Semi-supervised learning

Semester 1, 2021

Kris Ehinger

Importance of data

"memorization is a good policy if you have a lot of training data. … simple models and a lot of data trump more elaborate models based on less data."

– Halevy, Norvig, & Pereira (2009)

"The Unreasonable Effectiveness of Data"

Importance of data

- (Labelled) data is a bottleneck for machine learning
- In image classification, model depth has increased dramatically, but the size of "large-scale" datasets has not kept pace

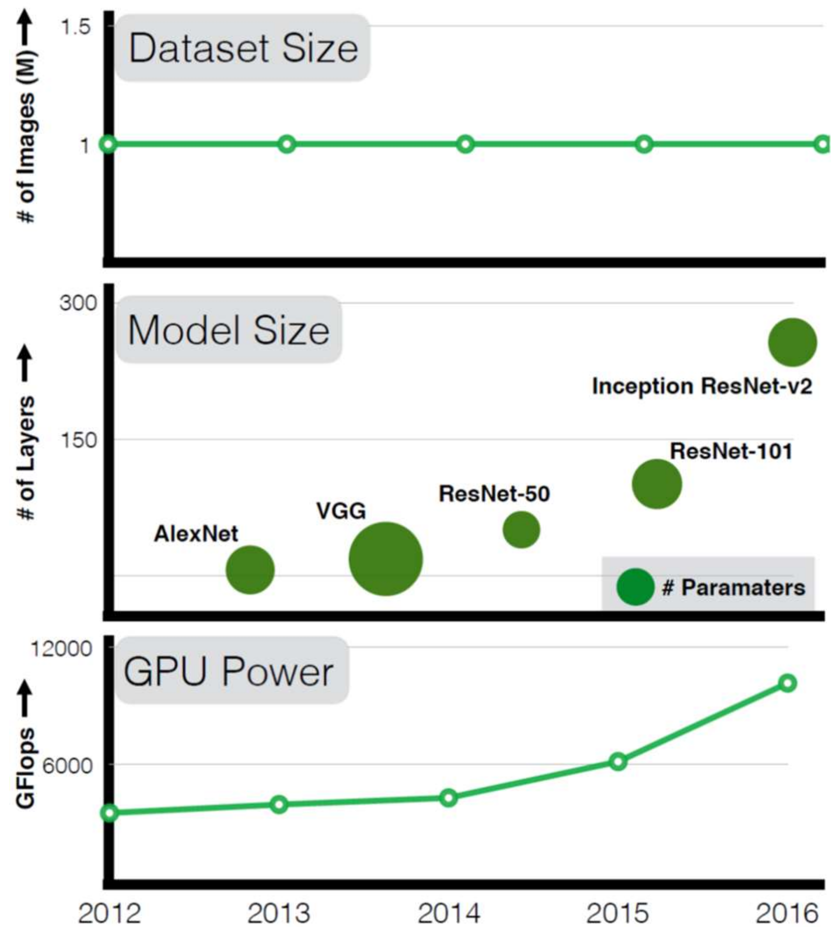


Figure: Sun, Shrivastava, Singh, & Gupta (2017)

