COMP 551 – Applied Machine Learning
Lecture 18: Semi-supervised learning

Instructor: Joelle Pineau (*jpineau@cs.mcgill.ca)*

Class web page: *www.cs.mcgill.ca/~jpineau/comp551*

Basic idea

- Traditional classifiers learn only from labeled data.
- Label data can be expensive / difficult to collect.
  - Human annotation is slow, boring!
  - Labels can require experts, or special devices to acquire.
- We prefer to get better performance for free: Unlabeled data!
- Goal of semi-supervised learning is to exploit both labeled and unlabeled examples.
- Most of today will be on semi-supervised classification; brief discussion of semi-supervised regression and semi-supervised clustering.

Example of hard-to-get labels

Task: speech analysis

- Switchboard dataset
- telephone conversation transcription
- 400 hours annotation time for each hour of speech

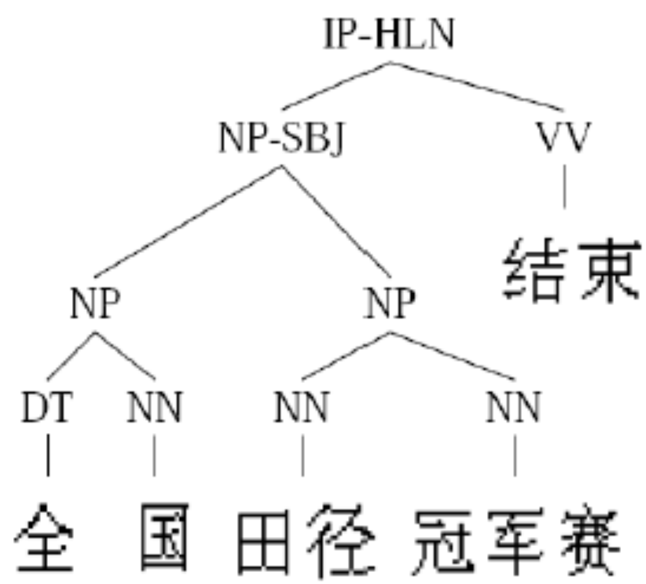
film $\Rightarrow$ `f ih_n uh_gl_n m`

be all $\Rightarrow$ `bcl b iy iy_tr ao_tr ao l_dl`

Example of hard-to-get labels

Task: natural language parsing

- Penn Chinese Treebank
- 2 years for 4000 sentences

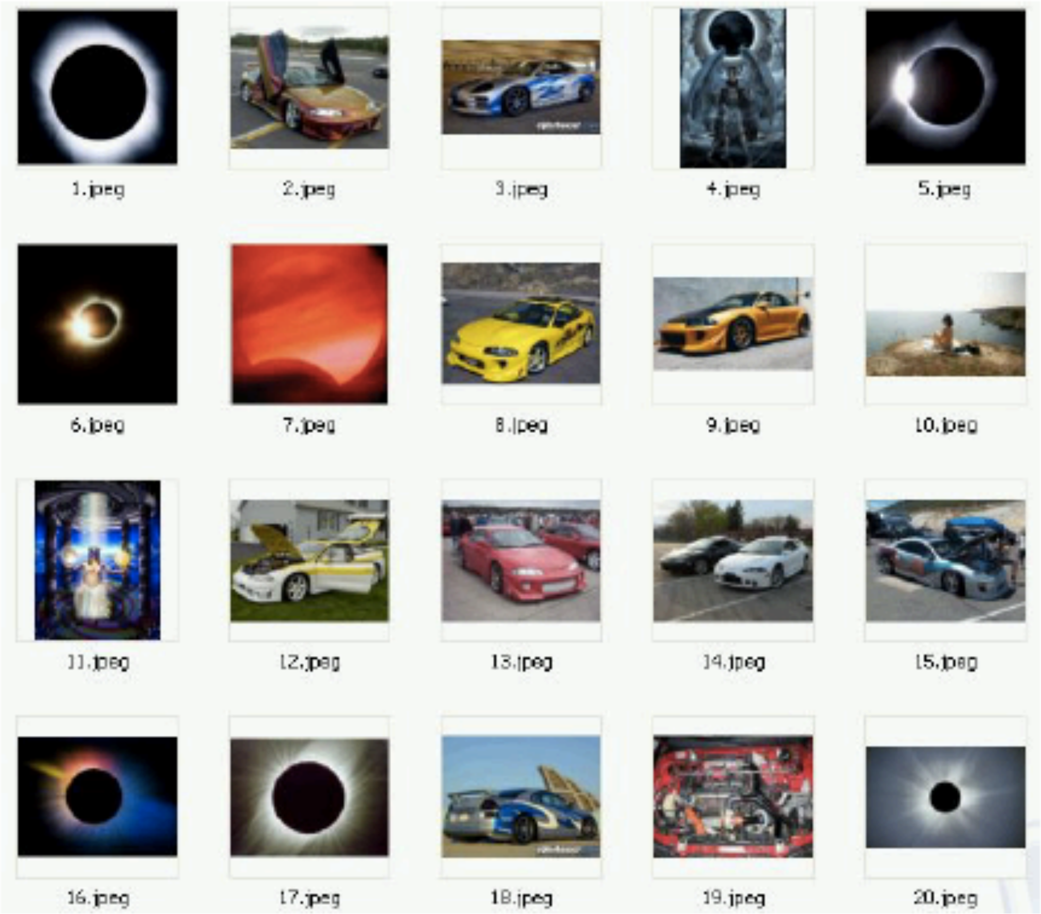


"The National Track and Field Championship has finished."

Example of not-so-hard-to-get labels

For some tasks, it may not be too difficult to label 1000+ instances.

Task: image categorization of "eclipse"



