On Practice of Machine Learning

Tricks or Principles ?

Summary

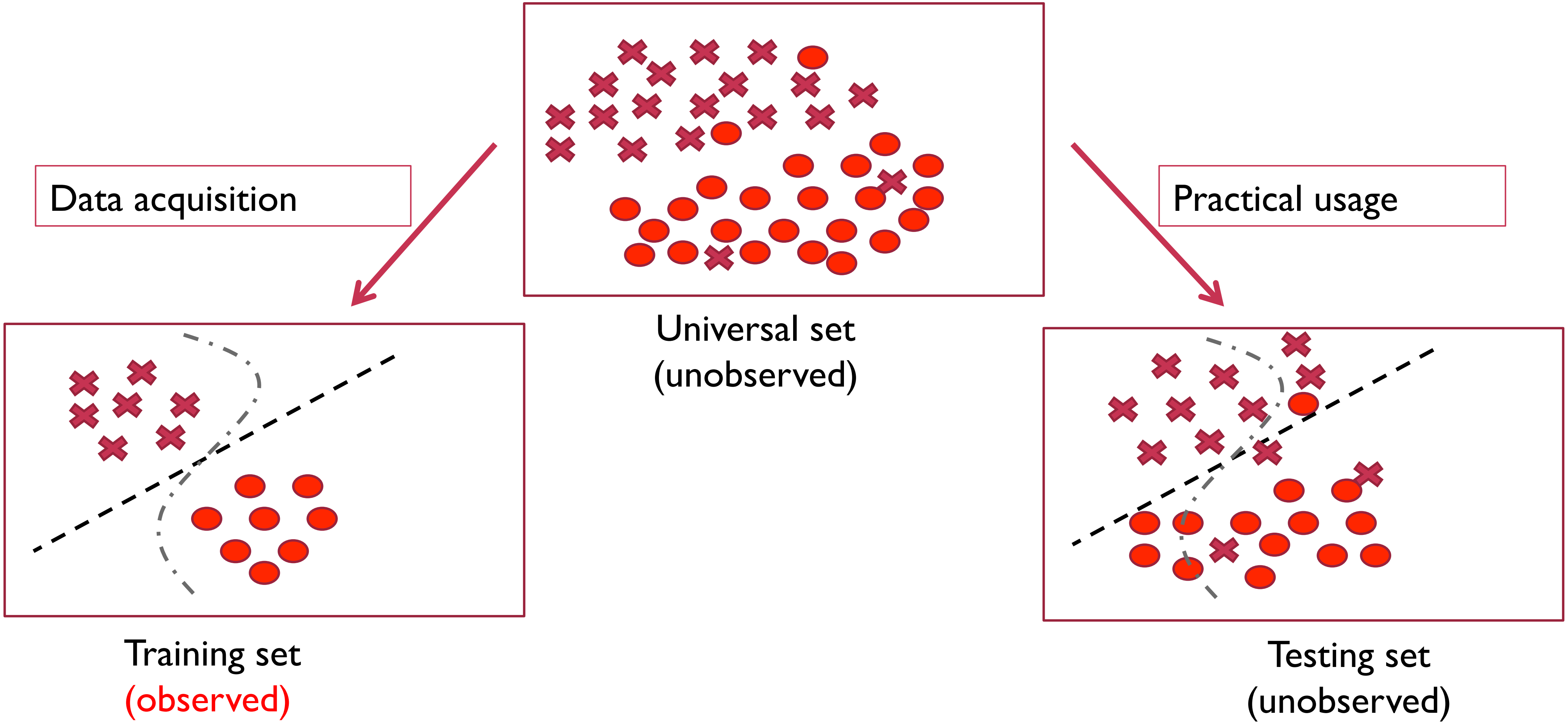
Problem

- Given X, Find Y
- Find F() such that for all examples $F(X_i)$ is as close as possible to Y_i
- F() is a learnable function with parameters W (many weights/coefficients).

