Semi-supervised Learning

Conal Sathi and Sameer Singh

BANA 290: ADVANCED DATA ANALYTICS

MACHINE LEARNING FOR TEXT

SPRING 2018

May 22, 2018

Upcoming...

Homework

- Homework 3 out!
- Due in ~1 week: May 29, 2017
- Focused on clustering and embeddings

Project

- Will send feedback regarding proposals by tomorrow
- Progress presentations in week 10 (~2 weeks)

Office Hours

- Conal's office hours 4pm-6pm tomorrow (Wed May 23rd)
- Feel free to come by to discuss HW3 or Final Project or Data Science post-graduation

Semi-supervised Learning

MOTIVATION AND CONCEPT

Supervised Learning

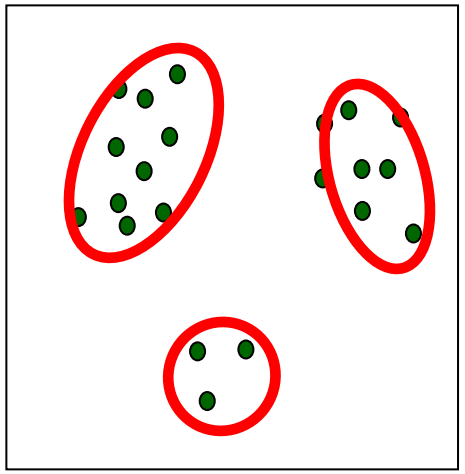
Given a set of labeled data, predict which label a new data point belongs to

Algorithms we've discussed in this class:

- Naïve Bayes
- K-Nearest Neighbors
- Logistic Regression
- Decision Trees
- Bagging and Boosting Algorithms

Unsupervised Learning

Given a set of data, understand how the data groups together (without any semantic meaning of which group means)



Algorithms we've discussed in this class:

- Clustering
  - K-Means
  - DBSCAN
  - Hierarchical
- Latent Semantic Analysis
- Word Embeddings

Semi-supervised Learning

Solving same problem as supervised learning:
Given a set of labeled data, predict which label a new data point belongs to

But… make use of both labeled data and unlabeled data

Why?

- Hard to get a large amount of labeled data (expensive)
  - Requires human expert to label the data
- There may be bias in how you get labeled data
  - E.g. Building a general purpose sentiment analyzer when you only have access to IMDB data
- Unlabeled data is generally much more plentiful
  - Just crawl the internet!
- Closer to how humans learn!

How to use unlabeled data?

Features

Self-training

Weak supervision

How to use unlabeled data?

Features

Self-training

Weak supervision

Using clusters as features

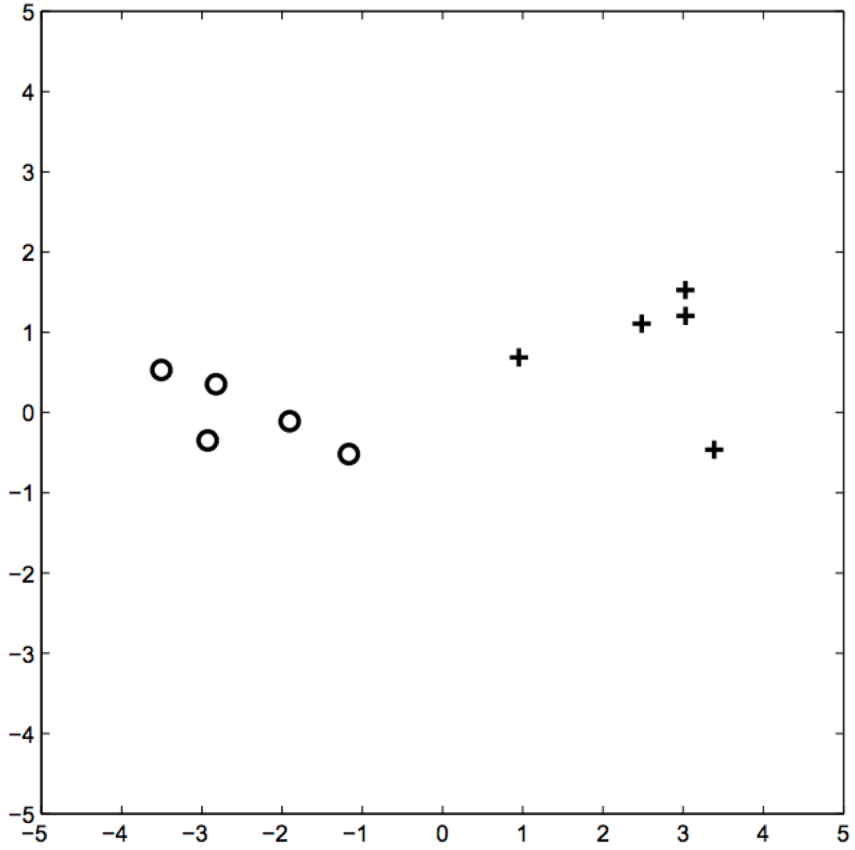
Run clustering algorithm on full data set (labeled and unlabeled)

