



Department of Statistics, University of Michigan

# An adventure in Semi-supervised learning

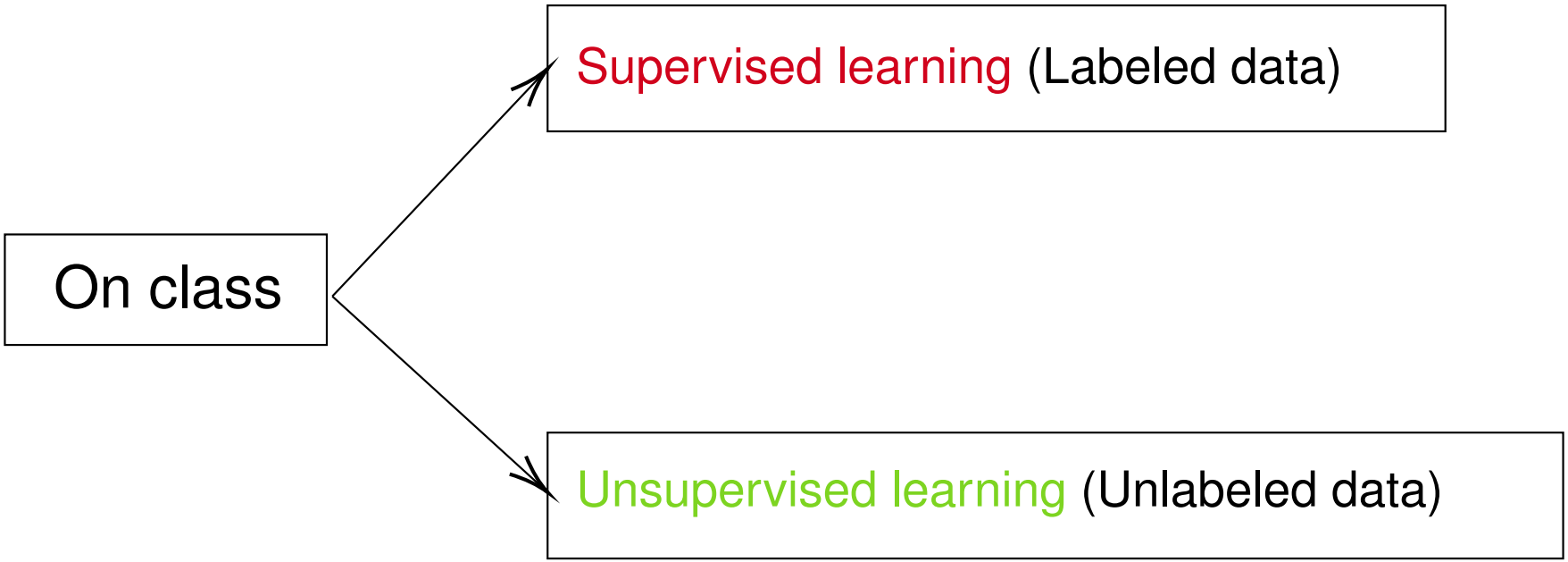
April 16, 2020

Trong Dat Do, Ziyi Song

Stats 601 Project Presentation

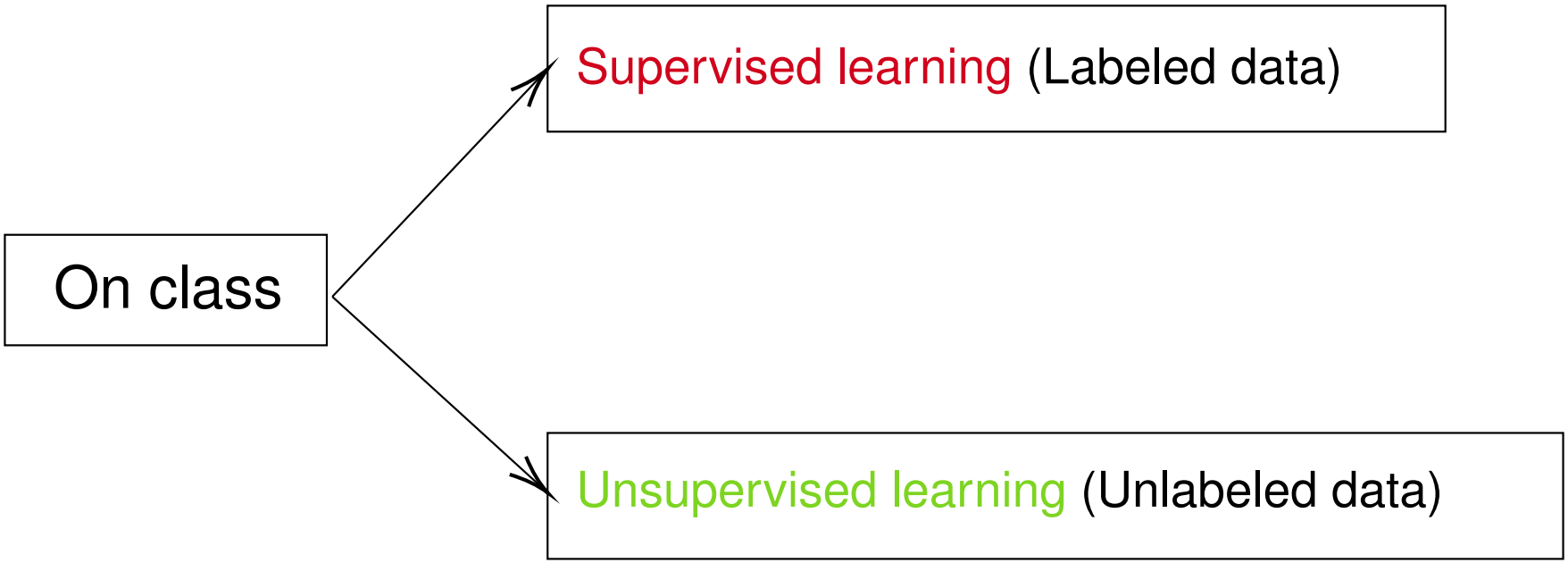
# Outline

# Motivation



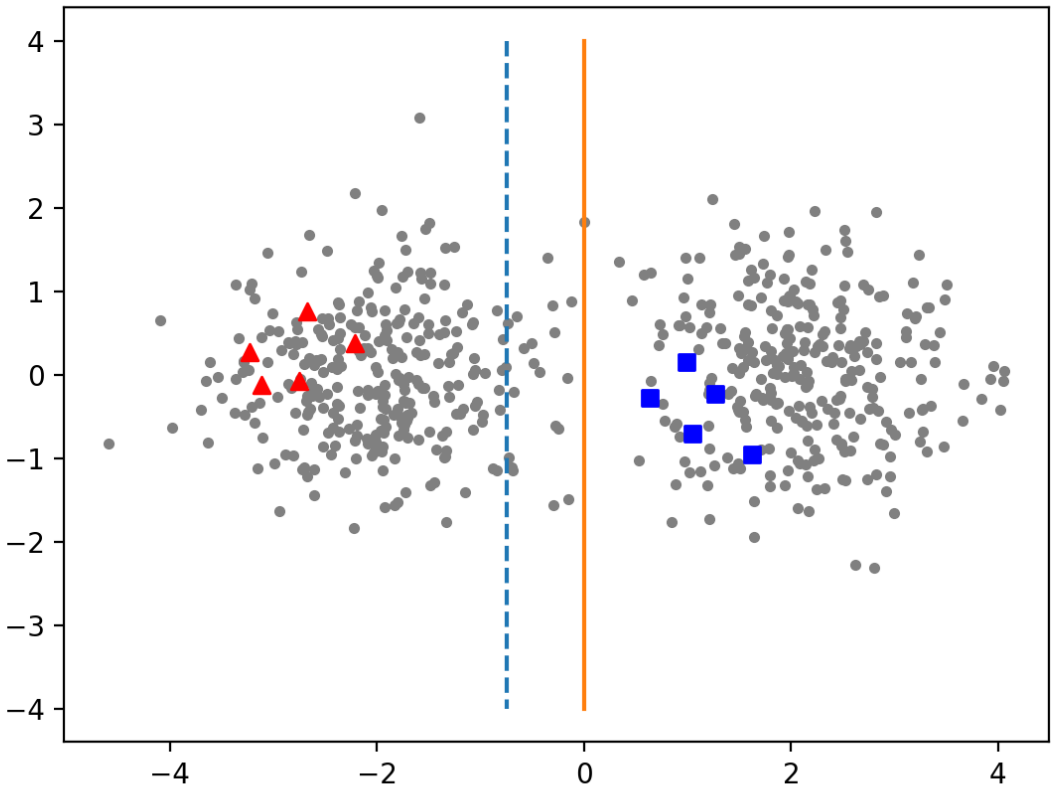

Question: What if our data contains both
labeled and unlabled data? (Medical data/Internet data/...)

# Motivation



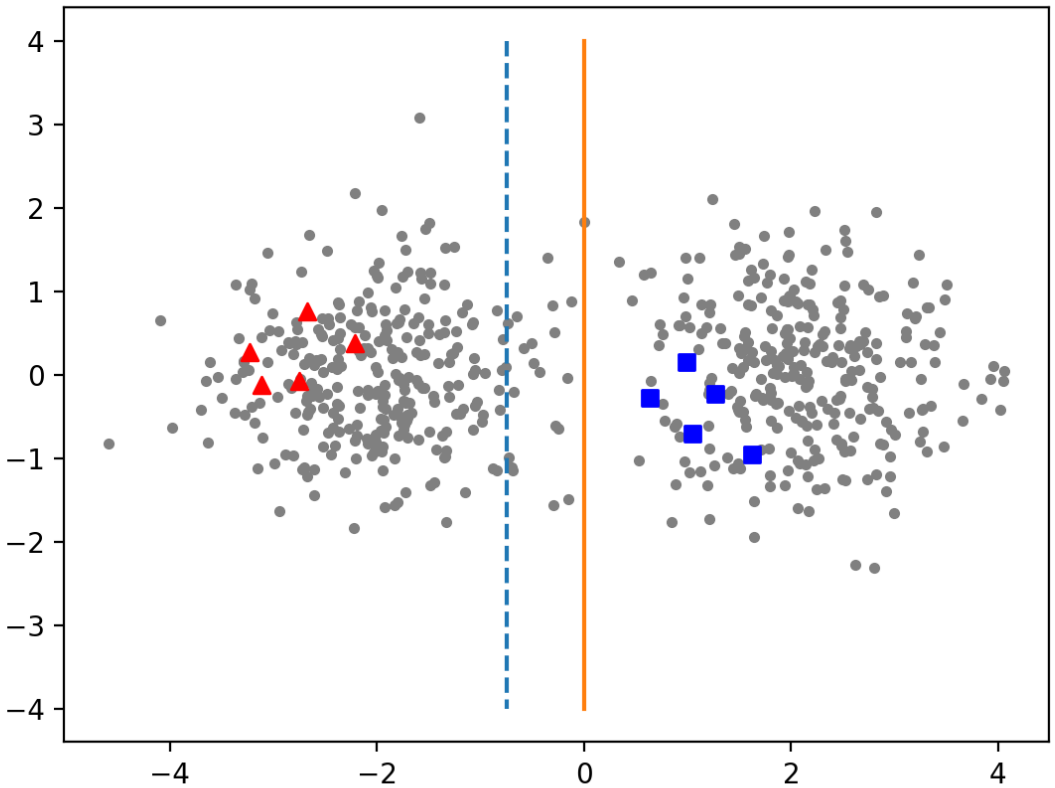

Question: What if our data contains both
labeled and unlabled data? (Medical data/Internet data/...)
$\rightarrow$ Semi-supervised learning

# An example




Blue dashed line: Supervised learning decision boundary.
Orange line: The decision boundary we deserve!

# An example




Blue dashed line: Supervised learning decision boundary.
Orange line: The decision boundary we deserve! How to make it?

# Semi-supervised learning, or How unlabeled data help predicting labels

- A research field following the trend of Machine Learning [Van Engelen and Hoos, 2020], two latest survey: [Van Engelen and Hoos, 2020], [Zhu, 2005]. My observation: Not very active compared to Supervised Learning research. Hardly find any paper comparing between them.
- Let's make a comparison: Do unlabeled data really help?

| Discriminant Analysis | Mixture Model |
|---|---|
| Supervised learning method | Semi-Supervised learning method |
| Any classifier | Self-training |
| Ensemble classifier | Co-training |
| SVM | S3VM |

# Semi-supervised learning, or How unlabeled data help predicting labels

- A research field following the trend of Machine Learning [Van Engelen and Hoos, 2020], two latest survey: [Van Engelen and Hoos, 2020], [Zhu, 2005]. My observation: Not very active compared to Supervised Learning research. Hardly find any paper comparing between them.
- Let's make a comparison: Do unlabeled data really help?

