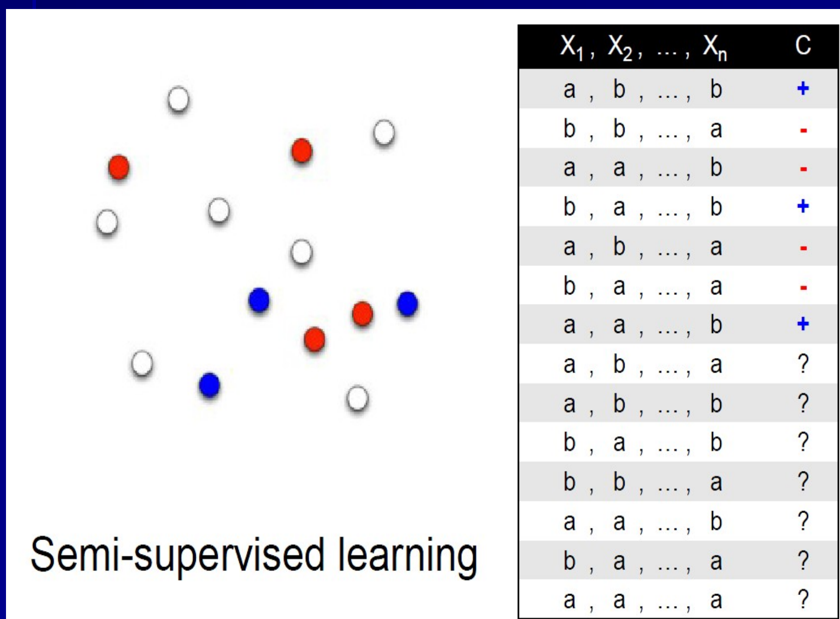
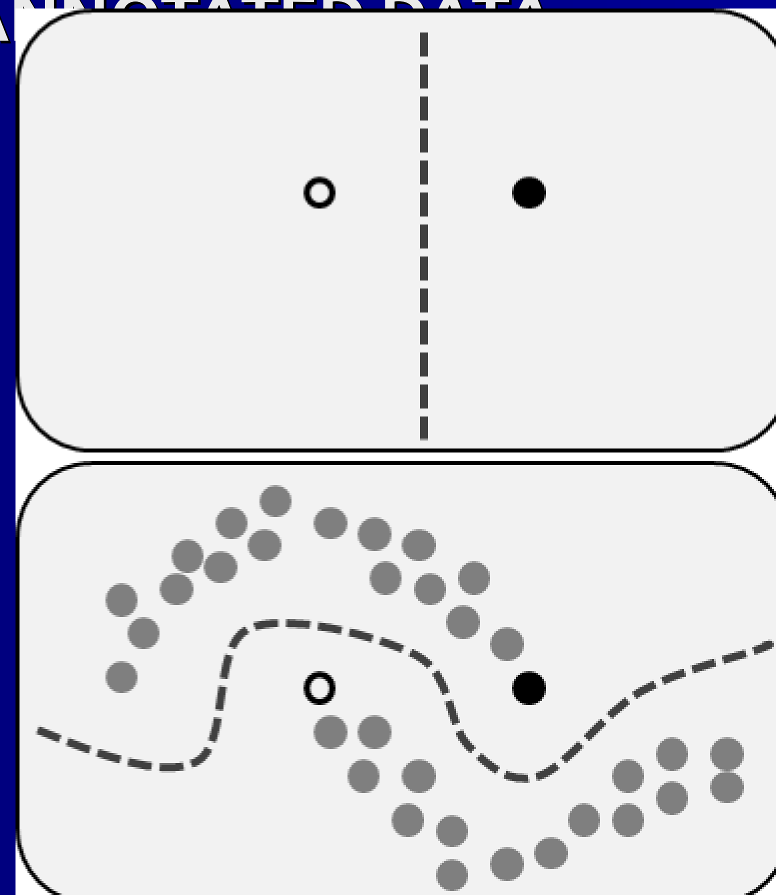
# SEMI SUPERVISED LEARNING

## A CRUCIAL TOOL FOR TEXT CLASSIFICATION WITH SCARCITY OF ANNOTATED DATA




| $X_1$ , $X_2$ , … , $X_n$ | C |
|---|---|
| a , b , … , b | + |
| b , b , … , a | - |
| a , a , … , b | - |
| b , a , … , b | + |
| a , b , … , a | - |
| b , a , … , a | - |
| a , a , … , b | + |
| a , b , … , a | ? |
| a , b , … , b | ? |
| b , a , … , b | ? |
| b , b , … , a | ? |
| a , a , … , b | ? |
| b , a , … , a | ? |
| a , a , … , a | ? |



