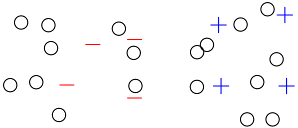
Study on Semi-Supervised Learning with the Framework of Generative Model
生成モデルを用いた半教師付き学習に関する研究

Nguyen Hoang Nghia
Supervisor: Professor Yamada Koichi

Nagaoka University of Technology
Information and Management Systems Engineering

From the View of Learning Data

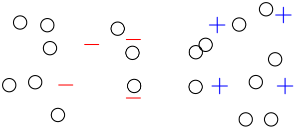
Semi-supervised learning (SSL)



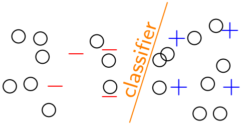
Labeled with a large amount of unlabeled data

From the View of Learning Data

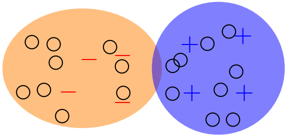
Semi-supervised learning (SSL)



Labeled with a large amount of unlabeled data



Discriminative: Direct solution



Generative: Data boundary

Outline

Multinomial Naive Bayes

- Inductive learning
- A class is constructed of many sub-components
- How do we initialize these sub-components?
- Apply on text data

Component Assumptions

Given a component set M, we have

$$P(x) = \sum_{M_j \in M} P(x|M_j)P(M_j)$$

with predefined $P(y|M_j) \in \{0, 1\}$

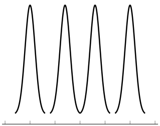
Component Assumptions

Given a component set M, we have

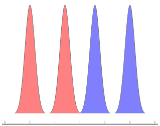
$$P(x) = \sum_{M_j \in M} P(x|M_j)P(M_j)$$

with predefined $P(y|M_j) \in \{0, 1\}$

Many-to-one assumption



$P(x)$



$P(x|y)$

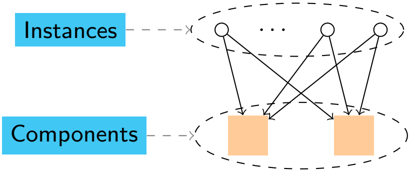
How Do We Initialize These Components?

Conventional methods set it randomly

$$P(M_j|y) \sim \mathcal{U}(0,1) \quad \text{such that} \quad \sum_{M_j : P(y|M_j)=1} P(M_j|y) = 1$$

In this way, it assumes that

- All components are the same
- Different outputs when we re-sampling



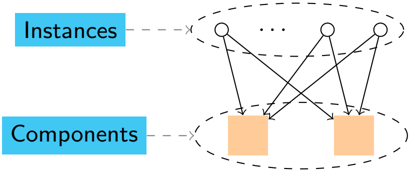
How Do We Initialize These Components?

Conventional methods set it randomly

$$P(M_j|y) \sim \mathcal{U}(0,1) \quad \text{such that} \quad \sum_{M_j:P(y|M_j)=1} P(M_j|y) = 1$$

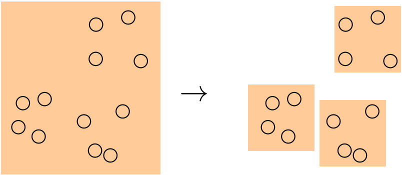
In this way, it assumes that

- All components are the same
- Different outputs when we re-sampling



Or it better to be seen as an unsupervised task

- Consider the structure of data
- Keep the same samples when re-sampling



Apply on Text Data

Let text feature x be the word count (Naive Bayes assumption) with dictionary $\mathbf{w}$. Then component conditional is multinomial distribution

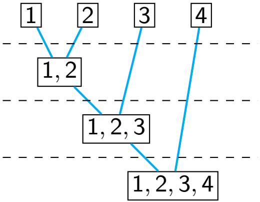
$$P(x|M_j, \theta) \sim Multinomial(x, P(\mathbf{w}|M))$$

