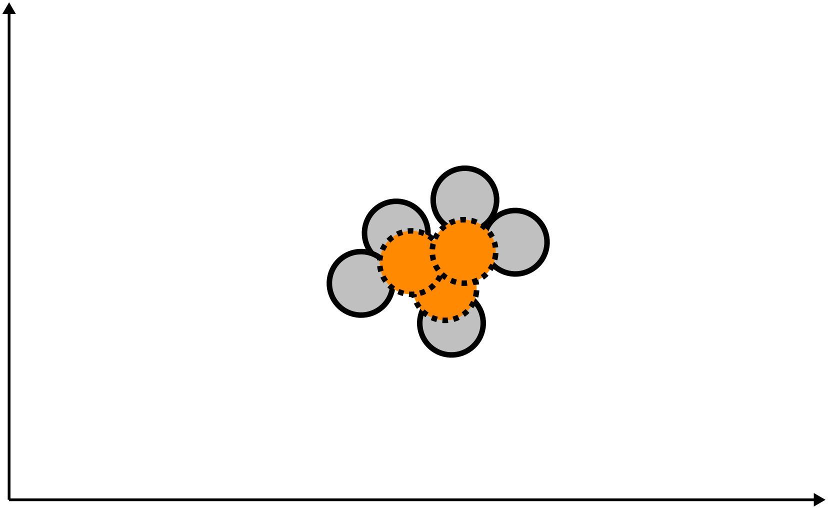
Some datasets manipulated $\rightarrow$ manipulated datasets excluded.



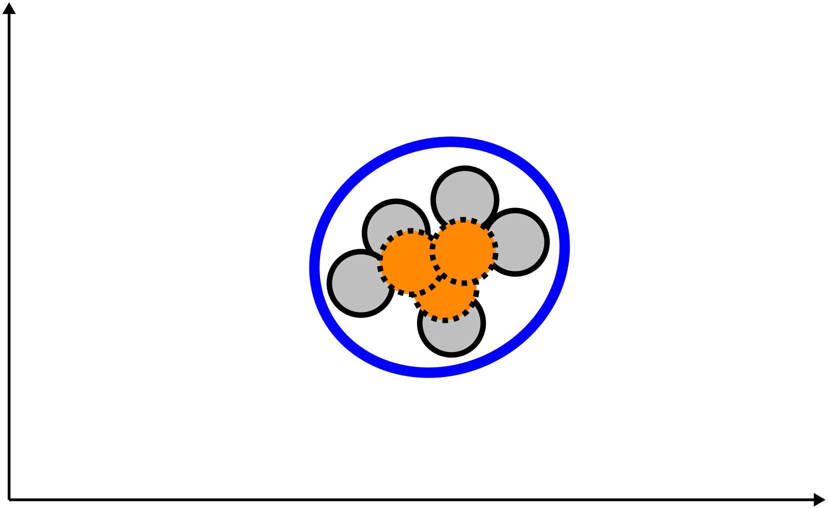
Some datasets manipulated in a consistent way



Some datasets manipulated in a consistent way → manipulated datasets excluded.



Some datasets manipulated to look like originals



Some datasets manipulated to look like originals $\rightarrow$ <u>all</u> datasets included.

Analysis: what properties does the distance measure d need?

Analysis: what properties does the distance measure d need?

1) S and S' are sampled from the same distribution $\Rightarrow$ $d(S, S')$ should be small (at least, if enough samples are available)

$\rightarrow$ 'clean' datasets will eventually get grouped together.

Analysis: what properties does the distance measure d need?

1) S and S' are sampled from the same distribution $\quad\Rightarrow\quad$ $d(S, S')$ should be small (at least, if enough samples are available)

$\rightarrow$ 'clean' datasets will eventually get grouped together.

2) $d(S, S')$ is small $\quad\Rightarrow\quad$ $\mathcal{L}(S') \approx \mathcal{L}(S)$

$\rightarrow$ if manipulated datasets are groups with the clean ones, they don't hurt the learning.

Analysis: what properties does the distance measure d need?

1) S and S' are sampled from the same distribution $\Rightarrow$ $d(S, S')$ should be small (at least, if enough samples are available)

$\rightarrow$ 'clean' datasets will eventually get grouped together.

2) $d(S, S')$ is small $\Rightarrow$ $\mathcal{L}(S') \approx \mathcal{L}(S)$

$\rightarrow$ if manipulated datasets are groups with the clean ones, they don't hurt the learning.