Scholar

[HTML] **Text classification** method based on **self-training** and LDA topic models

M Pavlinek, V Podgorelec - Expert Systems with Applications, 2017 - Elsevier

… The contributions of this **study** are as follows … Since too many mislabeled instances can have a negative effect on the further **learning** process, especially in early … In **self-training** it often turns out that the most reliable instances are classified predominantly only in certain categories …

☆ Cited by 91 Related articles

[PDF] Email classification with co-training

S Kiritchenko, S Matwin - Proceedings of the 2001 conference of the …, 2001 - Citeseer

The main problems in text classification are lack of labeled data, as well as the cost of labeling the unlabeled data. We address these problems by exploring co-training-an algorithm that uses unlabeled data along with a few labeled examples to boost the performance of a classifier. We experiment with co-training on the email domain. Our results show that the performance of co-training depends on the learning algorithm it uses. In particular, Support Vector Machines significantly outperforms Naive Bayes on email …

☆ 99 Cited by 297 Related articles All 20 versions

Scholar

[PDF] Seeing stars when there aren't many stars: **Graph-based semi-supervised** learning for sentiment **categorization**

AB Goldberg, X Zhu - … first workshop on **graph based** methods for natural …, 2006 - aclweb.org

… Unlike tra- ditional **text categorization based** on topics, senti … Pang and Lee showed that **supervised** machine learn- ing techniques (**classification** and regression) work well for rating … We demonstrate that the answer is 'Yes.' Our approach is **graph-based semi-supervised** learning …

☆ Cited by 398 Related articles

Scholar

Text classification from labeled and unlabeled documents using EM

K Nigam, AK McCallum, S Thrun, T Mitchell - Machine learning, 2000 - Springer

This paper shows that the accuracy of learned text classifiers can be improved by augmenting a small number of labeled training documents with a large pool of unlabeled documents. This is important because in many text classification problems obtaining training labels is expensive, while large quantities of unlabeled documents are readily available.

☆ Cited by 3684 Related articles

Scholar

[HTML] **Text classification** method based on **self-training** and LDA topic models

M Pavlinek, V Podgorelec - Expert Systems with Applications, 2017 - Elsevier

… The contributions of this **study** are as follows … Since too many mislabeled instances can have a negative effect on the further **learning** process, especially in early … In **self-training** it often turns out that the most reliable instances are classified predominantly only in certain categories …

☆ Cited by 91 Related articles

# OUTLINE

- Semi-supervised learning SSL→ type of data

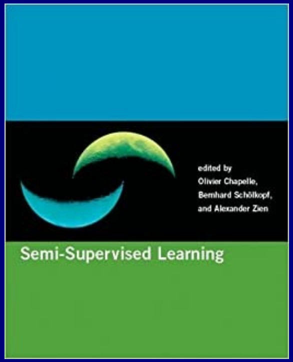
- “Learning with assumptions”
- Types of semi-supervised learning algorithms

- References and software

Machine Learning (2020) 109:373–440
https://doi.org/10.1007/s10994-019-05855-6

**A survey on semi-supervised learning**

Jesper E. van Engelen[1] · Holger H. Hoos[1,2]

