28 March 2023

CIS 5200: MACHINE LEARNING

SEMI-SUPERVISED & ACTIVE LEARNING

Surbhi Goel

Content here draws from material by Nina Balcan (CMU), Cynthia Rudin (Duke), and Kilian Weinberger (Cornell)

Spring 2023

OUTLINE - TODAY

- **Finish up Boosting**
- Semi-supervised Learning
- Active Learning
- Self-supervised Learning

GENERAL BOOSTING SCHEME

Weak learner $\mathscr{A}$ guarantees error $\leq 1/2 - \gamma$ for any distribution

Training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ where $y_i \in \{-1,1\}$

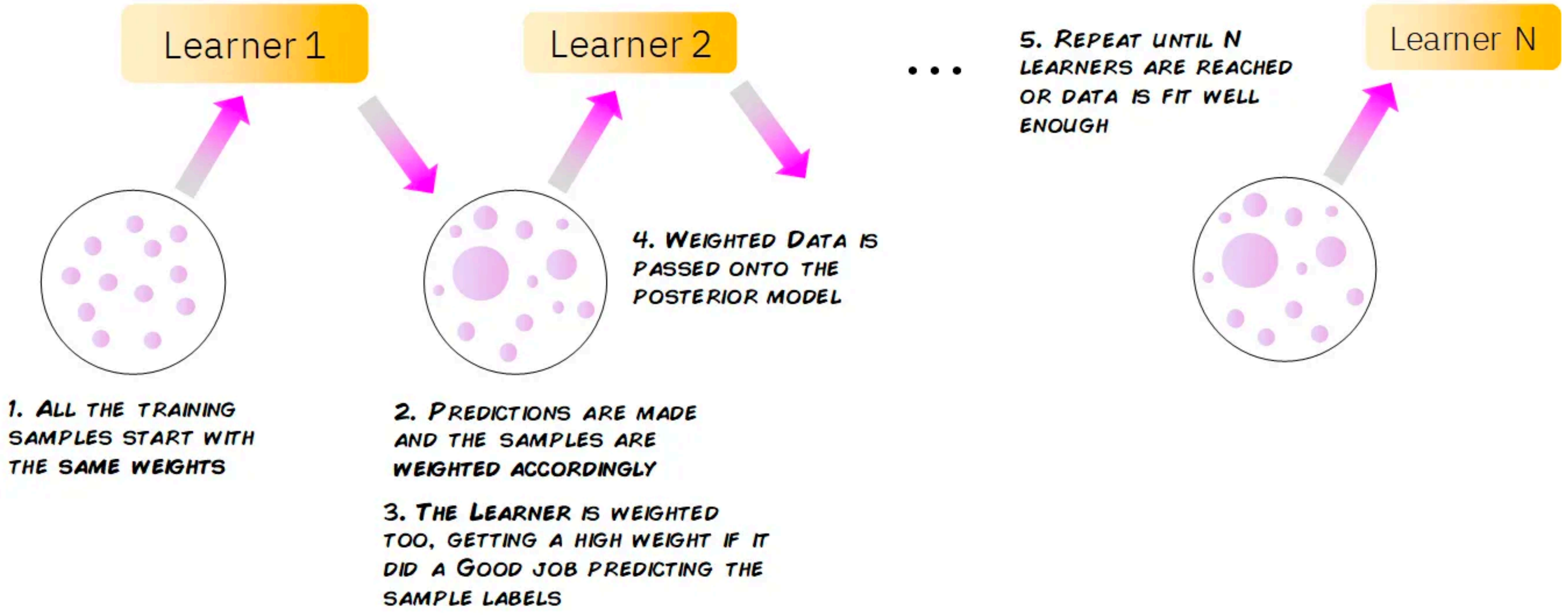


Image source: https://towardsdatascience.com/what-is-boosting-in-machine-learning-2244aa196682

ADABOOST - ADAPTIVE BOOSTING

Question 1: How do we choose μ_t?

For all $i \in [m]$,

$$\mu_{1,i} = \frac{1}{m}$$

Equal weight initially

$$\mu_{t+1,i} = \frac{\mu_{t,i}}{Z_t} \times \exp(-\alpha_t y_i f_t(x_i))$$

Normalizing factor

Weight increased if incorrect and decreased if correct

Optimal choice of shrinkage $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ where $\epsilon_t = \Pr_{i \sim \mu_t}[f_t(x_i) \neq y_i]$.

ADABOOST - ADAPTIVE BOOSTING

Question 2: How do we construct final classifier f using $f_1, \ldots, f_T$?

$$f(x) = \mathrm{sgn}\left(\sum_{t=1}^{T} \alpha_t f_t(x)\right)$$

Weighted combination of the weak learners

The weight is based on how good the weak learner is

TRAINING ERROR GUARANTEE

Weak learner $\mathscr{A}$ guarantees error $\leq 1/2 - \gamma$ for any distribution

Theorem:

Let f be the output of AdaBoost after T steps, then we have

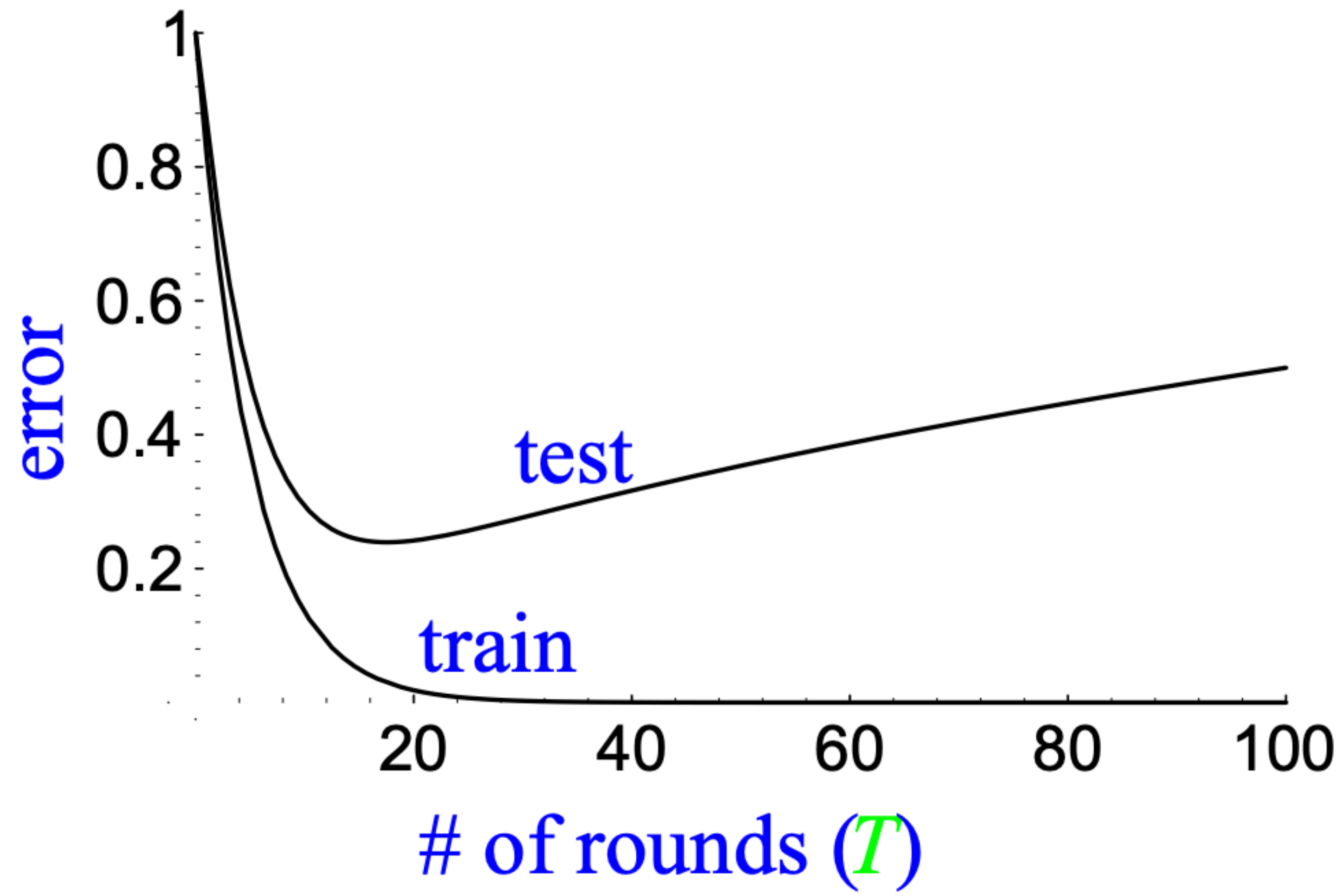
$$\hat{R}(f) = \frac{1}{m}\sum_{i=1}^{m} 1[f(x_i) \neq y_i] \leq \exp\left(-2\gamma^2 T\right).$$

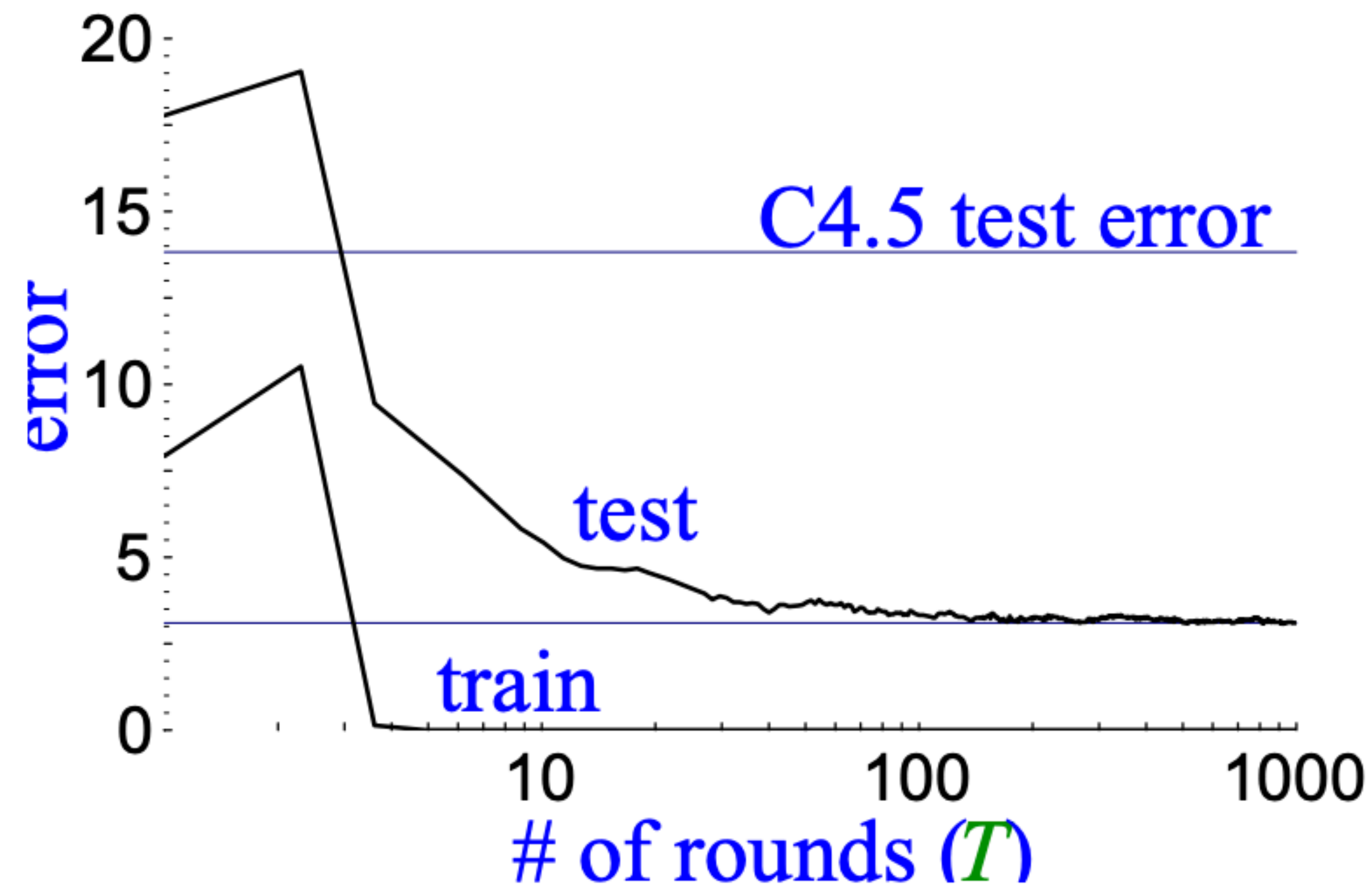
Training error goes down exponentially fast with the number of iterations

GENERALIZATION PERFORMANCE

We reduced bias by creating a more complex classifier

What about the variance of the final classifier for increasing T?