Apply on Text Data

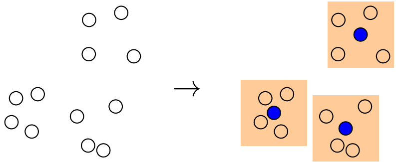
Let text feature x be the word count (Naive Bayes assumption) with dictionary $\mathbf{w}$. Then component conditional is multinomial distribution

$$P(x|M_j, \theta) \sim Multinomial(x, P(\mathbf{w}|M))$$

Component Initialization



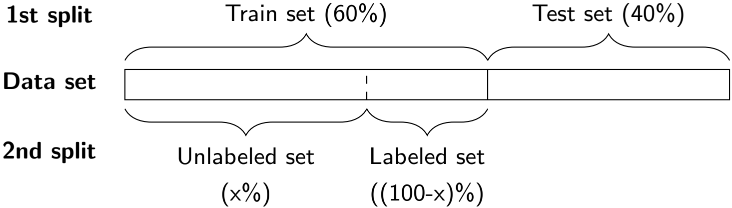
Agglomerative Tree



K-means

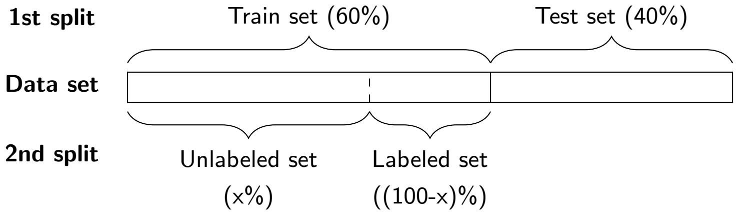
Experimental Results

Data splitting process using the split-x



Experimental Results

Data splitting process using the split-x



- Data: 20Newsgroup
- Default cross validation: 5 folds
- Similarity measure: Cosine for trees, Euclidean for K-means

Experimental Results (Thesis Page 11)

Results average in f1-score with split-x, (#labeled, #unlabeled)

Table: comp versus rec

Random	Tree	Kmeans
Split-70, (1064:2484)		
0.82	0.85	0.84

Table: comp versus sci

Random	Tree	Kmeans
Split-70, (1061:2476)		
0.91	0.91	0.90

Table: comp versus talk

Random	Tree	Kmeans
Split-70, (977:2280)		
0.93	0.93	0.93

Experimental Results (Thesis Page 11)

Results average in f1-score with split-x, (#labeled, #unlabeled)

Table: comp versus rec

Random	Tree	Kmeans
Split-70, (1064:2484)		
0.82	0.85	0.84
Split-97, (106:3442)		
0.74	0.77	0.77

Table: comp versus sci

Random	Tree	Kmeans
Split-70, (1061:2476)		
0.91	0.91	0.90
Split-97, (106:3431)		
0.86	0.89	0.89

Table: comp versus talk

Random	Tree	Kmeans
Split-70, (977:2280)		
0.93	0.93	0.93
Split-97, (97:3160)		
0.90	0.92	0.92

Outline

Graph-based Approach

- Transductive learning
- Present data on Graph
- Look for the separation between classes
- What happen if the separation is overlapped?

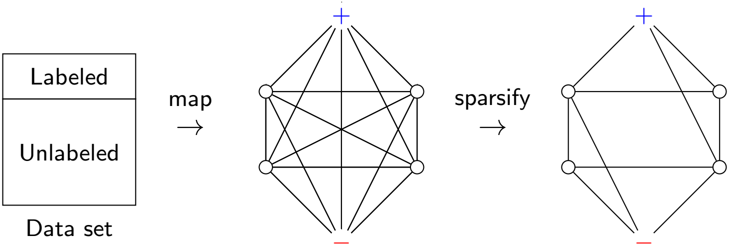
Graph Construction

$$G = (V, E, W)$$

Vertices $\{V_L, V_U\}$

Edges

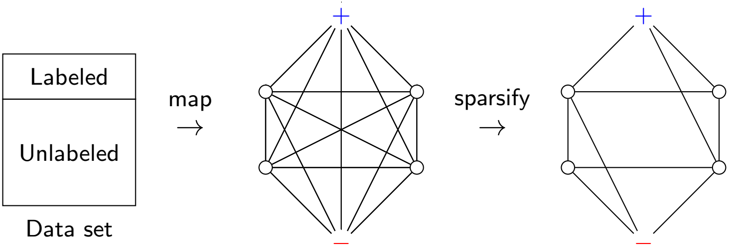
Similarity weights



Graph Construction

$$G = (V, E, W)$$

Vertices $\{V_L, V_U\}$ Edges Similarity weights



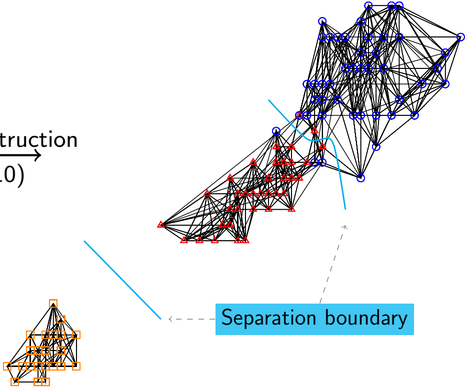
Some simple schedules

- K most similar vertices for each vertex (kNN)
- Spanning tree with maximum weight (MST)

