ESSCaSS 2022

Fair and Robust Machine Learning – Part 2

Christoph Lampert

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
- Interpretability
- 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
- Interpretability
- Abstract Reasoning
- Semantic Representations

Schedule

Lecture 1: Intro to Machine Learning

Lecture 1: Robust Machine Learning

Lecture 1: Fair Machine Learning

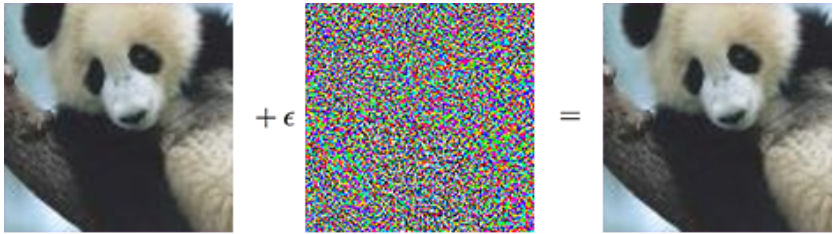
Lecture 2: Certified Robustness via Lipschitz Networks

Lecture 2: Robust and Fair Learning from Multiple Sources

Lecture 3: Behind the Scenes of (Machine Learning) Research

Certified Robustness via Lipschitz Networks

Reminder: neural networks are prone to adversarial examples.



- adding a tiny amount of adversarially constructed noise can change the model output

How to prevent adversarial examples

Add adversarial examples to the training set

Problem: does not work, new adversarial images emerge

Optimize robustified training error

Problem: optimization can't solve exactly, approximations leave vulnarabilities open

Prefilter input before applying the model, e.g. Gaussian smoothing

Problem: either loss of accuracy or prone to adversarial examples itself

Robust ensemble of randomized models

Problem: amount of randomization unclear, high computational cost to get guarantees

Alternative: use model architecture that guarantees no adversarial examples exist

Alternative: use model architecture that guarantees no adversarial examples exist

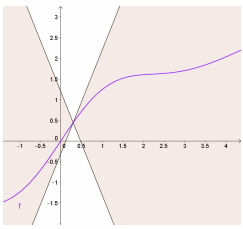
Setting: multi-class classification

- inputs $\mathcal{X} \subset \mathbb{R}^d$, outputs $\mathcal{Y} = \{1, \dots, K\}$
- model $g : \mathcal{X} \to \mathbb{R}^K$, from which we make predictions $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} g(x)_y$

Definition (Lipschitz constant)

A function $g : \mathbb{R}^d \to \mathbb{R}^k$ is called L-Lipschitz continuous, if

$$\|g(x) - g(x')\|_{\mathbb{R}^k} \leq L \|x - x'\|_{\mathbb{R}^d}$$



Note: for differentiable g, we can take $L = \left\|J_g(x)\right\|_2$ (operator norm of the Jacobian matrix).

Image: By Taschee - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=59500064

Alternative: use model architecture that guarantees no adversarial examples exist

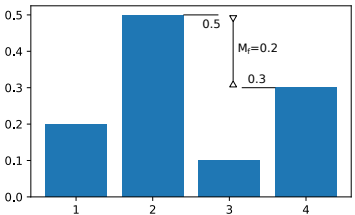
Setting: multi-class classification

- inputs $\mathcal{X} \subset \mathbb{R}^d$, outputs $\mathcal{Y} = \{1, \ldots, K\}$
- model $g : \mathcal{X} \to \mathbb{R}^K$, from which we make predictions $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} g(x)_y$

Definition (Margin)

For an example (x, y), the margin of a model g is defined as

$$M_g(x, y) = \begin{cases} g(x)_y - \max\limits_{z \neq y} g(x)_z & \text{if } \operatorname*{argmax}\limits_{z \in \mathcal{Y}} g(x)_z = y, \\ 0 & \text{otherwise.} \end{cases}$$



Alternative: use model architecture that guarantees no adversarial examples exist

Setting: multi-class classification

- inputs $\mathcal{X} \subset \mathbb{R}^d$, outputs $\mathcal{Y} = \{1, \ldots, K\}$
- model $g : \mathcal{X} \to \mathbb{R}^K$, from which we make predictions $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} g(x)_y$

Definition (Certified Robust Accuracy)

A model g with decision function $f(x) = \operatorname{argmax}_y g(x)_y$ is said to classify an example (x, y) ϵ-certified robustly if

$$f(x + \delta) = y \quad \text{for all } \delta \text{ with } \|\delta\| \leq \epsilon.$$

The ϵ-certified robust accuracy on a dataset S is the fraction of points in S that are ϵ-certified robustly classified.

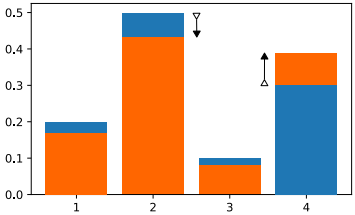
Alternative: use model architecture that guarantees no adversarial examples exist

Setting: multi-class classification

- inputs $\mathcal{X} \subset \mathbb{R}^d$, outputs $\mathcal{Y} = \{1, \ldots, K\}$
- model $g : \mathcal{X} \to \mathbb{R}^K$, from which we make predictions $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} g(x)_y$

Lemma

Let g be a model that is L-Lipschitz continuous. Then g classifies an example (x, y) ϵ-certified robustly, if $M_g(x, y) > \sqrt{2} L \epsilon$.



Lipschitz networks

How to make a network with prescribed Lipschitz constant (e.g. $L = 1$)?

Reminder: neural networks consist of layers

$$f(x) = f^{(L)}(f^{(L-1)}(\dots f^{(2)}(f^{(1)}(x)))) \quad \text{with} \quad f^{(l)}(x) = \sigma(W_l x + b_l) \quad \text{for } l = 1, \dots, L$$

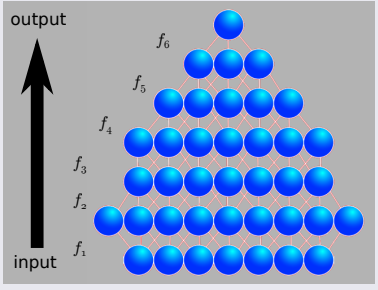
Lipschitz networks

Observation: $\mathrm{Lip}(f) \leq \prod_{l=1}^{L} \mathrm{Lip}(f_l)$

Reminder: $\mathrm{Lip}(f_l) \leq \left\| J_{f_l}(x) \right\|_2 \leq |\sigma'| \|W_l\|_2$

Conclusion: it suffices to choose

- σ with $|\sigma'| \leq 1$, e.g. $\sigma(t) = \max\{0, t\}$
- W_l with $\|W_l\|_2 \leq 1$



But: how to ensure the norm constraint when W_l is learned from data?

Almost-Orthogonal Layers for Efficient General-Purpose Lipschitz Networks

Bernd Prach

[B. Prach, CHL. *"Almost-orthogonal Layers for Efficient General-Purpose Lipschitz Networks"*, ECCV 2022; https://arxiv.org/abs/2208.03160]

$$f(x) = f^{(L)}(f^{(L-1)}(\dots f^{(2)}(f^{(1)}(x)))) \quad \text{with} \quad f^{(l)}(x) = \sigma(W_l x + b_l) \quad \text{for } l = 1, \dots, L$$

Self-normalizing Layers

Main idea: new layer parametrization $W_l = P_l D_l$

- $P_l \in \mathbb{R}^{n_l \times n_{l-1}}$ arbitrary parameter matrix
- $D_l = \text{diag}(d_1, \dots, d_{n_{l-1}})$ with $d_i = \left(\sum_j \left| P_l^\top P_l \right|_{ij} \right)^{-1/2}$

Observation:

- for any P_l, it holds that $\left\| W_l \right\|_2 \leq 1$
  $\rightarrow$ each $f^{(l)}$ is 1-Lipschitz continuous
  $\rightarrow$ the network itself is 1-Lipschitz continuous
- if P_l is orthogonal, $D_l = \text{Id}$ and inequality is tight.