Example of not-so-hard-to-get labels

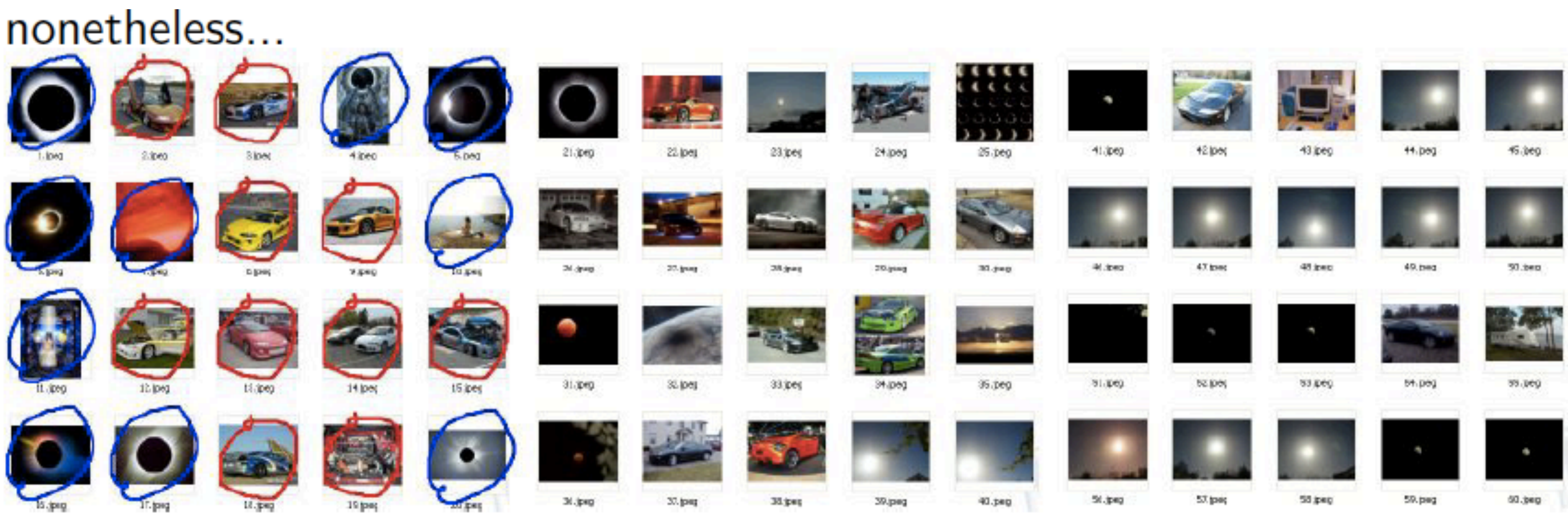
For some tasks, it may not be too difficult to label 1000+ instances.

Task: image categorization of "eclipse"

There are ways like the EPS game (*www.epsgame.org*) to encourage "human computation" for more labels.

Example of not-so-hard-to-get labels

For some tasks, it may not be too difficult to label 1000+ instances.



Goal: Use both labeled and unlabeled data to build better learners, than using each one alone.

Notation

- Given:
  - Labeled data: $(X_l, Y_l) = \{ x_{1:l}, y_{1:l} \}$ available during training
  - Unlabeled data: $X_u = \{ x_{l+1:n} \}$ available during training
  - Test data: $X_{test} = \{ x_{n+1:N} \}$ NOT available during training
- Usually $l<<n$, so much more unlabeled data than labeled data.

Notation

supervised learning (classification, regression) $\{(x_{1:n}, y_{1:n})\}$

$\updownarrow$

semi-supervised classification/regression $\{(x_{1:l}, y_{1:l}), x_{l+1:n}, x_{test}\}$

transductive classification/regression $\{(x_{1:l}, y_{1:l}), x_{l+1:n}\}$

$\updownarrow$

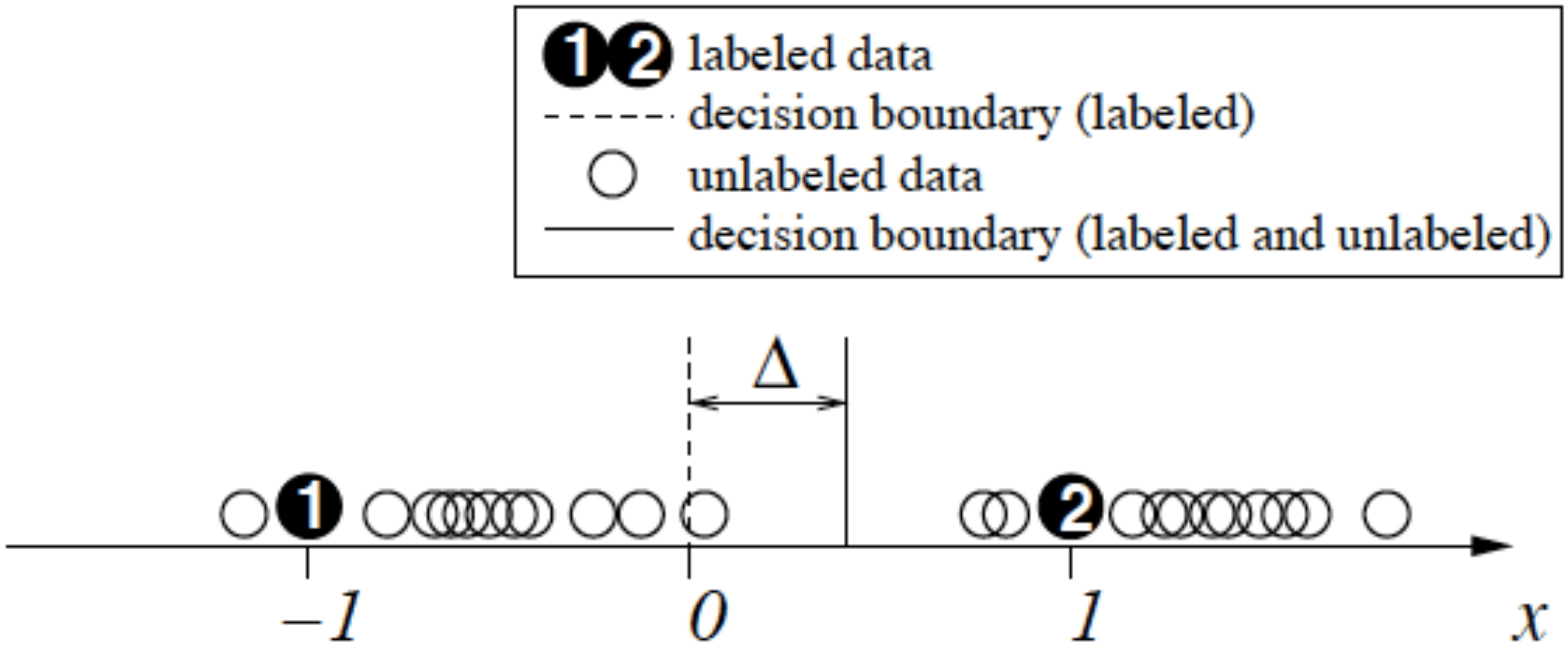
semi-supervised clustering $\{x_{1:n}, \text{must-, cannot-links}\}$

$\updownarrow$

unsupervised learning (clustering) $\{x_{1:n}\}$

How can unlabeled data help?

- Assuming each class is a coherent group (e.g. Gaussian)
- With vs without unlabeled data: Decision boundary shifts.



Self-training algorithm

- Assume: One's own high confidence predictions are correct.
- Basic algorithm:
  - Train f from (X_l, Y_l).
  - Predict for $x \in X_u$.
  - Add $(x, f(x))$ to labeled data.
  - Repeat.

Self-training algorithm

- Assume: One's own high confidence predictions are correct.
- Basic algorithm:
  - Train f from (X_l, Y_l).
  - Predict for $x \in X_u$.
  - Add $(x, f(x))$ to labeled data.
  - Repeat.
- Variations:
  - Add a few most confident $(x, f(x))$ to labeled data.
  - Add all $(x, f(x))$ to labeled data.
  - Add all $(x, f(x))$ to labeled data, weigh each by confidence.