- K-Means
- DBSCAN
- Hierarchical

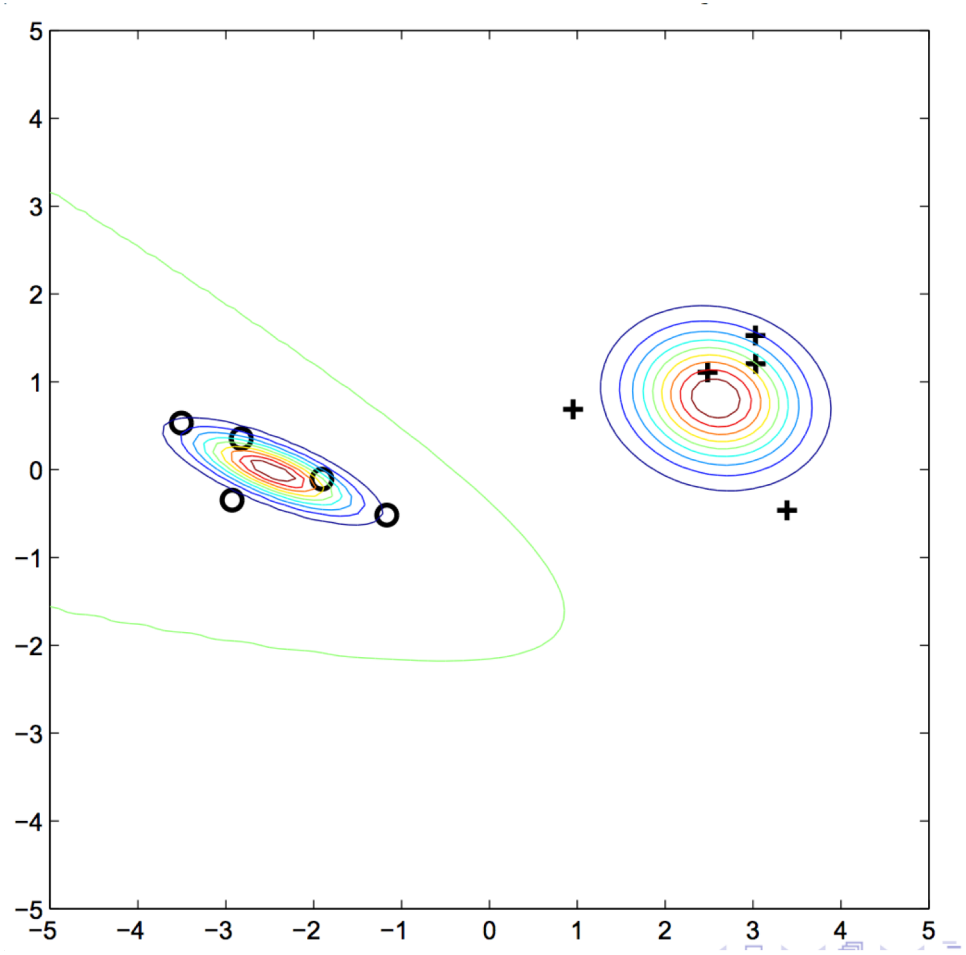
This gives us a new feature (cluster id)

Train a supervised model with this new feature on the labeled data set.