Two Stages

- Training:
  - Given labeled examples { (X_i, Y_i) }, Find $F_w()$
  - Concerns:
    - Choose the nature of function F()
    - Find the best/appropriate W (i.e., Optimization)
- Testing/Inference
  - Given a new sample X, Predict Y

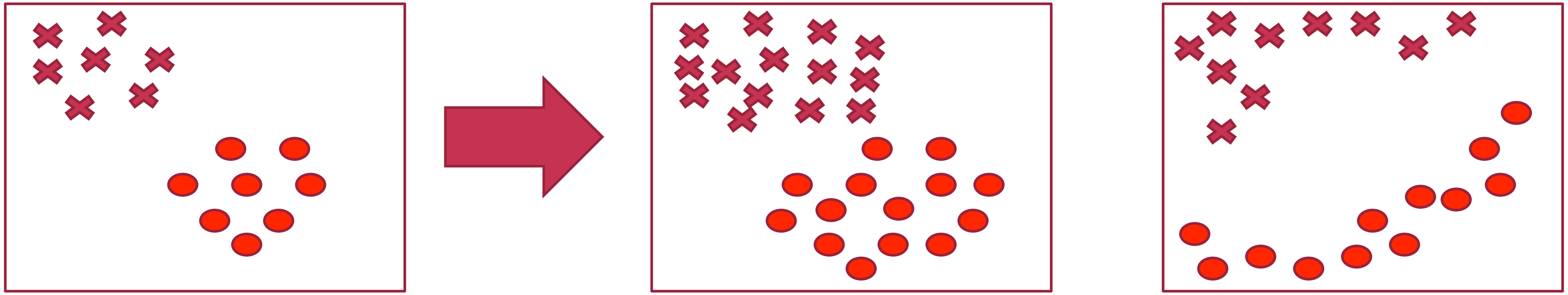
Training and testing



Training and testing

- Training is the process of making the system able to learn.

- Assumptions:
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias



Set of Concerns

- How to identify whether a problem is good for ML or not.
- Training:
  - How to obtain reliable supervision?
  - How to optimize my business performance measures?
- Testing/Inference
  - Fit the solution into Memory/Computational constraints.
- How to learn continously with user feedbacks, access patterns, availability of more data etc.?

Spectrum of Problems: F: X->Y

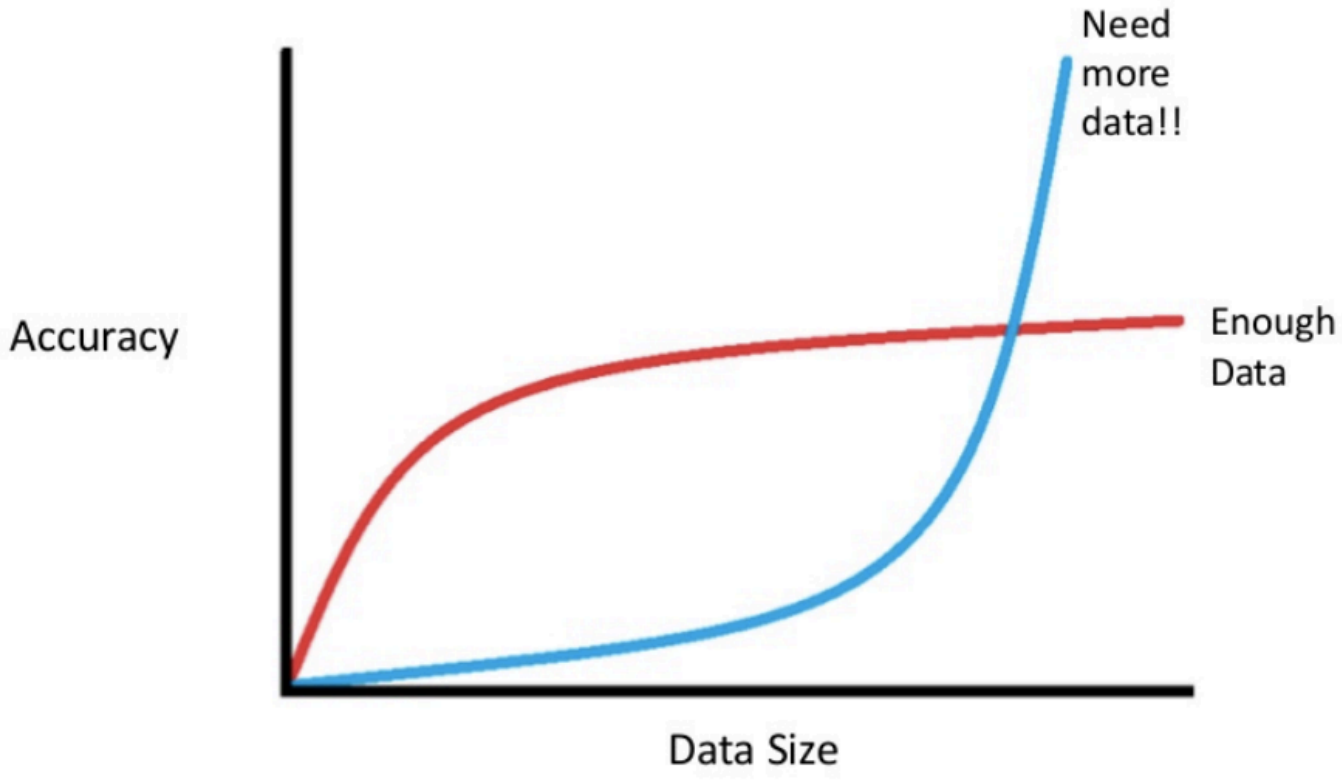
- X come from a small set (eg. Dictionaries)
- F(X) can be defined with some simple rules
- Popular ML problem space
- Y is some what independent of X
  - X is your name, gender, height. Y is your wealth.
- F(X) is not smooth $F(X+\Delta x)$ far from $Y+\Delta y$
  - Small change in X is taking predictions taking too far