Importance of data

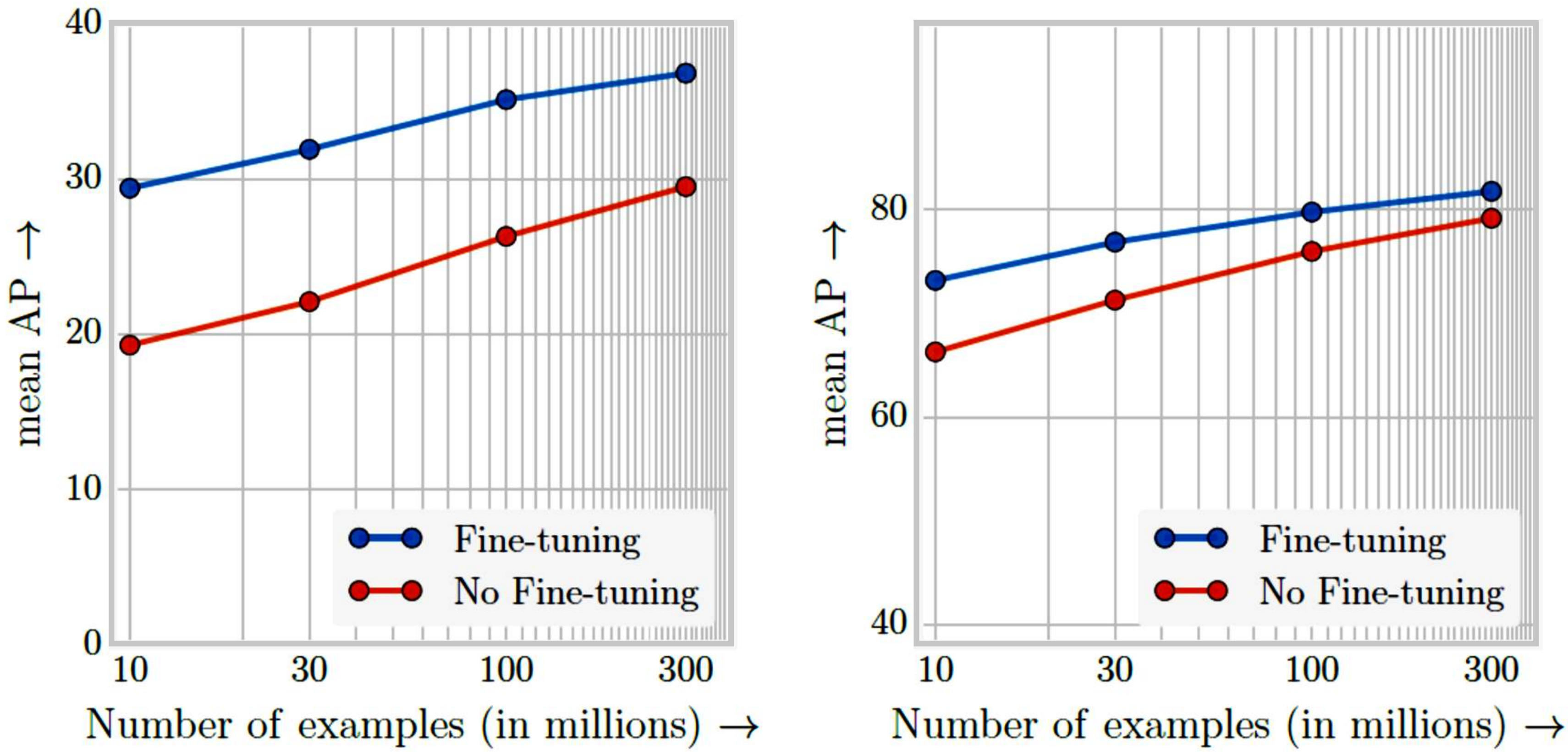


Figure: Sun, Shrivastava, Singh, & Gupta (2017)

Importance of data

- Adding data is nearly as effective as adding layers
- More parameters are not helpful unless you have more data to train them

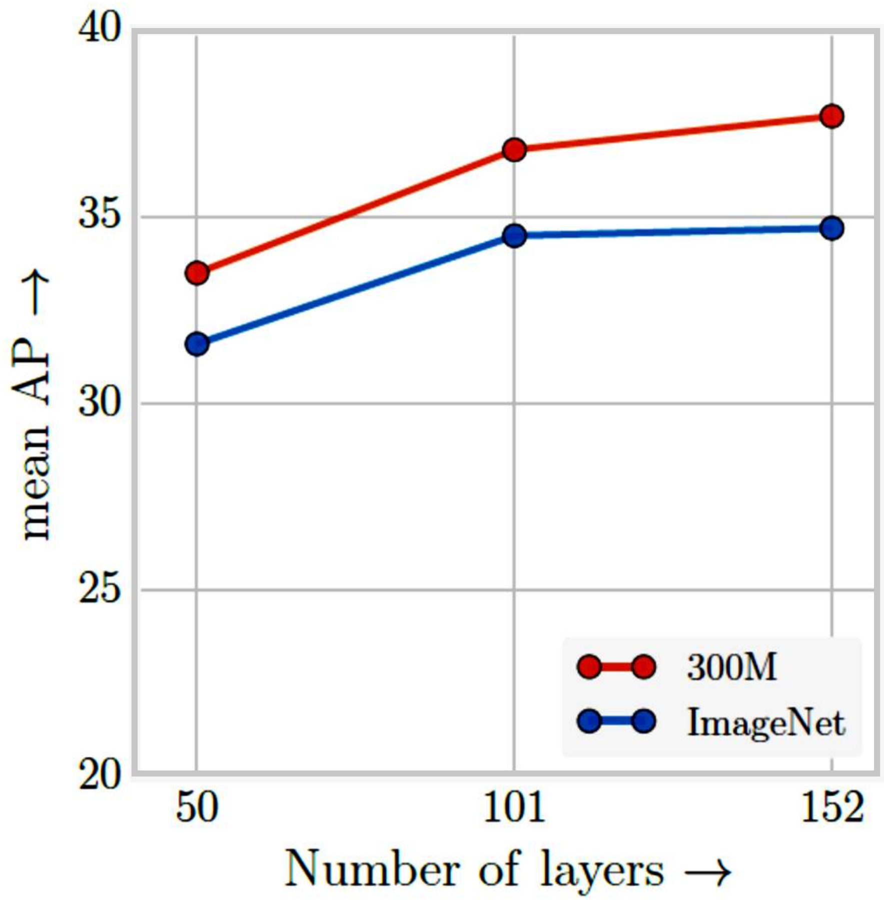


Figure: Sun, Shrivastava, Singh, & Gupta (2017)

Outline

- Semi-supervised learning
- Active learning
- How to get more data?