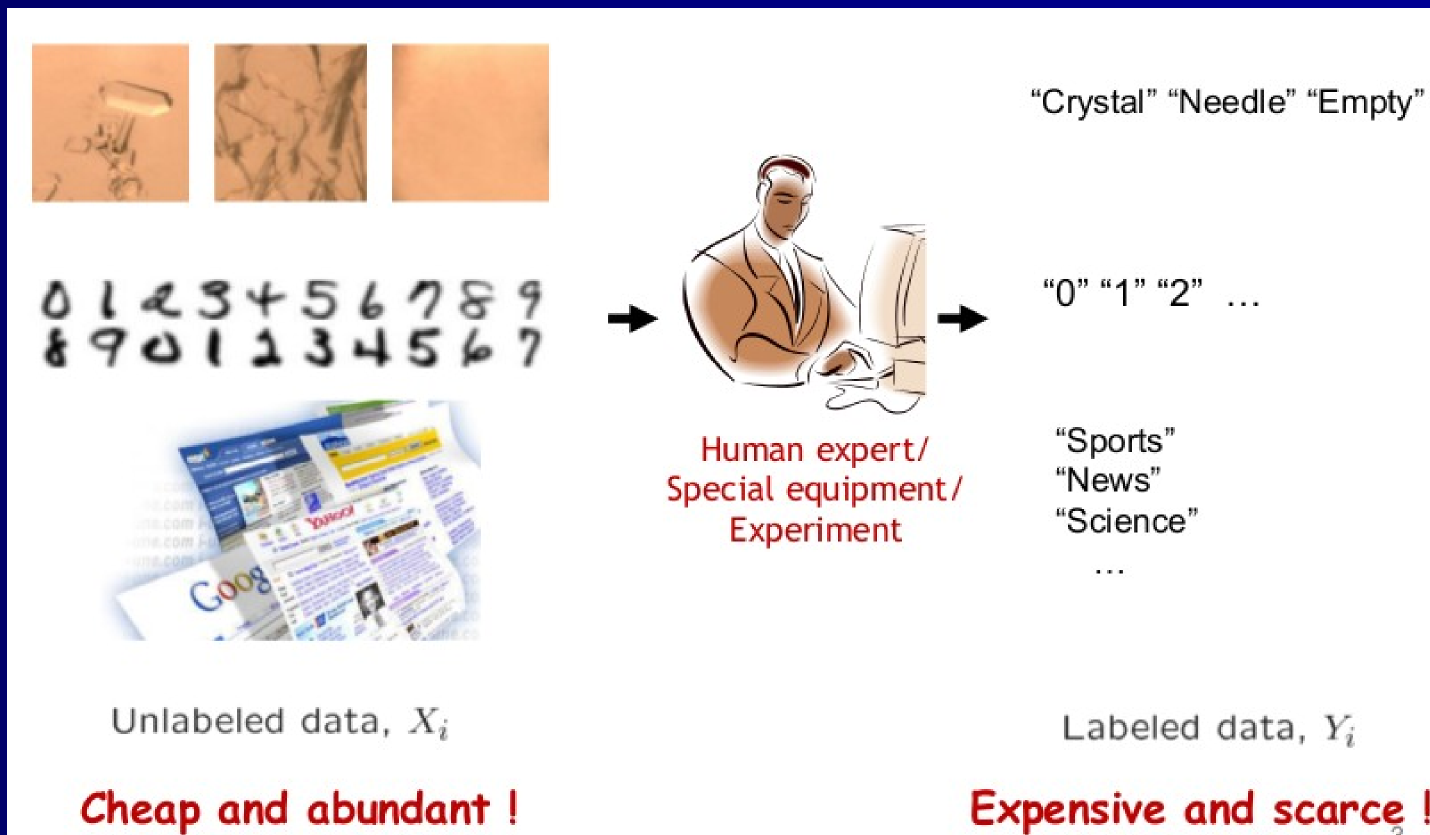



**RSSL: Semi-supervised Learning in R**

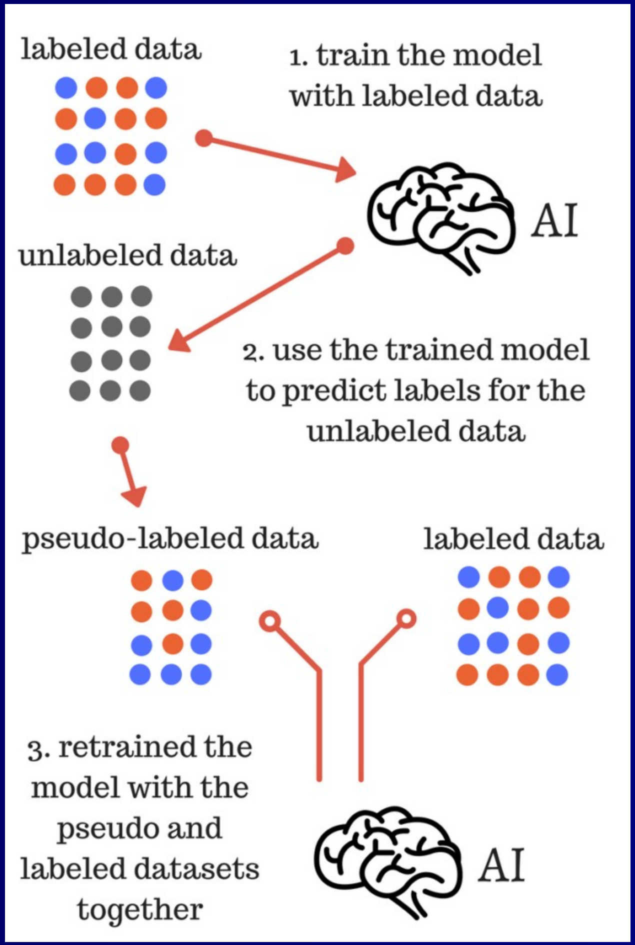
Jesse H. Krijthe[1,2]