Setting: multi-class classification

- inputs $\mathcal{X} \subset \mathbb{R}^d$, outputs $\mathcal{Y} = \{1, \dots, K\}$
- model $g : \mathcal{X} \to \mathbb{R}^K$, from which we make predictions $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} g(x)_y$
- training set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$

Margin-enforcing loss function

Main idea: use a loss function that enforces a large margin

- write $\vec{y} = \delta_{z=y}(z)$ "one-hot" representation of y

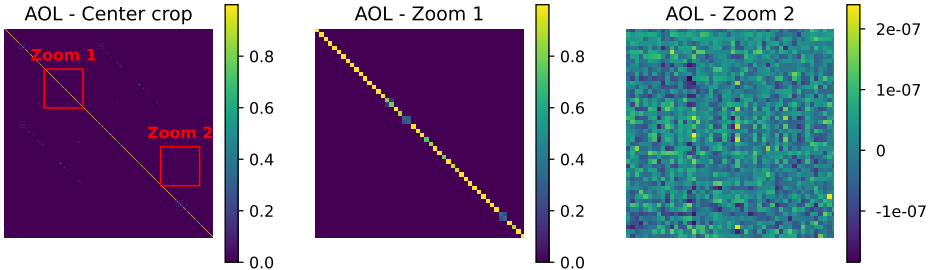
$$\ell(y, g(x)) = \text{crossentropy}\left(\vec{y}, \text{softmax}\left(g(x) - u\vec{y}\right)\right),$$

with offset parameter $u \geq 0$

Observation:

- y-component of $g(x)$ is shifted down by u before computing usual crossentropy-loss.
  $\to$ to achieve small loss, learning must make $g(x)_y$ bigger by u than otherwise

Observation: learned parameter matrices P are close to orthogonal



visualization of $P^\top P$

Reasoning:

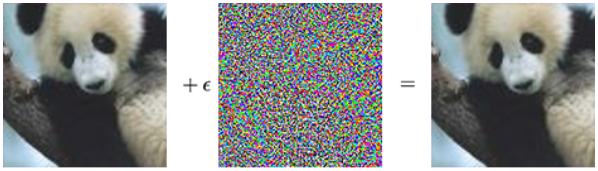
- to achieve low error in training, output values must have large dynamic range
- the normalization step restricts the dynamic range of the layers
- for orthogonal matrices P, the bound is tight and dynamic range is maximal

Almost-Orthogonal Layers (AOL)

Experimental results: image classification on CIFAR-10 dataset. Proposed AOL and methods from the literature.

Method	Standard Accuracy	Certified Robust Accuracy $\epsilon = \frac{36}{255}$	$\epsilon = \frac{72}{255}$	$\epsilon = \frac{108}{255}$	$\epsilon = 1$
Standard CNN	83.4%	0%	0%	0%	0%
BCOP Large [Li et al., 2019]	72.2%	58.3%	-	-	-
GloRo 6C2F [Leino et al., 2021]	77.0%	58.4%	-	-	-
Cayley Large [Trockman and Kolter, 2021]	75.3%	59.2%	-	-	-
SOC-20 [Singla and Feizi, 2021]	76.4%	61.9%	-	-	-
SOC-25 [Yu et al., 2022]	-	60.2%	43.7%	28.6%	-
ECO-25 [Yu et al., 2022]	*75.7%*	*66.1%*	*55.6%*	*45.3%*	-
SOC-15 [Singla et al., 2022]	*76.4%*	*63.0%*	*48.5%*	*35.5%*	-
AOL-Small	69.8%	62.0%	54.4%	47.1%	21.8%
AOL-Medium	71.1%	63.8%	56.1%	48.6%	23.2%
AOL-Large	71.6%	64.0%	56.4%	49.0%	23.7%

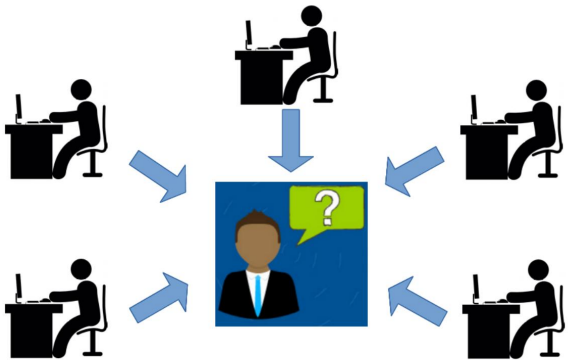
Summary



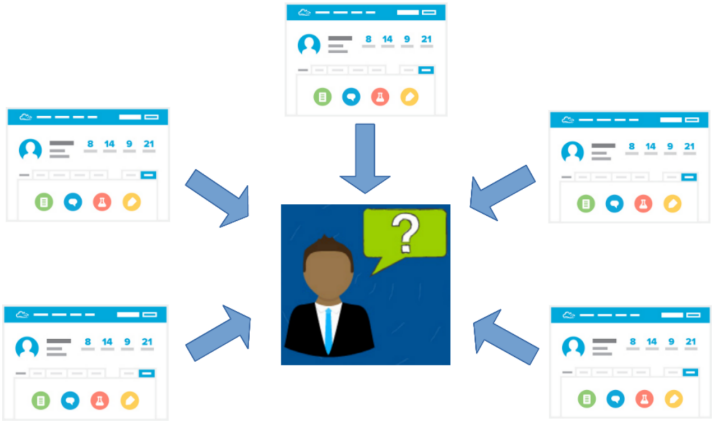
- Most neural networks are vulnerable to adversarial examples.
- Most empirical methods to prevent them do not work very well.
- Lipschitz-networks trained with margin loss can guarantee robustness (though usually at a certain loss of non-robust accuracy).
- AOL [Prach and Lampert, 2022] is easy to use, flexible and works well.

Learning from Multiple Sources

Modern machine learning systems are often trained on data collected from many different sources.

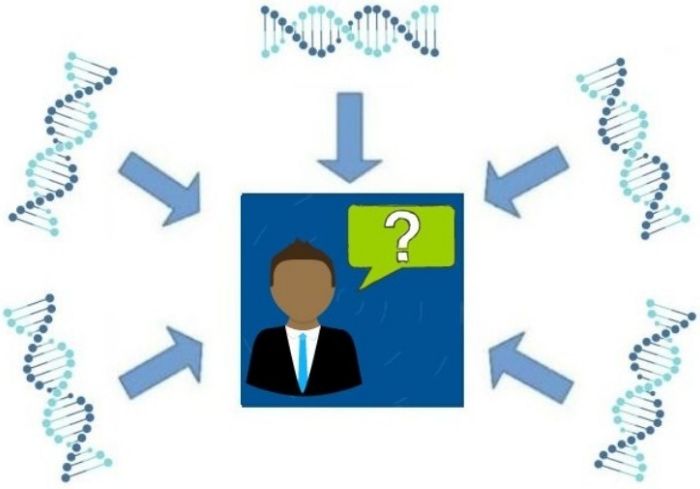


Modern machine learning systems are often trained on data collected from many different sources.



tens of different online resources (Wikipedia, Twitter, Reddit, . . .)

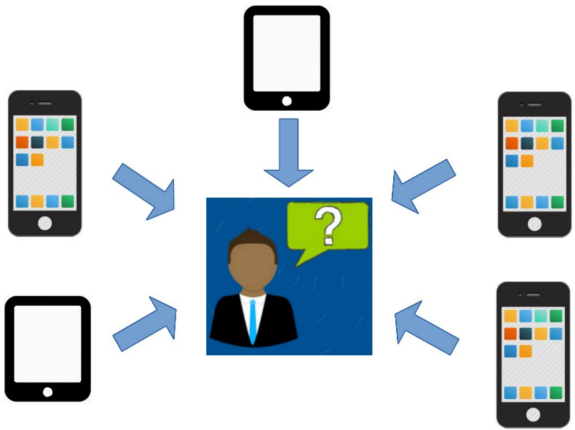
Modern machine learning systems are often trained on data collected from many different sources.



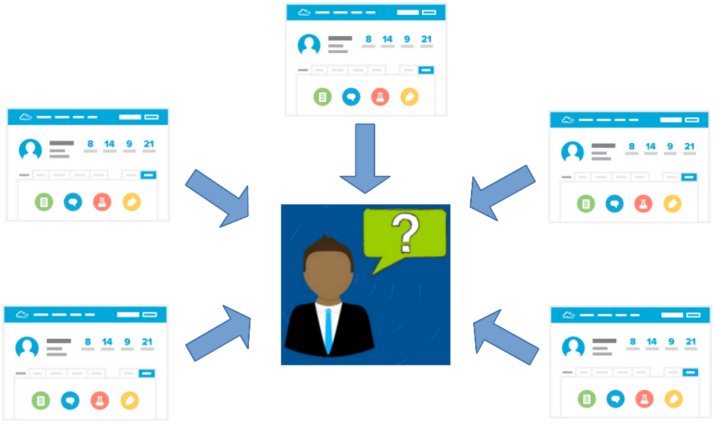
hundreds of different hospitals or medical labs

Learning from Multiple Sources

Modern machine learning systems are often trained on data collected from many different sources.