Semi-supervised learning

- We’ve covered several methods for supervised learning with fully-labelled datasets
- Last time, we covered some unsupervised learning methods for unlabelled datasets
- What if we had a mix of labelled and unlabelled data?
- What if we had a large unlabelled dataset and limited budget (time and/or money) for labelling?

Semi-supervised learning

- Semi-supervised learning is learning from both labelled and unlabelled data
- Semi-supervised classification
  - Training data consists of L labelled instances $\langle x_i, y_i \rangle$ and U unlabelled instances $\langle x_j \rangle$
  - Often $U \gg L$
  - Goal: learn a better classifier from $L \cup U$ than is possible from L alone

Why is it important?

- Data is (often) abundant but labelling is expensive
  - Switchboard corpus: 400 hours of annotation time per hour of speech data
  - Image labelling: often 30-60 minutes per image for a complete segmentation

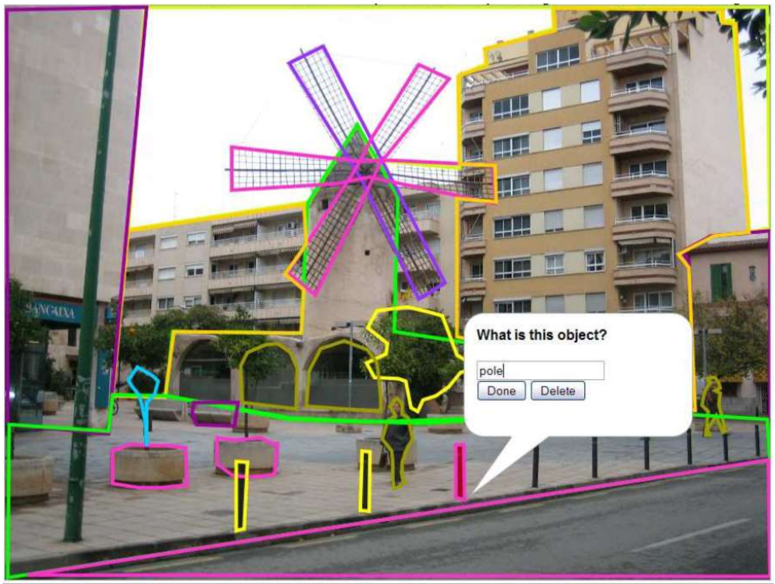
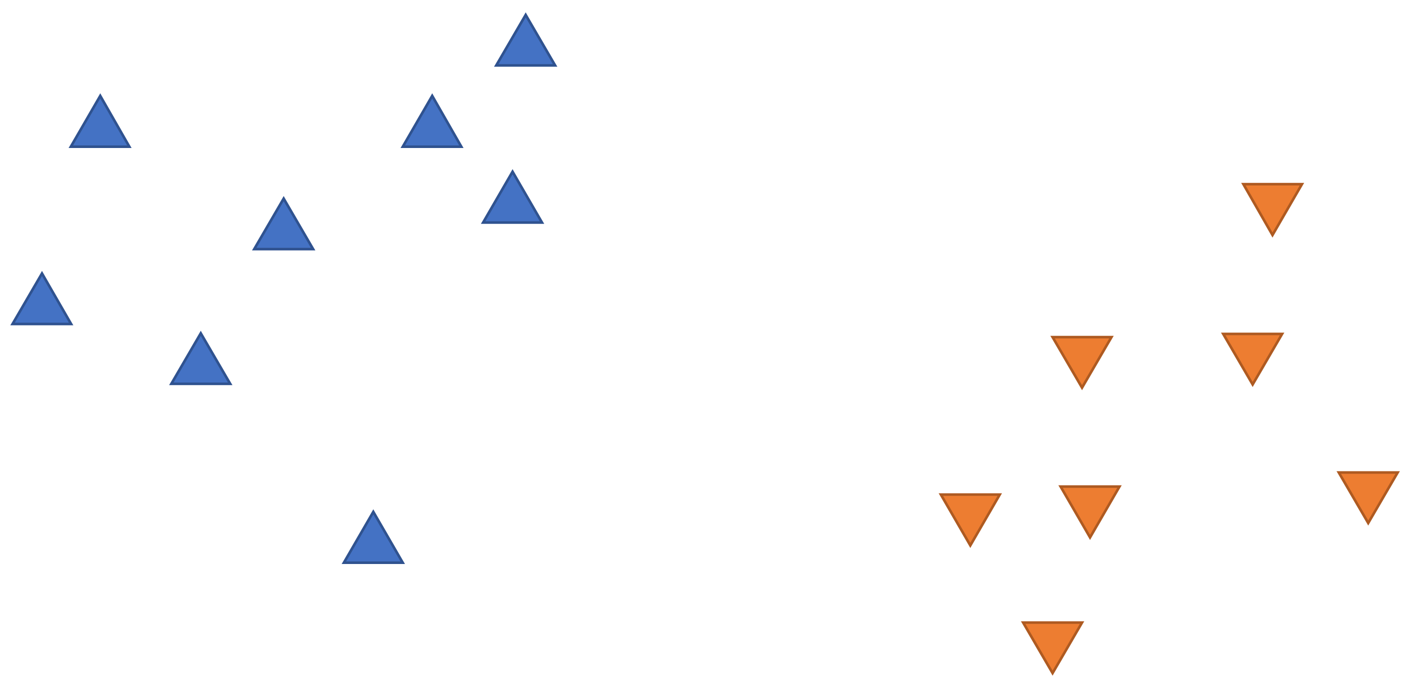
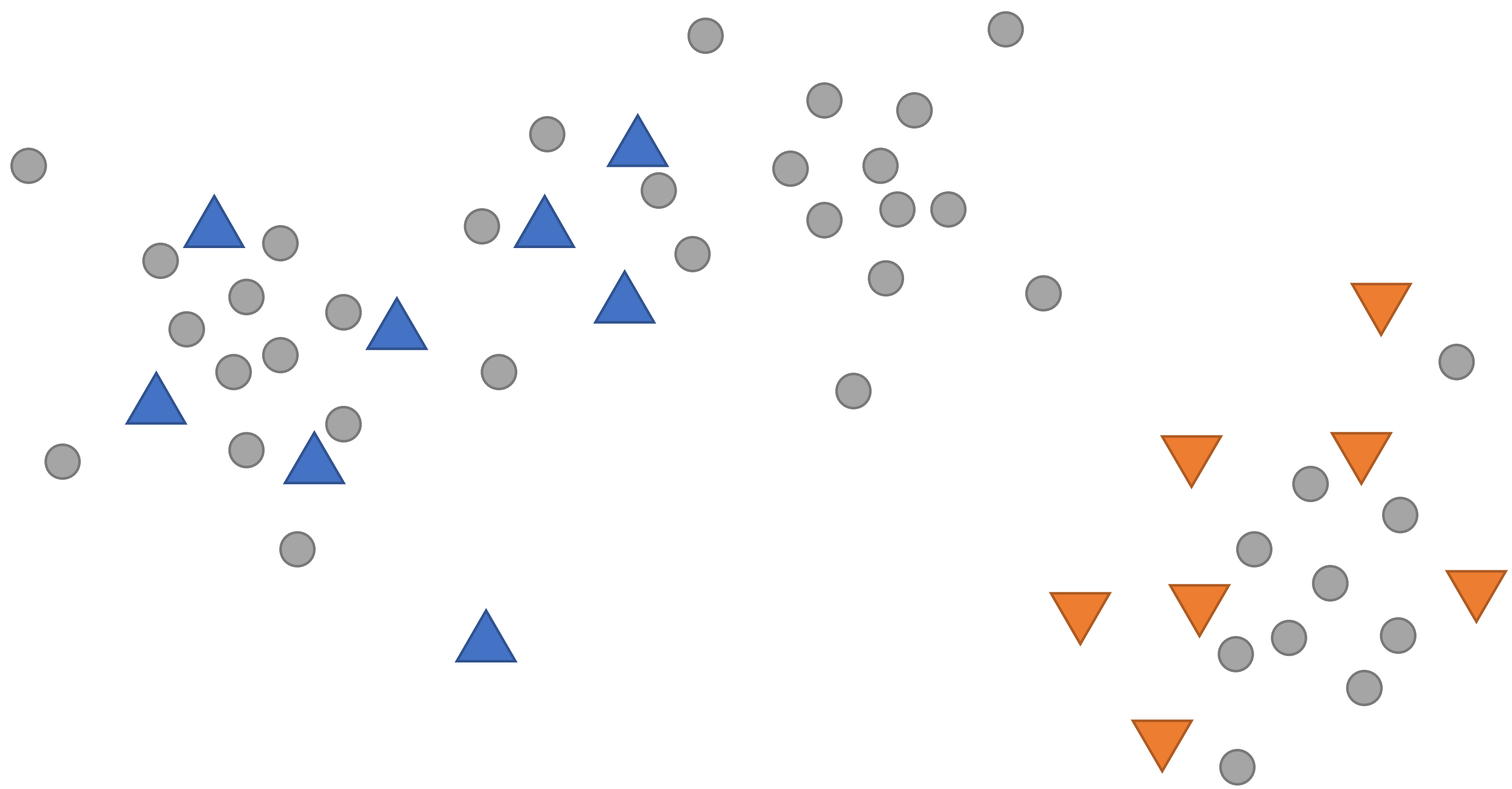