Test error improves even after training error is 0!

Image source: https://media.nips.cc/Conferences/2007/Tutorials/Slides/schapire-NIPS-07-tutorial.pdf

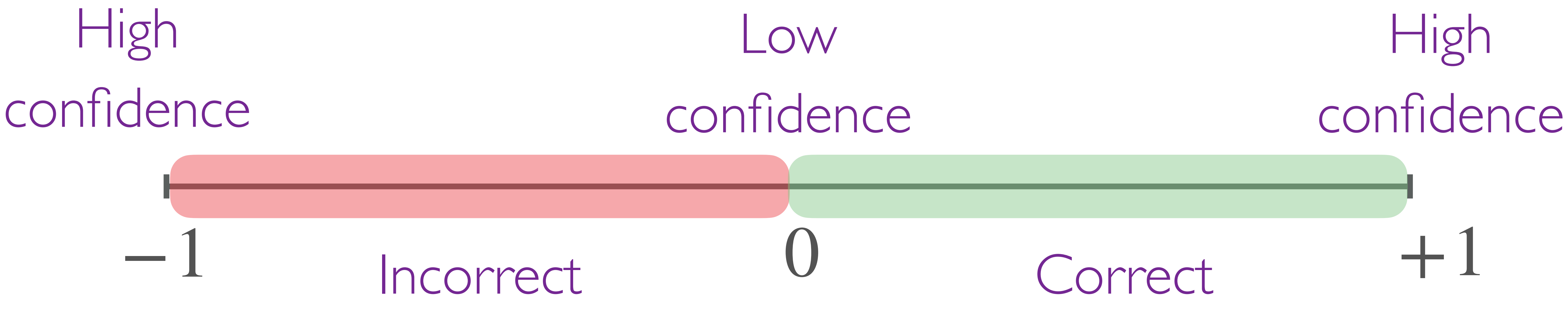
BIAS/VARIANCE - WHY NO TRADEOFF?

AdaBoost ensures large margin! (By Schapire, Freund, Bartlett & Lee)

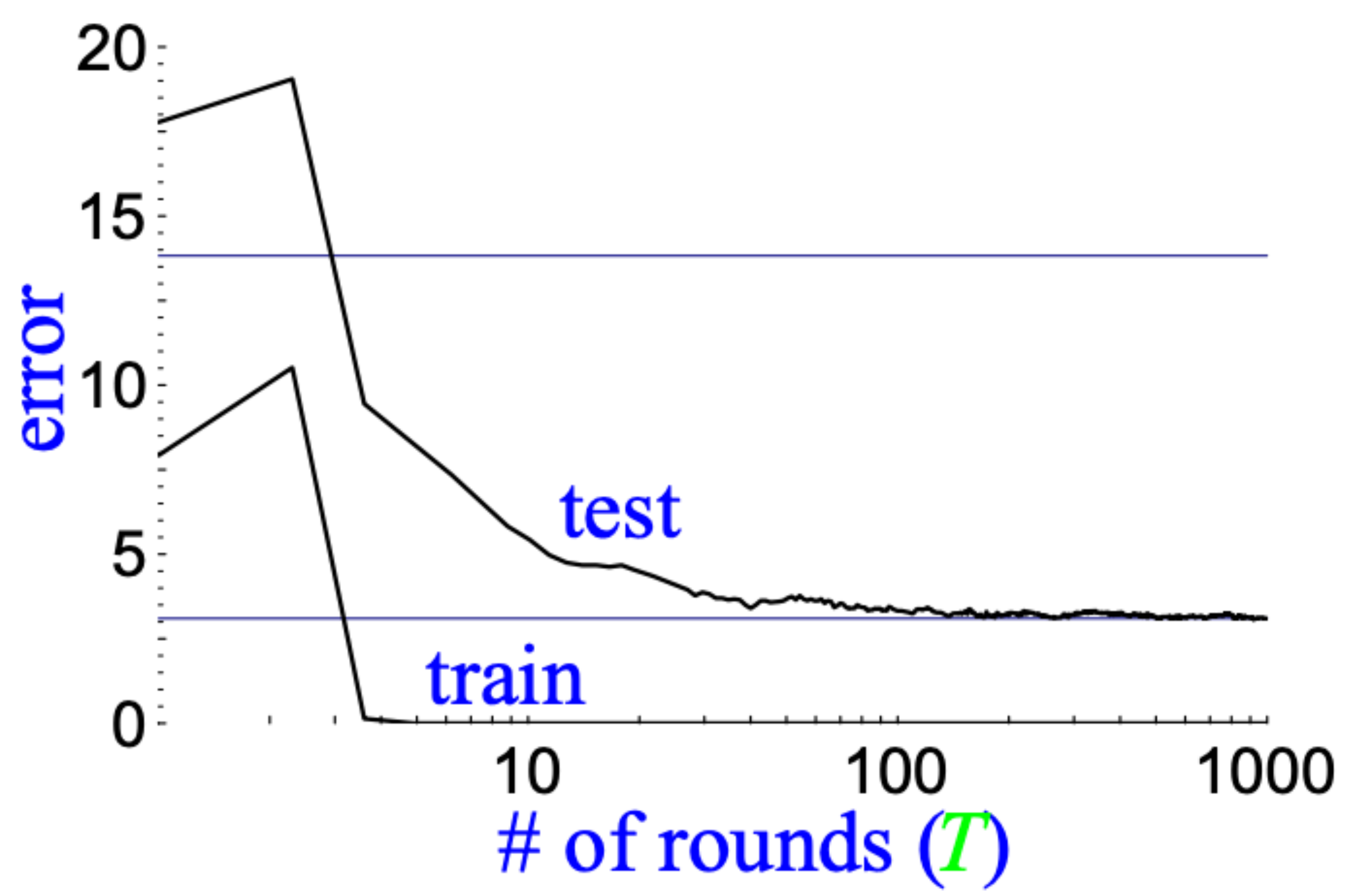
- Training error measures only correctness of prediction
- A better notion is confidence, how sure is the learner about the prediction
- In AdaBoost, the final classifier is a weighted vote of the weak learners

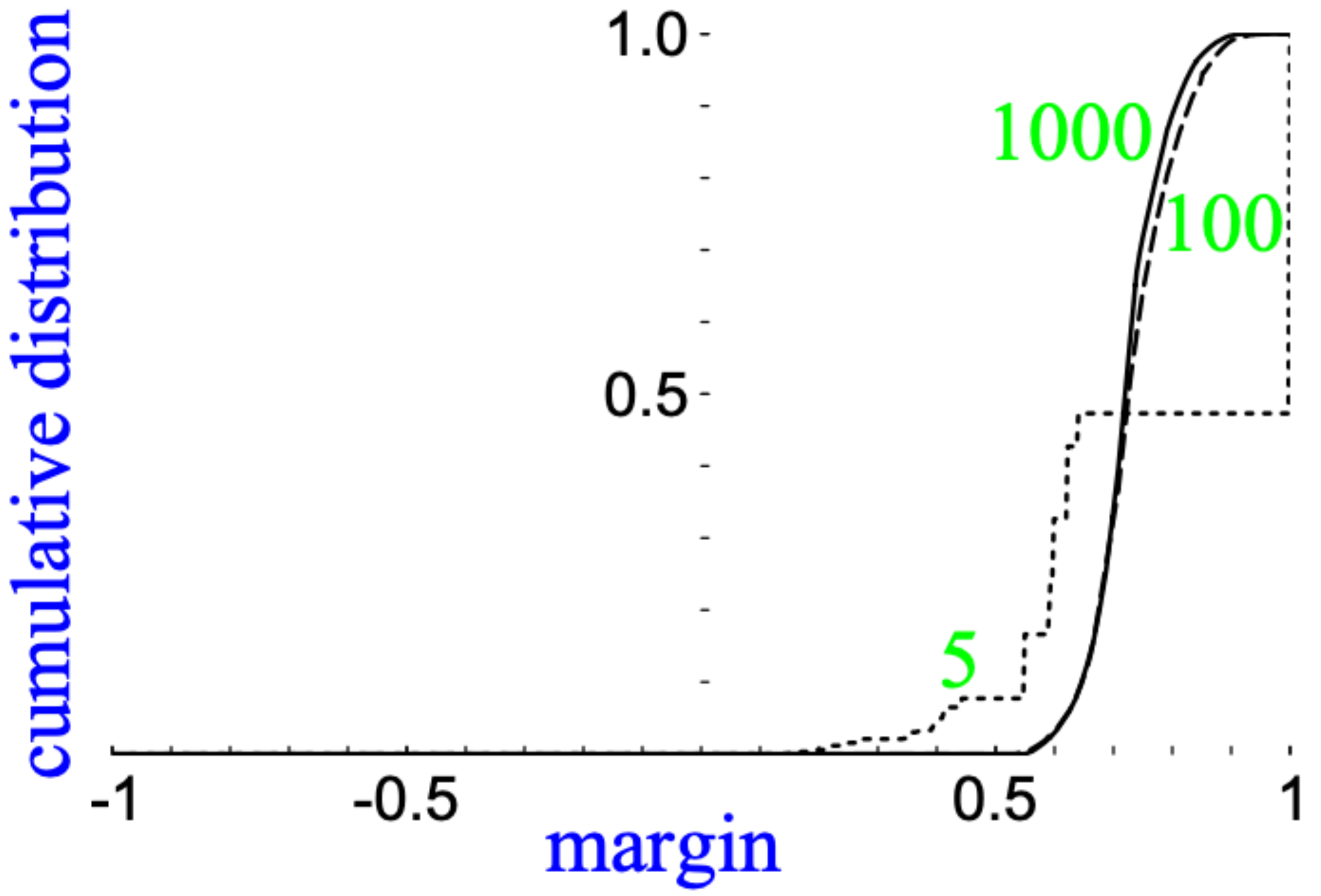
Margin - how strong is the vote?

= total weight of correct weak learners - total weight of incorrect weak learners



MARGIN





All points have margin at least 0.5

	# rounds		
	5	100	1000
train error	0.0	0.0	0.0
test error	8.4	3.3	3.1
% margins ≤ 0.5	7.7	0.0	0.0
minimum margin	0.14	0.52	0.55

Large margin $\Longrightarrow$ simpler classifier and better generalization

Image source: https://media.nips.cc/Conferences/2007/Tutorials/Slides/schapire-NIPS-07-tutorial.pdf

OPTIMIZATION VIEWPOINT OF BOOSTING

AdaBoost can be viewed as coordinate descent on a loss function over the space of linear combinations of weak classifiers

Recall that $\hat{R}(f) \leq \prod_{t=1}^{T} Z_t = \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i f(x_i))$ where $f(x) = \sum_t \alpha_t f_t(x)$

- Coordinate descent would choose a coordinate and find the corresponding α to maximally decrease the loss
- AdaBoost is essentially doing coordinate descent on this loss

PROS AND CONS

Benefits of AdaBoost

- Fast
- Simple
- Only hyper-parameter is T
- Flexible - can use any weak learning algorithm
- Do not need to know how good the weak learner is
- Powerful - only weak learners needed

Caveats of AdaBoost

- Performance dependent on data and weak learner
- Can overfit if weak learner is too complex
- Can also underfit if weak learner is not good
- Not robust to noise