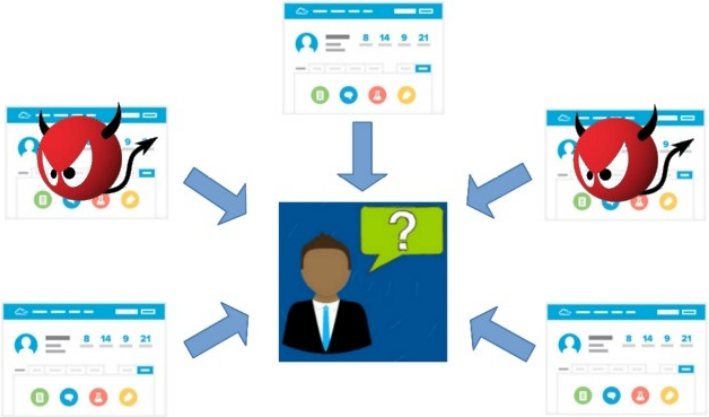


millions or billions of user devices

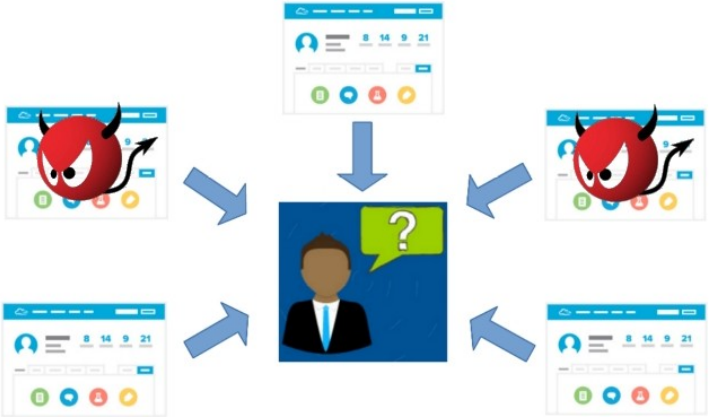


Ideally, all sources are i.i.d. samples from the correct data distribution

- best strategy: merge all datasets and train on resulting dataset



What, if some sources are not reliable?



What, if some sources are not reliable?

- a fraction of the data might be biased, noisy or manipulated
- classic result [Kearns and Li, 1993]: if we merge all data no algorithm can ensure optimal learning!

Is there a better way than merging all data?

Robust Learning from Unreliable or Manipulated Sources

Nikola Konstantinov

Elias Frantar

Dan Alistarh

[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 $S_i = \{(x_1^i, y_1^i), \ldots, (x_n^i, y_n^i)\} \overset{i.i.d.}{\sim} p$
- set of possible models $\mathcal{F}$
- multi-source learning algorithm $\mathcal{L} : (\mathcal{X} \times \mathcal{Y})^{N \times n} \to \mathcal{F}$
  - input: training sets, $S_1, S_2, \ldots, S_N$
  - output: one hypothesis $\mathcal{L}(S_1, \ldots, S_N) \in \mathcal{F}$ (= a trained model).

Learning from Multiple Unreliable/Manipulated Sources

- multiple training sets $S_1, S_2, \ldots, S_N$
  - each $S_i = \{(x_1^i, y_1^i), \ldots, (x_n^i, y_n^i)\} \overset{i.i.d.}{\sim} p$
- set of possible models $\mathcal{F}$
- multi-source learning algorithm $\mathcal{L} : (\mathcal{X} \times \mathcal{Y})^{N \times n} \to \mathcal{F}$
  - input: training sets, $S'_1, S'_2, \ldots, S'_N = \mathfrak{A}(S_1, \ldots, S_N)$
  - output: one hypothesis $\mathcal{L}(S'_1, S'_2, \ldots, S'_N) \in \mathcal{F}$ (= a trained model).
- adversary $\mathfrak{A}$
  - input: data sets $S_1, \ldots, S_N$
  - output: data sets $S'_1, \ldots, S'_N$,
    of which $\lceil (1-\alpha)N \rceil$ are identical to before and $\lfloor \alpha N \rfloor$ are arbitrary
  - the adversary knows the training algorithm

Learning from Multiple Unreliable/Manipulated Sources

- multiple training sets $S_1, S_2, \ldots, S_N$
  - each $S_i = \{(x_1^i, y_1^i), \ldots, (x_n^i, y_n^i)\} \overset{i.i.d.}{\sim} p$
- set of possible models $\mathcal{F}$
- multi-source learning algorithm $\mathcal{L} : (\mathcal{X} \times \mathcal{Y})^{N \times n} \to \mathcal{F}$
  - input: training sets, $S'_1, S'_2, \ldots, S'_N = \mathfrak{A}(S_1, \ldots, S_N)$
  - output: one hypothesis $\mathcal{L}(S'_1, S'_2, \ldots, S'_N) \in \mathcal{F}$ (= a trained model).
- adversary $\mathfrak{A}$
  - input: data sets $S_1, \ldots, S_N$
  - output: data sets $S'_1, \ldots, S'_N$,
    of which $\lceil (1-\alpha)N \rceil$ are identical to before and $\lfloor \alpha N \rfloor$ are arbitrary
  - the adversary knows the training algorithm

Is there a universal learning algorithm that learns an optimal model in the limit $n \to \infty$?

Answer: yes!

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

There exists a learning algorithm, $\mathcal{L}$, such that

$$\mathrm{er}(\,\mathcal{L}(S'_1,\ldots,S'_N)\,) \leq \min_{f\in\mathcal{F}} \mathrm{er}(f) + \underbrace{\widetilde{\mathcal{O}}\Big(\frac{1}{\sqrt{(1-\alpha)Nn}} + \alpha\frac{1}{\sqrt{n}}\Big)}_{\to 0 \text{ for } n=|S|\to\infty},$$

with $S'_1,\ldots,S'_N = \mathfrak{A}(S_1,\ldots,S_N)$ for any adversary $\mathfrak{A}$ 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 has to adhere to the source structure

Algorithm idea: exploit law of large numbers

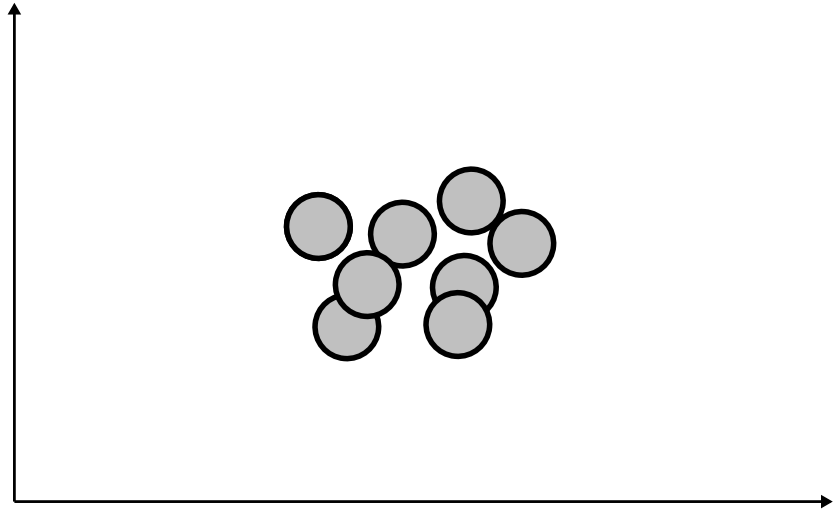
- majority of datasets are unperturbed
- for $n \to \infty$ these start to look more and more similar
- we can identify (at least) the unperturbed datasets

Robust multi-source learning algorithm:

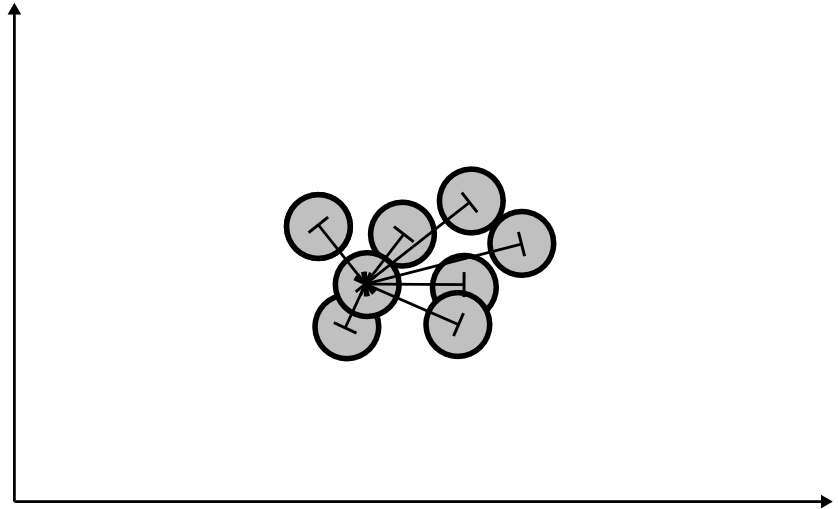
- Step 1) identify which sources to trust
  - compute all pairwise distance d_{ij} between datasets $S'_1, \ldots, S'_N$ (with a suitable distance measure d)
  - for any i: if $d_{ij} < \theta$ for at least $\lfloor \frac{N}{2} \rfloor$ values of $j \neq i$, then $T \leftarrow T \cup \{i\}$ (with a suitable 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:

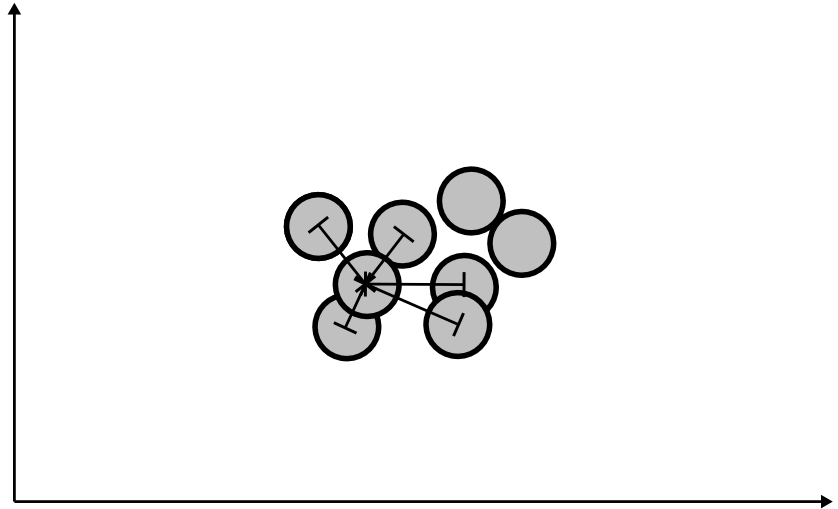
- distance measure d (discussed later)
- threshold θ (not discussed, see paper)