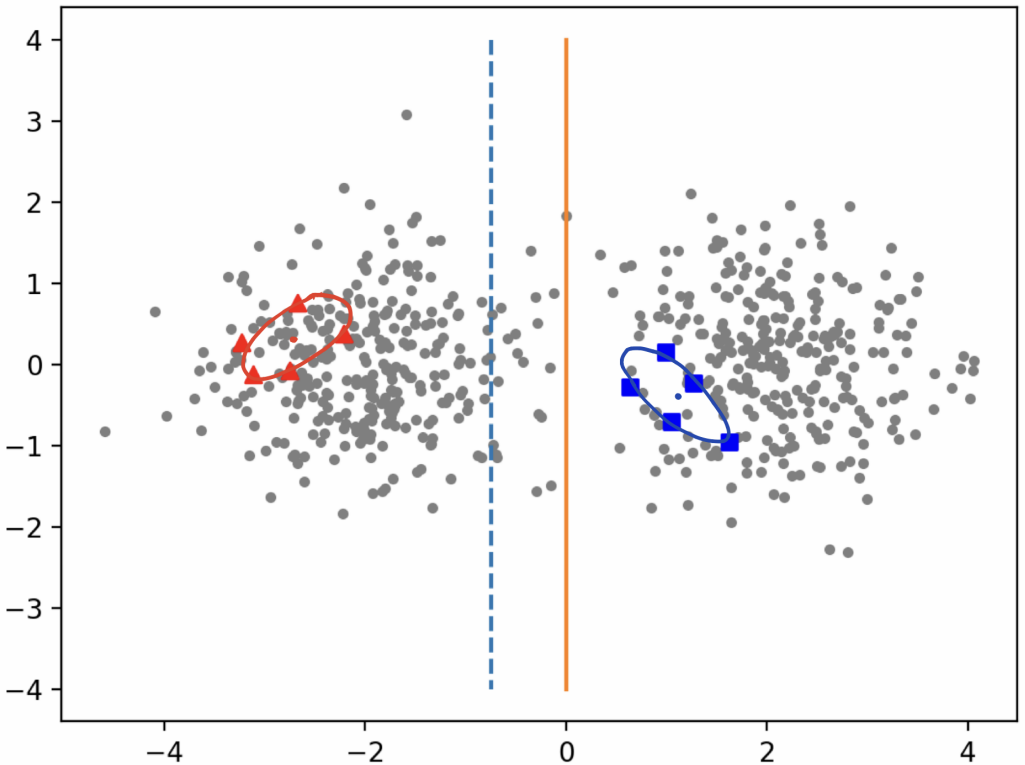
| Discriminant Analysis | Mixture Model |
|---|---|
| Supervised learning method | Semi-Supervised learning method |
| k Nearest Neighborhood | Self-training with kNN |
| AdaBoost/Bagging | Co-training by committee |
| SVM | S3VM |

# Familiar data sets from UCI

Table: Data set from UCI, labels partition by KEEL

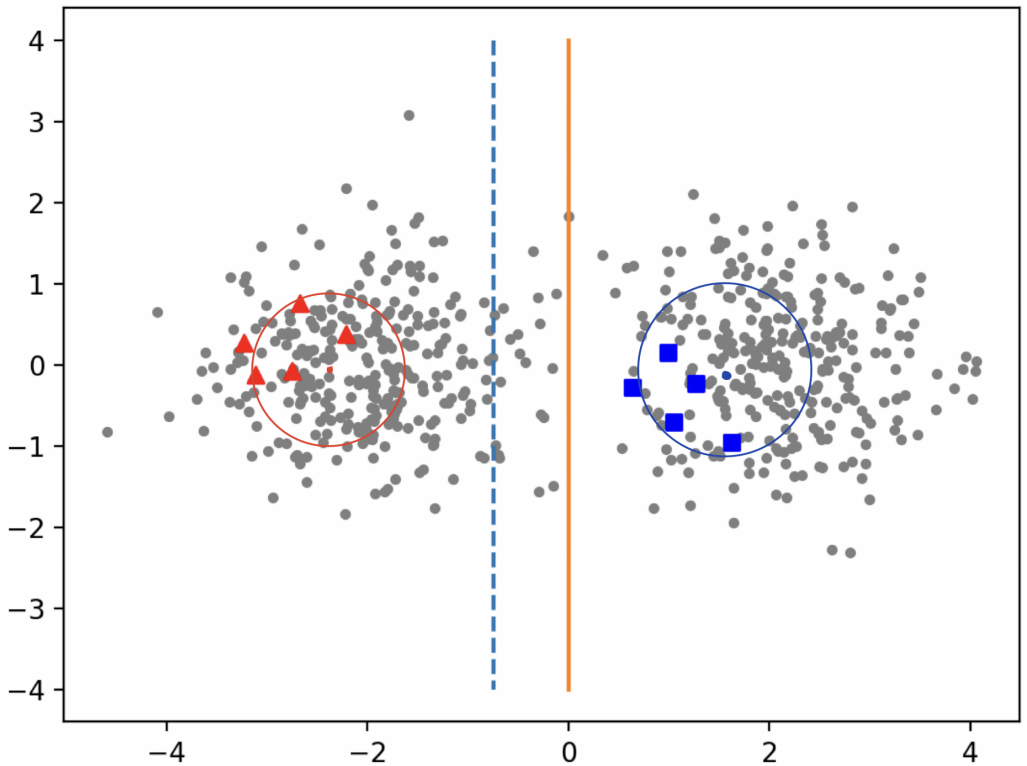
| Data set | Sample size | Features | Classes |
|---|---|---|---|
| appendicitis | 106 | 7 | 2 |
| banana | 5300 | 2 | 2 |
| cleveland | 303 | 13 | 2 |
| ecoli | 336 | 7 | 8 |
| glass | 214 | 9 | 7 |
| iris | 150 | 4 | 3 |
| led7digit | 500 | 7 | 10 |
| pima | 786 | 8 | 2 |
| titanic | 2201 | 3 | 2 |
| wine | 178 | 13 | 3 |