Type of Data and Concerns

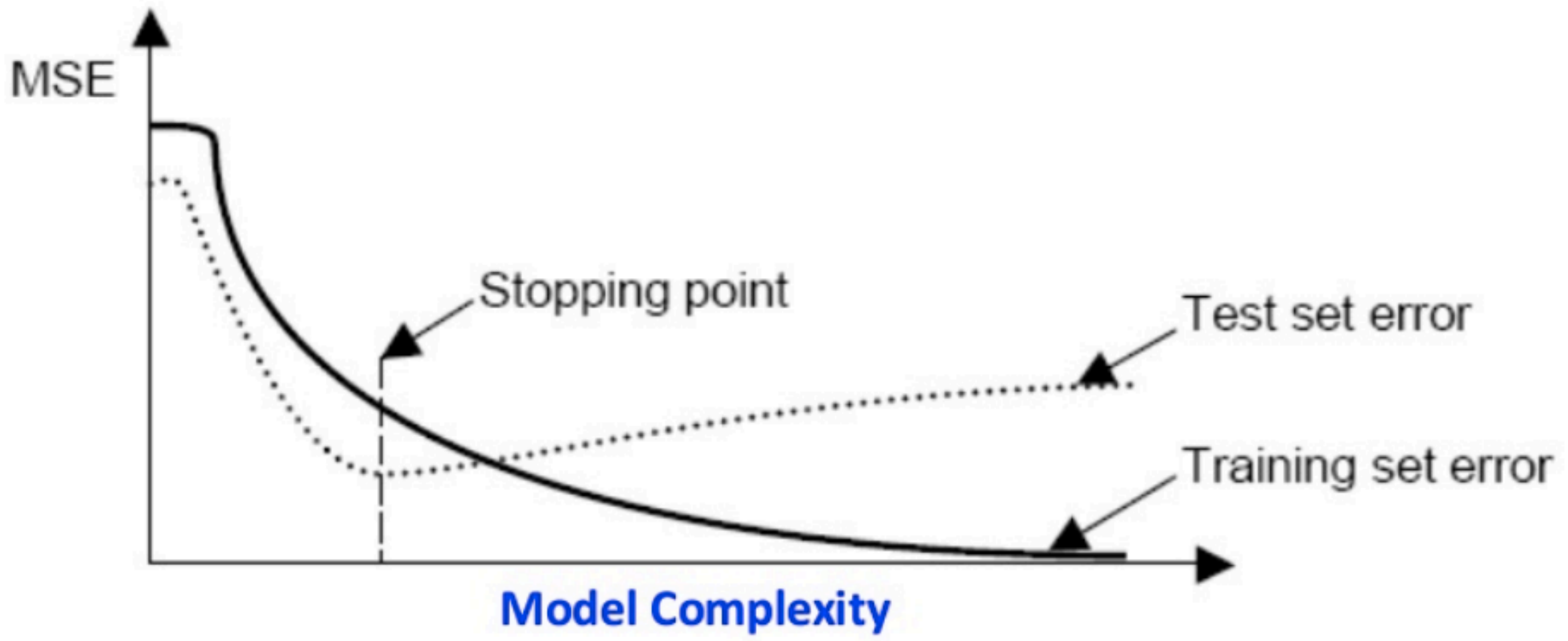
- Size
  - Small Vs Large data
- Sparsity
  - Sparse Vs Dense
- Balance
  - Balance Vs imbalance
- Quality
  - Noise, missing values etc.