Self-training example: image categorization

- Train a Naïve Bayes classifier on two initial labeled images:

0.00

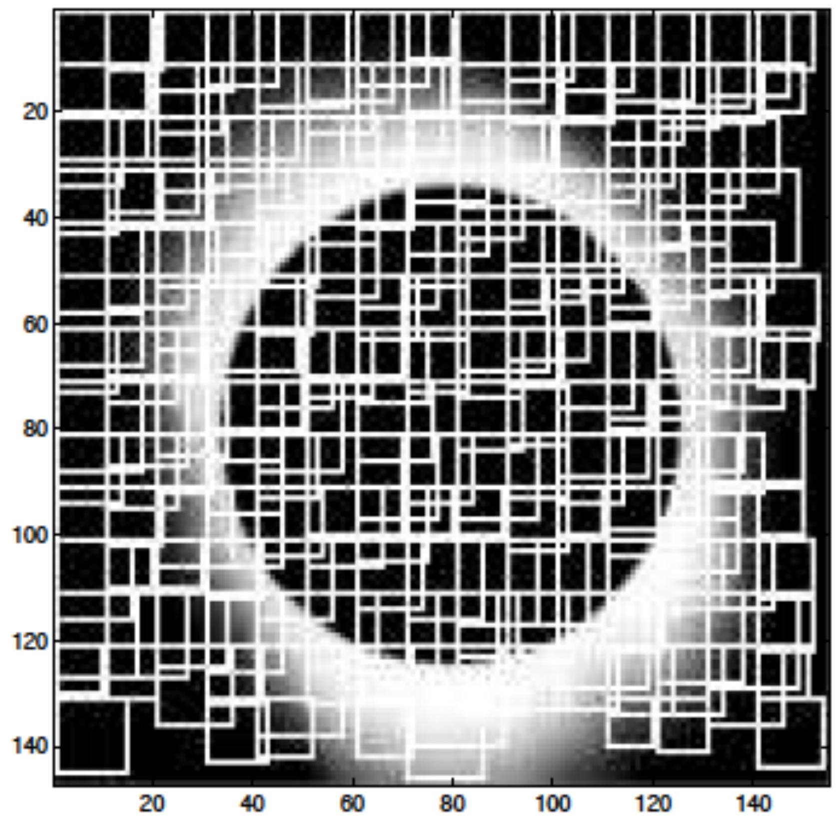
1.jpeg

-Inf

14.jpeg

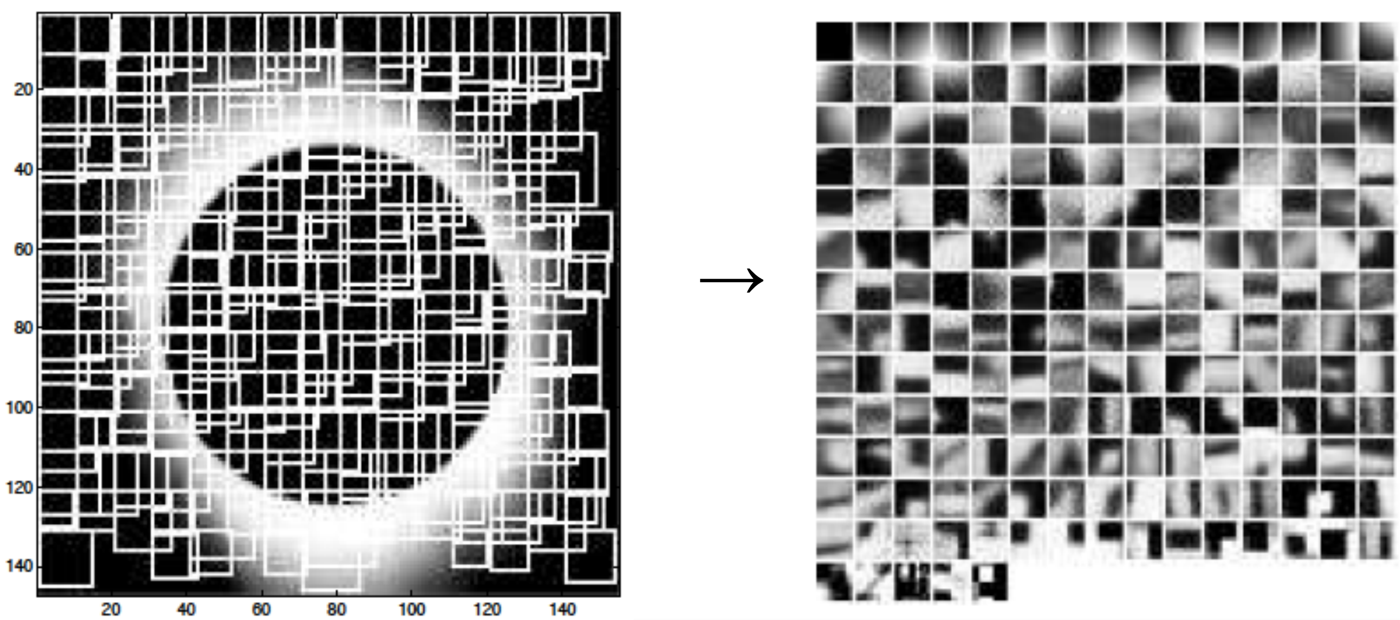
Self-training example: image categorization