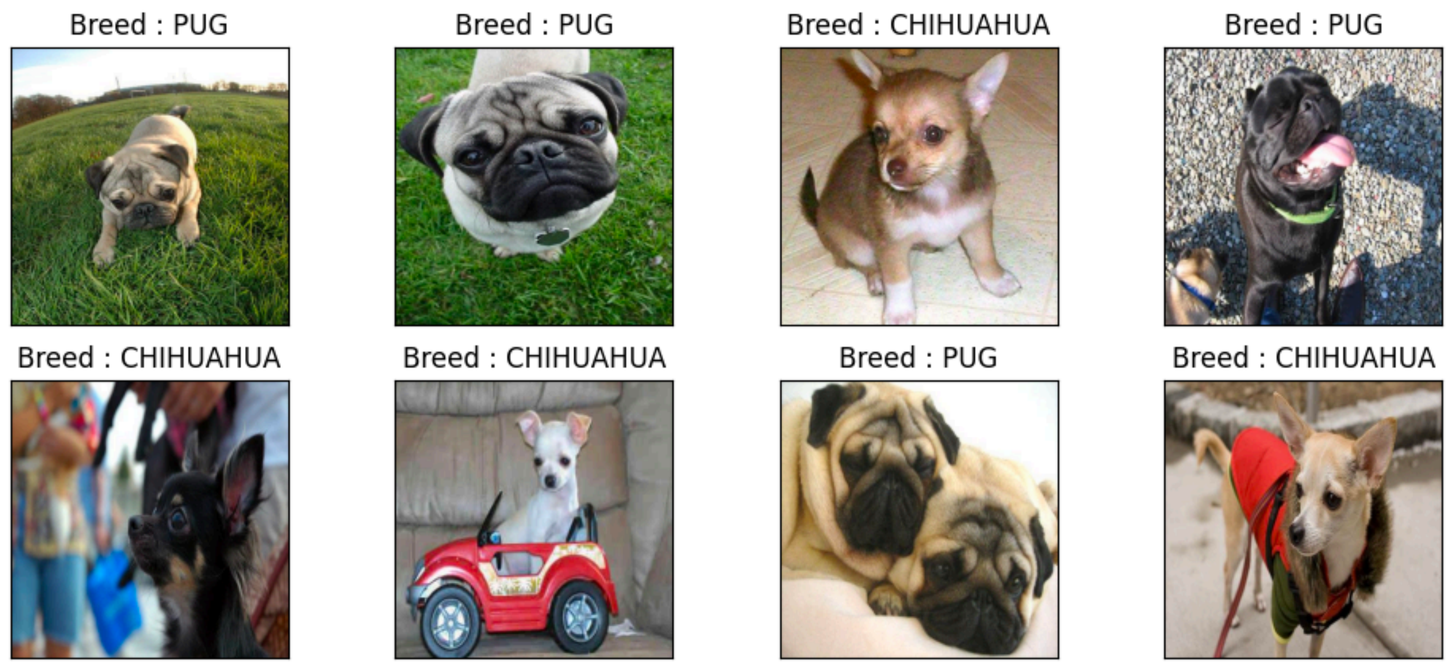
OUTLINE - TODAY

- Finish up Boosting
- Semi-supervised Learning
- Active Learning
- Self-supervised Learning

SUPERVISED LEARNING - DATA

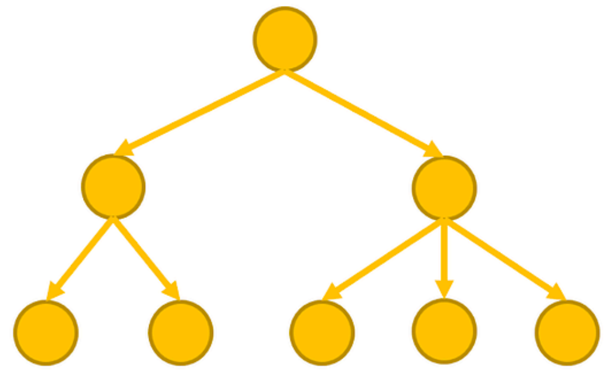
Training dataset

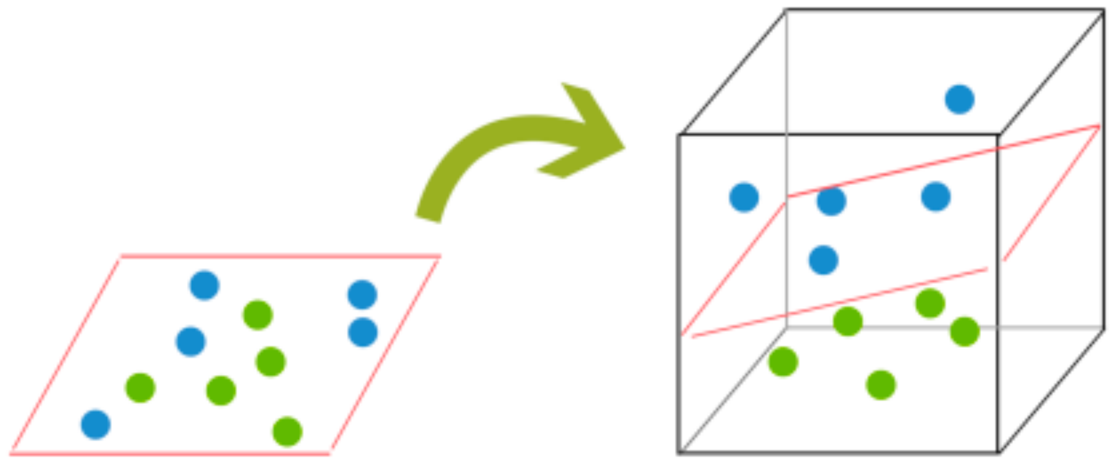
$\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$



Machine Learning Method

Prediction function $\hat{f}$





How do we actually get labels?

Domain experts

amazon mechanical turk

Crowdsource

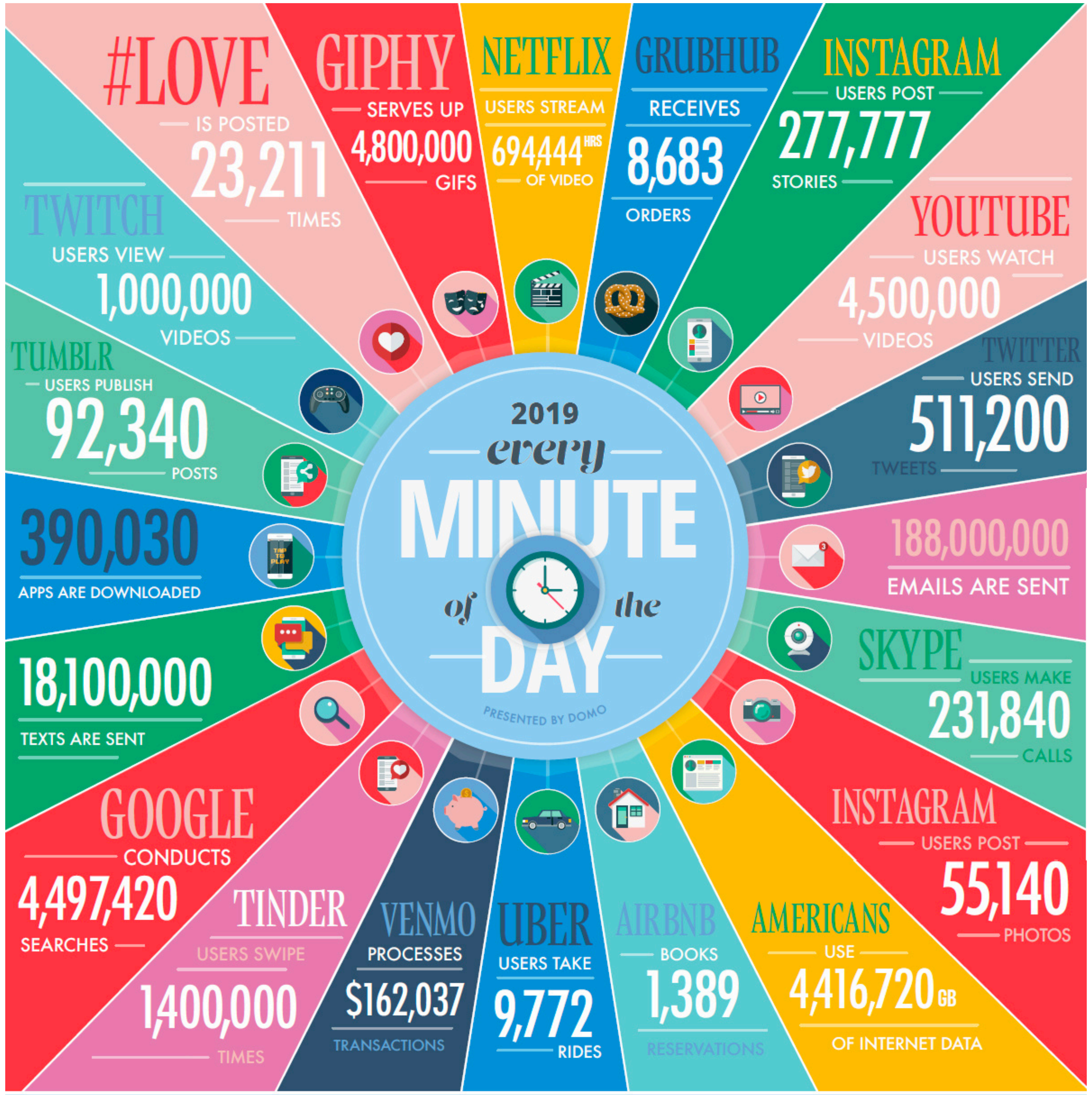
High cost in terms of effort/time

Need for domain experts

Inconsistencies/Errors

CHALLENGE - LARGE AMOUNTS OF DATA

Massive amount of data but limited supply of domain experts



- Images
- Text on websites
- Videos
- Protein sequences
- DNA
- ….

Image source: https://www.visualcapitalist.com/big-data-keeps-getting-bigger/

MODERN ML - REDUCE RELIANCE ON LABELS

Can we train machine learning models with less human supervision?

No labelled data → **Fully labelled data**

Unsupervised Learning

Clustering (K-means)
Density Estimation (GMM)
Dimensionality Reduction (PCA)

Can identify structures or patterns in data

Self-supervised Learning

Semi-supervised Learning

Active Learning

Supervised Learning

Regression (Linear Regression)
Classification (SVM, Perceptron, Logistics Regression)

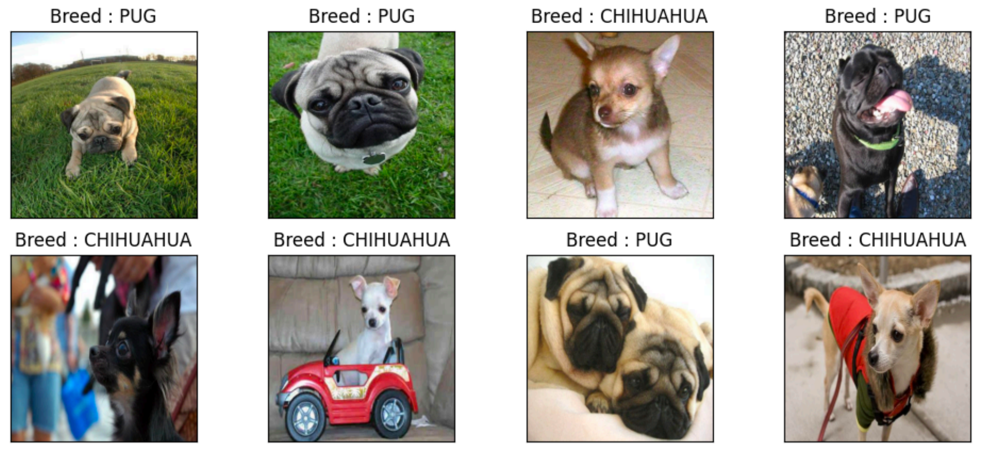
Can learn mapping from input to label

SEMI-SUPERVISED LEARNING

$m_l << m_u$

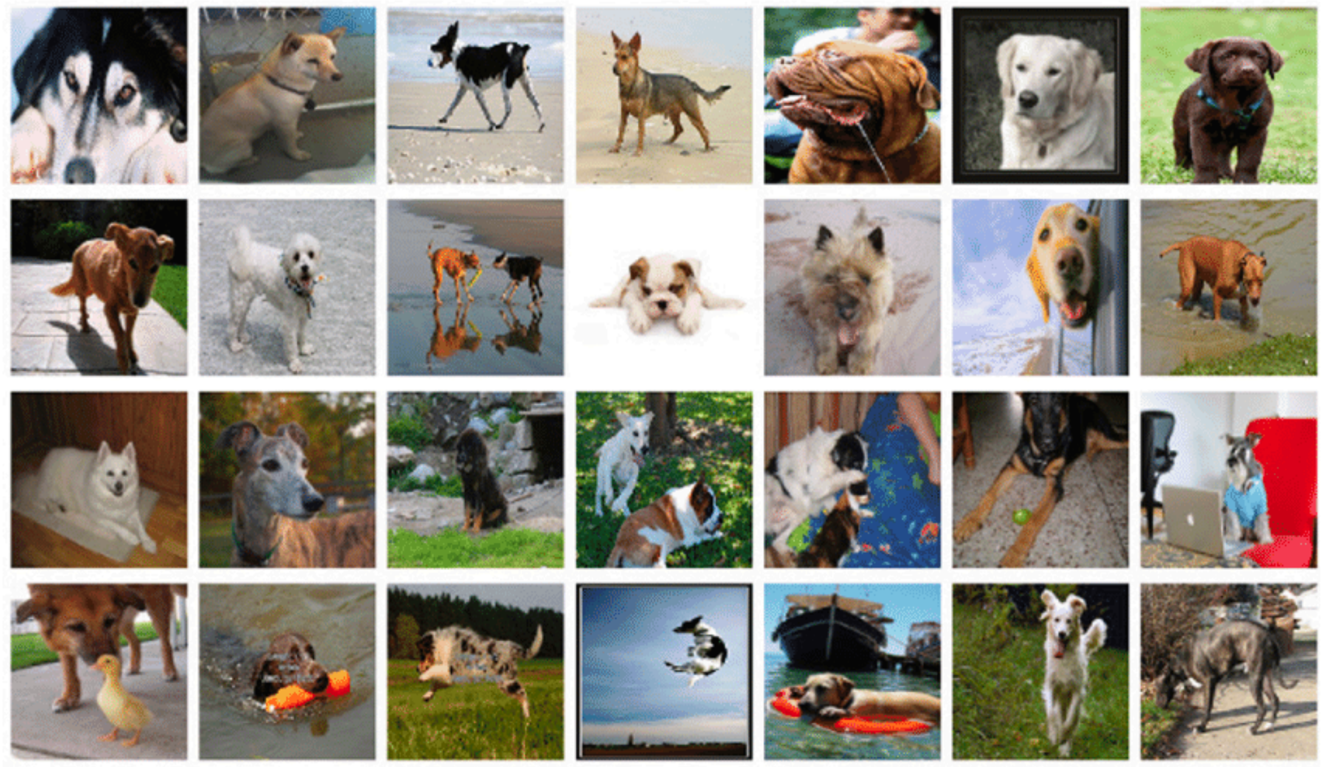
Labelled training dataset

$$\mathcal{S}_l = \left\{ (x_1, y_1), (x_2, y_2), \ldots, (x_{m_l}, y_{m_l}) \right\}$$