Image: Torralba, Russell, & Yuen (2010)

Semi-supervised learning



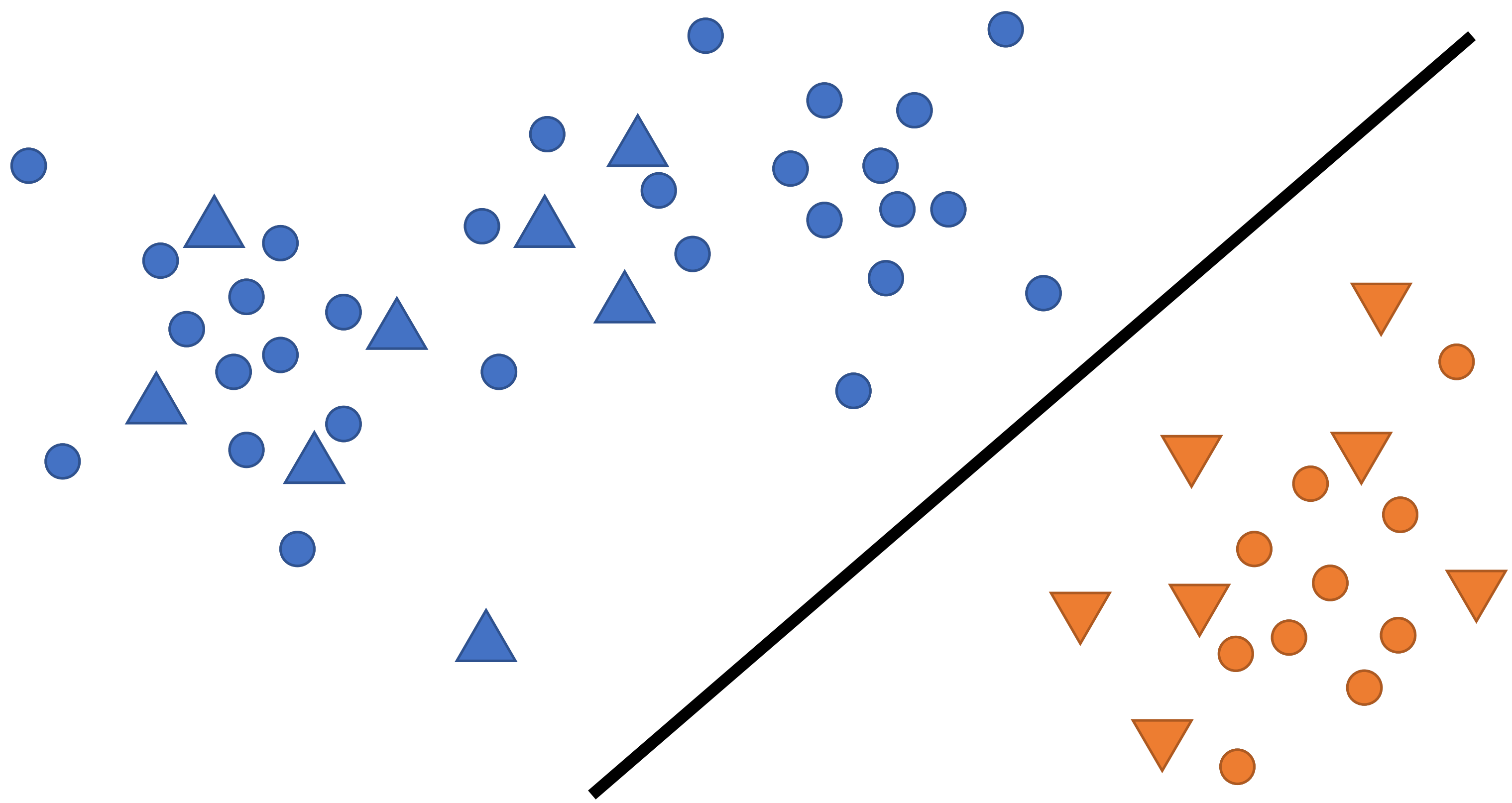
Semi-supervised learning



Semi-supervised learning

- A simple approach: combine a supervised and unsupervised model
- For example: Find clusters, choose a label for each (most common label?) and apply it to the unlabelled cluster members