- Simple model or complex models
- Dense is easier?
- Special treatment for minority.
- Reduce sensitivity. Remove noise.

Do we have enough data?



Do we have enough parameters?



How do I “push” the accuracy?

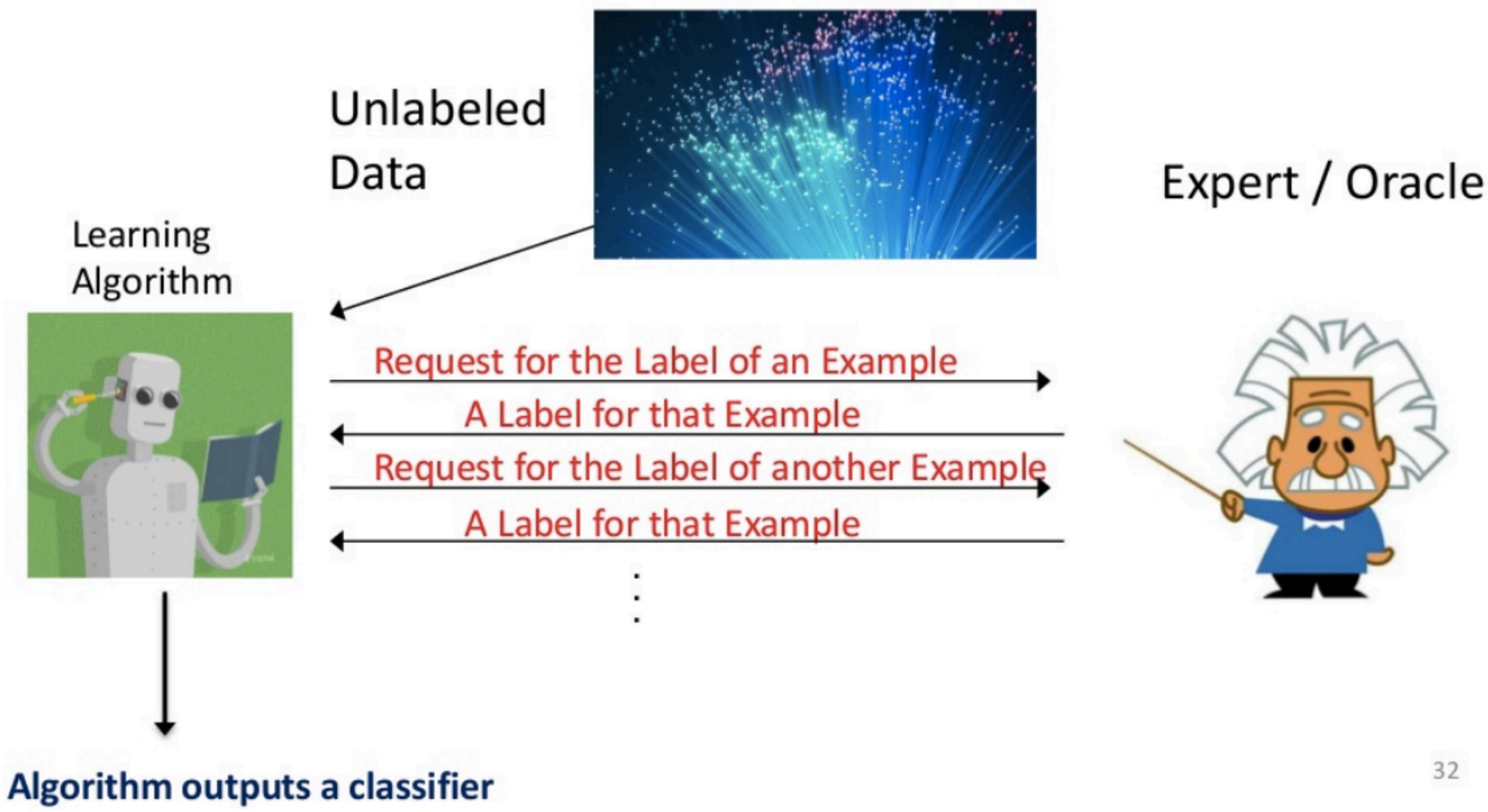
- Combine a variety of models
- If the models are diverse, and performances are similar,
  - High chance that the ensemble will do better