Observation:

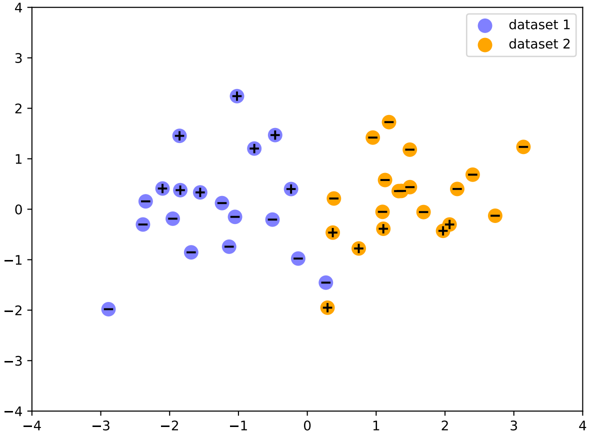
- many candidate distances do not fulfill both conditions simultaneously:
  - geometric: average Euclidean distance, Chamfer distance, Haussdorf distance, ...
  - probabilistic: Wasserstein distance, total variation, Kullback-Leibler divergence, ...
- discrepancy distance does fulfill the conditions!

Discrepancy Distance [Mansour *et al.* 2009]

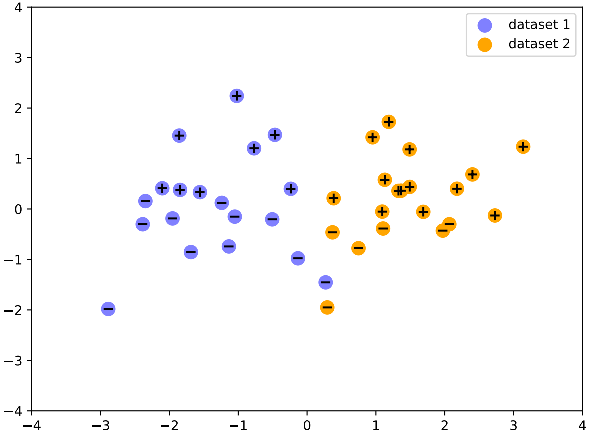
For a set of classifiers $\mathcal{H}$ and datasets S_i, S_j, define

$$\mathrm{disc}(S_i, S_j) = \max_{h \in \mathcal{H}} \left| \widehat{\mathrm{er}}_{S_i}(h) - \widehat{\mathrm{er}}_{S_j}(h) \right| .$$

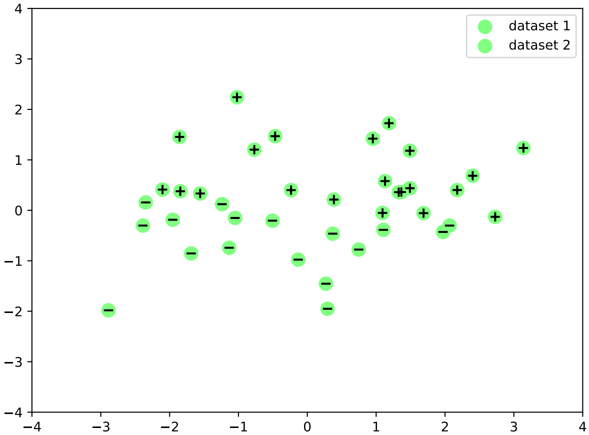
- maximal amount any classifier, $h \in \mathcal{H}$, can disagree between S_i, S_j
- for binary classification, discrepancy can be computed by training a classifier:
  - $S_j^{\pm} \;\leftarrow\; S_j$ with all ± 1 labels flipped to their opposites
  - $\tilde{S} \;\leftarrow\; S_i \cup S_j^{\pm}$
  - $\mathrm{disc}(S_i, S_j) \;\leftarrow\; 1 - 2\min_{h \in \mathcal{H}} \widehat{\mathrm{er}}_{\tilde{S}}(h)$ (minimal training error of any $h \in \mathcal{H}$ on $\tilde{S}$)