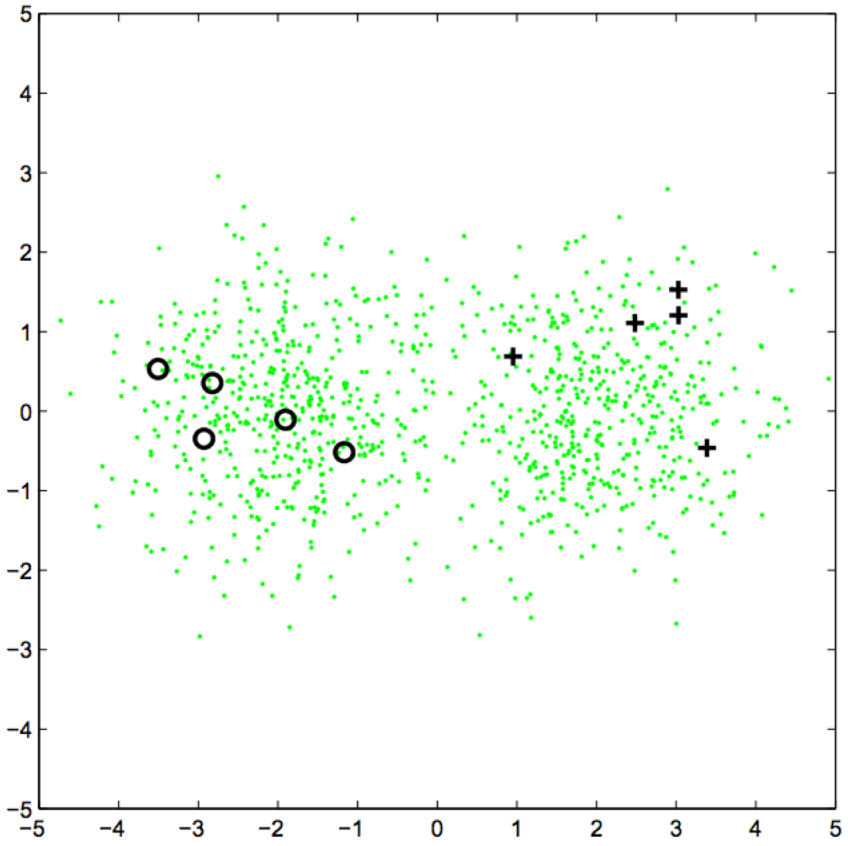
Why might this work?



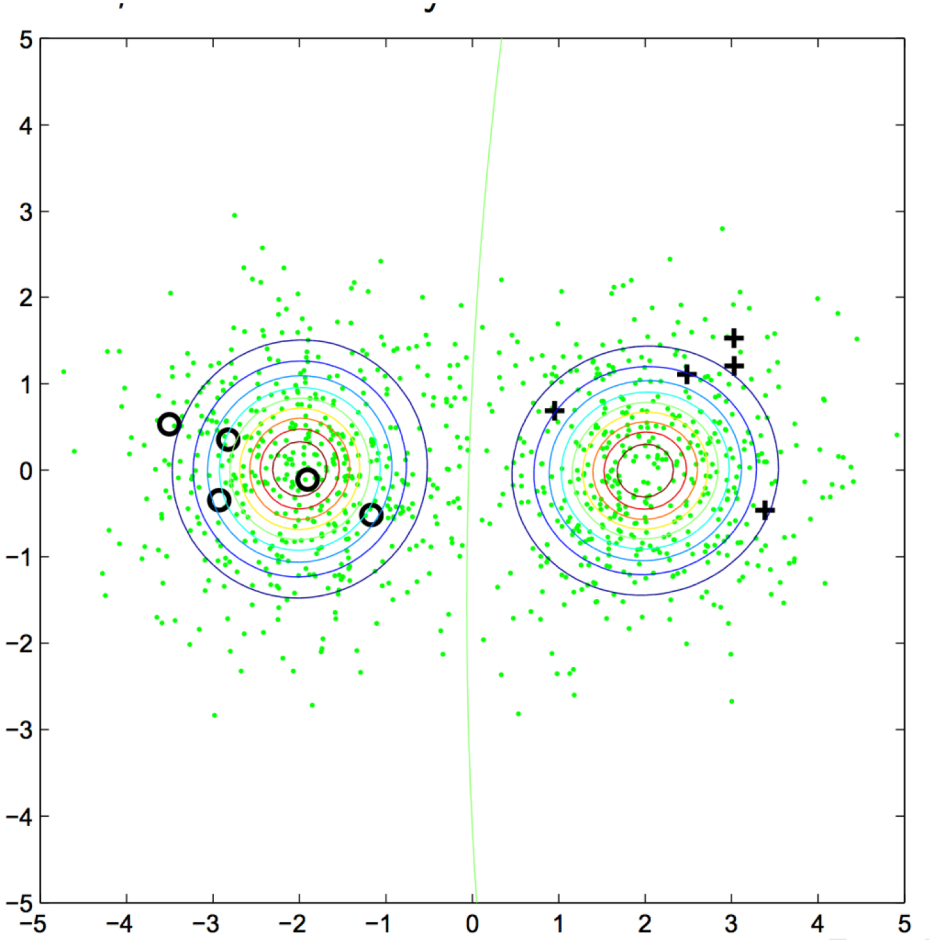
Why might this work?