Supervision

Challenges with Supervision

- I have too much data. But most of them are unlabelled. What do we do?
- I have labeled data. But a good percentage of the labels are erroneous. What do I do?
- I have labellings from experts itself. But they do not agree. What do we do?
- My supervisors are too costly. How do I do minimize the cost of supervision?
- …

Learn with minimal # of examples ?



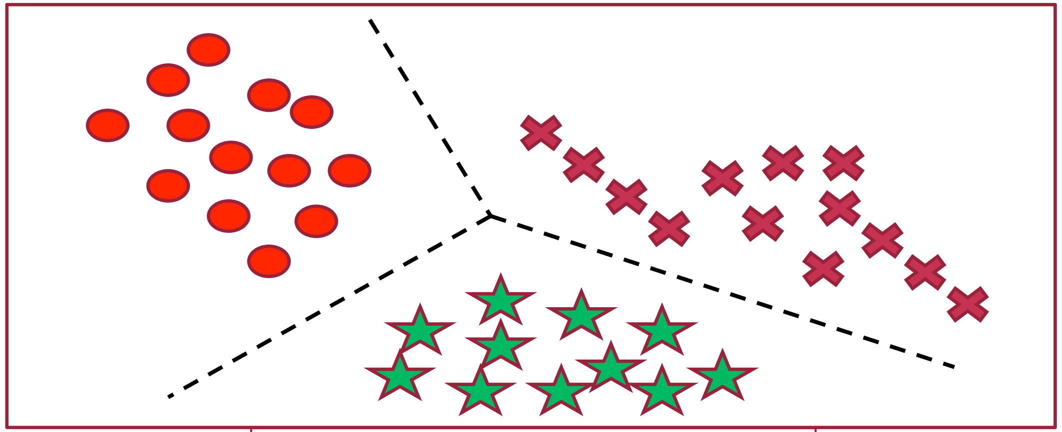
Active Learning

Eg. Learn the notion of a rectangle.

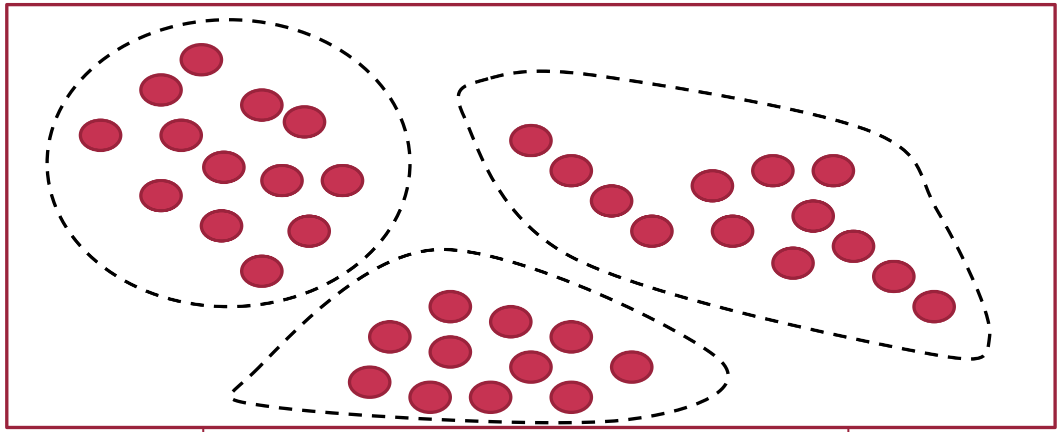
Semi Supervised Learning

- I have a small quantity of labelled data and large quantity of unlabeled data.
  - How do I take advantage of the unlabeled data?