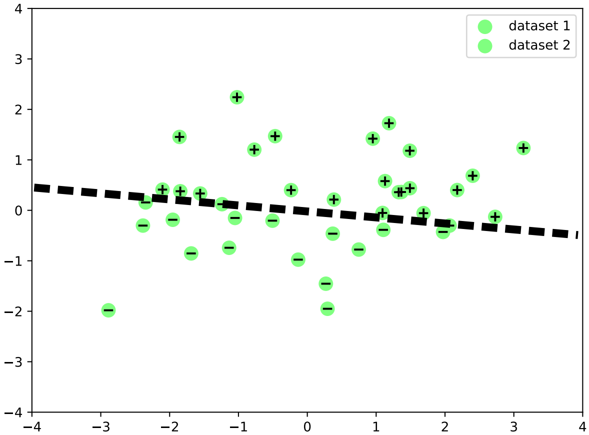
Two datasets, S_i, S_j



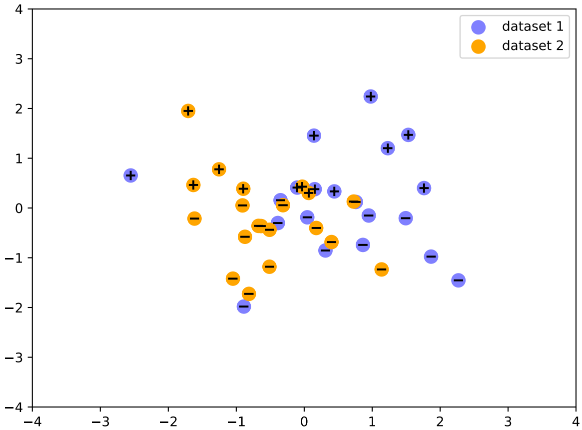
Flip signs of S_j



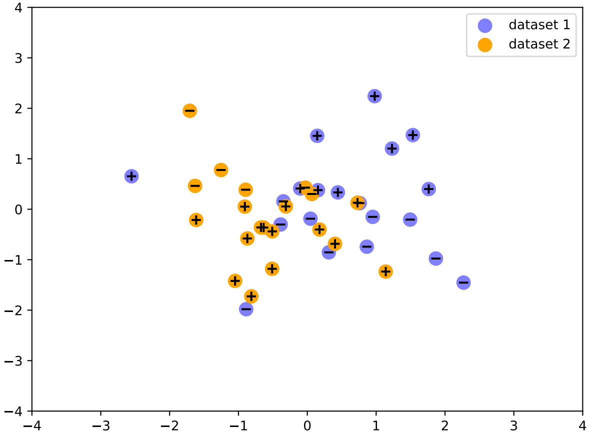
Merge both datasets



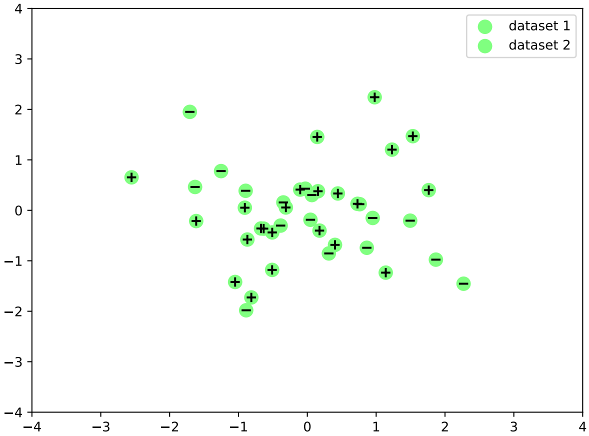
Classifier with small training error $\rightarrow$ large discrepancy



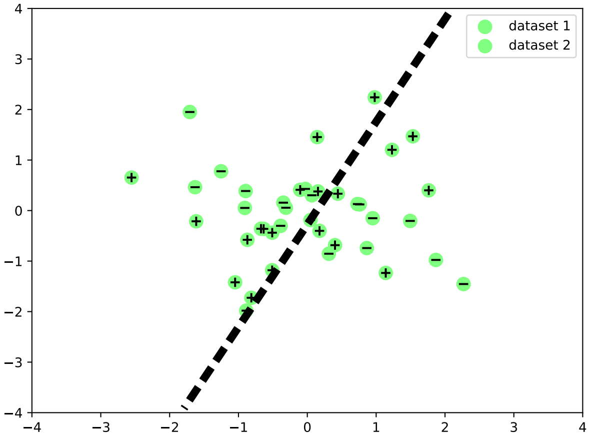
Two datasets, S_i, S_j



Flip signs of S_j



Merge both datasets



No classifier with small training error $\rightarrow$ small discrepancy

Observation: discrepancy distance has both property we need:

1) Datasets from the same distribution (eventually) gets grouped together
   - if S_i and S_j are sampled from the same distribution, then

$$\mathrm{disc}(S_i, S_j) \to 0 \quad \text{for} \quad |S_i|, |S_j| \to \infty$$

2) Datasets that are grouped together do not hurt the learning (much)

   Assume:
   - training set $S_{\text{trn}} \overset{i.i.d.}{\sim} p$
   - arbitrary set S', potentially manipulated but with $\mathrm{disc}(S_{\text{trn}}, S') \leq \theta$
   - test set $S_{\text{tst}} \overset{i.i.d.}{\sim} p$

Then, for every $h \in \mathcal{H}$:

$$\widehat{\text{er}}_{S_{\text{tst}}}(h) \leq \widehat{\text{er}}_{S'}(h) + \underbrace{\mathrm{disc}(S_{\text{trn}}, S')}_{\leq \theta} + \underbrace{\mathrm{disc}(S_{\text{trn}}, S_{\text{tst}})}_{\text{small by prop. 1)}}$$

Robust Multi-Source Learning: Final Result

Theorem [N. Konstantinov, E. Frantar, D. Alistarh, CHL. ICML 2020]

Let $S_1, \dots, S_N$ are training sets of size m, out of which at most $N-k$ can be arbitrarily manipulated (so k datasets are not manipulated). Denote $\alpha = \frac{N-k}{N}$. Let h^* be the result of the robust multi-source learning algorithm. Then

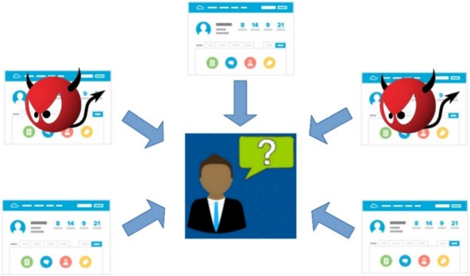
$$\mathrm{er}(h^*) \leq \min_{h\in\mathcal{H}} \mathrm{er}(h) + \underbrace{\widetilde{\mathcal{O}}\Big(\frac{1}{\sqrt{km}} + \alpha\frac{1}{\sqrt{m}}\Big)}_{\to 0 \text{ for } m\to\infty},$$

($\widetilde{\mathcal{O}}$-notation hides constant and logarithmic factors)

Discussion:

- km is the number of "clean" samples $\to \frac{1}{\sqrt{km}}$ is the "normal" speed of learning
- $\alpha\frac{1}{\sqrt{m}}$ is a slow-down due to α-manipulation
- lower bounds exists that show that $O(\alpha\frac{1}{\sqrt{m}})$ slowdown is unavoidable

Summary



- Learning from multiple unreliable sources now commonplace
- Can be studied formally: learning with an adversary of a certain power
- Group structure enables statistical learnability, even against a strong adversary
- Unfortunately: no statement about computational efficiency

Thanks to. . .

My research group and collaborators:

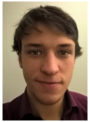

Paul Henderson


Niko Konstantinov

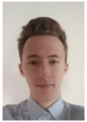

Alex Peste


Dan Alistarh


Mary Phuong


Bernd Prach


Amélie Royer


Elias Frantar

Funding Sources:

IST AUSTRIA
Institute of Science and Technology

erc

D. Alistarh, Z. Allen-Zhu, and J. Li. Byzantine stochastic gradient descent. In *NeurIPS*, 2018.

A. Blum, N. Haghtalab, A. D. Procaccia, and M. Qiao. Collaborative pac learning. In *NIPS*. 2017.

A. Jain and A. Orlitsky. Optimal robust learning of discrete distributions from batches. In *ICML*, 2020.

M. Kearns and M. Li. Learning in the presence of malicious errors. In *SIAM Journal on Computing*, 1993.

N. Konstantinov and C. H. Lampert. Robust learning from untrusted sources. In *ICML*, 2019.

N. Konstantinov, E. Frantar, D. Alistarh, and C. H. Lampert. On the sample complexity of adversarial multi-source PAC learning. In *ICML*, 2020.

M. Qiao. Do outliers ruin collaboration? In *ICML*, 2018.

M. Qiao and G. Valiant. Learning discrete distributions from untrusted batches. In *ITCS*, 2018.

D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *ICML*, 2018.