# Quick review: Semi-supervised Mixture Model



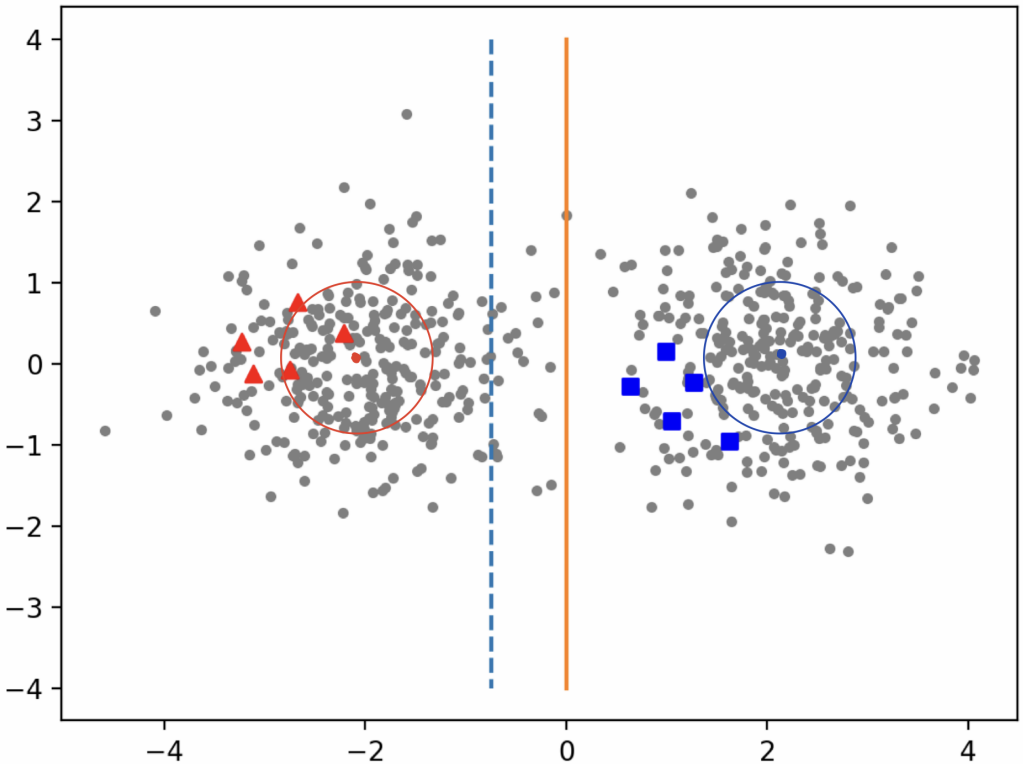

- Combination of Supervised and Unsupervised Mixture Model
- Assumption: Data follows the mixture model
- Implemented from [Zhu and Goldberg, 2009]

# Quick review: Semi-supervised Mixture Model



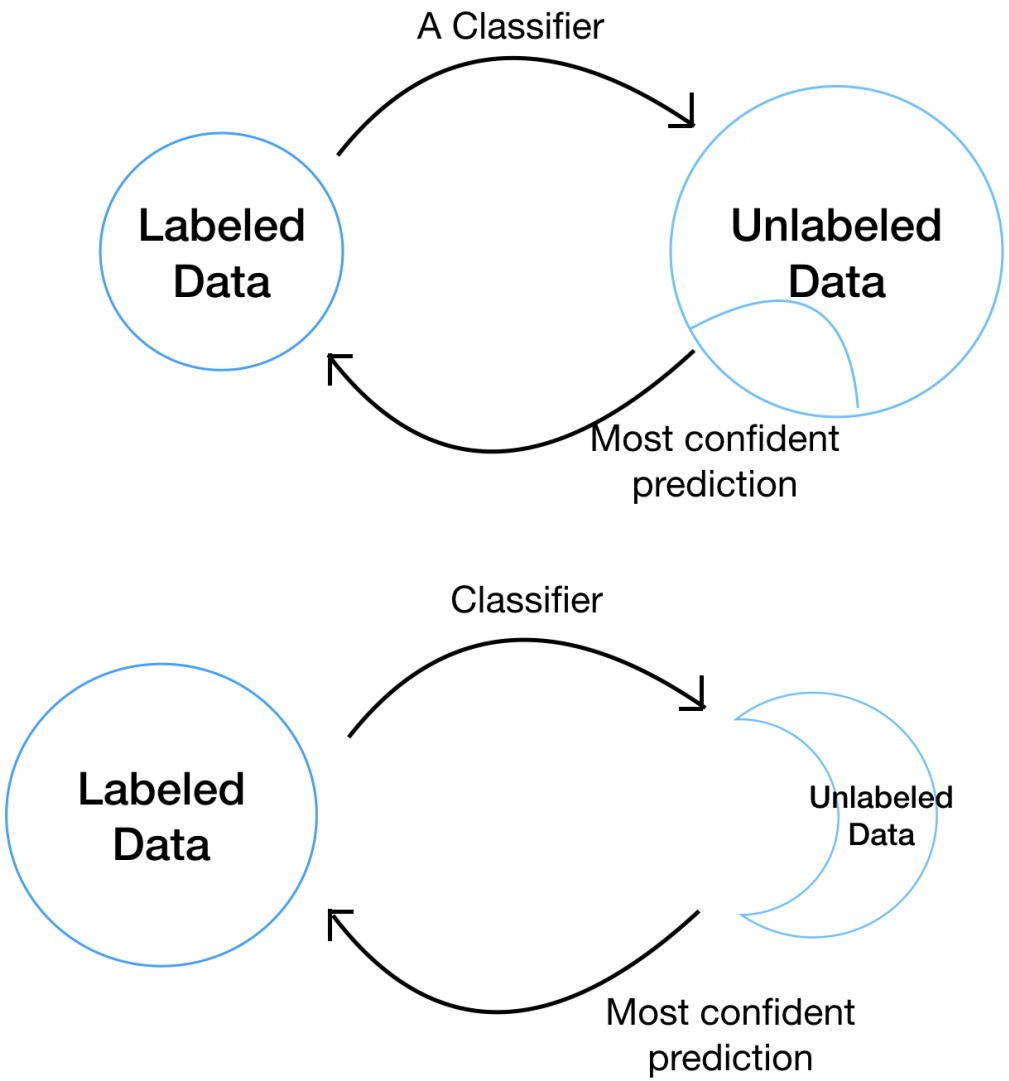

- Combination of Supervised and Unsupervised Mixture Model
- Assumption: Data follows the mixture model
- Implemented from [Zhu and Goldberg, 2009]