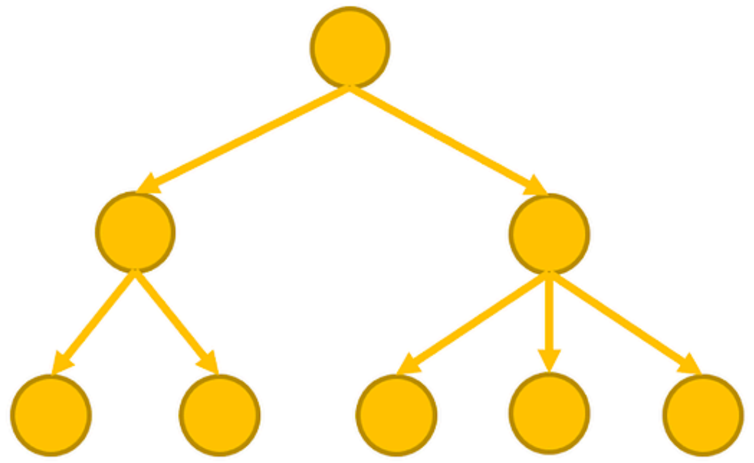
Unlabelled training dataset

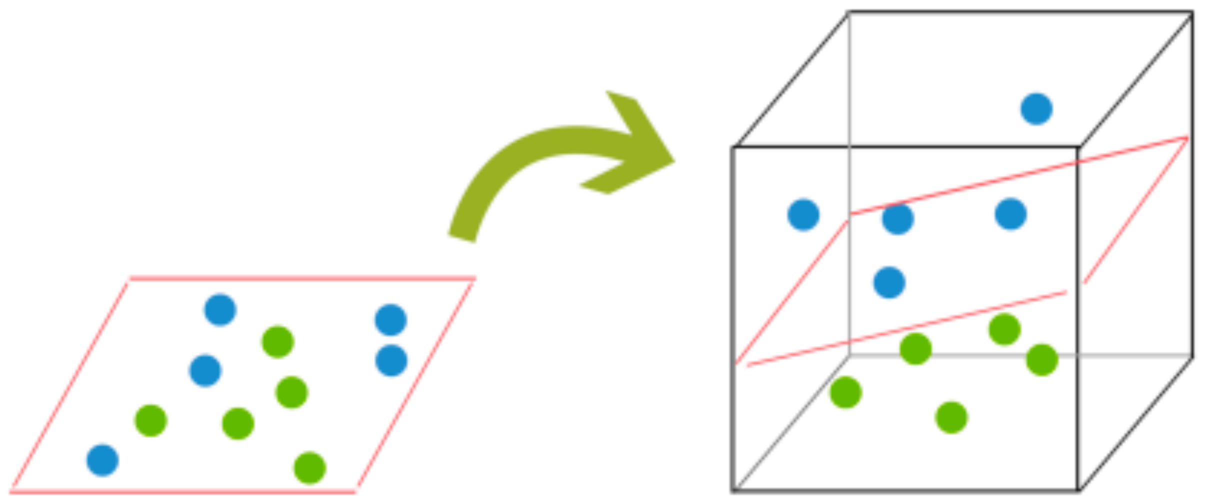
$$\mathcal{S}_u = \{ x'_1, \ldots, x'_{m_u} \}$$



Machine Learning Method

Prediction function $\hat{f}$

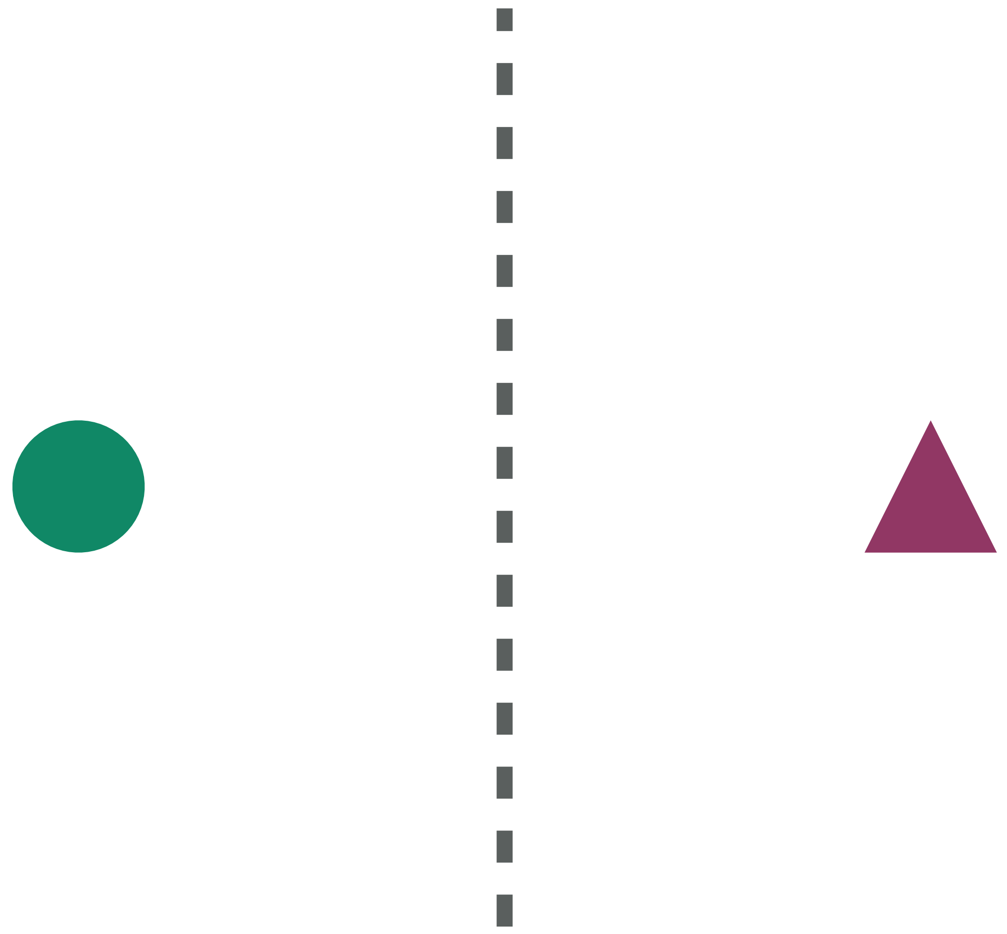


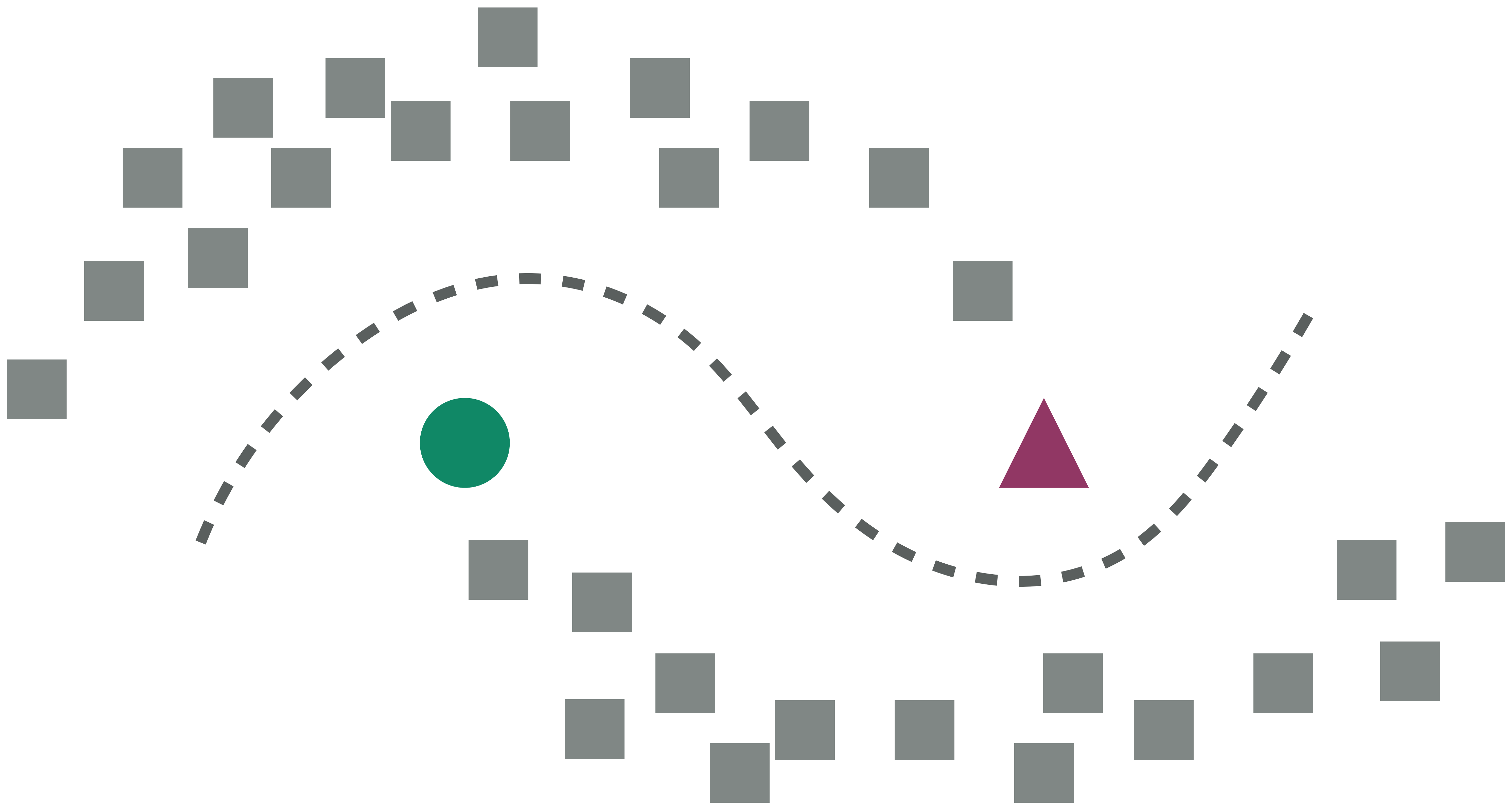


SEMI-SUPERVISED LEARNING - WHY?

We have access to a lot of unlabelled data, but why is it helpful?