- Each image is divided into small patches.
- 10x10 grid, random size of 10 ~ 20



Self-training example: image categorization

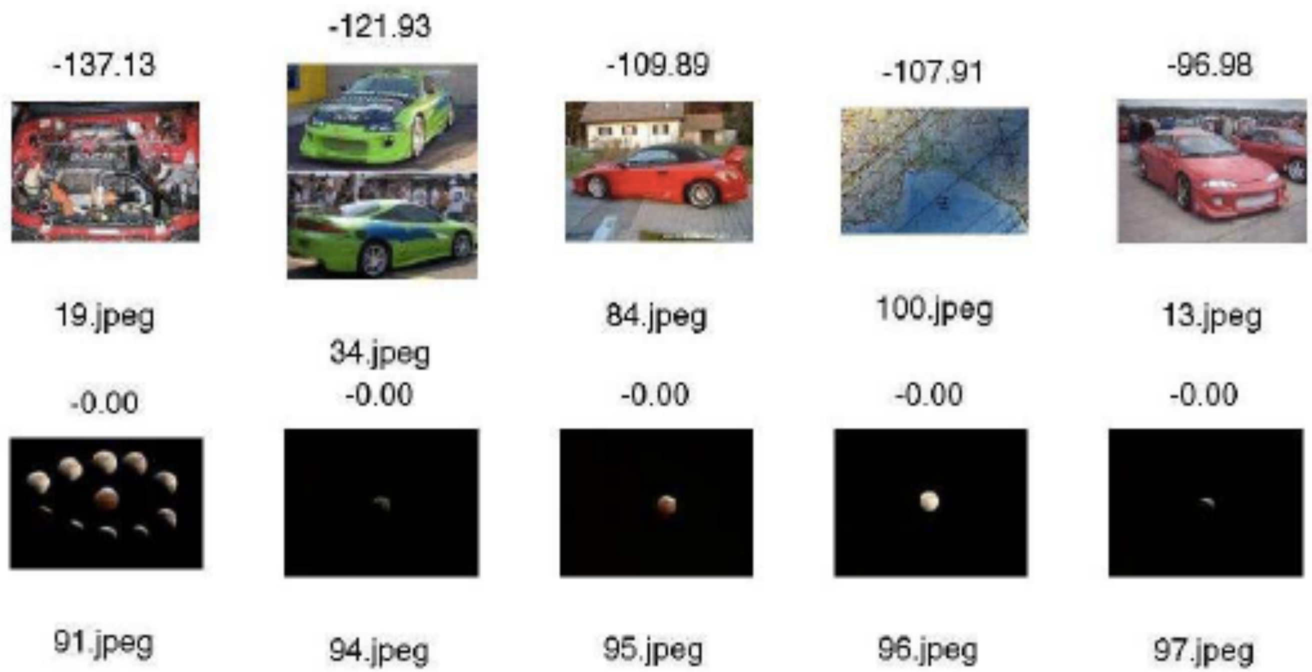
- All patches are normalized.
- Define a dictionary of 200 “visual words” (cluster centroids) with 200-means clustering on all patches.
- Represent a patch by the index of its closest visual word.



Self-training example: image categorization

- Train a Naïve Bayes classifier on two initial labeled images:

- Classify unlabeled data, sort by confidence *log Pr(y=astronomy | x)*.



Advantages of self-training

- The simplest semi-supervised learning method.
- A wrapper method, applies to existing (complex) classifiers.
- Often used in real tasks like natural language processing.

Disadvantages of self-training?

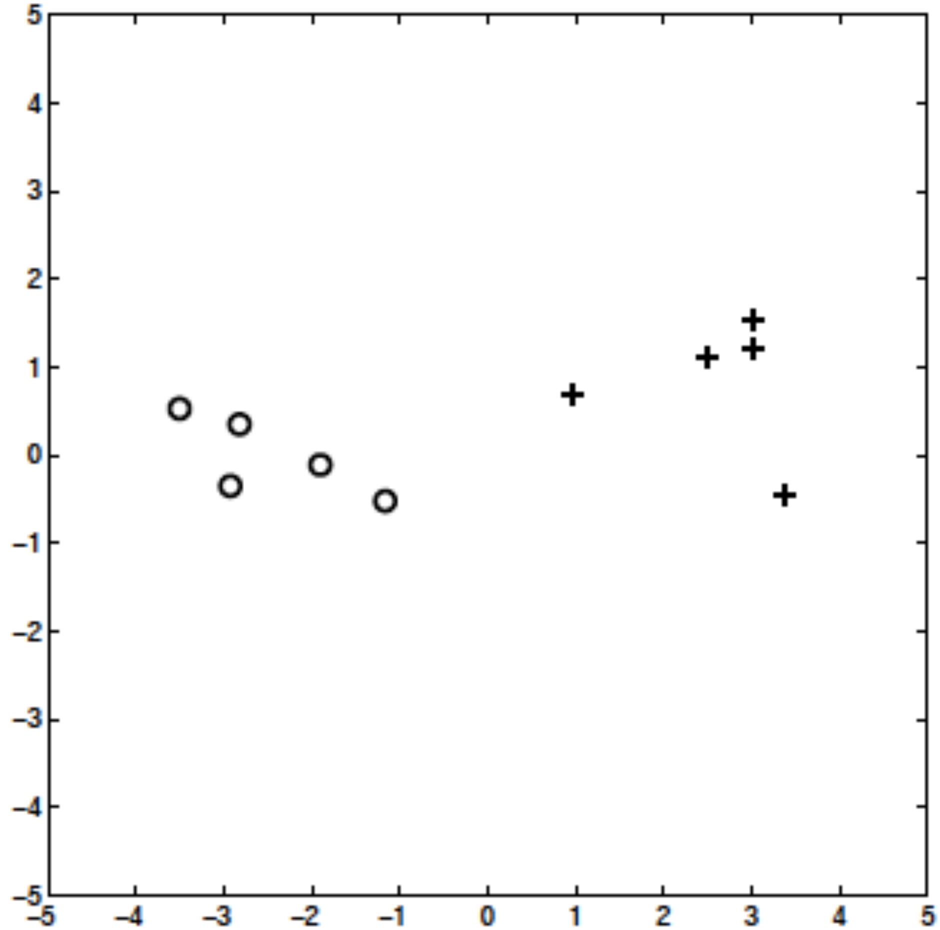
Disadvantages of self-training?

- Early mistakes could reinforce themselves.
  - Heuristic solutions, e.g. “un-label” an instance if its confidence falls below a threshold.
- Cannot say too much in terms of convergence.
  - But there are special cases when self-training is equivalent to the Expectation-Maximization (EM) algorithm.
  - There are also special cases (e.g. linear functions) when the closed-form solution is known.

Alternatives?