Sparse Graph Separation Assumption

Iris data
(150 instances, 4 features)

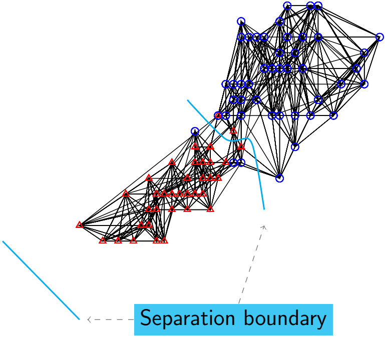
kNN construction
(k=10)

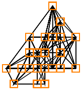


Sparse Graph Separation Assumption

Iris data
(150 instances, 4 features)

kNN construction $\xrightarrow{\hspace{2em}}$ (k=10)





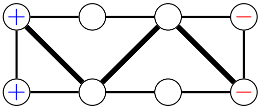
Objective function

$$\text{argmin}_f \quad \frac{1}{2} \sum_{(i,j)\in E} W_{i,j} |f(i) - f(j)|$$

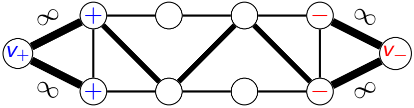
where $f(i) \in \{-1, +1\}$ represents label for $i \in V$.

Minimum Cut Separation

Blum and Chawla in “Learning from Labeled and Unlabeled Data Using Graph Mincuts” gives a solution with a minimum cut on G



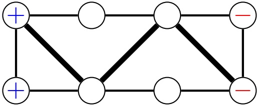
Input graph G



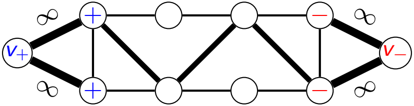
Step 1, add pseudo vertices v_+, v_- and corresponding edges

Minimum Cut Separation

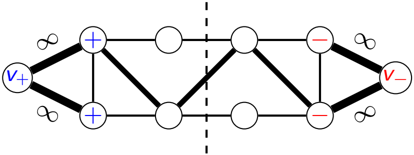
Blum and Chawla in “Learning from Labeled and Unlabeled Data Using Graph Mincuts” gives a solution with a minimum cut on G



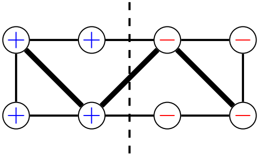
Input graph G



Step 1, add pseudo vertices v_+, v_- and corresponding edges



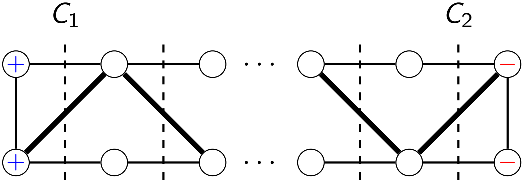
Step 2, find a minimum cut in G with source v_+ and sink v_-



Step 3, use this cut to label unlabeled vertices

When the Boundary Is Collapsed

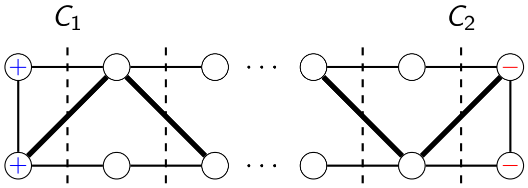
A graph may have more than one positive solution and can cause the degenerate inference.



$\rightarrow$ The need for a balanced separation.

When the Boundary Is Collapsed

A graph may have more than one positive solution and can cause the degenerate inference.



$\rightarrow$ The need for a balanced separation.