- We can learn some structure of the data using the unlabelled points
- It can be helpful if we some knowledge of how the labels are related to the data distribution





Points in the same cluster are labelled similarly

SELF-TRAINING

Assumption: One's own predictions are good

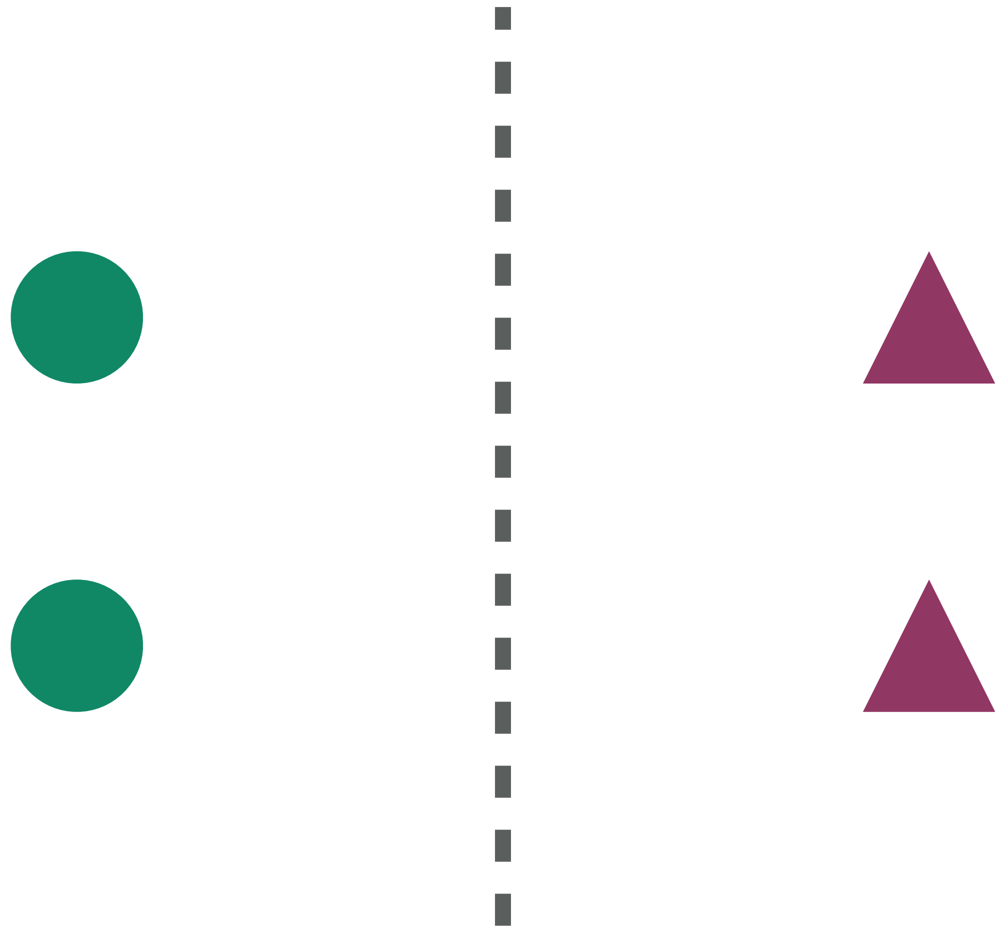
- Train f using the labelled training set S_l
- Predict pseudo-label $y_i' = f(x_i')$ for unlabelled examples $i \in [m_u]$
- Add a subset of the pseudo-labelled training set to the labelled training set
- Repeat

Alternative 1: Add only the most confident points

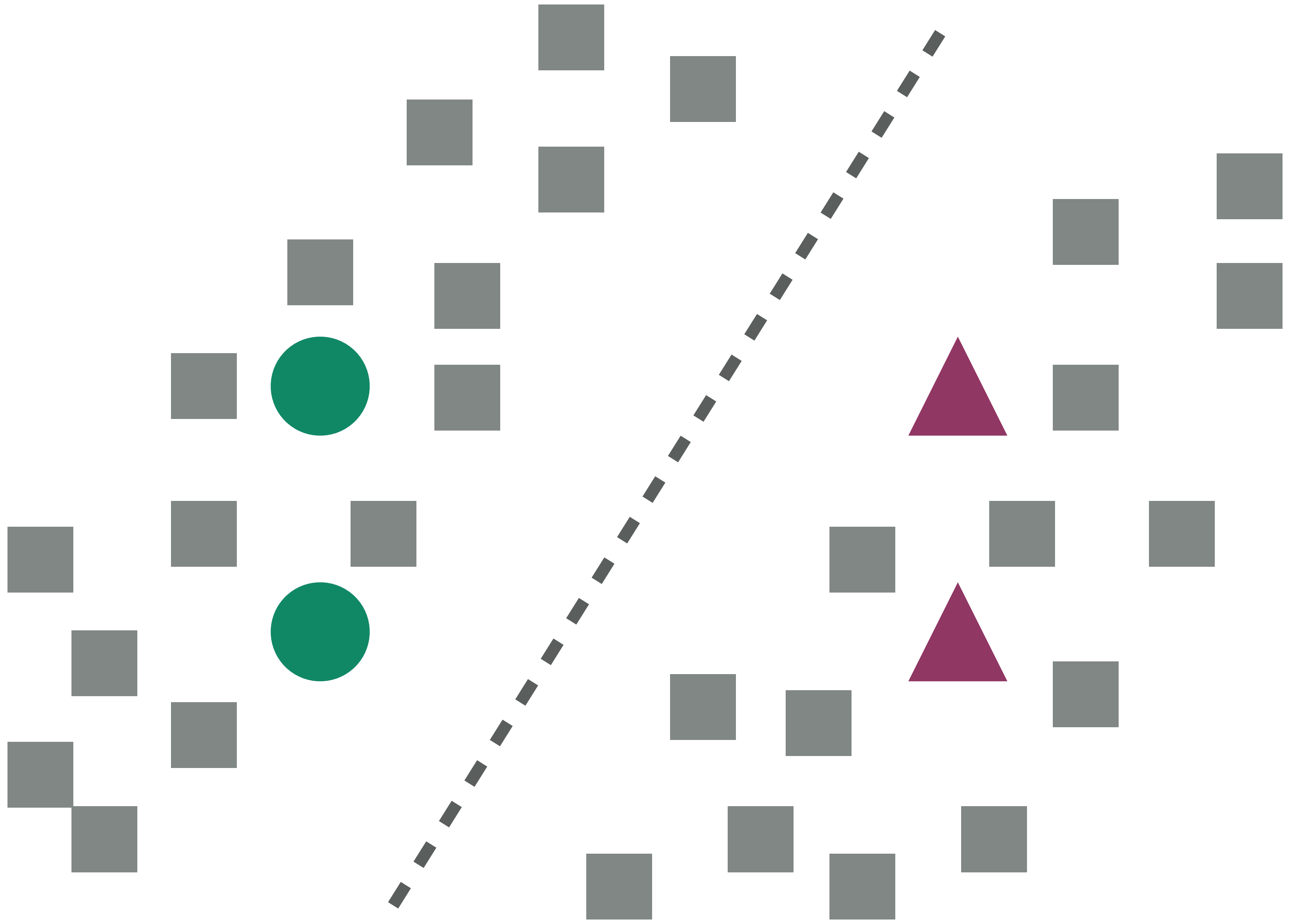
Alternative 2: Add points weighted by the confidence

SEMI-SUPERVISED SVM - MARGIN

Assumption: The classifier has large margin



SVM with only labelled data



Transductive SVM

SEMI-SUPERVISED SVM - MARGIN

Joachims'99

Labelled training dataset

$$\mathcal{S}_l = \left\{(x_1, y_1), (x_2, y_2), \ldots, (x_{m_l}, y_{m_l})\right\}$$

Unlabelled training dataset

$$\mathcal{S}_u = \{x'_1, \ldots, x'_{m_u}\}$$

SVM with only labelled data

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2$$

such that $\quad y_i(w^\top x_i + b) \geq 1, \forall i \in [m_l]$

SVM with unlabelled data

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2$$

such that

$$y_i(w^\top x_i + b) \geq 1, \forall i \in [m_l]$$

$$y'_i(w^\top x'_i + b) \geq 1, \forall i \in [m_u]$$

$$y'_i \in \{-1,1\}, \forall i \in [m_u]$$

Find a labelling y'_i for the unlabelled samples and w, b that maximize margin over all samples

SEMI-SUPERVISED SVM

Joachims'99

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2$$

such that
$$y_i(w^\top x_i + b) \geq 1, \forall i \in [m_l]$$
$$y_i'(w^\top x_i' + b) \geq 1, \forall i \in [m_u]$$
$$y_i' \in \{-1,1\}, \forall i \in [m_u]$$

Not a convex problem!
Convex once you fix y_i' for all $i \in [m_u]$.

Find a labelling y_i' for the unlabelled samples and w, b that maximize margin over all samples

- Heuristic method:
  - First maximize margin with only labelled samples
  - Use the w, b found to label the unlabelled samples
  - Try flipping labels of unlabelled points and see if margin increases
  - Keep going till no more improvements

Joachims'99

SEMI-SUPERVISED SVM - FAILURES

It is not always helpful though!

- If there is no margin
- If margin is satisfied in multiple ways

Margin but hard to know which one if we only have a few labelled points

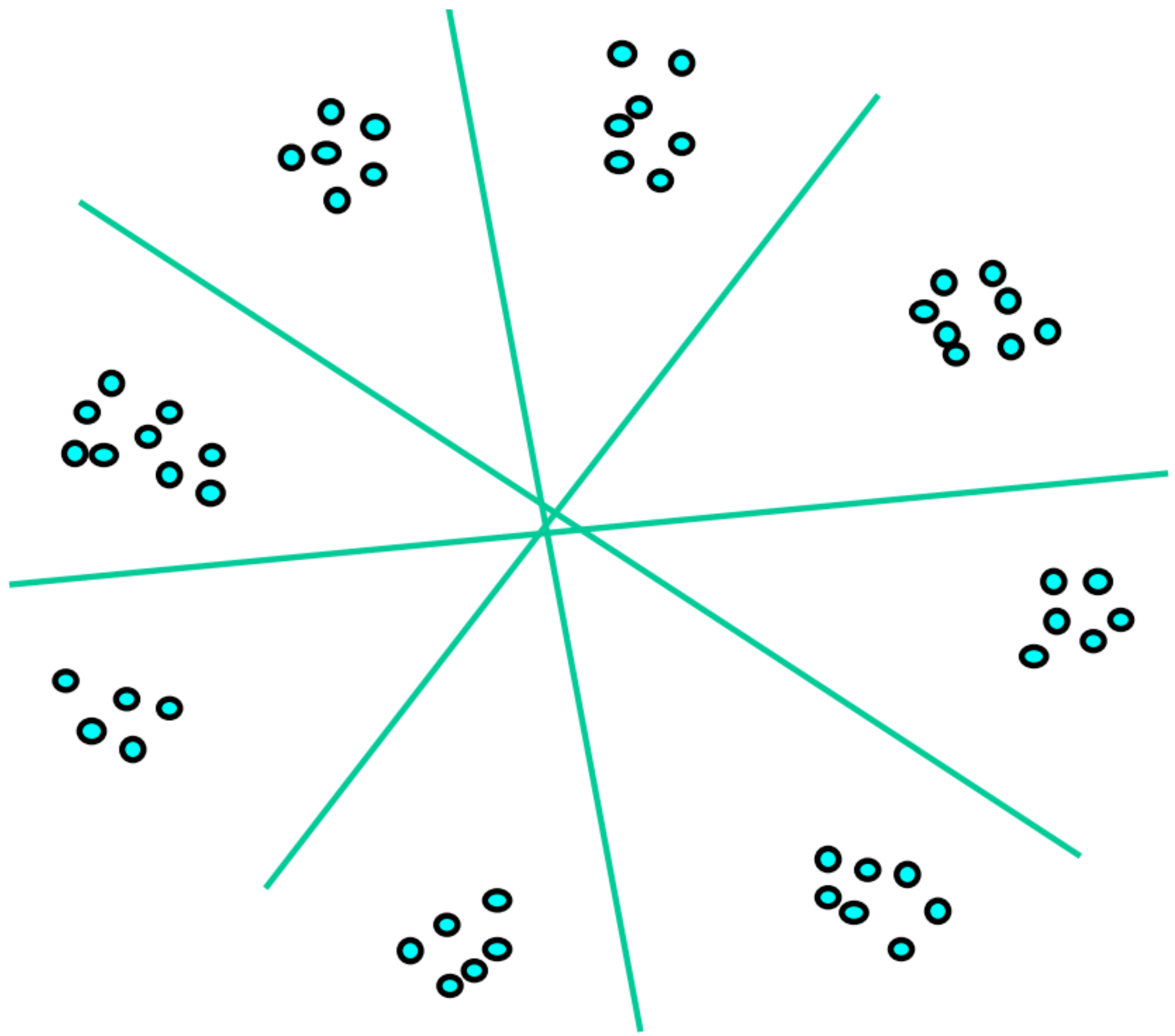
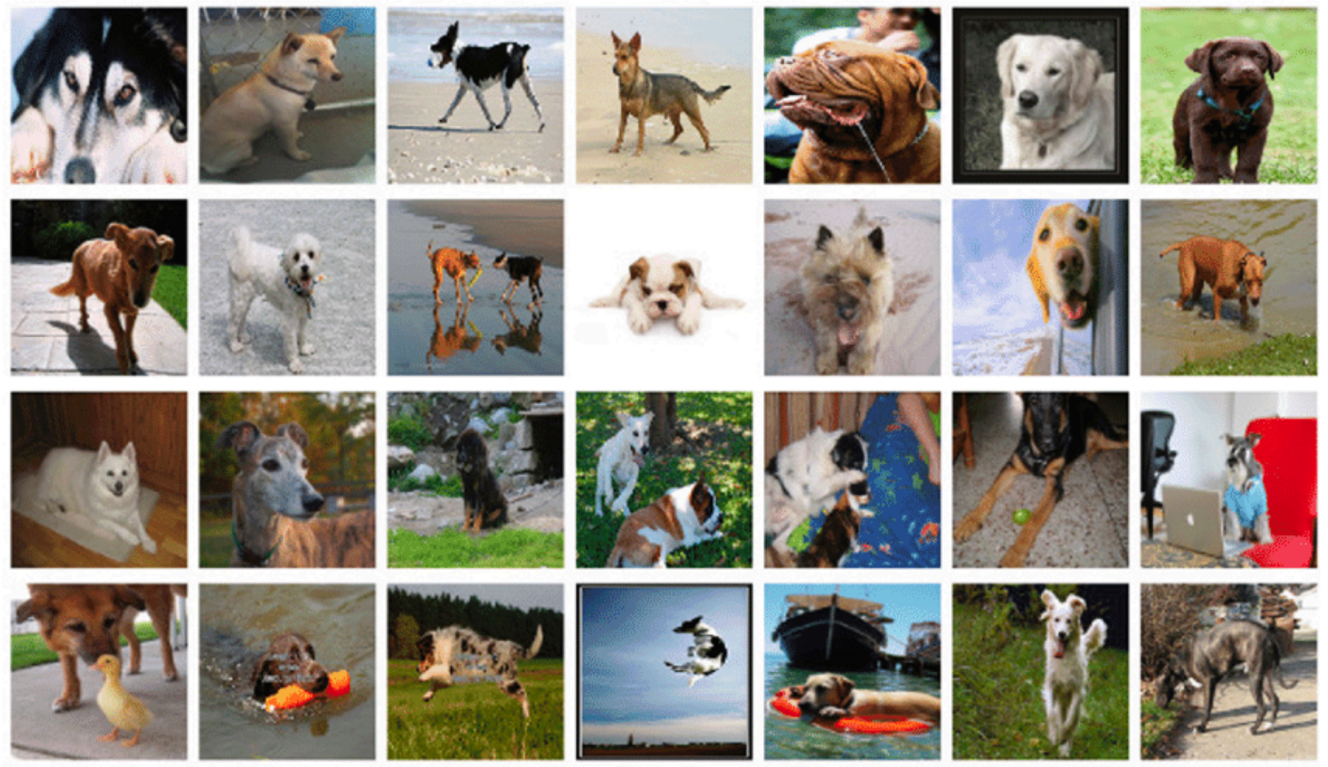