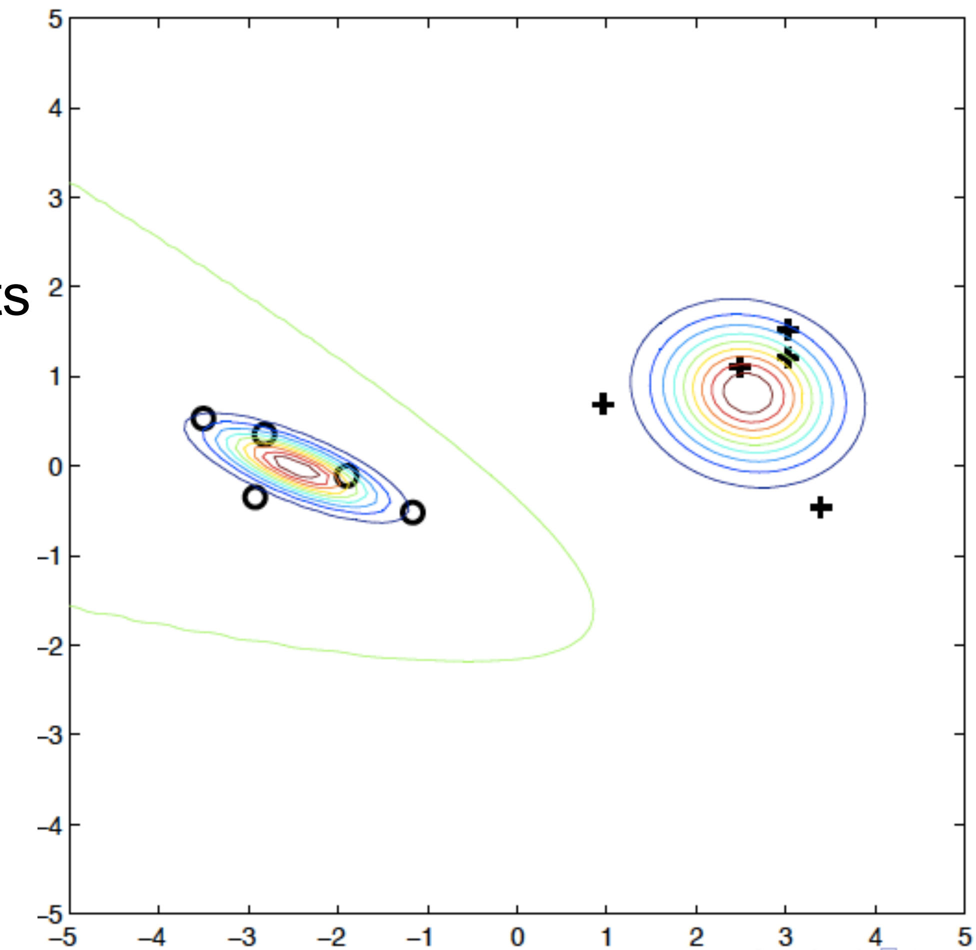
The generative approach

- Given labeled data, assume each class has a Gaussian distribution.



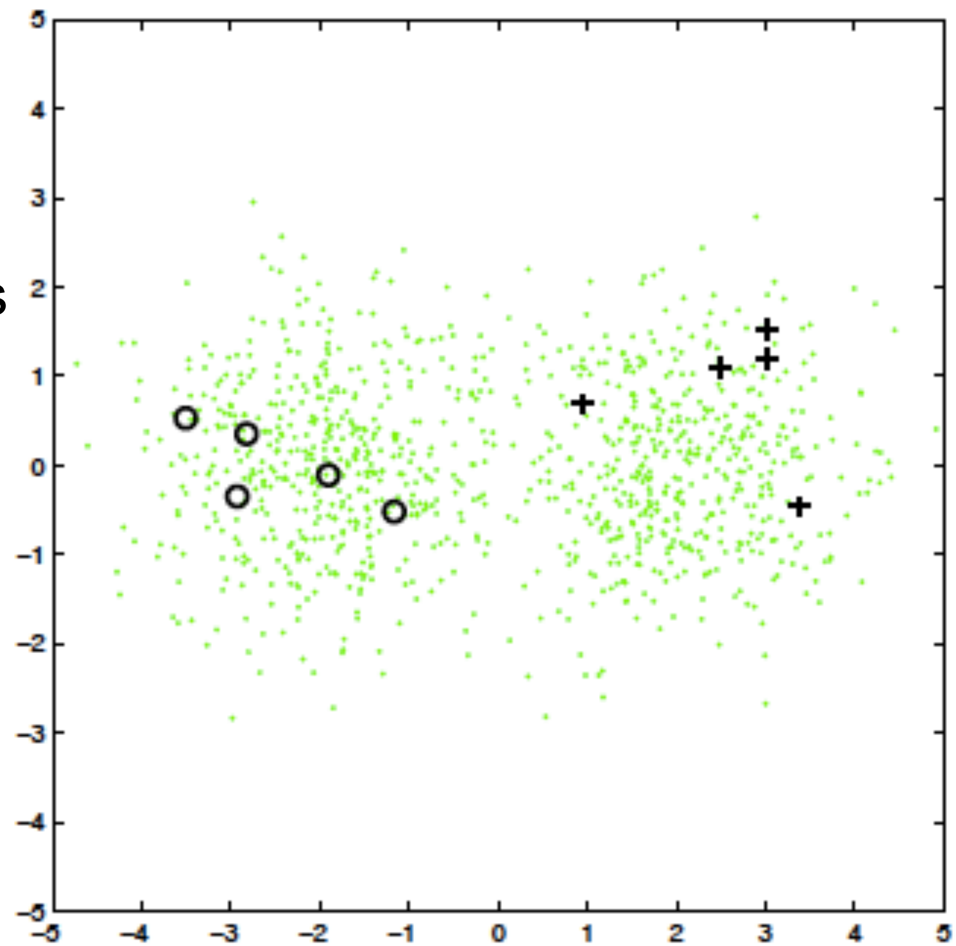
The generative approach

- Given labeled data, assume each class has a Gaussian distribution.
- The most likely model and its decision boundary:



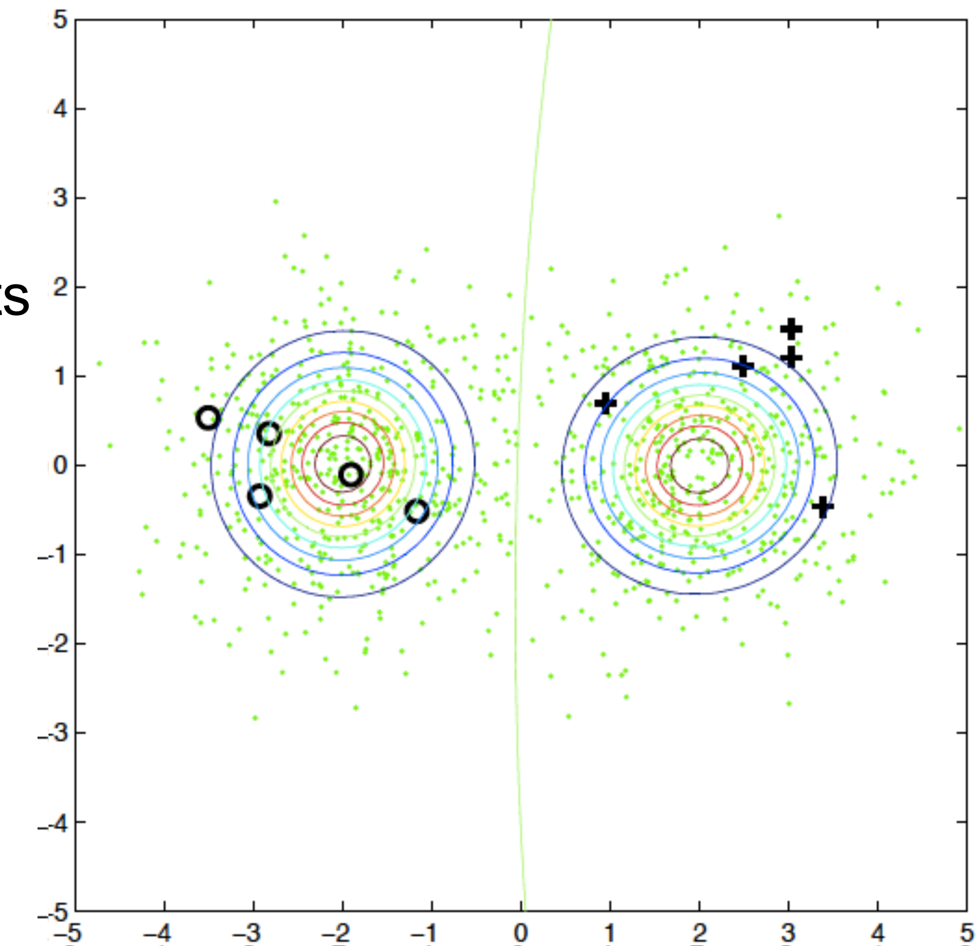
The generative approach

- Given labeled data, assume each class has a Gaussian distribution.
- The most likely model and its decision boundary.
- Add unlabeled data:



The generative approach

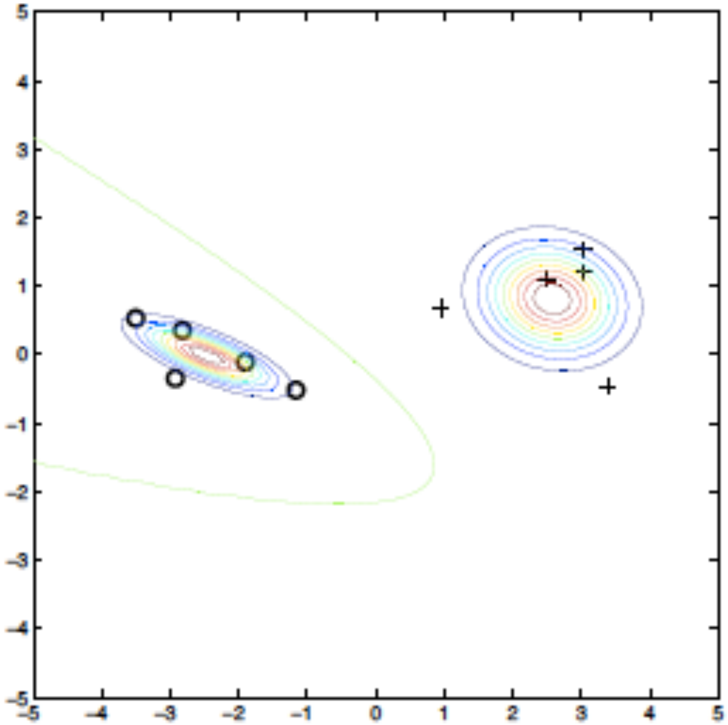
- Given labeled data, assume each class has a Gaussian distribution.
- The most likely model and its decision boundary:
- Add unlabeled data.
- The most likely model and decision boundary change.



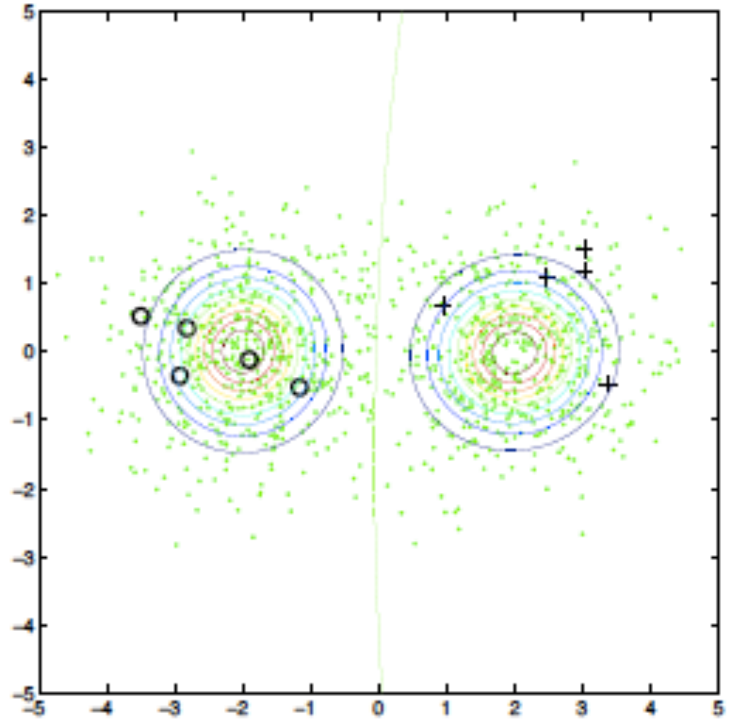
The generative approach

- Decision boundaries are different because they maximize different quantities.

$$p(X_l, Y_l|\theta)$$



$$p(X_l, Y_l, X_u|\theta)$$



Revisiting the EM algorithm

- **Setup**:
  - Observed data: $D = (X_l, Y_l, X_u)$
  - Hidden data: $H = Y_u$
  - $P(D|\theta) = \sum_H p(D, H \mid \theta)$
- **Goal**: Find θ to maximize $p(D|\theta)$

Revisiting the EM algorithm

- **Setup**:
  - Observed data: $D = (X_l, Y_l, X_u)$
  - Hidden data: $H = Y_u$
  - $P(D|\theta) = \sum_H p(D, H \mid \theta)$
- **Goal**: Find θ to maximize $p(D|\theta)$
- **Algorithm**:
  - Start with some arbitrary θ_0.
  - **E-step**: Estimate $p(H|D,\theta)$
  - **M-step**: Find $argmax_\theta \sum_H p(D,H|\theta)$
- **Comments**: EM iteratively improves $p(D|\theta)$. Converges to a local minima of θ. K-means is a special case of this.

Comments on the generative approach

- This offers a clear, well-studied, probabilistic framework.
- Can be very effective if the model is close to correct.

Comments on the generative approach

- This offers a clear, well-studied, probabilistic framework.
- Can be very effective if the model is close to correct.
- Often difficult to verify the correctness of the model. Unlabeled data can hurt the solution if the generative model is wrong.

- EM converges to a local optima.
- There are other ways than EM to find parameters, e.g. variational approximation.

Alternate method: Cluster-and-label

Instead of running EM with the probabilistic generative model using the labeled data:

- Run the clustering algorithm assuming all data is unlabeled.
- Label all points within a cluster by the majority of labeled points in that cluster.