Semi-supervised learning



Self training

- Assume you have labelled data $L = \langle x_i, y_i \rangle$ and unlabelled data $U = \langle x_j \rangle$
- Repeat:
  - Train a model f_i on L using supervised learning method
  - Apply f_i to predict labels on each instance in U
  - Identify a subset U' of U with "high-confidence" labels
  - Remove U' from the unlabelled and add it to the labelled set with the classifier predictions as "ground truth" labels ($U \leftarrow U \setminus U'$ and $L \leftarrow L \cup U'$)
  - Until L does not change
- Also known as "bootstrapping"

Self-training example: 1-NN

- 1-nearest neighbour with labelled data $L = \langle x_i, y_i \rangle$ and unlabelled data $U = \langle x_j \rangle$
- Repeat:
  - Find neighbours for unlabelled points in U
  - For points x whose nearest neighbour is in the labelled set, take the labels y' from the nearest neighbour
  - $U \leftarrow U \setminus \langle x \rangle$
  - $L \leftarrow L \cup \langle x, y' \rangle$
  - Until there is no change in the labelling

Self-training example: NB

- Naïve Bayes with labelled data $L = \langle X_i, Y_i \rangle$ and unlabelled data $U = \langle X_j \rangle$
- Initialization: Train on labelled data to learn P(X|Y) and P(Y) for all attributes X and all classes Y
- Run EM algorithm:
  - E(xpectation): For each unlabelled instance, compute a probability distribution over classes
  - M(aximisation): Recompute P(X|Y) and P(Y) with all data, weighting the unlabelled instances by their probability of being in each class

Self-training example: NB

- Problem: if the unlabelled dataset is much larger than the labelled dataset, probability estimates will be based almost entirely on unlabelled data
- Solution: add only a small amount of unlabelled data initially and gradually add more in later EM iterations

Assumptions

- Propagating labels requires some assumptions about the distribution of labels over instances:
  - Points that are nearby are likely to have the same label
- Not really creating data from nothing
- Classification errors are also propagated
  - One option: move points back to the “unlabelled” pool if the classification confidence falls below a threshold

Active Learning

Supervised classification

“Yes”	“No”