Image source: http://www.cs.cmu.edu/~ninamf/courses/315sp19/lectures/4_22-SSL.pdf

ACTIVE LEARNING

$m_l << m$

Unlabelled training dataset
$\mathcal{S} = \{x_1, \ldots, x_m\}$



Query labels of m_l points
from the training set

Machine
Learning
Method

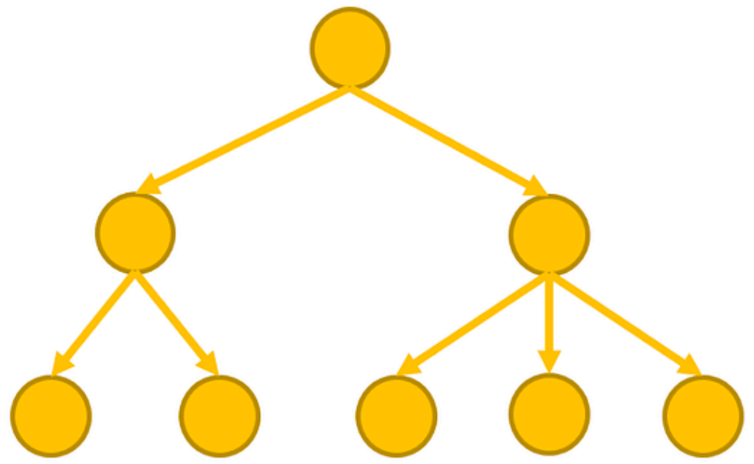
Pug

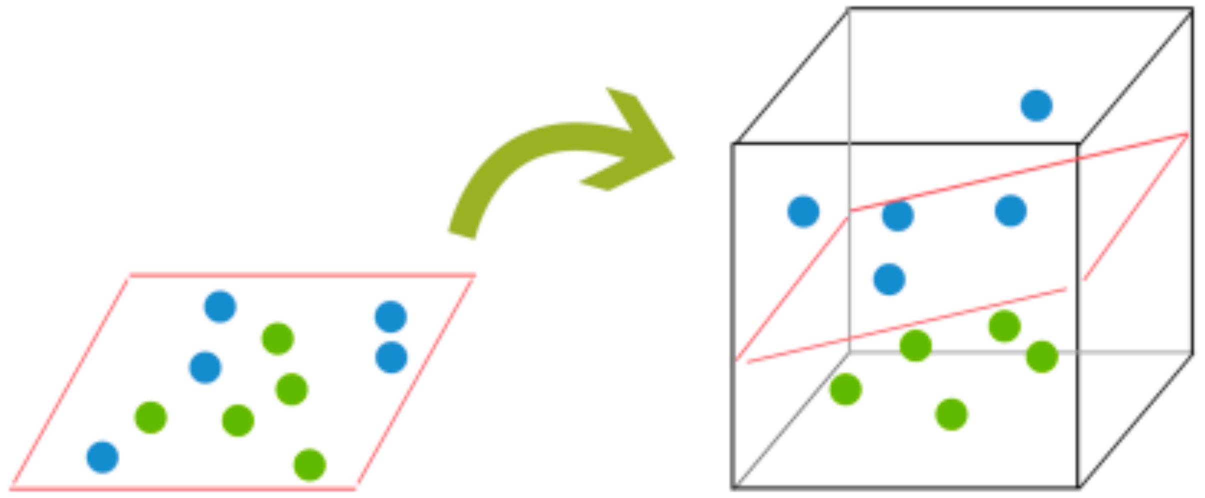
• • •

Chihuahua

Domain expert

Prediction function $\hat{f}$





ACTIVE LEARNING - WHY?

We have access to choosing which examples to label, but why is it helpful?

- We can choose more informative examples to query
- Reduce sample complexity compared to passive algorithms

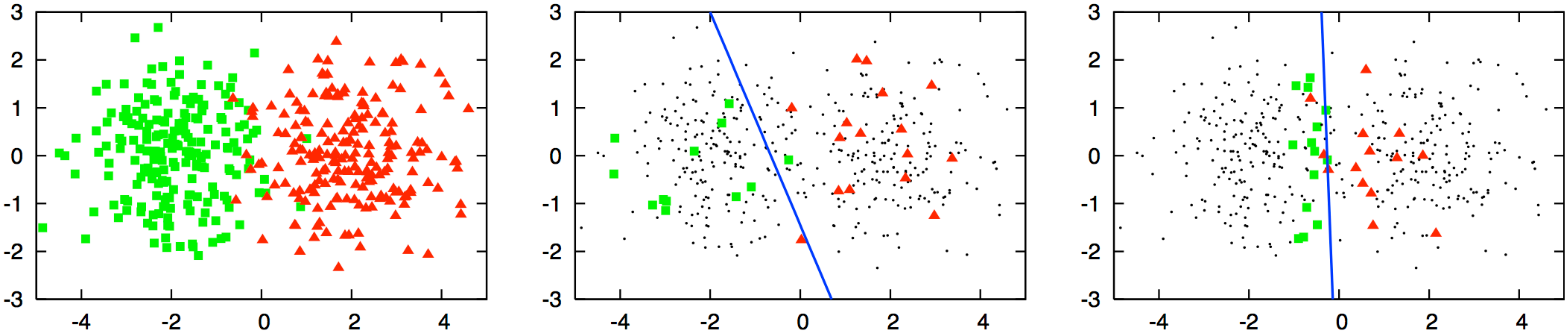
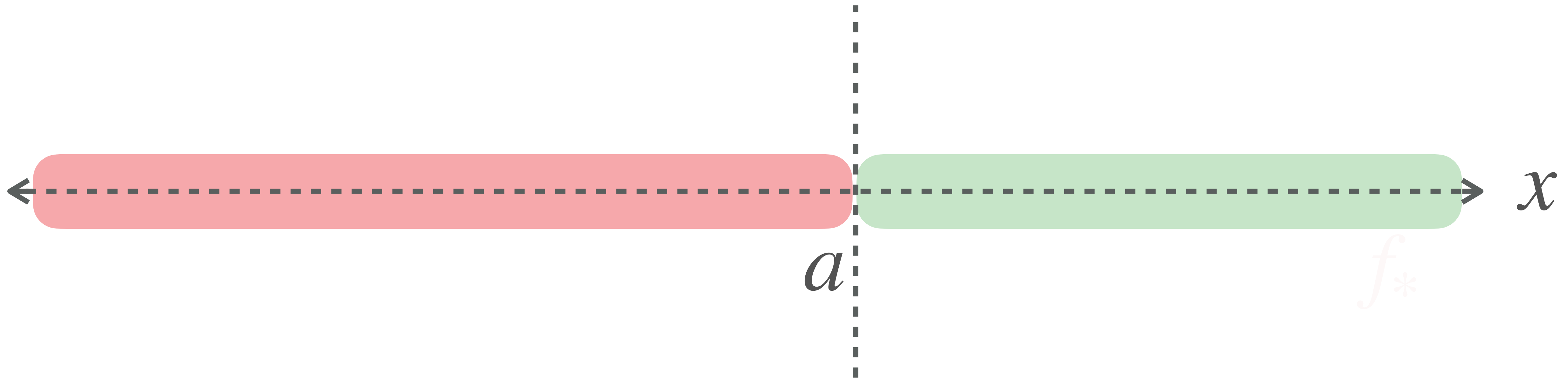


Image source: https://burrsettles.com/pub/settles.activelearning.pdf

ACTIVE LEARNING - THRESHOLDS

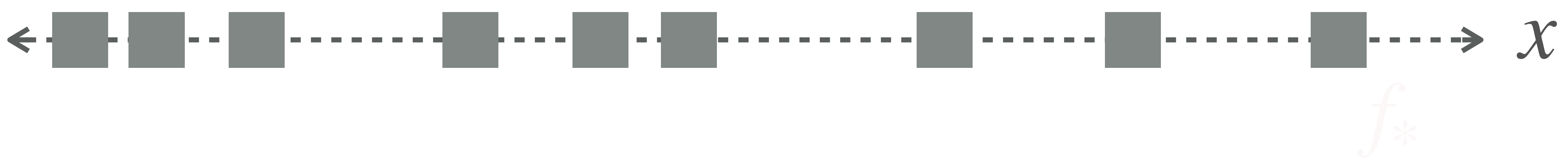


$$f_a(x) = \begin{cases} 1 & \text{if } x \geq a \\ -1 & \text{otherwise.} \end{cases}$$

VC dimension is $\mathbf{1}$

- Recall, from VC dimension bounds, we know ERM over $O(1/\epsilon)$ samples is enough to get ϵ error
- Passive learning would require $O(1/\epsilon)$ samples
- Active learning can do this with $O(\log(1/\epsilon))$ samples

ACTIVE LEARNING - THRESHOLDS



$$f_a(x) = \begin{cases} 1 & \text{if } x \geq a \\ -1 & \text{otherwise.} \end{cases}$$

Algorithm:

- Do binary search on the m unlabelled points

  Start by querying the median

  If + then recurse on the left half of the points

  If - then recurse on the right half of the points

ACTIVE SVM

Tong & Koller'00

Unlabelled training dataset

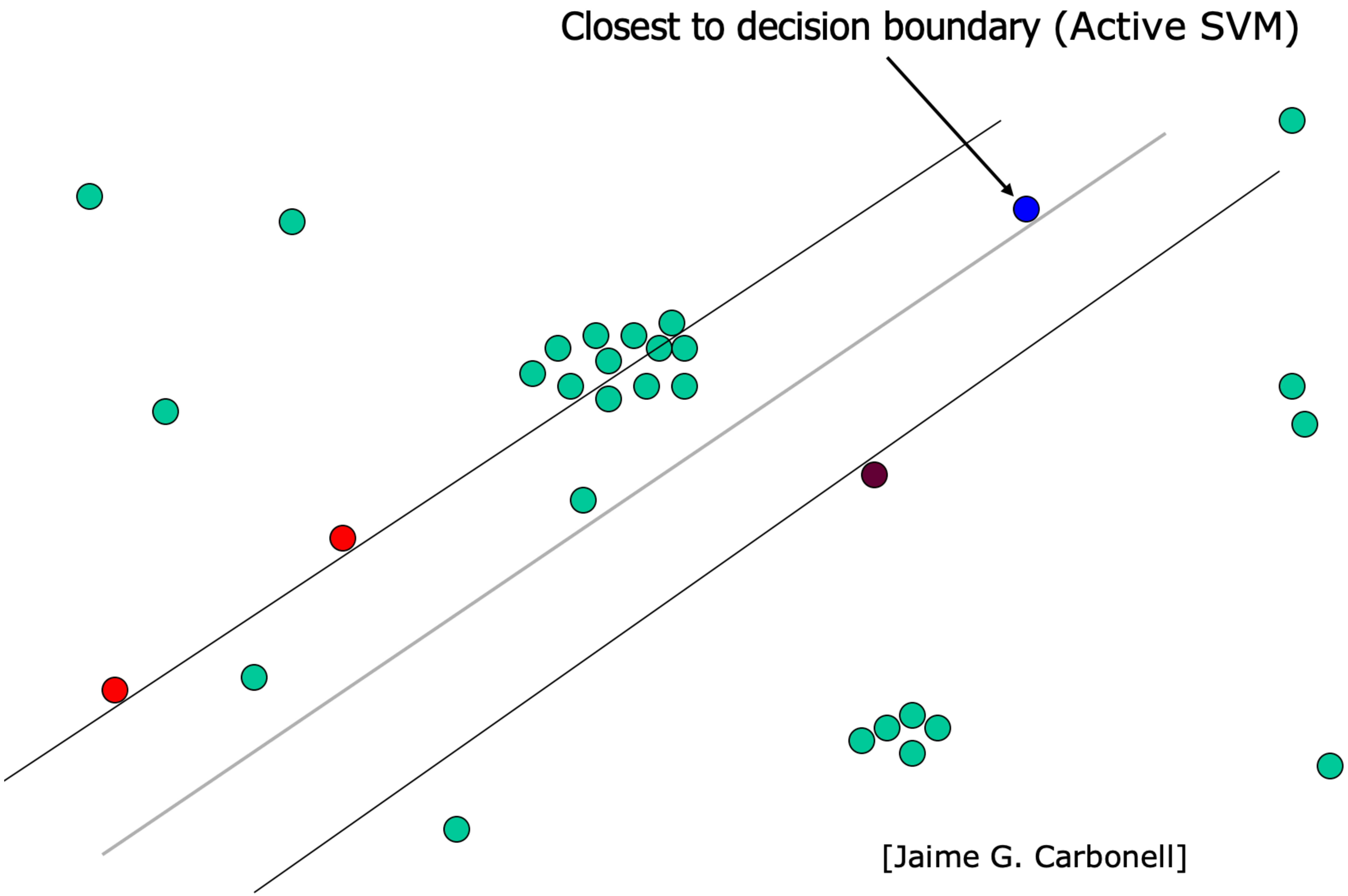
$$\mathcal{S} = \{x_1, \ldots, x_m\}$$

Algorithm:

- Query a few random examples to start
- Repeat for T iterations:
  - Find the max-margin classifier for all the labelled examples so far
  - Identify the closest unlabelled example to the decision boundary and query its label

Uncertainty estimation - closest to boundary, most uncertain

ACTIVE SVM - UNCERTAINTY BASED



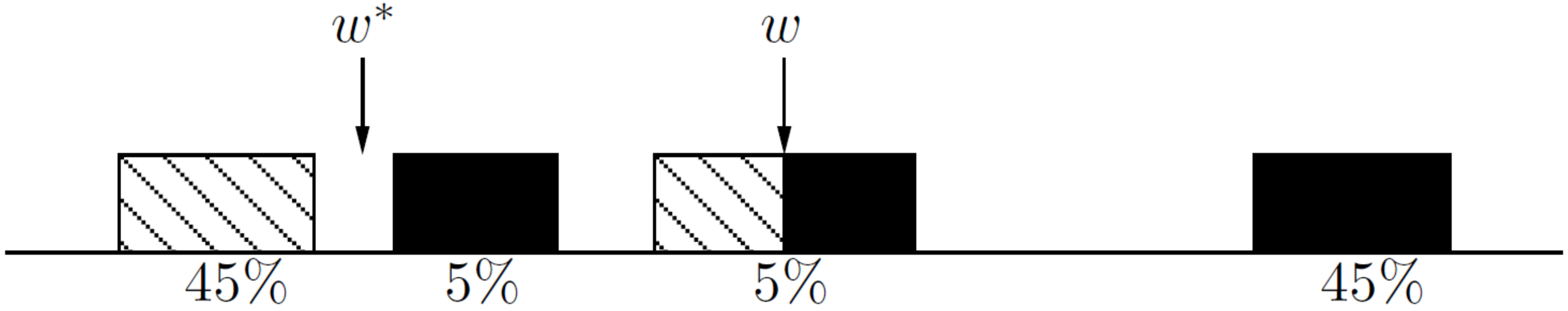
ACTIVE SVM - FAILURES

Tong & Koller'00

Can suffer from sampling bias

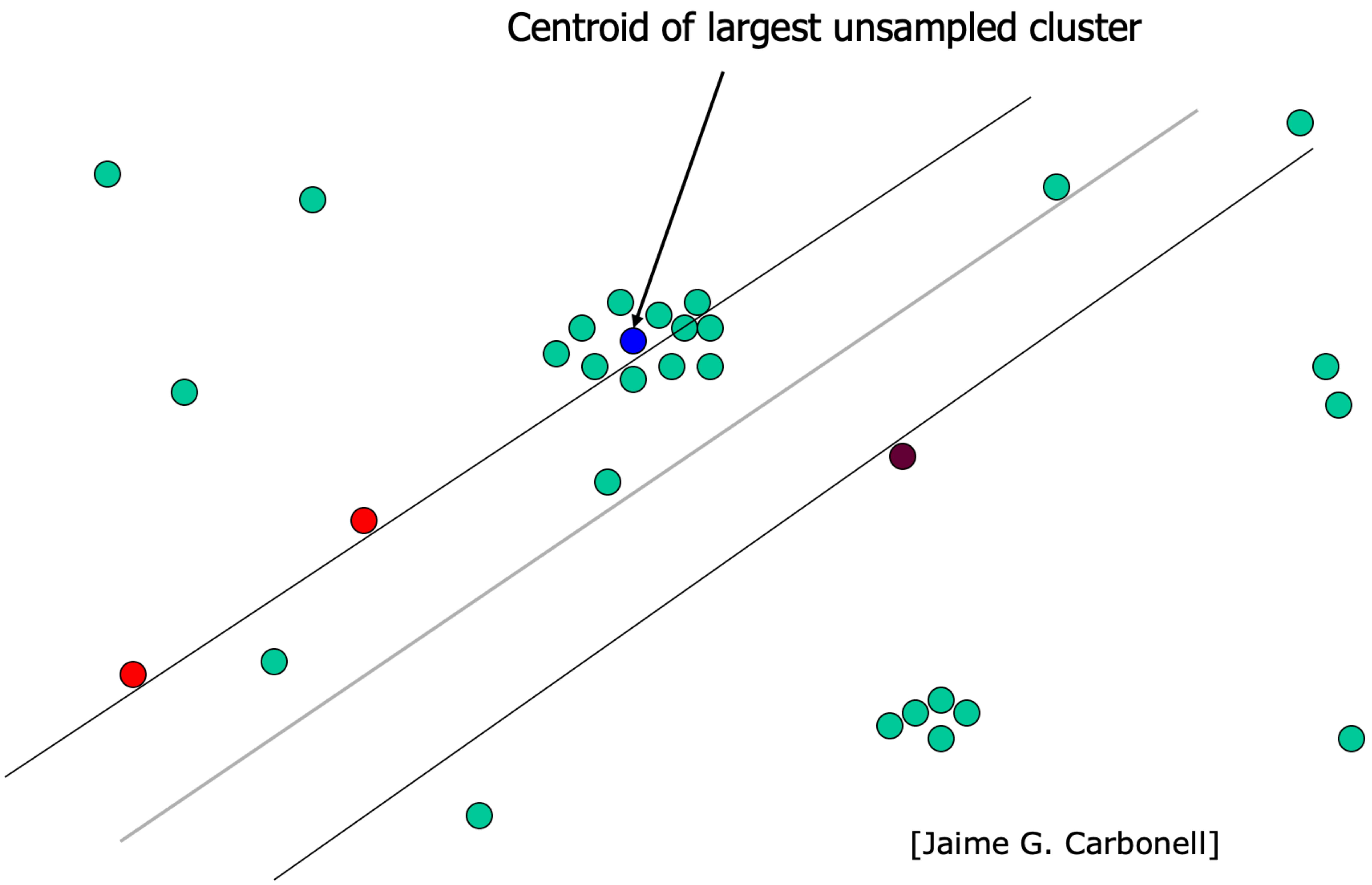
- As we run the algorithm, the labelled sample become less representative of the tru distribution

[Dasgupta10]

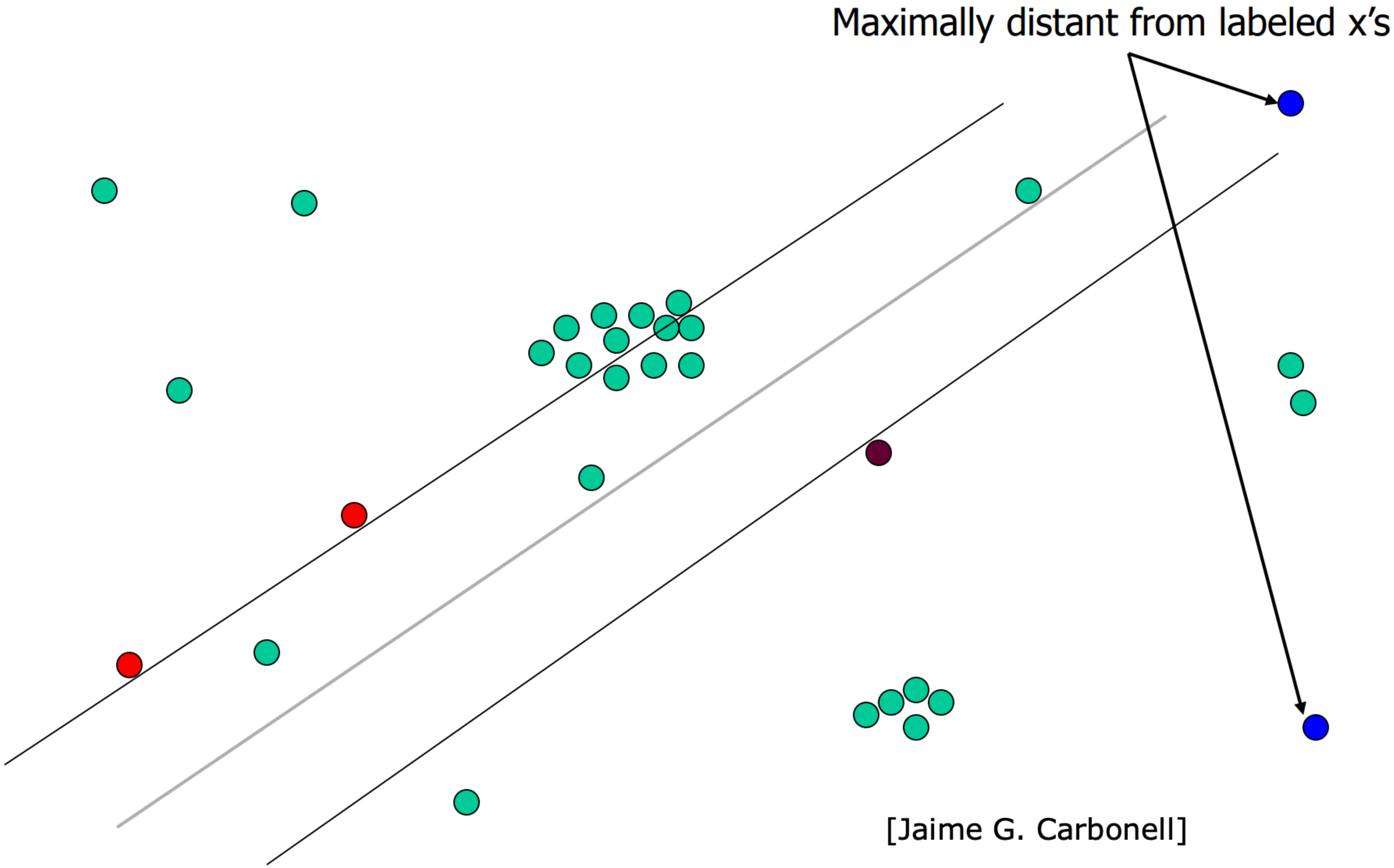


Happens in practice too!