- Loop for all positive separations and choose the best
- Add random noise to have better change of the right boundary (*randomized mincut*, Blum *et al.*)
- **Intervene the reasoning process**

Graphical Model

Consider the vertices as random variables

$$X = \{X_i : i \in V, X_i = f(i)\}$$

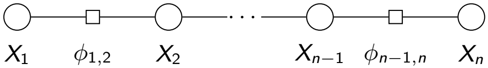
Graphical Model

Consider the vertices as random variables

$$X = \{X_i : i \in V, X_i = f(i)\}$$

We define a set of independent factors. Assume that the implied distribution of X only depends on Φ

$$\Phi = \{\phi_{i,j} : (i,j) \in E, \phi_{i,j} = \exp(-W_{i,j}(X_i - X_j)^2)\}$$



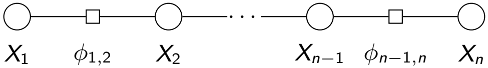
Graphical Model

Consider the vertices as random variables

$$X = \{X_i : i \in V, X_i = f(i)\}$$

We define a set of independent factors. Assume that the implied distribution of X only depends on Φ

$$\Phi = \{\phi_{i,j} : (i,j) \in E, \phi_{i,j} = \exp(-W_{i,j}(X_i - X_j)^2)\}$$

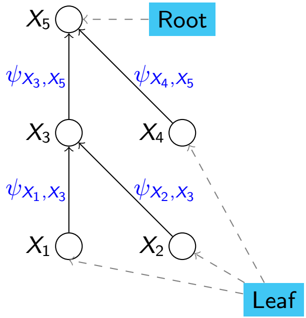


We are looking for a configuration X_{max} of X that

$$P_\Phi(X_{max}) = \max_X P_\Phi$$

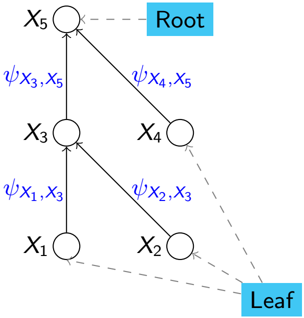
Influence Index

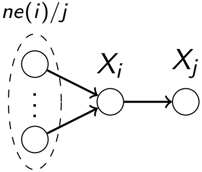
The inference can be represented in a message sending schedule



Influence Index

The inference can be represented in a message sending schedule



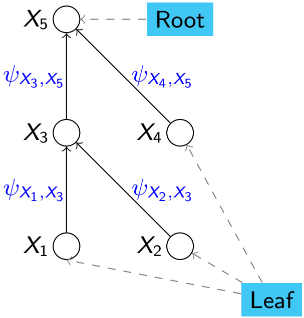


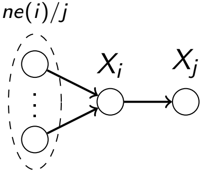
More possible solutions with equal condition

$$\psi_{-1,X_j} = \psi_{+1,X_j}$$

Influence Index

The inference can be represented in a message sending schedule





More possible solutions with equal condition

$$\psi_{-1,X_j} = \psi_{+1,X_j}$$

We can replace it with the Influence index

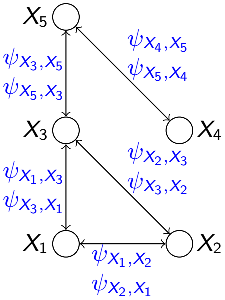
$$\text{influence}_{i \in V_U} = \text{argmax}_{X_i} \sum_{X_i = X_j, \forall j \in V_L} W_{i,j}$$

Loopy Belief Propagation Algorithm

Message sending is not guaranteed when we have circles in G

Loopy Belief Propagation Algorithm

Message sending is not guaranteed when we have circles in G

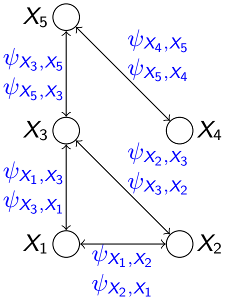


The approximate loopy process

- Messages will be sent from both directions, all at the same time.
- The decision will be made on all neighbor vertices $P_{\Phi}(X_i) = \max_{X_i} \prod_{k \in ne(i)} \psi_{X_k,X_i}$

Loopy Belief Propagation Algorithm

Message sending is not guaranteed when we have circles in G



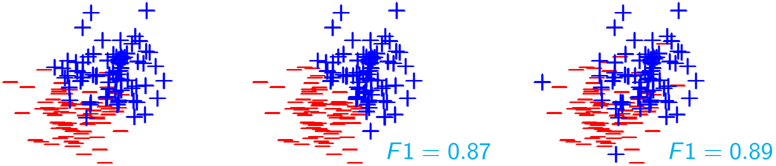
The approximate loopy process

- Messages will be sent from both directions, all at the same time.
- The decision will be made on all neighbor vertices $P_\Phi(X_i) = \max_{X_i} \prod_{k \in ne(i)} \psi_{X_k,X_i}$