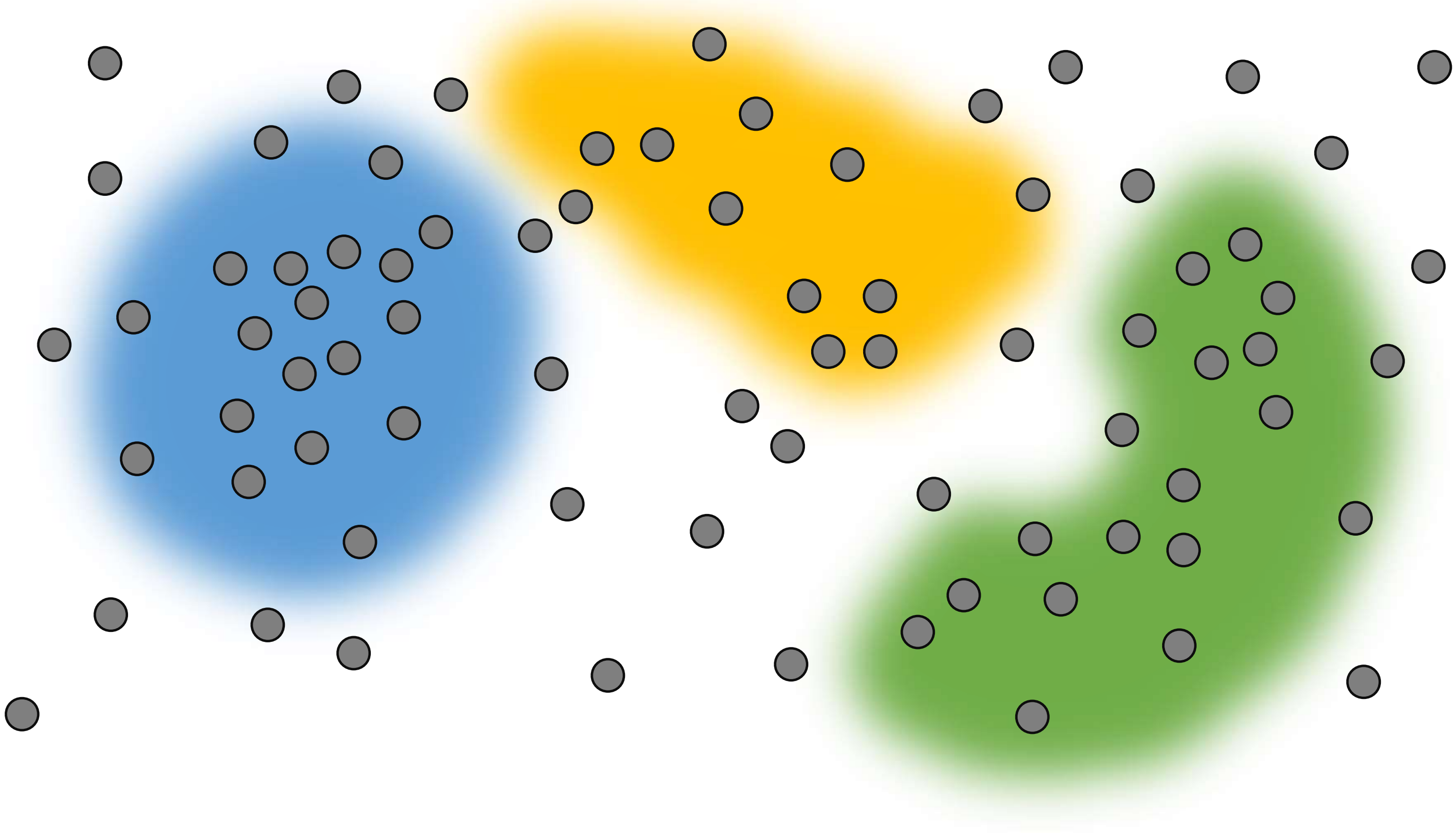
Supervised classification

“Yes”	“No”

Active learning

- Hypothesis: a classifier could achieve higher accuracy with fewer training instances if it were allowed to have some say in the selection of the training instances
- Labelling is a finite resource, so instances should be labelled in a way that maximizes learning
- Active learners pose **queries** (unlabelled instances) for labelling by an **oracle** (e.g., a human annotator)

Active learning in theory



Active learning in theory

- Ideally, we'd the instances that are most effective for distinguishing between competing models
  - To do this most efficiently, we should have some sense of the likelihood of different models, or knowledge of how labels are distributed over instances, which usually isn't the case
- In machine learning, querying generally focuses on instances with high uncertainty, e.g.:
  - Instances near the boundaries between classes
  - Instances in regions with few labels

Query strategies

- Which unlabelled instances will be most useful for learning?
- One simple strategy: query instances where the classifier is least confident of the classification

$$x = \arg\max_{x}(1 - P_\theta(\hat{y}|x))$$

where $\hat{y} = \arg\max_{y} P_\theta(y|x)$

Query strategies

- Alternatively, margin sampling selects queries where the classifier is least able to distinguish between two categories, e.g.:

$$x = \arg\min_x (P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x))$$

  where $\hat{y}_1$ and $\hat{y}_2$ are the first- and second-most-probable labels for x

- Or better still, use entropy as an uncertainty measure:

$$x = \arg\max_x - \sum_i P_\theta(\hat{y}_i|x) \log_2 P_\theta(\hat{y}_i|x)$$

Query strategies

- A more complex strategy, if you have multiple classifiers: **query by committee (QBC)**
- Train multiple classifiers on a labelled dataset, use each to predict on unlabelled data, and select instances with the highest disagreement between classifiers
- Assumes that all the classifiers learn something different, so can provide different information
- Disagreement can be measured by entropy

Active learning: Practicalities

- Advantages
  - Empirically, seems to be a robust strategy to increase accuracy
- Disadvantages
  - Often difficult to justify these strategies theoretically
  - Introduces bias, resulting in a dataset that might not be as useful for other machine learning tasks
  - May be sensitive to label noise

How to get more data?

Data augmentation

- There are various ways to expand a labelled training dataset
- General: resampling methods
- Dataset-specific: add artificial variation to each instance, without changing ground truth label