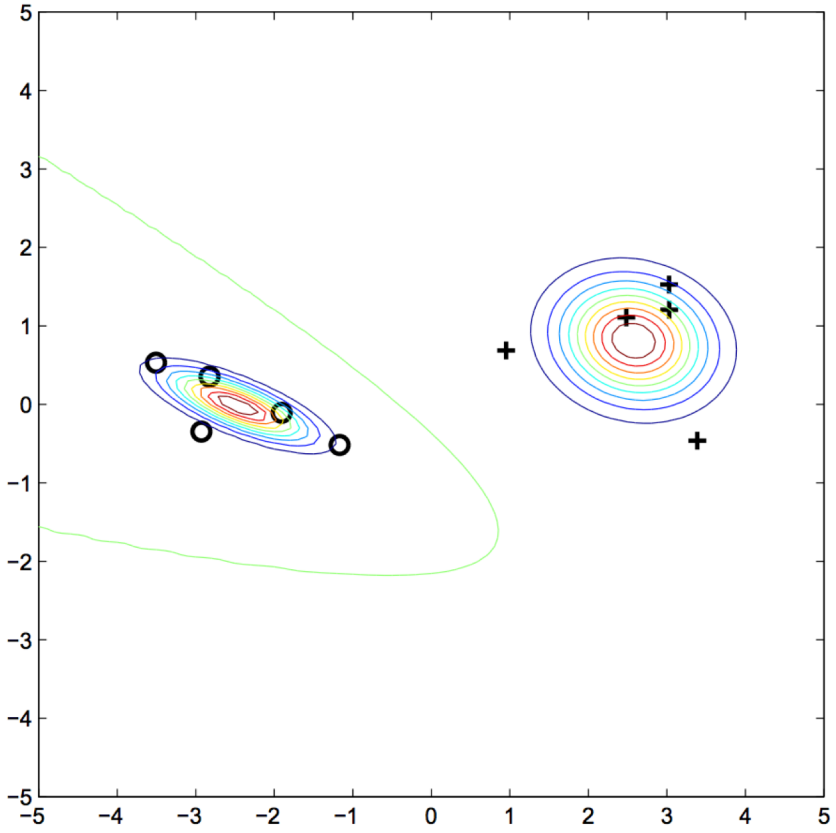
Adding unlabeled data

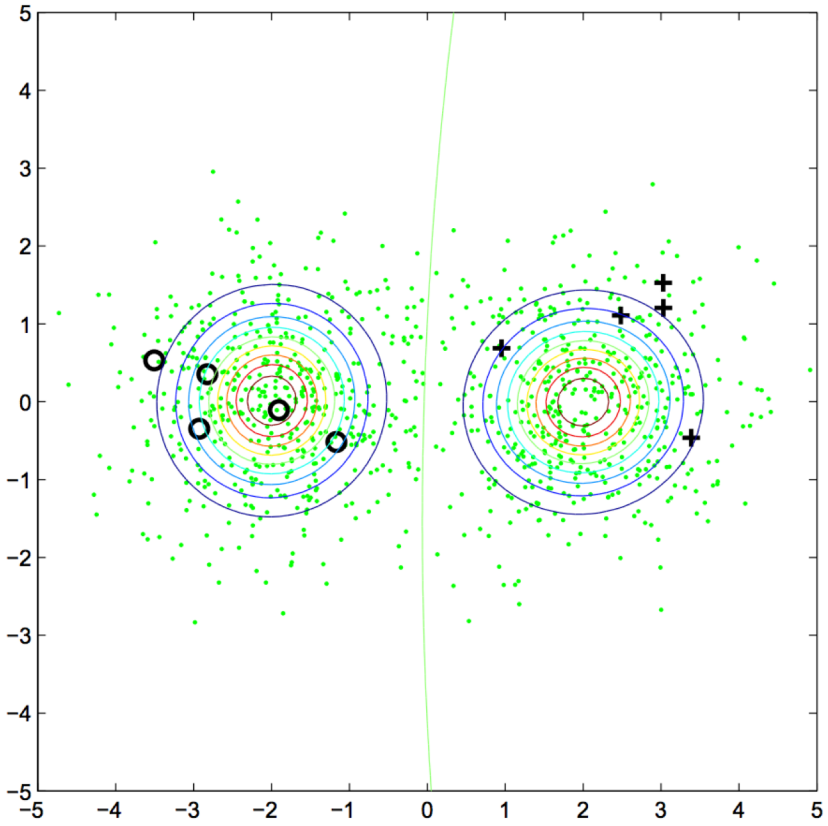


Adding unlabeled data



Using only labeled vs. labeled + unlabeled





*these charts came from Xiaojin Zhu's slides http://pages.cs.wisc.edu/~jerryzhu/pub/sslicml07.pdf

Using “dense” vectors

Convert ”sparse” vectors to “dense” vectors on full data set using:

- Latent Semantic Analysis using Singular Value Decomposition (SVD)
- Word Embeddings

Train model with new features on labeled data set

Why would this help?

- Helps with *synonymy* and *polysemy*

In-Class Activity 1

How to use unlabeled data?

Features

Self-training

Weak supervision

Self-training

Assumption: if the model is highly confident on a classification (e.g. probability is high), then the classification is correct

- Remember the precision/recall tradeoff discussed earlier

Self-training algorithm:

1. Train a model from the labeled data
2. Use the model to classify unlabeled data
3. Add the confident classifications from the unlabeled data to the labeled data
4. Repeat

Variations in Self-training

Self-training algorithm:

1. Train a model from the labeled data
2. Use the model to classify unlabeled data
3. Add the confident classifications from the unlabeled data to the labeled data
4. Repeat

Different versions of self-training:

- In step 3, add only classifications (not just confident ones)
- In step 3, add all classifications to labeled data, weigh each by confidence

Advantages of Self-training

- Increases the size of the labeled training data without human effort!
- Can be used on many different classifiers (Naïve Bayes, Logistic Regression, Decision Trees, Neural Networks, etc.) provided that there is some sort of confidence metric
- Pretty simple to implement

Disadvantages of Self-training

What if you add mistakes to your training data?

Distribution of your training data is changing automatically and may diverge from the distribution of data in the wild

Related concept: Co-training

If your features can neatly split into two sets,

1. From the labeled data, train one classifier on one set and train the other classifier on the second set
2. Classify the unlabeled data with each of the two classifiers separately
3. Add the first classifier's most confident classifications to the second classifier's labeled data
4. Add the second classifier's most confident
5. Repeat

“If your features can neatly split into two sets”

Example?