# Quick review: Semi-supervised Mixture Model



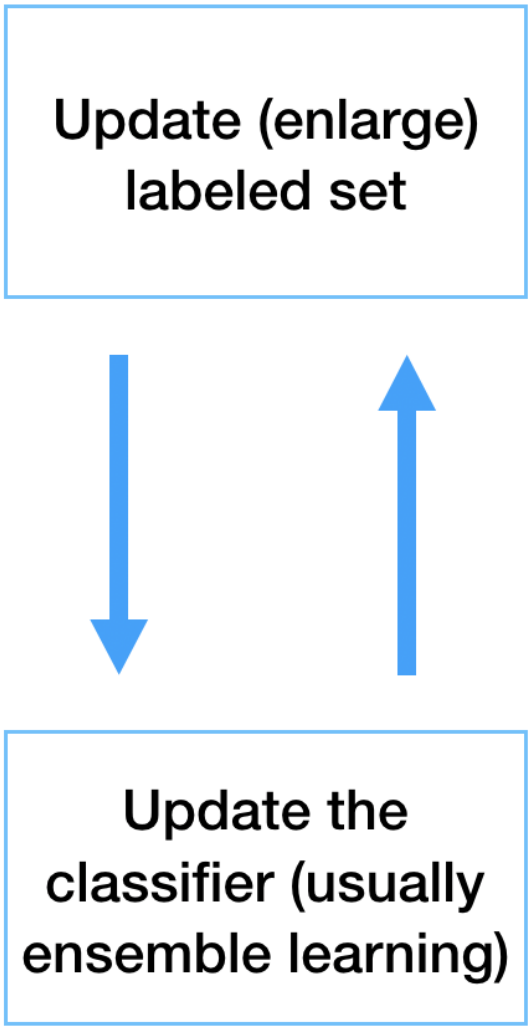

- Combination of Supervised and Unsupervised Mixture Model
- Assumption: Data follows the mixture model
- Implemented from [Zhu and Goldberg, 2009]

# Quick review: Self-training




- Naive technique
- Assumption: The classifier is good. (The inference from bad self-labeled data can hurt the labeled)
- Implemented from [[Zhu and Goldberg, 2009]]

# Quick review: Co-training by Committee (AdaBoost)




- AdaBoost is good when prediction error of ensemble members are different
- By using unlabeled data, ensemble members have chance to make different errors and communicate with each other (increase diversity)
- Implemented from [[Hady and Schwenker, 2008]]

# Result: Semi-supervised mixture model

Table: Test error of Gaussian Mixture Model for 10% and 20% labeled data set

| data_name | 10% Super | 10% Semi-Super | 20% Super | 20% Semi-Super |
|---|---|---|---|---|
| appendicitis | 21.7% | 20.75% | 24.53% | 22.64% |
| banana | 44.68% | 41.51% | 44.42% | 39.0% |
| cleveland | 46.13% | 37.37% | 46.13% | 45.45% |
| ecoli | 56.85% | 53.27% | 57.44% | 55.06% |
| glass | 64.49% | 70.09% | 63.08% | 65.89% |
| iris | 66.67% | 3.33% | 59.33% | 2.67% |
| led7digit | 89.6% | 61.8% | 85.0% | 38.4% |
| pima | 30.6% | 34.24% | 33.2% | 32.55% |
| titanic | 32.08% | 32.08% | 32.08% | 32.08% |
| wine | 59.55% | 6.18% | 33.71% | 3.37% |

# Result: Semi-supervised mixture model (cont.)

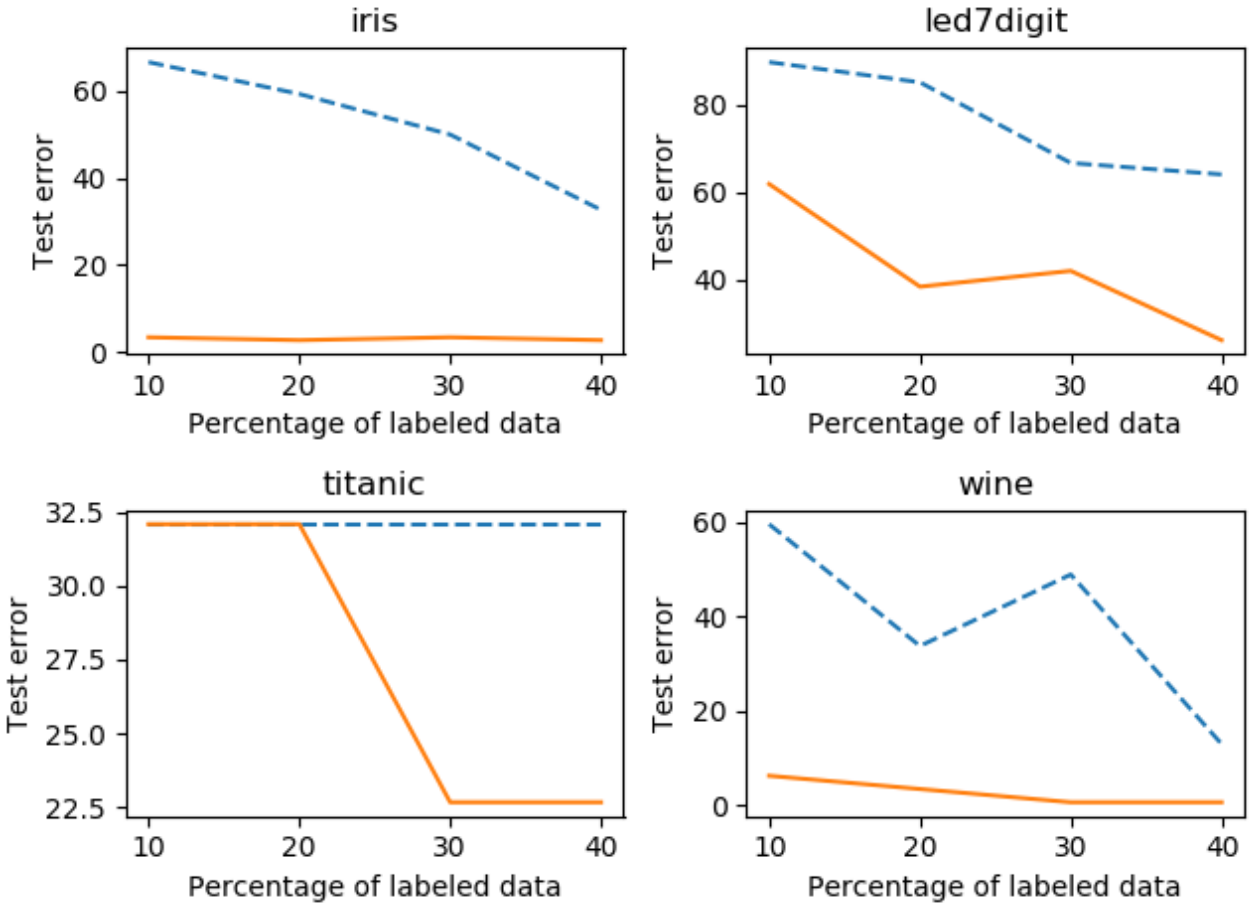



Figure: Blue line: Supervised learning error, Orange: Semi-supervised learning error