$\rightarrow$ We can only check the equal condition (when using influence index) after the loop is finished

$$\prod_{k \in ne(i)} \psi_{X_k,-1} = \prod_{k \in ne(i)} \psi_{X_k,+1}$$

Synthetic Data

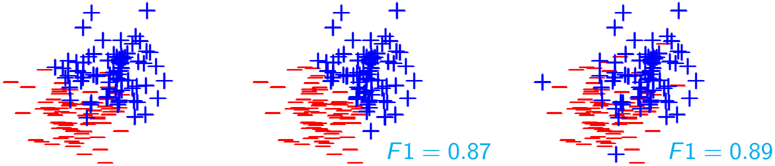


(a) True data (b) Mincut approach (c) Graphical model

$\mathcal{N}(0, 0.6^2)$ & $\mathcal{N}(1, 0.6^2)$

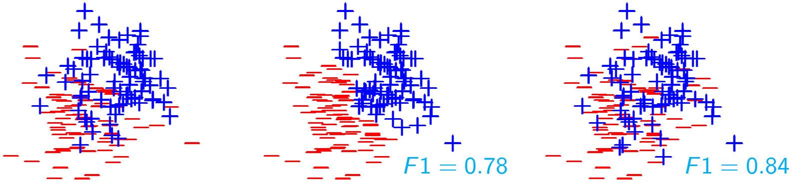
(75 instances for each label: 7 labeled and 68 unlabeled)

Synthetic Data



(a) True data (b) Mincut approach (c) Graphical model

$\mathcal{N}(0, 0.6^2)$ & $\mathcal{N}(1, 0.6^2)$



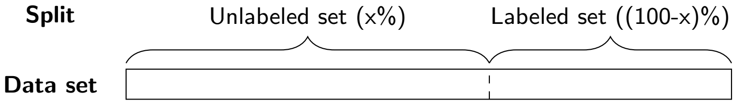
(a) True data (b) Mincut approach (c) Graphical model

$\mathcal{N}(0, 0.8^2)$ & $\mathcal{N}(1, 0.8^2)$

(75 instances for each label: 7 labeled and 68 unlabeled)

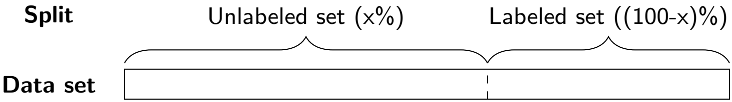
Experimental Results

Data splitting process using the split-x



Experimental Results

Data splitting process using the split-x



- Data: Abalone, Digit, 20Newsgroup
- Default cross validation: 5 folds
- Similarity measure: Cosine for text, Euclidean similarity and Gaussian kernel for others

$\rightarrow$ Assume that we have pre-examined a fit graph construction method for our model.

Experimental Results (Thesis Page 29-31)

Results average in f1-score with split-x, (#labeled, #unlabeled)

Table: Abalone data

Mincut	**Graphical**
Split-70, (1152:2690)	
0.76	0.73
Split-97, (115:3727)	
0.69	0.72

Table: Digit data

Mincut	**Graphical**
Split-70, (539:1258)	
0.99	0.99
Split-97, (53:1744)	
0.95	0.95

Experimental Results (Thesis Page 29-31)

Results average in f1-score with split-x, (#labeled, #unlabeled)

Table: Abalone data

Mincut	Graphical
Split-70, (1152:2690)	
0.76	0.73
Split-97, (115:3727)	
0.69	0.72

Table: Digit data

Mincut	Graphical
Split-70, (539:1258)	
0.99	0.99
Split-97, (53:1744)	
0.95	0.95

Table: comp vs rec, 20Newsgroups data

Mincut	Graphical
Split-70, (2661:6209)	
0.85	0.85
Split-97, (266:8604)	
0.41	0.84

Conclusion

Our assumption

The label values are distributed as dense components and these components are separated in its data representation form.

Multinomial Naive Bayes

- Component initialization problem

→ Unsupervised component assignment

Graph-based

- Component separation problem

→ Graphical model with influence index

Long term view

"Semi-supervised learning is much more a practical than a theoretical problem." (Olivier, Schölkopf, and Zien, *Semi-Supervised Learning*).

Resources & References

Resources and references including full thesis, experimental implementations and results are online at

https://github.com/nghiapickup/master_thesis

Thank You!

Questions and Discussions?