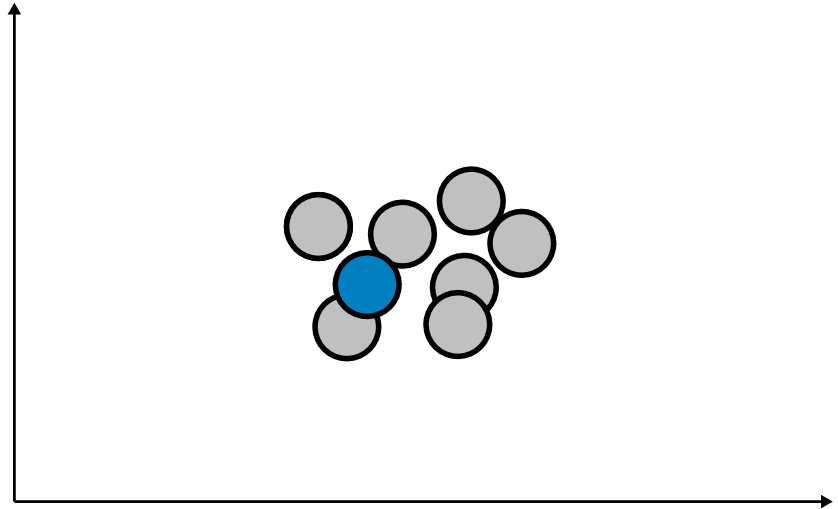
Example: All datasets clean



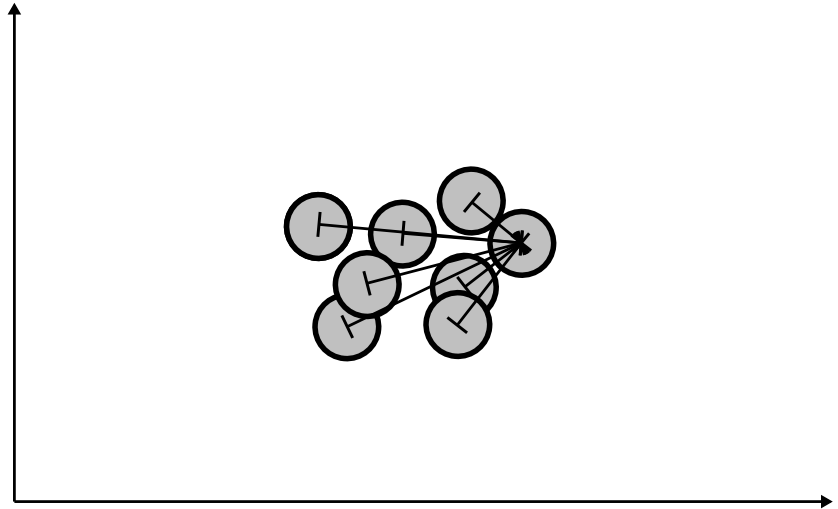
Example: All datasets clean



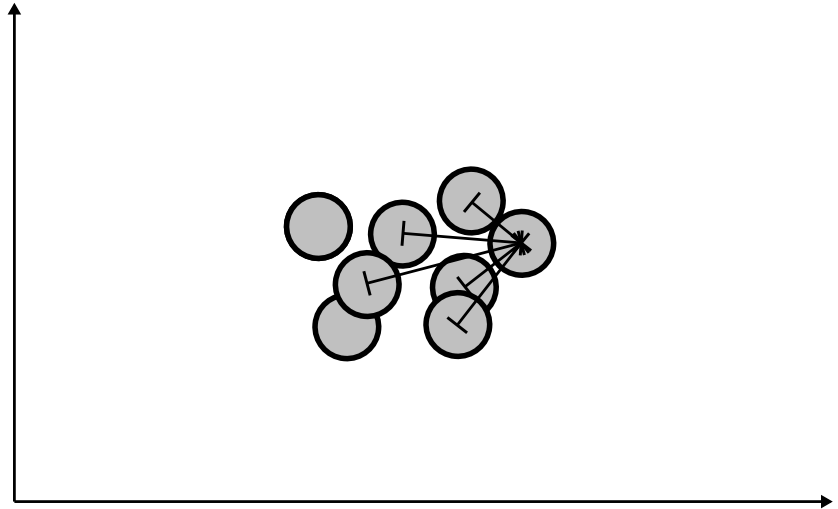
Example: All datasets clean



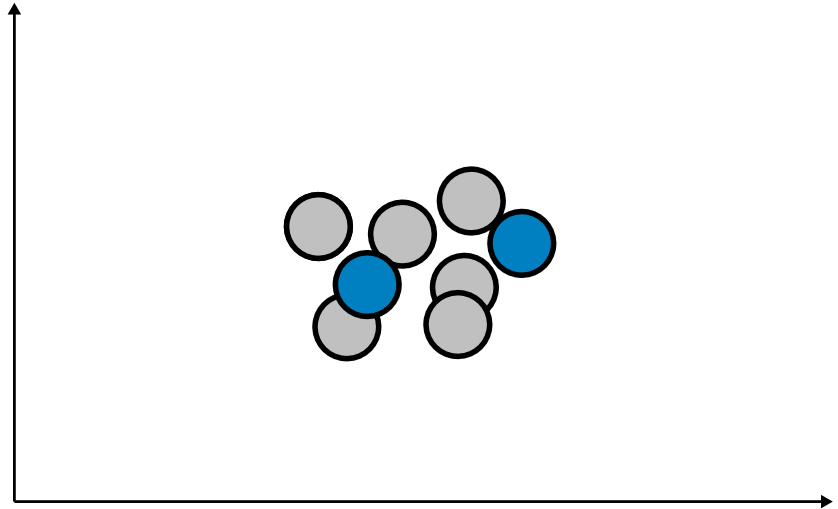
Example: All datasets clean



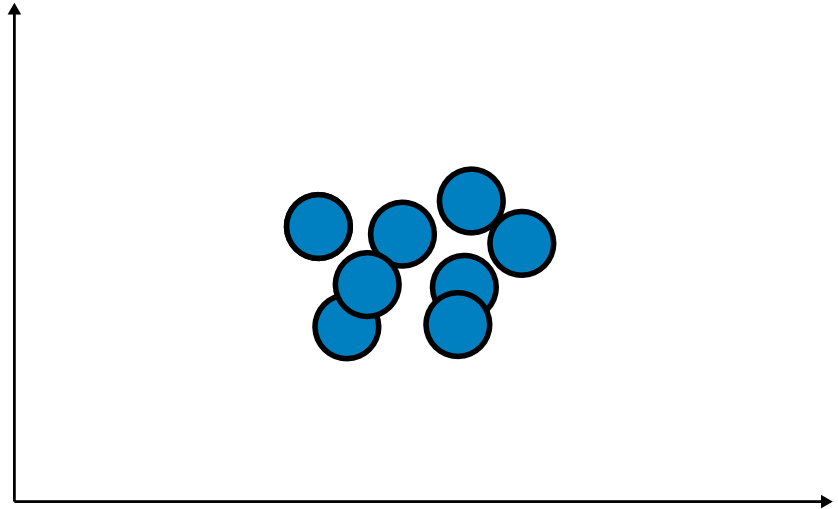
Example: All datasets clean



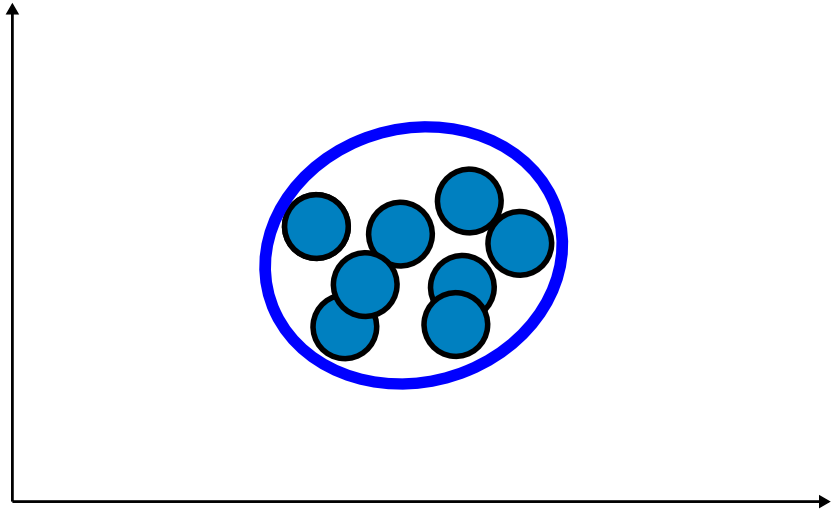
Example: All datasets clean



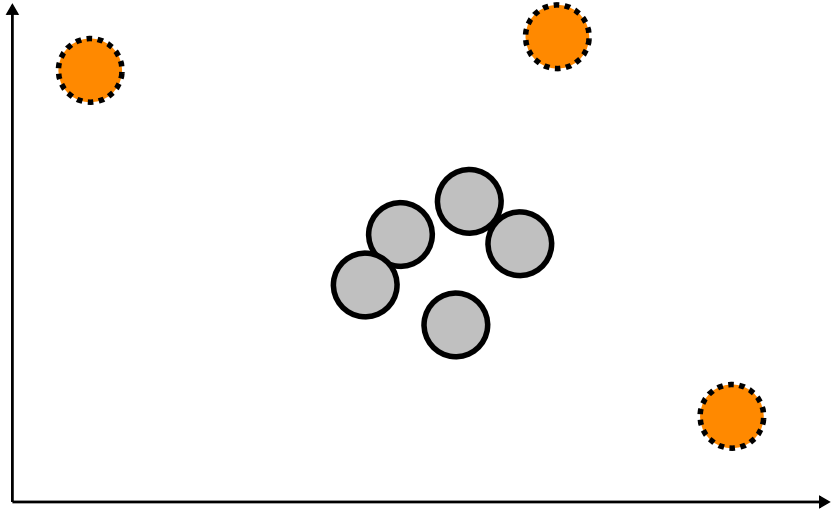
Example: All datasets clean



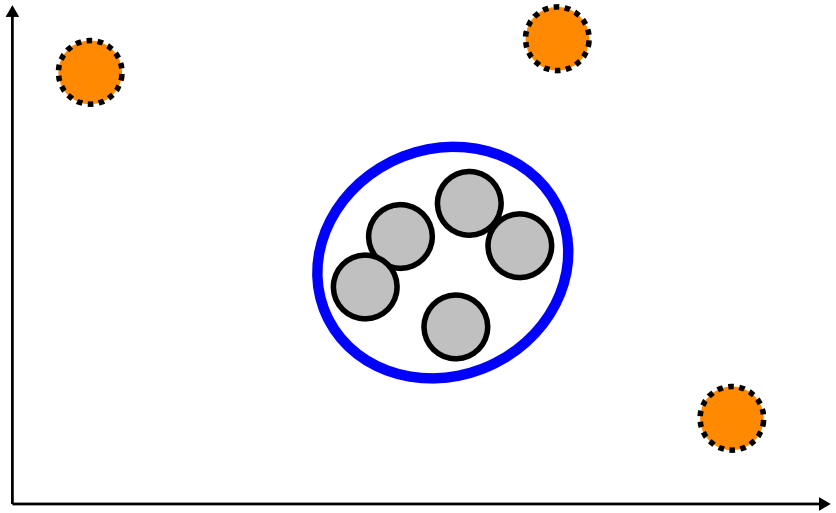
Example: All datasets clean



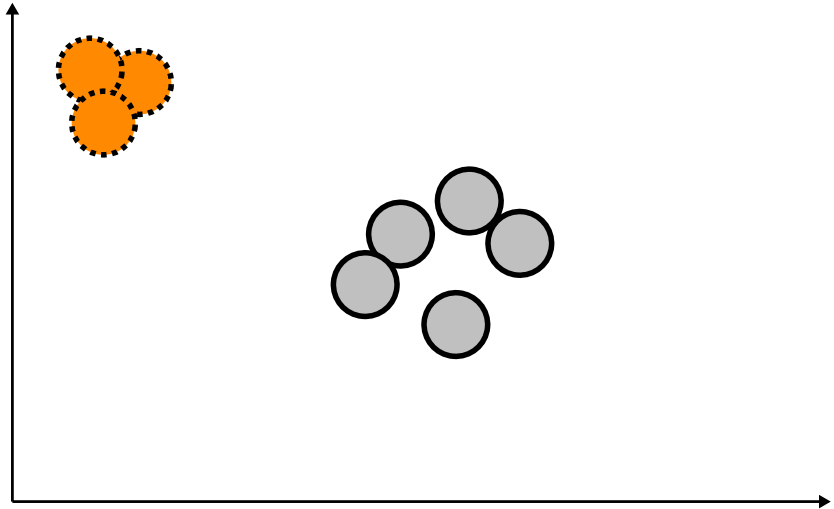
Example: All datasets clean → all datasets included → same as (optimal) naive algorithm