Alternate method: Cluster-and-label

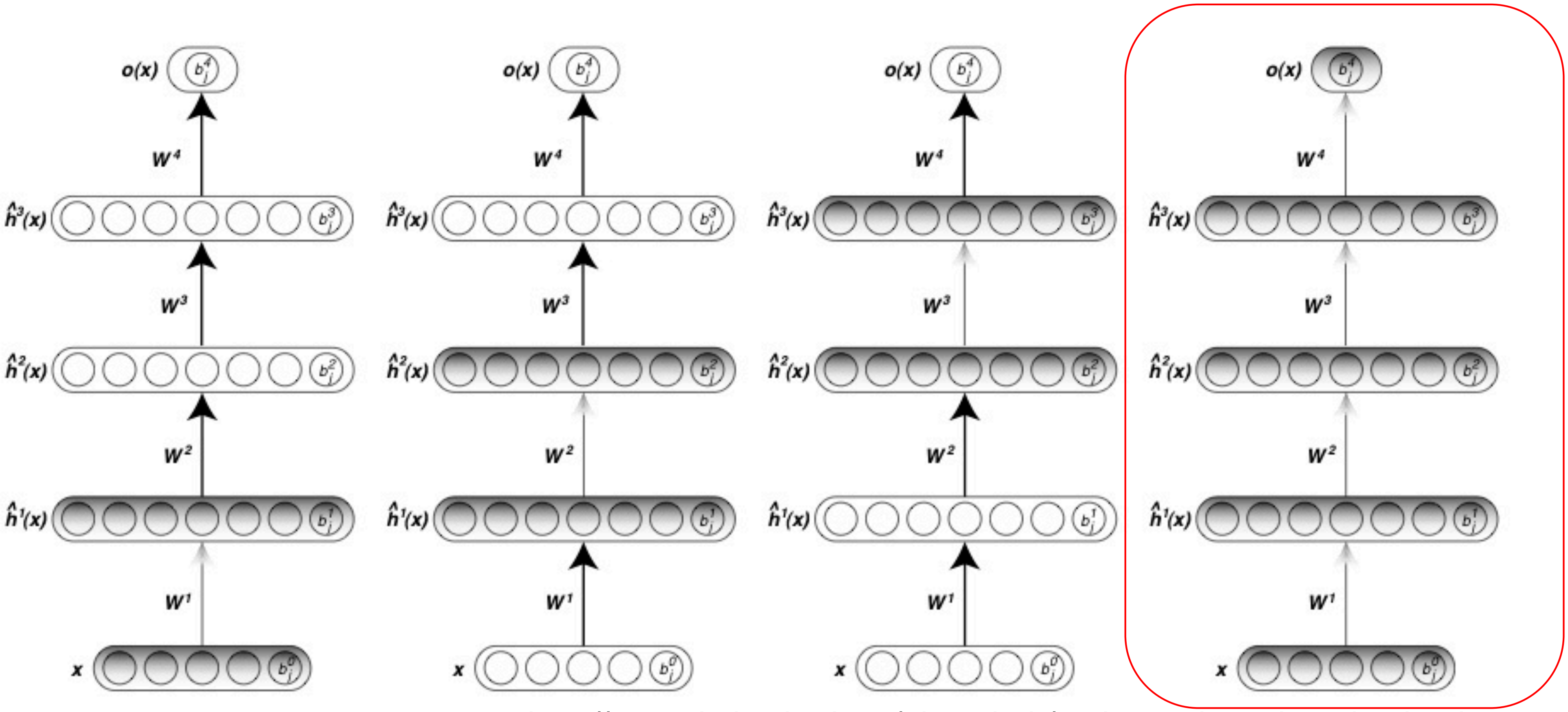
Instead of running EM with the probabilistic generative model using the labeled data:

- Run the clustering algorithm assuming all data is unlabeled.
- Label all points within a cluster by the majority of labeled points in that cluster.

- **Pro**: Another simple wrapper method.
- **Con**: Can be difficult to analyze; labels within a cluster may disagree.

Recall: Autoencoder + supervised layer

Train an autoencoder, then add a supervised layer and train the full network with backpropagation using error on the predicted output, $Err(W) = \sum_{i=1:n} L [y_i , o(x_i)]$

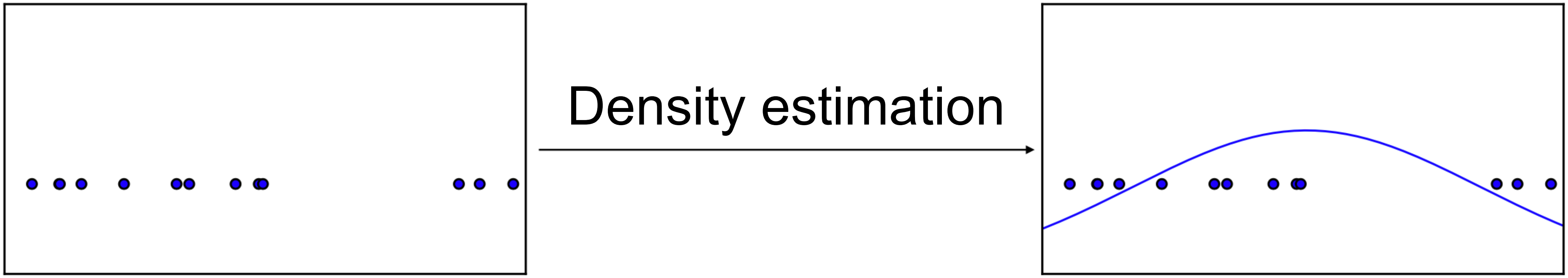


http://www.dmi.usherb.ca/~larocheh/projects_deep_learning.html

Many more methods!

- Co-training.
- Semi-supervised SVMs.
- Graph-based algorithms.
- Etc.