OTHER TECHNIQUES - DENSITY BASED

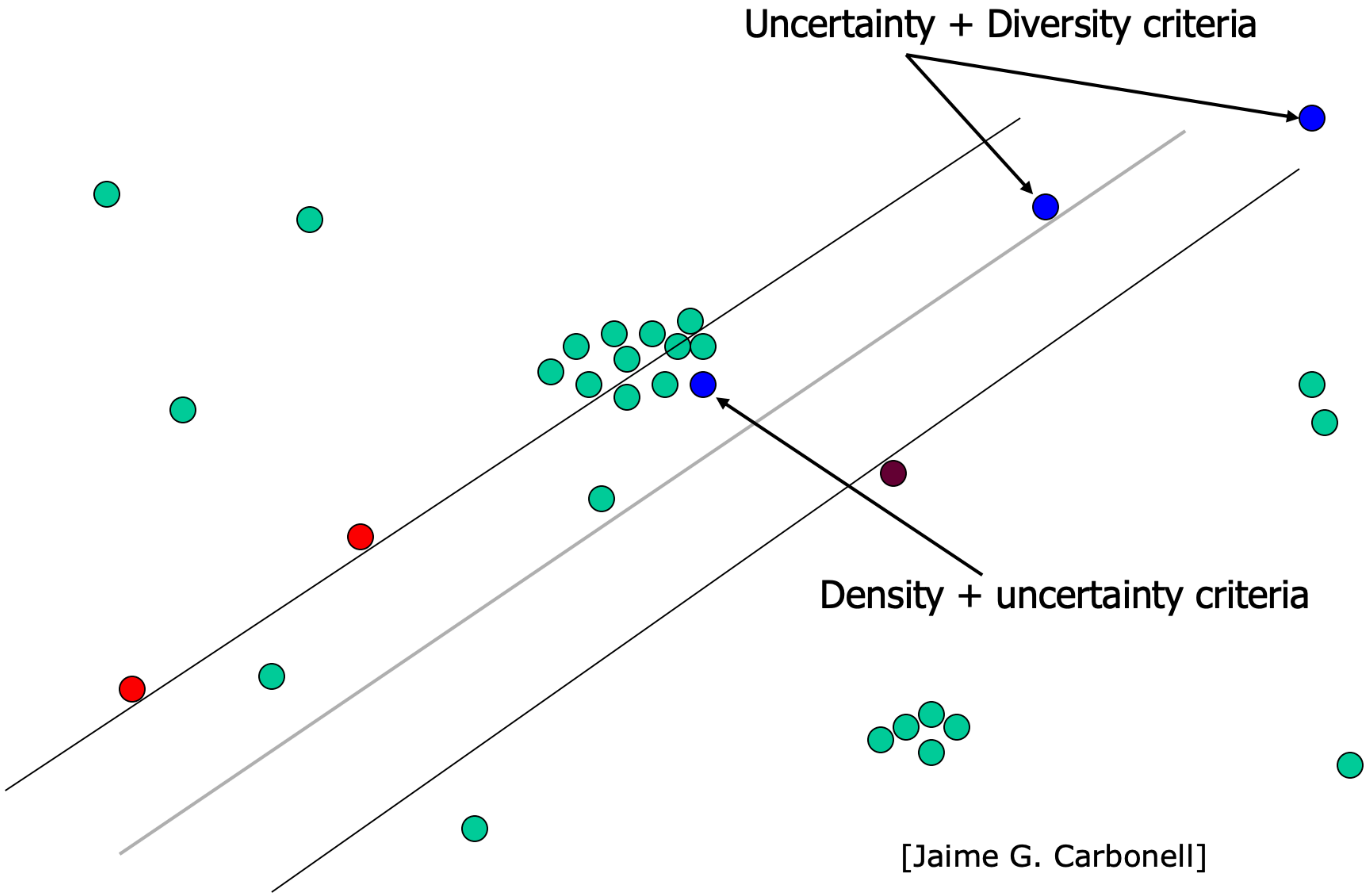


OTHER TECHNIQUES - MAXIMAL DIVERSITY



[Jaime G. Carbonell]

OTHER TECHNIQUES - ENSEMBLE



[Jaime G. Carbonell]

SAFE ACTIVE LEARNING - DISAGREEMENT BASED

Overcome the tension between choosing uncertain points and being faithful to the true distribution

Let us assume the realizable setting:

- Fix a function class $\mathcal{F}$
- There is some $f_* \in \mathcal{F}$ such that for all inputs $x \in \mathcal{X}$, $y = f_*(x)$

Definition (Version Space)

For a given set of labelled examples $(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})$ with $y_i = f_*(x_i)$, version space $VS(\mathcal{F})$ is the set of functions that are consistent with the labels, that is,

$$f \in \mathcal{F} \text{ iff } f(x_i) = f_*(x_i) \text{ for all } i \in [m_l].$$

SAFE ACTIVE LEARNING - DISAGREEMENT BASED

Definition (Version Space)

For a given set of labelled examples $(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})$ with $y_i = f_*(x_i)$, version space $VS(\mathcal{F})$ is the set of functions that are consistent with the labels, that is,

$$f \in \mathcal{F} \text{ iff } f(x_i) = f_*(x_i) \text{ for all } i \in [m_l].$$

Definition (Disagreement Region)

Part of the version space that still has uncertainty, that is,

$$x \in DIS(VS(\mathcal{F})) \text{ iff } \exists f_1, f_2 \in VS(\mathcal{F}) \text{ such that } f_1(x) \neq f_2(x).$$

SAFE ACTIVE LEARNING - DISAGREEMENT BASED

CAL'92

Algorithm:

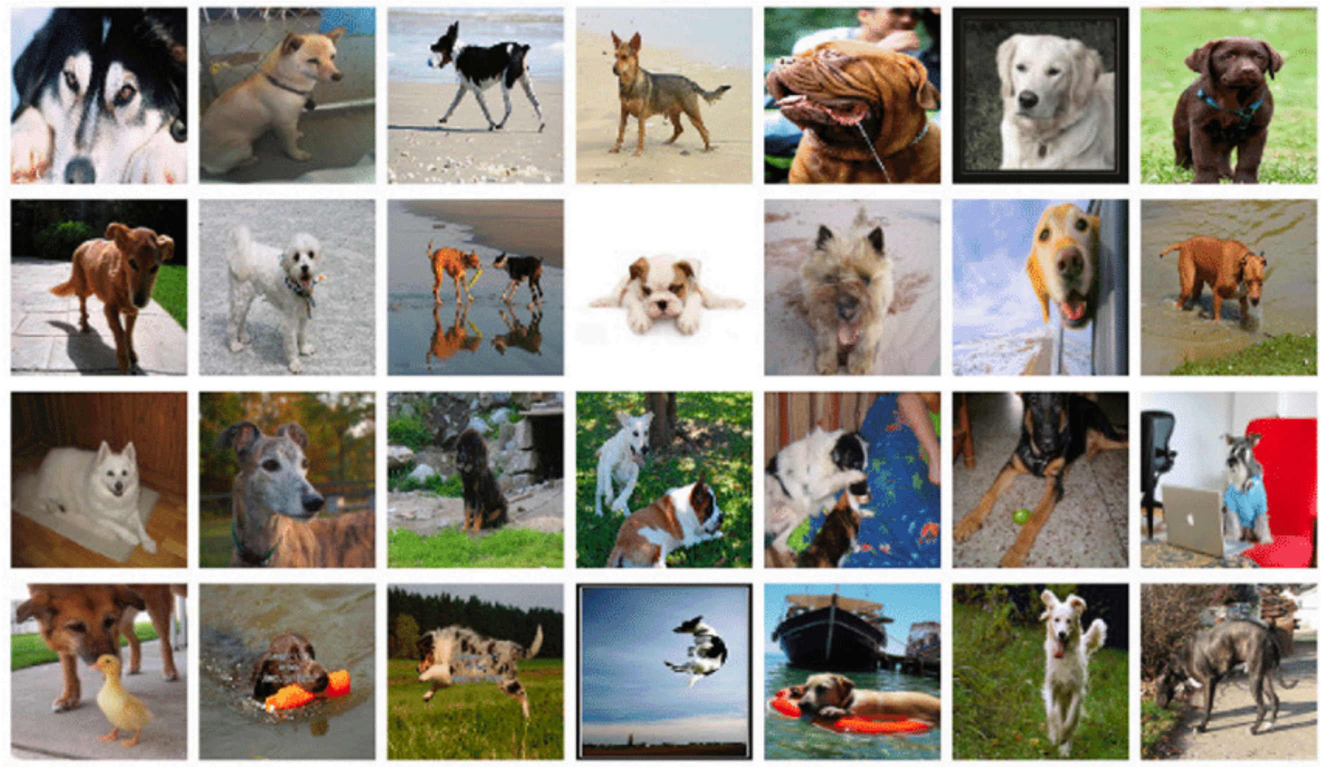
- Query a few random points to start
- Let V_1 be the current version space
- For $t = 1, \ldots$
    - Query a few random points in $DIS(V_t)$
    - Update the version space to V_{t+1} using these points

Can stop when the disagreement region has very small mass/number of points

SELF-SUPERVISED LEARNING

Unlabelled training dataset

$$\mathcal{S} = \{x_1, \ldots, x_m\}$$



Create pretext tasks

Machine Learning Method

Prediction $\hat{f}$

Extract representation

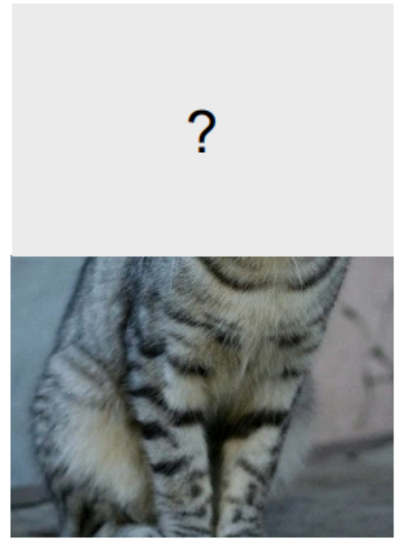
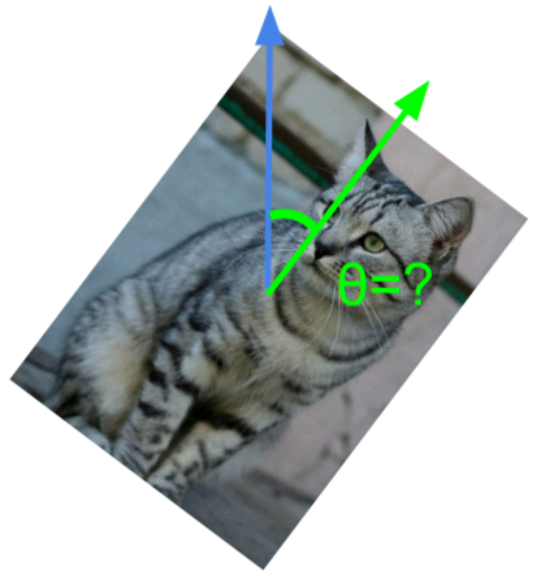
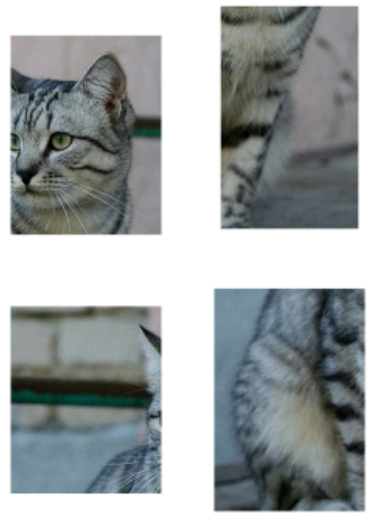


image completion



rotation prediction



"jigsaw puzzle"

colorization

MODERN ML - REDUCE RELIANCE ON LABELS

Can we train machine learning models with less human supervision?

No labelled data → Fully labelled data

Unsupervised Learning

Clustering (K-means)
Density Estimation (GMM)
Dimensionality Reduction (PCA)

Can identify structures or patterns in data

Self-supervised Learning

Semi-supervised Learning

Active Learning

Supervised Learning

Regression (Linear Regression)
Classification (SVM, Perceptron, Logistics Regression)

Can learn mapping from input to label