# Result: Self-training

Table: CV and test error of Self-training for 10% labeled data set,

$$\text{improve} = \frac{\text{Error Supervised} - \text{Error Semi-supervised}}{\text{Error Semi-supervised}}$$

| Data set | CV super | CV semi-super | Test super | Test semi-super | improve-ment |
|---|---|---|---|---|---|
| appendicitis | 1.0% | 1.0% | 18.87% | 16.98% | 11.11% |
| banana | 4.91% | 4.82% | 5.81% | 5.72% | 1.65% |
| cleveland | 19.13% | 17.21% | 21.55% | 21.55% | 0.0% |
| ecoli | 3.98% | 3.98% | 13.99% | 13.39% | 4.44% |
| glass | 6.36% | 6.36% | 19.63% | 19.16% | 2.44% |
| iris | 0.77% | 0.0% | 5.33% | 4.0% | 33.33% |
| led7digit | 24.44% | 24.89% | 26.4% | 27.4% | -3.65% |
| pima | 4.93% | 4.49% | 20.96% | 20.05% | 4.55% |
| titanic | 24.09% | 21.26% | 20.95% | 21.67% | -3.35% |
| wine | 0.62% | 0.62% | 5.06% | 4.49% | 12.5% |

# Result: Co-training

Table: Test error of co-training with parameters set up as in [Hady and Schwenker, 2008]

| Data set | Test Super | test_SS Semi-Super | Improve -ment |
|---|---|---|---|
| appendicitis | 7.55% | 8.4% | -12.82% |
| banana | 15.19% | 12.13% | 20.14% |
| cleveland | 23.3% | 20.37% | 12.47% |
| ecoli | 12.47% | 12.14% | 2.52% |
| glass | 16.03% | 16.12% | -0.91% |
| iris | 4.67% | 4.67% | 0.0% |
| led7digit | 24.84% | 24.4% | 1.76% |
| pima | 24.74% | 21.74% | 12.11% |
| titanic | 21.08% | 21.08% | 0.0% |
| wine | 8.31% | 7.19% | 12.68% |

# Result: Conclusions and Comments

- The results of Self-training and Co-training are not as good as in the paper [Triguero et al., 2015] and [Hady and Schwenker, 2008]. Will set the parameters more carefully
- Self-training and co-training may take lots of time to train (computationally cost)
- Self-training is not always good. Self-training by a bad model can even make the result worse. (Because of its assumption)
- Will implement S3VM ([Gieseke et al., 2012]) to compare and draw a conclusion about semi-supervised learning in final report.

# How limited labeled data help clustering

For partially labeled datasets, the limited labels help semi-supervised clustering, e.g., identifying the number of clusters, etc.
One natural and naive method is *cluster-then-label* algorithm [X.Zhu Goldberg 2009]:

---

**Input:** labeled data $(x_1, y_1), ..., (x_m, y_m)$, unlabeled data $x_{m+1}, ..., x_{m+n}$, a clustering method $\mathcal{A}$, a supervised method $\mathcal{L}$
**Output:** labels on unlabeled data $y_{m+1}, ..., y_{m+n}$
1. Cluster $x_1, ..., x_{m+n}$ using $\mathcal{A}$
2. For each resulting cluster, let $\mathcal{S}$ be the set of labeled instances in this cluster:
**if** $\mathcal{S}$ *is non-empty* **then**
  learn a supervised model from $\mathcal{S}$, $f_{\mathcal{S}} = \mathcal{L}(\mathcal{S})$, apply $f_{\mathcal{S}}$ to all unlabeled data in this cluster
**end**
**if** $\mathcal{S}$ *is empty* **then**
  use predictor $f$ trained from all labeled data
**end**

# small experiment on Wine data

- Wine data(13 dimensions, 178 observations)
  - 20% data are labeled with 3 different labels, rest 80% data are unlabeled
- What I did:
  - use hierarchical agglomerative clustering with complete linkage on the whole data
  - stop the algorithm once only 3 clusters remain,because the labeled data contains only 3 classes.
  - Find the majority label within each cluster, and this label to all unlabeled instances in this cluster
- clustering accuracy
  - 0.84, seems nice
- **Big Concerns**
  - **In real world, labeled data always doesn't contain all the classes, not all clusters are partially labeled, the number of clusters is unknown!**

# Backbone Disease data

We randomly remove most labels, make it a partially labeled data.

(Background: there are different types of backbone diseases, e.g., DH, SL, LMD, LST, MLD, etc.)

- partially labeled dataset on patients having backbone diseases, with 6 biomedical variables for each person
- labeled data has 3 classes: Normal, DH, SL.
- to cluster this dataset,
  we cannot set the number of clusters $K$ in advance