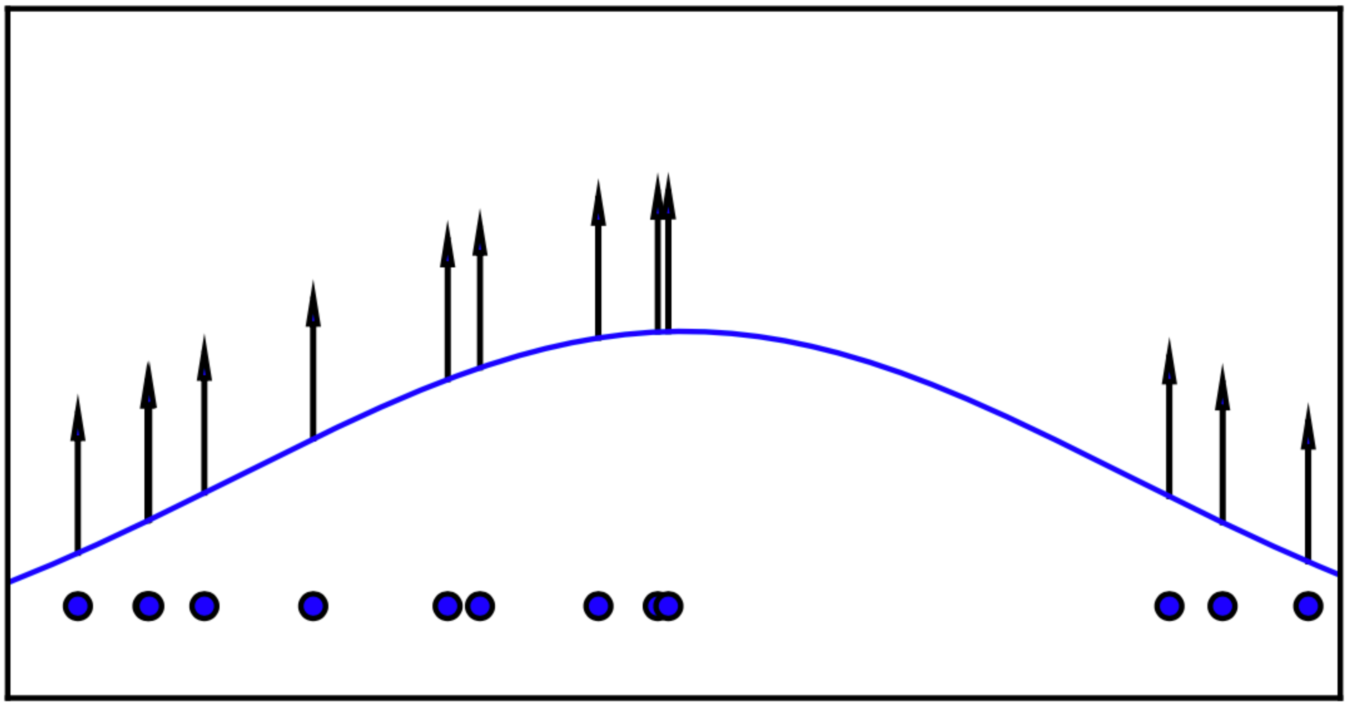
Generative Models



Training examples Model samples

Material from Ian Goodfellow, Montreal Summer School 2017

Maximum Likelihood Criteria



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

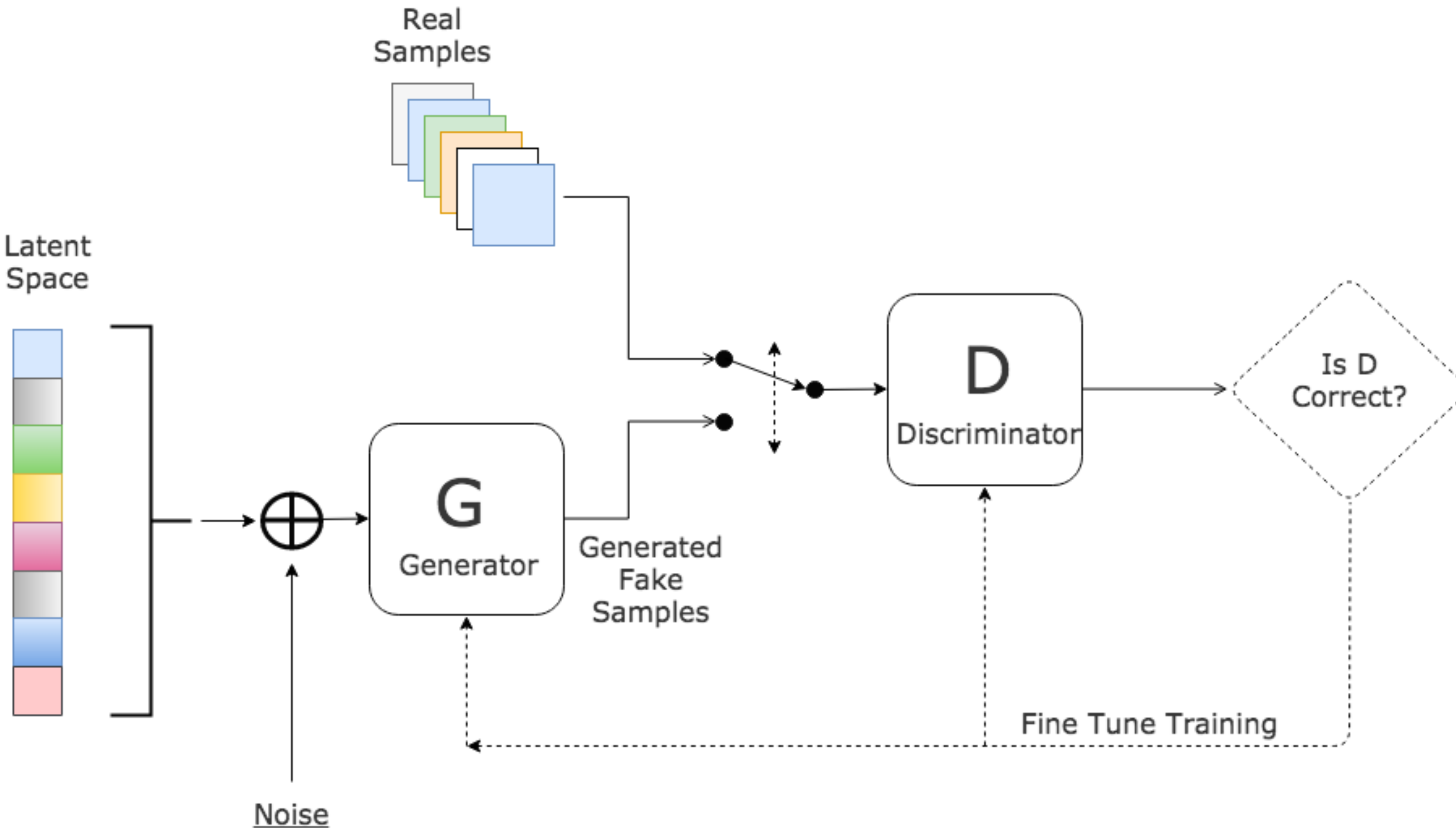
Material from Ian Goodfellow, Montreal Summer School 2017

What can you do with generative models?

- Semi-supervised learning
- Missing data
- Multiple correct answers
- Realistic generation tasks
- Simulation by prediction
- Simulated environments and training data
- Learn useful embeddings

Material from Ian Goodfellow, Montreal Summer School 2017

Generative Adversarial Nets



Picture from *https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html*

Generative Adversarial Nets

- Very realistic samples! *http://research.nvidia.com/publication/2017-10_Progressive-Growing-of*

- Also used to generate voice, natural language, robot behaviors, ...

Does unlabeled data always help?

- There's no free lunch! Semi-supervised learning typically makes strong model assumptions (to compensate for lack of labels).
- Performance can degrade by addition of unlabeled data when the modeling assumptions are not appropriate. This has been empirically observed by many researchers.

Does unlabeled data always help?

- There's no free lunch! Semi-supervised learning typically makes strong model assumptions (to compensate for lack of labels).
- Performance can degrade by addition of unlabeled data when the modeling assumptions are not appropriate. This has been empirically observed by many researchers.

- So far, we have discussed **missing labels**.
- In many problems, we are missing some of the features.
- More on this later in the semester.

Final notes

- You should know:
  - Problem definition for semi-supervised learning.
  - Self-training method, pros/cons
  - Generative approach, generative models
  - Concept of Generative Adversarial Nets
- Significant material for these slides was taken from:
  - *http://pages.cs.wisc.edu/~jerryzhu/icml07tutorial.html*
  - *http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf*
  - *http://www.cs.cmu.edu/~tom/10701_sp11/slides/LabUnlab-3-17-2011.pdf*
  - *https://drive.google.com/file/d/0ByUKRdiCDK7-bTgxTGoxYjQ4NW8/view?usp=drive_web*