Bootstrap sampling

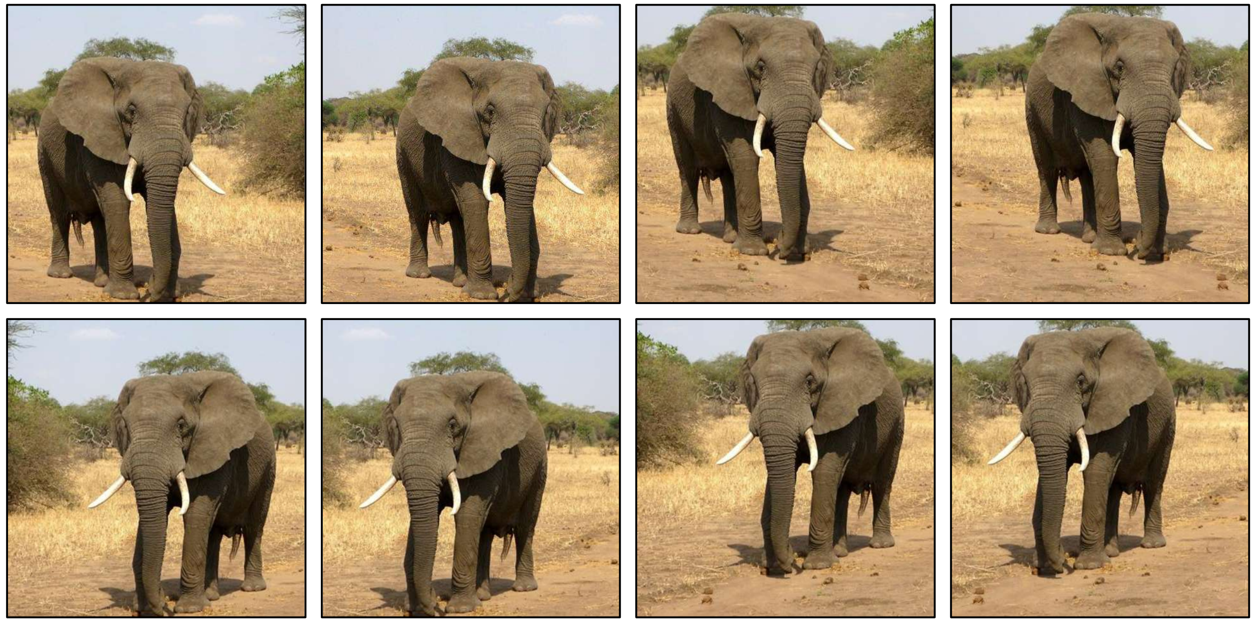
- Bootstrap sampling: create “new” datasets by resampling existing data, with or without replacement
- Common in perceptron and neural network training (“mini-batch,” “batch size”), methods that involve stochastic gradient descent
- Each “batch” has a slightly different distribution of instances, forces model to use different features and not get stuck in local minima

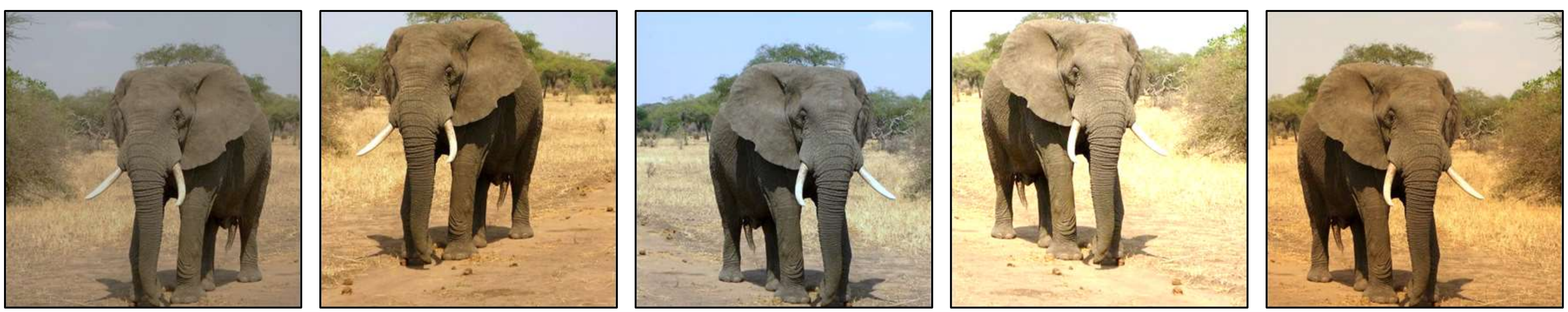
Data manipulation

- Another option: add a small amount of noise to each instance to create multiple variations:
  - Images: adjust brightness, flip left-right, shift image up / down / left / right, resize, rotate
  - Audio: adjust volume, shift in time, adjust frequencies
  - Text: synonym substitution
- These perturbations should not change the instance’s label
- Generally, they should be the same kind of variations you expect in real-world data

Common image manipulations

Original





Data synthesis

- Create artificial data using another machine learning method:
  - Train a probability distribution on labelled data
  - Sample the probability distribution to produce new instances
- Generative adversarial network: neural network trained to create samples from a distribution
- Exploit algorithms designed for other tasks, e.g.:
  - Computer-generated images
  - Automatic translation


Grand Theft Auto V, Rockstar Games

Data augmentation

- Advantages:
  - More data nearly always improves learning
  - Most learning algorithms have some robustness to noise (e.g., from machine-translation errors)
- Disadvantages
  - Biased training data
  - May introduce features that don't exist in the real world
  - May propagate errors
  - Increases problems with interpretability and transparency

Summary

- Labelled data is a major bottleneck for machine learning
- There are various strategies for making use of unlabelled data, or making more effective use of the data and labelling resources we already have
- These strategies generally improve performance, but sometimes with a trade-off in terms of error propagation, bias, and interpretability