**Dirichlet process! Dirichlet Process Gaussian Mixture Model (DPGMM)!**

# Dirichlet Process Gaussian Mixture Model (DPGMM)

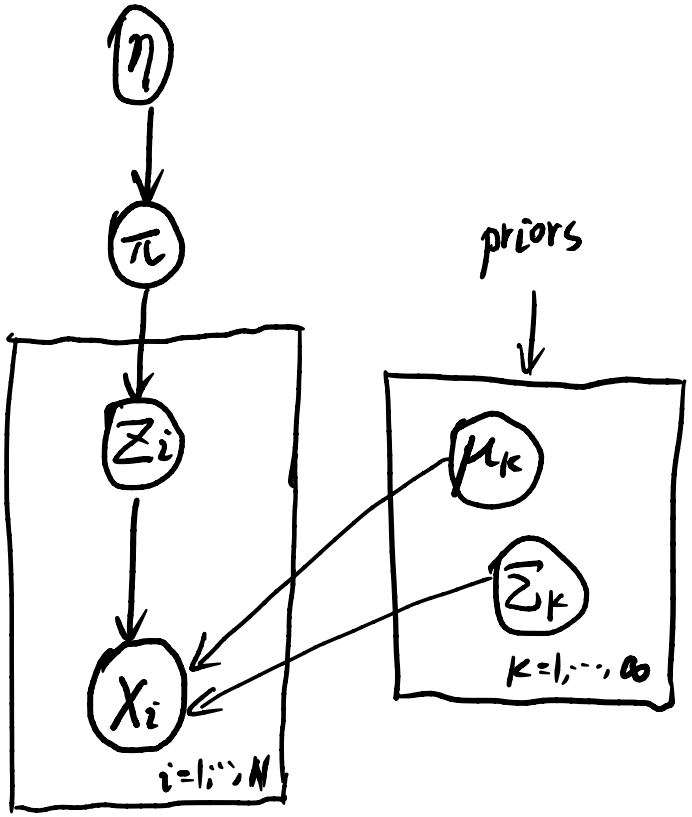
Basic idea

- DPGMM has **infinite** components of Gaussian Mixture Model and does not require us to specify the number of components $K$ at first
- The goal is to use the **posterior** distributions of $K$ to find an optimal choice for clustering

# Dirichlet Process Gaussian Mixture Model (DPGMM)

$\pi \sim$ *Dirichlet*(K,$\alpha$) or GEM($\eta$)
$\mu_k \sim \mathcal{N}(\mu_0, \Sigma_0)$
$\Sigma_k \sim$Inv-Wishart($\Psi_0, \nu_0$)
$\boldsymbol{z}_i|\pi \sim$Categorical($\pi$)
$\boldsymbol{x}_i|\boldsymbol{z}_i =$k, $\mu, \Sigma \sim p_x(\cdot|\boldsymbol{z}_i =$k, $\mu, \Sigma)$
$k = 1, ..., K$
$i = 1, ..., N$

where $\pi \sim$ *GEM*($\eta$) is equivalent to
$\beta_k \sim$Beta(1,$\eta$) for $\forall k$
$\pi_k = \beta_k \prod_{i=1}^{k-1}(1 - \beta_i)$



# DPGMM on Backbone disease data

To draw inference on this probabilistic model, we compute estimate the posterior distribution

$$p(\boldsymbol{z}_{1:N}, \mu, \Sigma, \pi | \boldsymbol{x}_{1:N})$$

using

- Markov Chain Monte Carlo methods, e.g., Gibbs sampling, etc
- variational inference to approximate

Applying DPGMM on Backbone disease data gives clustering accuracy as 0.54. It is not good. Why?

Because $\boldsymbol{z}_i | \pi \sim \mathit{Categorical}(\pi)$. Loosely speaking, it means we treat all the data in the same way. We didn't utilize the knowledge/labels of the labeled data.

# Digression a bit on Accuracy for Clustering

Knowing the ground truth of cluster labels, to calculate the accuracy of our clustering methods, we have to find the best match between the cluster labels and true labels. The accuracy is defined as:

$$accuracy(y, \hat{y}) = \max_{permutation \in P} \frac{1}{N} \sum_{i=1}^{N} 1(permutation(\hat{y}_i = y_i)$$

where P is the set of all permutations of $[1, ..., K]$, $K$ is the number of clusters

e,g,: if true labels are [1,1,2,2,3,3,3], and our cluster labels are [2,2,1,1,3,4,4], then the accuracy is 1 - 1/7 = 6/7

FYI: There are $O(K!)$ permutations, but Hungarian Algorithm computes it in $O(K^3)$

# Semi-DPGMM [[Amine Echraibi, 2019]]

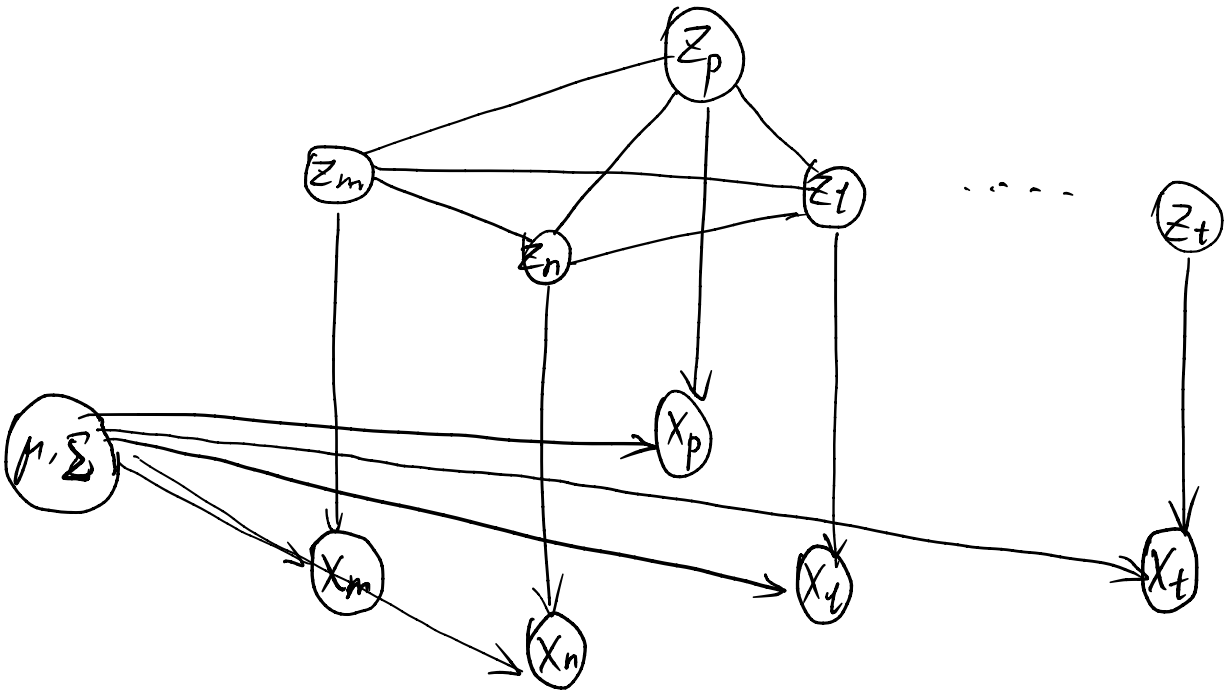
To utilize the information of limited labeled data, we need to consider the statistical dependency between the hidden random variables $\boldsymbol{z}_n$ with same label $\ell_n$:

$\boldsymbol{z}_n$ is neighbor of $\boldsymbol{z}_m \Longleftrightarrow \ell_n = \ell_m$, and we later need to quantify the "connection/relation" between $\boldsymbol{z}_n$ and $\boldsymbol{z}_m$

Such statistical dependency is used to construct the joint prior distribution of $\boldsymbol{z}_{1:N}$, different from the $\boldsymbol{z}_i|\boldsymbol{\pi} \sim$Categorical($\boldsymbol{\pi}$) in DPGMM

# Semi-DPGMM

And each $\boldsymbol{x}_n$ is independent of $\boldsymbol{x}_m$ given $\boldsymbol{z}_n$, $\mu$, and $\Sigma$



$\pi \sim$ *Dirichlet*(K,$\alpha$) or GEM($\eta$)
$\mu_k \sim \mathcal{N}(\mu_0, \Sigma_0)$
$\Sigma_k \sim$Inv-Wishart($\Psi_0, \nu_0$)
$\boldsymbol{z}_{1:N}|\pi \sim p_{\boldsymbol{z}_{1:N}}(\cdot|\pi)$
$\boldsymbol{x}_i|\boldsymbol{z}_i =$k$, \mu, \Sigma \sim p_x(\cdot|\boldsymbol{z}_i =$k$, \mu, \Sigma)$

where joint pdf
$\mathrm{p}_x(\cdot|\boldsymbol{z}_i =$k$, \mu, \Sigma) =$
$\frac{1}{\Gamma} \prod_{n=1}^{N} \pi_{\boldsymbol{z}_n}^{1[\mathcal{N}_n=\emptyset]} \prod_{n\sim m} e^{-\lambda\nu(\boldsymbol{z}_n,\boldsymbol{z}_m)}$

where $\mathcal{N}_n$ is neighborhood of $\boldsymbol{z}_n$
$n \sim m$ means $\boldsymbol{z}_n$ is neighbor of $\boldsymbol{z}_m$

Intuition of this joint pdf is that:

take $\nu(\boldsymbol{z}_n, \boldsymbol{z}_m)$ as KL divergence of densities of two clusters, or say, as a "distance"

if $\boldsymbol{z}_n = \boldsymbol{z}_m$, then $\nu(\boldsymbol{z}_n, \boldsymbol{z}_m) = 0$, then $e^{-\lambda\nu(\boldsymbol{z}_n,\boldsymbol{z}_m)} = 1$

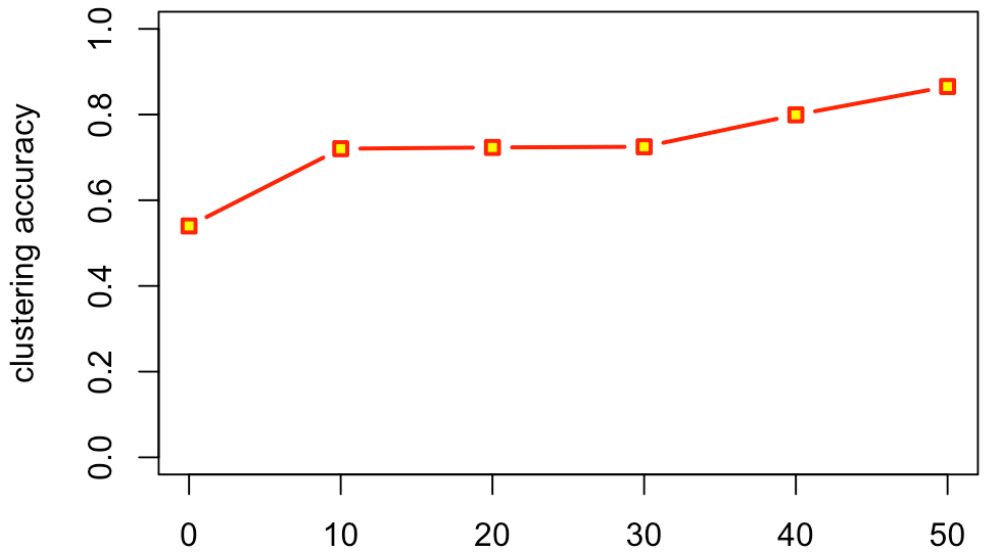
if $\boldsymbol{z}_n \neq \boldsymbol{z}_m$, then $\nu(\boldsymbol{z}_n, \boldsymbol{z}_m) > 0$, then $e^{-\lambda\nu(\boldsymbol{z}_n,\boldsymbol{z}_m)} < 1$. Thus data log-likelihood gets smaller, and $\nu(\boldsymbol{z}_n, \boldsymbol{z}_m)$ plays a role of penalty.

# Apply Semi-DPGMM on Backbone disease data

Use 5 types of partially labeled datasets, with 10%, 20%, 30%, 40%, 50% of labeled data, respectively. Each type has 10 copies.

The accuracy, with respect to each type of labeled data, are 0.72033, 0.72324, 0.72484, 0.79935, 0.86547

Recall when not using labeled data, the accuracy of DPGMM is about 0.54. It shows the limited labeled data does help in the the semi-supervised clustering.




Amine Echraibi, Joachim Flocon-Cholet, S. G. S. V. (2019).
Bayesian mixture models for semi-supervised clustering.
Technical report, Orange Labs IMT Atlantique, France.


Gieseke, F., Airola, A., Pahikkala, T., and Kramer, O. (2012).
Sparse quasi-newton optimization for semi-supervised support vector machines.
In *ICPRAM*.


Hady, M. F. A. and Schwenker, F. (2008).
Co-training by committee: a new semi-supervised learning framework.
In *2008 IEEE International Conference on Data Mining Workshops*, pages 563–572. IEEE.


Triguero, I., García, S., and Herrera, F. (2015).
Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study.
*Knowledge and Information systems*, 42(2):245–284.


Van Engelen, J. E. and Hoos, H. H. (2020).
A survey on semi-supervised learning.
*Machine Learning*, 109(2):373–440.


Zhu, X. and Goldberg, A. B. (2009).
Introduction to semi-supervised learning.
*Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.


Zhu, X. J. (2005).
Semi-supervised learning literature survey.
Technical report, University of Wisconsin-Madison Department of Computer Sciences.

THANK YOU FOR LISTENING!

Any comment can be sent to: dodat@umich.edu and ziyisong@umich.edu