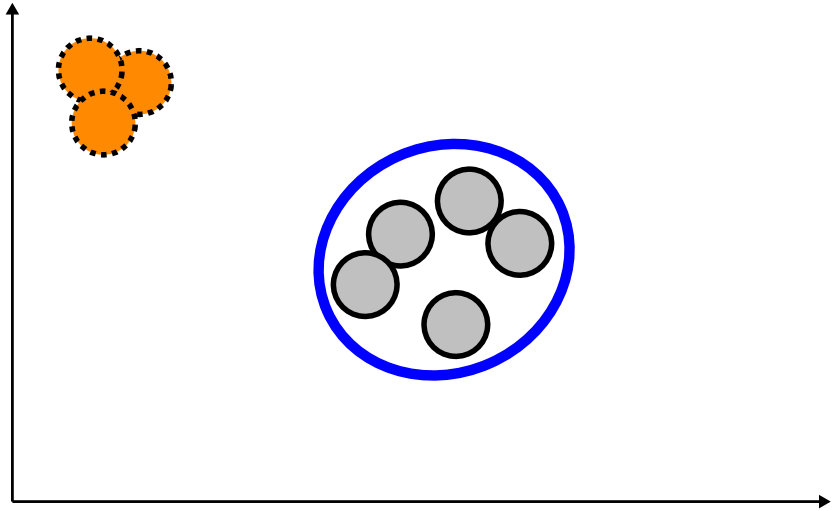
Example: Some datasets manipulated



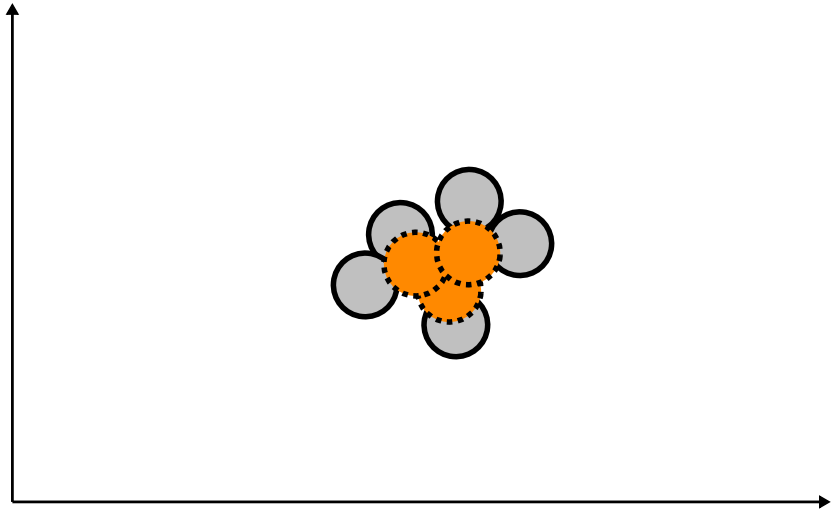
Example: Some datasets manipulated $\rightarrow$ manipulated datasets excluded.



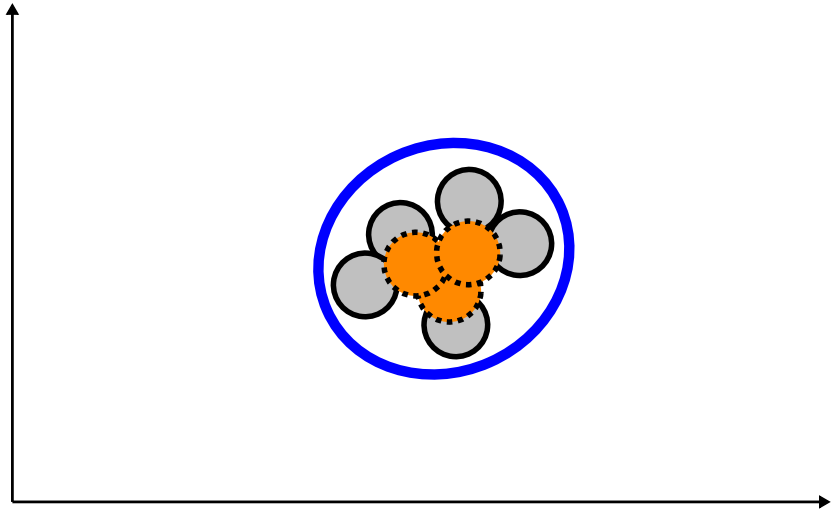
Example: Some datasets manipulated in a consistent way



Example: Some datasets manipulated in a consistent way $\rightarrow$ manipulated datasets excluded.



Example: Some datasets manipulated to look like originals



Example: Some datasets manipulated to look like originals $\rightarrow$ all 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 $\Rightarrow$ $d(S, S')$ 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.

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:

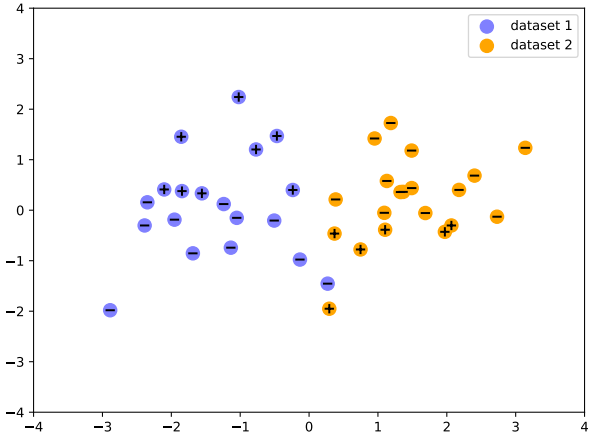
- many candidate distances do not fulfill both conditions simultaneously:
  - geometric: average Euclidean distance, Chamfer distance, Haussdorf distance, ...
  - probabilistic: Wasserstein distance, total variation, Kullback-Leibler divergence, ...
- discrepancy distance does fulfill the conditions!

Discrepancy Distance [Mansour *et al.* 2009]

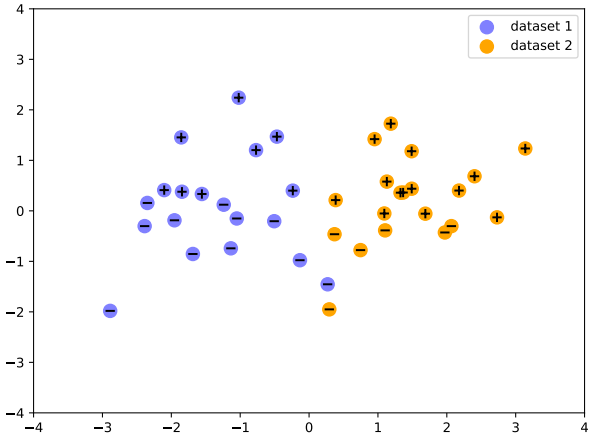
For a set of classifiers $\mathcal{H}$ and datasets S_i, S_j, define

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

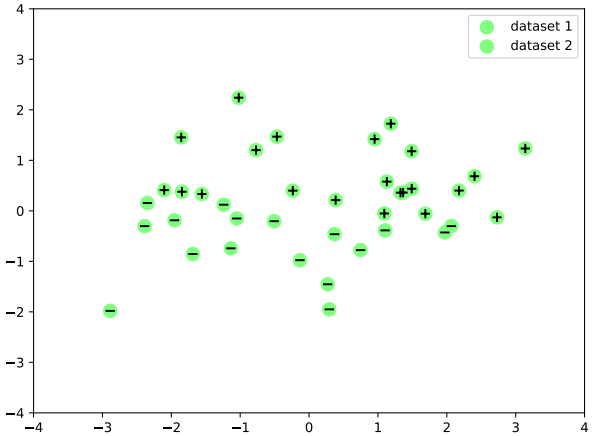
- maximal amount any classifier, $f \in \mathcal{F}$, can disagree between S_i, S_j
- for binary classification, discrepancy can be computed by training a classifier itself:
  - $S_j^{\pm} \leftarrow S_j$ with all ± 1 labels flipped to their opposites
  - $\tilde{S} \leftarrow S_i \cup S_j^{\pm}$
  -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