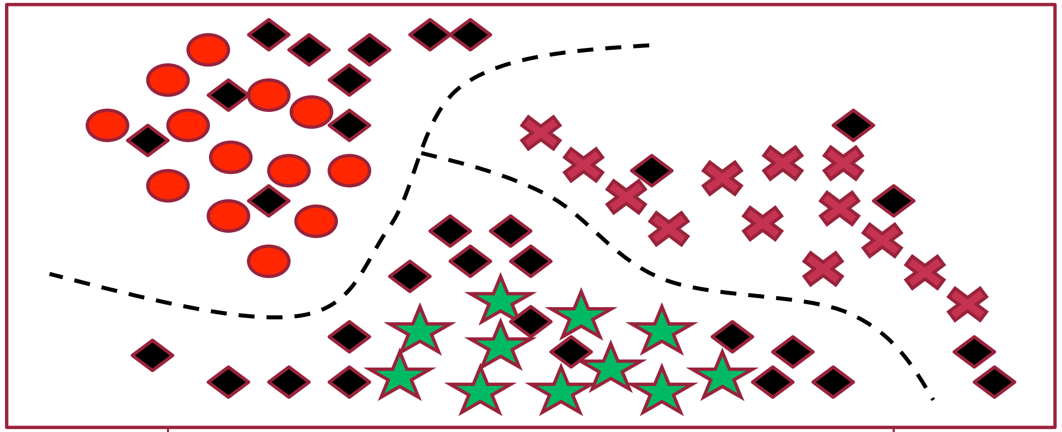
Algorithms



Supervised learning



Unsupervised learning



Semi-supervised learning

Self Training: Naïve

- Train a supervised learner on available labelled data (X_l, Y_l).
- Label all points in unlabelled data X_u.
- Retrain the classifier using the new labels for documents where the classifier is most confident.
- Coninue until labels do not change any more.

Self Training: Refined

- **Assumption:** One's own high confidence predictions are correct.
- **Self-Training Algorithm**
  - Train on labeled examples
  - Predict on unlabeled examples
  - Add $(x, f(x))$ to the labeled data
    - Add all
    - Add a few most confident pairs
    - Add weight for each pairs
  - Repeat the process

Co-Training

- Co-training assumed two “views” of the data where each input x is a pair

$$x = (x_1, x_2)$$

- Eg. In the context of web page classification, x_1 may be metadata associated with the web page such as tite etc. x_2 be the words in the link pointing to this page.
- Assume there exists fuctions c_1, c_2 and c such that

$$c_1(x_1) = c_2(x_2) = c(x)$$

- Two sets of features x_1 and x_2 are conditionally independent given the class.

1998 paper demonstrates, with 12 labeled examples, 788 web pages could be classified with 95% accuracy.

Co-Training

1. Use the labeled data to learn the initial h_1, h_2
2. First use h_1 to label examples that it is confident about and then feed these to our learner to update h_2
3. Then use h_2 to label examples that it is confident about and then feed these to our learner to update h_1
4. Keep repeating this process

Summary: Questions?

- Varying amount and quality of supervision
  - Many wrapper style methods.
  - Intuitive
- Many principled formulations
  - Formal extensions of existing methods
    - (eg. Transductive SVMs; Semi Supervised Random Forest)
  - Many newer learning problems
    - (eg. Multiple Instance Learning,)