- If you’re classifying web pages, maybe using the image features as one set of features and the text features as a second set of features
- If you’re classifying an email, maybe use the text features in the subject as one set of features and the text features in the body as another set of features

Pros and Cons of Co-Training

Pros:

- Increases the size of the labeled training data without human effort!
- Can be used on many different classifiers (Naïve Bayes, Logistic Regression, Decision Trees, Neural Networks, etc.) provided that there is some sort of confidence metric
- Pretty simple to implement
- Less sensitive to mistakes than self-training

Cons:

- Natural feature splits may not exist
- Models using both features should do better

Variants of Co-Training

- Add all examples, not just most confident (but weighed by probability)
- Artificial feature split (just split all features randomly)
- Don’t split features but just train multiple classifiers with very different assumptions (e.g. a linear classifier and a nonlinear one)

Active Learning

- If you have resources (human domain experts) to label your data but they're in limited supply, one option is to only have them label examples where your classifier is unconfident

In-Class Activity 2

How to use unlabeled data?

Features

Self-training

Weak supervision

Weak supervision

- Use heuristics
  - E.g. create a regex/rule (all emails from this domain are spam, all emails from that domain aren't, etc.)
- Use dictionaries
  - If a review only contains positive words, then it's positive
  - If an email contains a a baseball team or a baseball player, then the category is baseball
- Use mappings from other domain
  - If you have training data for Amazon's taxonomy, then create a mapping from Amazon's taxonomy to your own category taxonomy
- Use weak learners (underfitting classifiers)
- Use crowdsourcing
- There will be errors in this training data (won't be perfect), but it may be okay if it significantly increases the size of your training data

Snorkel

New tool that allows you to insert heuristics and noisy training data from weak supervision

It will handle cases where the heuristics/sources disagree by assessing the reliability of each heuristic/source

Motivation is that the state-of –the-art models that are coming out (deep learning) require very large training sets

snorkel

A training data creation and management system focused on information extraction