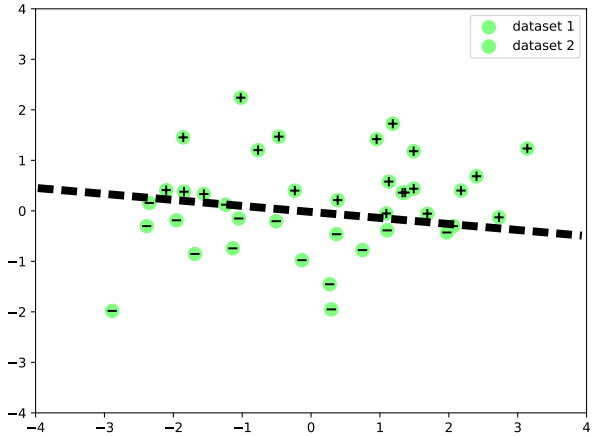
Two (dissimilar) datasets, S_1, S_2



Flip signs of S_2

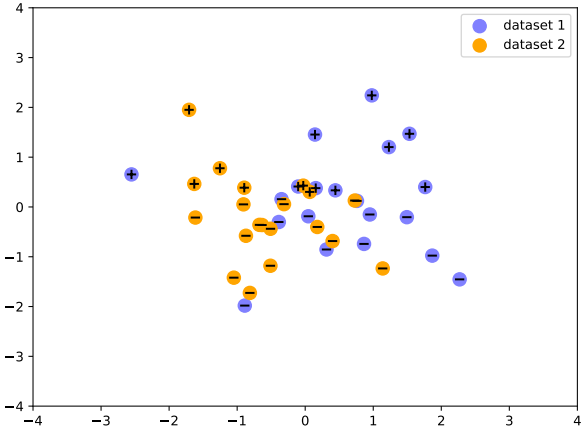


Merge both datasets

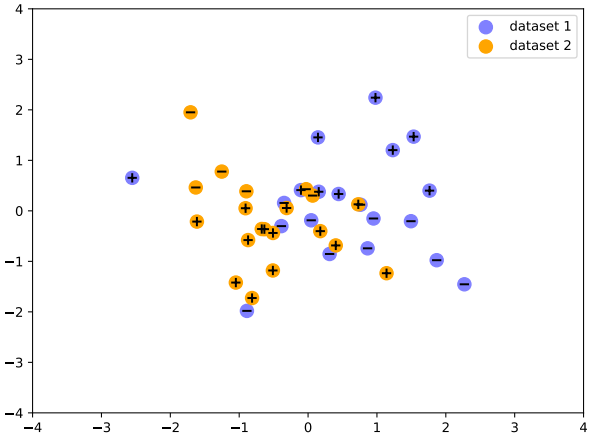


Classifier with small training error $\rightarrow$ large discrepancy

Discrepancy illustration

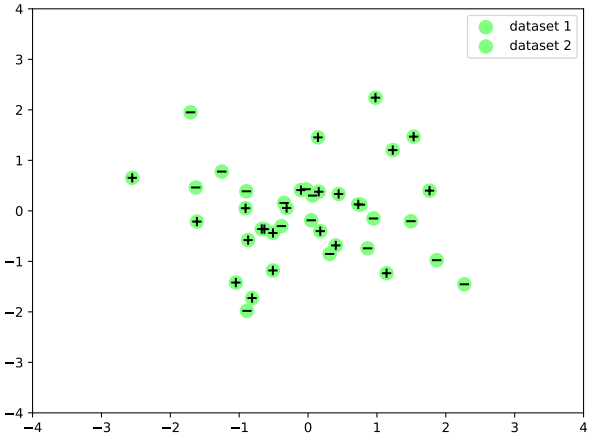


Two (similar) datasets, S_1, S_2



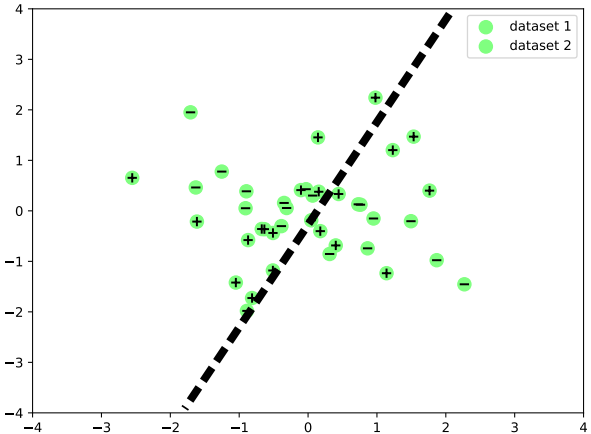
Flip signs of S_2

Discrepancy illustration



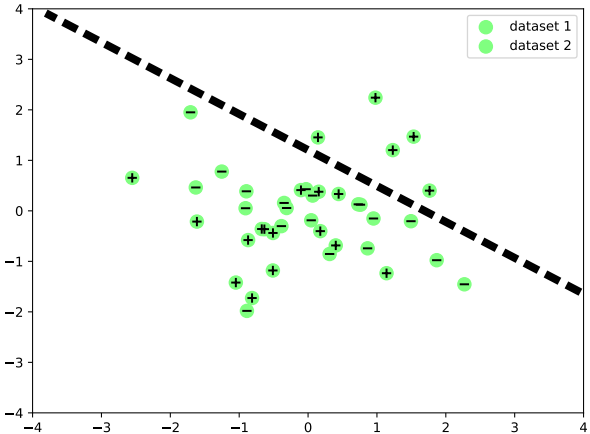
Merge both datasets

Discrepancy illustration



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

Discrepancy illustration



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

Then, for every $f \in \mathcal{F}$: $\quad \widehat{\mathsf{er}}_{S_{\mathsf{tst}}}(f) \leq \widehat{\mathsf{er}}_{S'}(f) + \underbrace{\mathrm{disc}(S_{\mathsf{trn}}, S')}_{\leq \theta} + \underbrace{\mathrm{disc}(S_{\mathsf{trn}}, S_{\mathsf{tst}})}_{\text{small by prop. 1)}}$

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 f^* be the result of the robust multi-source learning algorithm. Then

$$\mathsf{er}(f^*) \le \min_{f \in \mathcal{F}} \mathsf{er}(f) + \underbrace{\widetilde{\mathcal{O}}\Big(\frac{1}{\sqrt{km}} + \alpha\frac{1}{\sqrt{m}}\Big)}_{\to 0 \text{ for } m \to \infty},$$

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

Discussion:

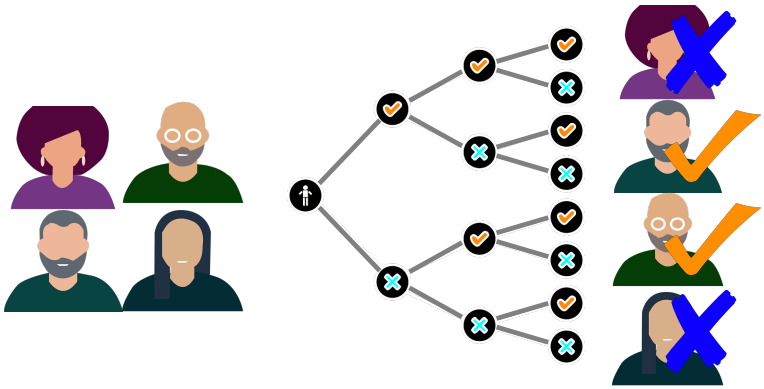
- km is the number of "clean" samples $\to$ $\frac{1}{\sqrt{km}}$ is the "normal" speed of learning
- $\alpha\frac{1}{\sqrt{m}}$ is a slow-down due to α-manipulation
- lower bounds exists that show that $O(\alpha\frac{1}{\sqrt{m}})$ slowdown is unavoidable

Fairness-Aware Learning from Unreliable or Manipulated Data

Jen Iofinova

Nikola Konstantinov

[N. Konstantinov, CHL. *"Fairness-Aware PAC Learning from Corrupted Data"*, JMLR 2022; https://arxiv.org/abs/2102.06004]
[E. Iofinova*, N. Konstantinov*, CHL. *"Robust Learning from Untrusted Sources"*, TMLR 2022; https://arxiv.org/abs/2106.11732]



How to ensure that a classifier does not discriminate against certain groups?

Reminder:

- Inputs: $x \in \mathcal{X}$, e.g. strings, images, vectors, ...
- Protected attribute: $a \in \mathcal{A}$, e.g. gender, age, race, ...
- Outputs: $y \in \mathcal{Y}$ (for simplicity: $\mathcal{Y} = \{0, 1\}$)
- Probability distribution: $p(x, a, y)$ over $\mathcal{X} \times \mathcal{A} \times \mathcal{Y}$
- Loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ (for simplicty: 0/1-loss)

Abstract Goal:

- find a prediction function, $f : \mathcal{X} \to \mathcal{Y}$ with low expected loss

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

that in addition fulfills some condition of (group) fairness.

Group Fairness:

- demographic parity (independence): "all groups have the same success rate"

$$\forall a, b \in \mathcal{A} \quad p(f(X) = 1 | A = a) = p(f(X) = 1 | A = b)$$

- equality of opportunity: "all groups have the true positive rate"

$$\forall a, b \in \mathcal{A} \quad p(f(X) = 1 | A = a, Y = 1) = p(f(X) = 1 | A = b, Y = 1)$$

and many others.