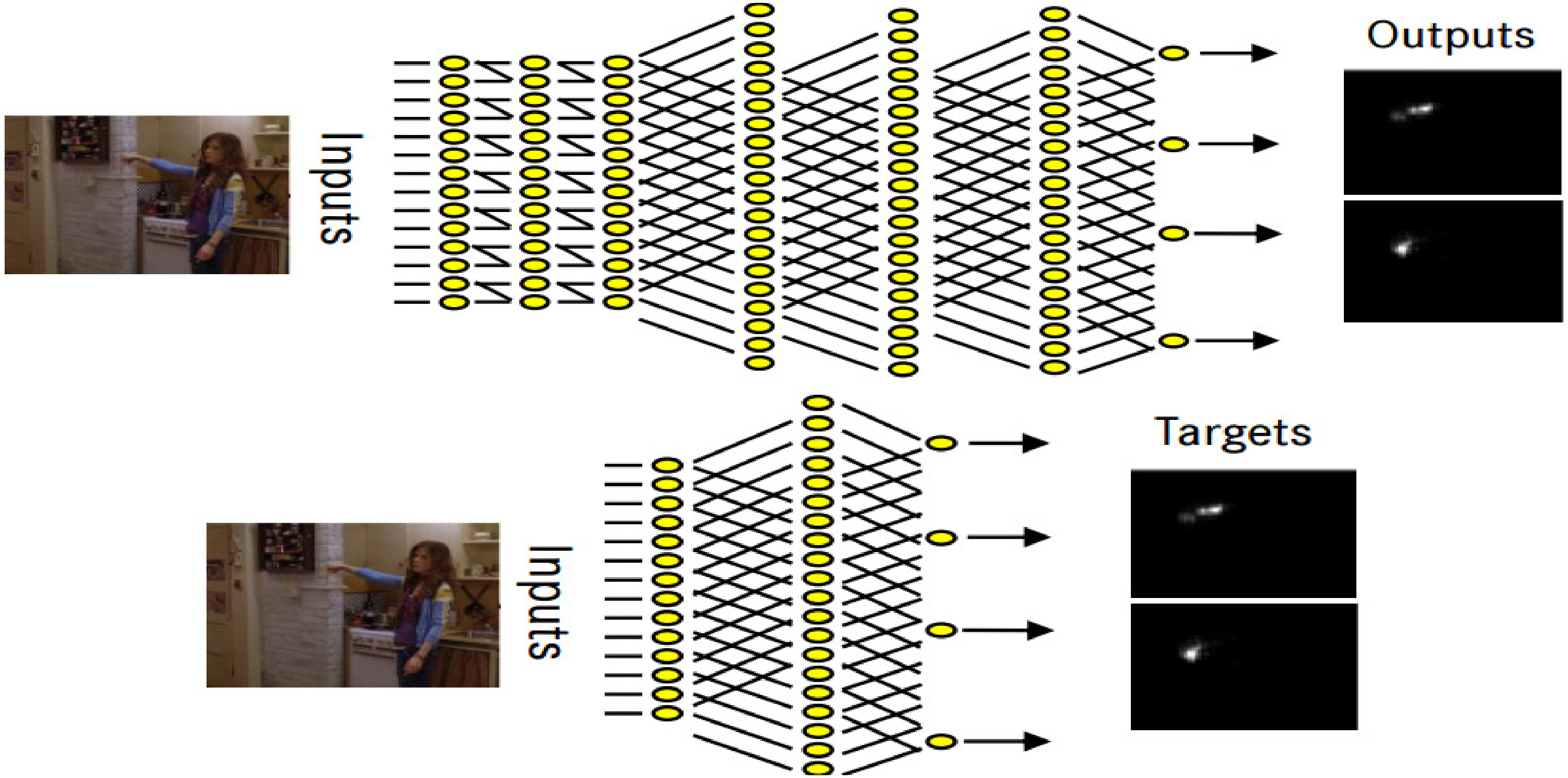
Compression of DL Models

- At Test time.
- Why?
- **Popular:**
  - Pruning
  - Quantization
  - Architectural Modifications

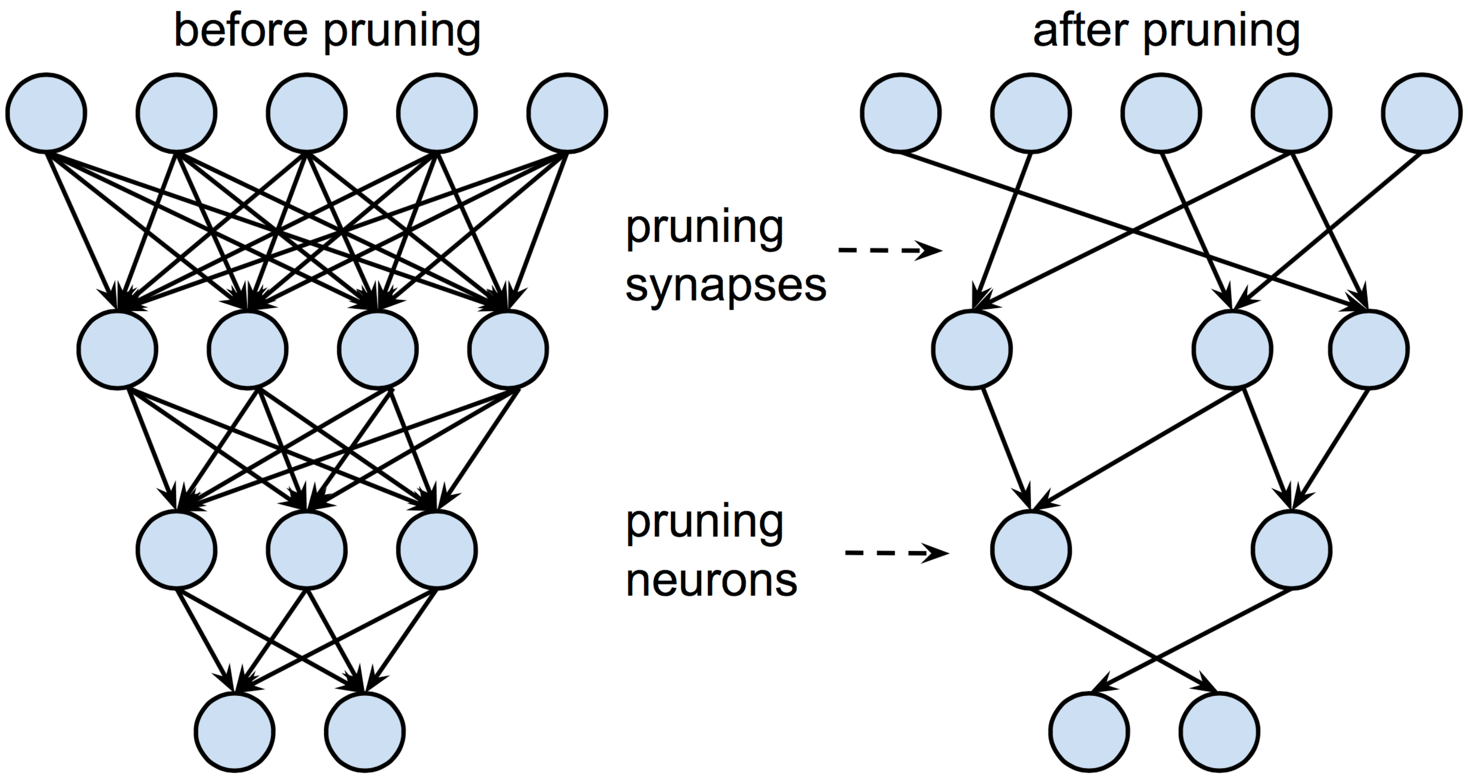
Quantization

- What do we store/use at test time.
  - Primarily weights (interconnect of neurons)
- Convert weights (say double floats) to
  - Integers
  - Characters (say 8 bit)
  - 0/1 (or -1, 1)
- Objective: Round such that accuracies will not decrease much.

Student-Teacher Network



Iterative Pruning + Retraining



Han, Song, et al. "Learning both weights and connections for efficient neural network." NIPS. 2015.

Iterative Pruning + Retraining

- 1. Choose a neural network architecture.
- 2. Train the network until a reasonable solution is obtained.
- 3. Prune the weights of which magnitudes are less than a threshold τ.
- 4. Train the network until a reasonable solution is obtained.
- 5. Iterate to step 3.

Han, Song, et al. "Learning both weights and connections for efficient neural network." NIPS. 2015.

Summary: Questions?

- Problem and Setting:
  - How hard the problem?
  - Amount of data, parameters, ?
- Training
  - Availability and Reliability of Supervision
- Testing
  - Memory, FLOPS, etc.
  - Compress DL models

Thanks. Questions?