Several fairness-aware learning methods exist to enforce these criteria.

[S. Barocas, M. Hardt, A. Narayanan. "Fairness and Machine Learning. Limitations and Opportunities", fairmlbook.org, 2019]

Fair Learning from one unreliable/manipulated dataset:

- original training set: $S = \{(x_1, a_1, y_1), \ldots, (x_m, a_m, y_m)\}$
- adversary $\mathfrak{A}$ can manipulate a fraction α of the dataset
- actual training set: $S' = \mathfrak{A}(S)$

Question: Can a fairness-aware learner overcome the manipulation?

Fair Learning from one unreliable/manipulated dataset:

- original training set: $S = \{(x_1, a_1, y_1), \ldots, (x_m, a_m, y_m)\}$
- adversary $\mathfrak{A}$ can manipulate a fraction α of the dataset
- actual training set: $S' = \mathfrak{A}(S)$

Question: Can a fairness-aware learner overcome the manipulation?

Answer: No!

Theorem [Konstantinov and Lampert, 2022]

There exists a learning situation and (even finite) hypothesis space for which

- No learning algorithm can guarantee optimal fairness.
- This effect is independent of whether accuracy is also affected or not.
- The smaller the minority group, the stronger the bias.

Fair Learning from multiple sources:

- multiple training sets: $S_1, S_2, \ldots, S_N \subset \mathcal{X} \times \mathcal{A} \times \mathcal{Y}$
- adversary $\mathfrak{A}$ can manipulate $K = \lfloor \alpha N \rfloor$ of the datasets for $\alpha < \frac{1}{2}$
- actual training sets: $S'_1, \ldots, S'_N = \mathfrak{A}(S_1, \ldots, S_N)$

Is there a fairness-aware learning algorithm that overcomes such manipulations?

Fair Learning from multiple sources:

- multiple training sets: $S_1, S_2, \ldots, S_N \subset \mathcal{X} \times \mathcal{A} \times \mathcal{Y}$
- adversary $\mathfrak{A}$ can manipulate $K = \lfloor \alpha N \rfloor$ of the datasets for $\alpha < \frac{1}{2}$
- actual training sets: $S'_1, \ldots, S'_N = \mathfrak{A}(S_1, \ldots, S_N)$

Is there a fairness-aware learning algorithm that overcomes such manipulations?

Answer: Yes!

Theorem [Iofinova et al., 2022]

There exists a learning algorithm, $\mathcal{L}$, such that for $f^* = \mathcal{L}(\mathfrak{A}(S_1, \ldots, S_N))$ with high probability

$$\mathsf{er}(f^*) \leq \min_{f \in \mathcal{F}} \mathsf{er}(f) + \widetilde{\mathcal{O}}(\frac{1}{\sqrt{m}}), \qquad \Gamma(f^*) \leq \min_{f \in \mathcal{F}} \Gamma(f) + \widetilde{\mathcal{O}}(\frac{1}{\sqrt{m}})$$

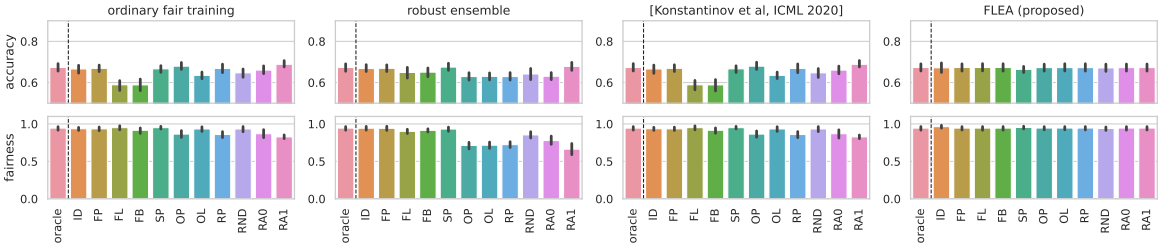
where Γ is a quantitative measure of *demographic parity* fairness.

[E. Iofinova, N. Konstantinov, CHL, "FLEA: Provably Robust Fair Multisource Learning", TMLR 2022; https://arxiv.org/abs/2106.11732]

FLEA (Fair LEarning against Adversaries):

- **Input:** datasets $S'_1, \ldots, S'_N$
- **Input:** $\beta \leq \frac{1}{2}$ upper bound on fraction of malignant sources
- **Define:** distance measure $d(S, \hat{S}) = \text{disc}(S, \hat{S}) + \text{disp}(S, \hat{S}) + \text{disb}(S, \hat{S})$
  - $\text{disc}(S, \hat{S})$: discrepancy as before
  - $\text{disp}(S, \hat{S})$: maximal fairness difference of any classifier between S and $\hat{S}$
  - $\text{disb}(S, \hat{S})$: difference in protected group proportions
- Step 1) identify which sources to trust
  - compute all pairwise distance d_{ij} between datasets $S'_1, \ldots, S'_N$
  - for any $i = 1, \ldots, N$: $\quad q_i \leftarrow \beta\text{-quantile}(d_{i1}, \ldots, d_{iN})$
  - $T \leftarrow \{i : q_i \leq \beta\text{-quantile}(q_1, \ldots, q_N)\}$
- Step 2) merge data from all sources S'_i with $i \in T$ into a new dataset $\tilde{S}$
- Step 3) train fairness-aware learning algorithm on $\tilde{S}$

Experimental Results

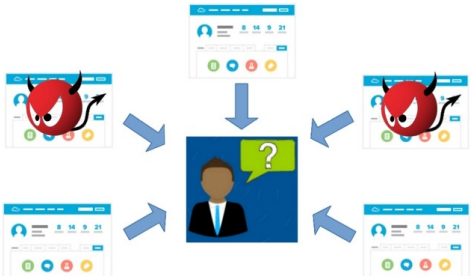


- bars are different data manipulations, designed to hurt accuracy or fairness
- simply training on all data often suboptimal
- other baselines often fail to overcome problems
- FLEA reliably recovers fairness and accuracy

	COMPAS	
method	accuracy	fairness
naive	$63.5_{\pm 2.1}$	$78.9_{\pm 2.3}$
robust ensemble	$65.0_{\pm 1.1}$	$88.4_{\pm 2.9}$
DRO (Wang et al., 2020)	$54.5_{\pm 1.2}$	$70.9_{\pm 5.7}$
(Konstantinov et al., 2020)	$63.5_{\pm 2.1}$	$78.9_{\pm 2.3}$
FLEA (proposed)	$65.9_{\pm 1.1}$	$95.3_{\pm 2.3}$
oracle	$66.2_{\pm 1.1}$	$96.2_{\pm 1.3}$

[E. Iofinova, N. Konstantinov, CHL, "FLEA: Provably Robust Fair Multisource Learning", TMLR 2022; https://arxiv.org/abs/2106.11732]

Summary



- Learning from multiple unreliable sources now commonplace
- Can be studied formally: learning with an adversary of a certain power
- Group structure allow robust and fair learning, even against a strong adversary

References I

E. Iofinova, N. Konstantinov, and C. H. Lampert. FLEA: Provably robust fair multisource learning. *Transactions of Machine Learning Research (TMLR)*, 2022.

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

N. Konstantinov and C. H. Lampert. Fairness-aware PAC learning from corrupted data. *Journal of Machine Learning Research (JMLR)*, 2022.

K. Leino, Z. Wang, and M. Fredrikson. Globally-robust neural networks. In *International Conference on Machine Learing (ICML)*, 2021.

B. Li, C. Chen, W. Wang, and L. Carin. Certified adversarial robustness with additive noise. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

B. Prach and C. H. Lampert. Almost-orthogonal layers for efficient general-purpose lipschitz networks. In *European Conference on Computer Vision (ECCV)*, 2022.

S. Singla and S. Feizi. Skew orthogonal convolutions. In *International Conference on Machine Learing (ICML)*, 2021.

S. Singla, S. Singla, and S. Feizi. Improved deterministic l_2 robustness on CIFAR-10 and CIFAR-100. In *International Conference on Learning Representations (ICLR)*, 2022.

A. Trockman and J. Z. Kolter. Orthogonalizing convolutional layers with the Cayley transform. In *International Conference on Learning Representations (ICLR)*, 2021.

T. Yu, J. Li, Y. CAI, and P. Li. Constructing orthogonal convolutions in an explicit manner. In *International Conference on Learning Representations (ICLR)*, 2022.