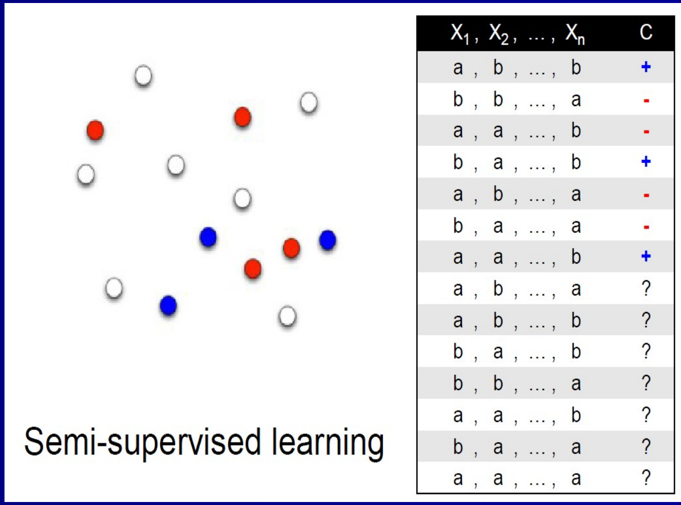
[1] Pattern Recognition Laboratory, Delft University of Technology
[2] Department of Molecular Epidemiology, Leiden University Medical Center
jkrijthe@gmail.com

# SEMI SUPERVISED LEARNING
## – DATA TYPE –



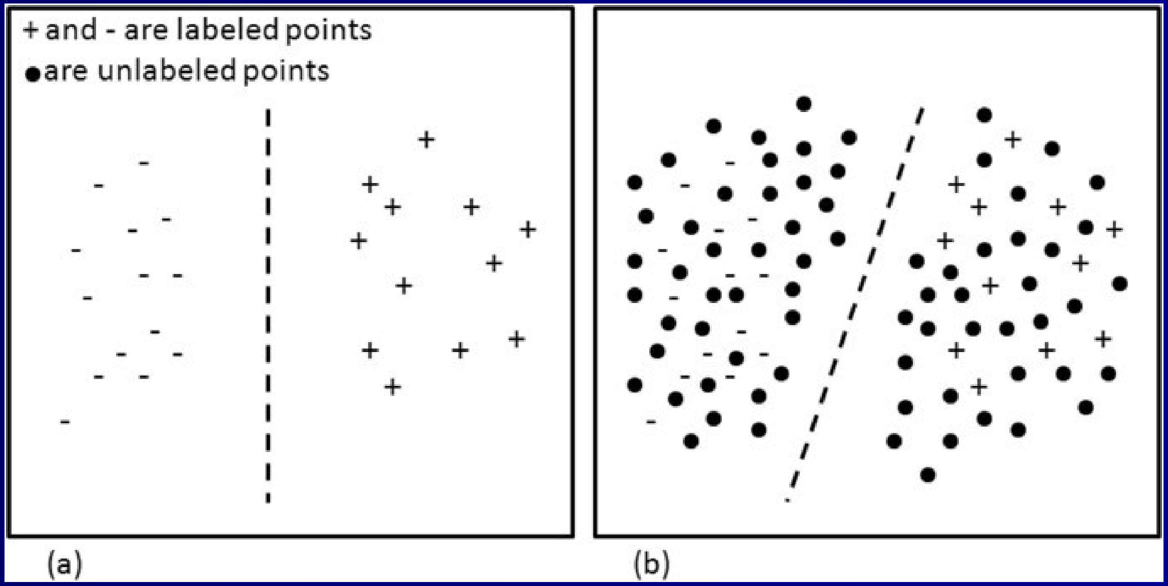
# SEMI SUPERVISED LEARNING
## – DATA TYPE –



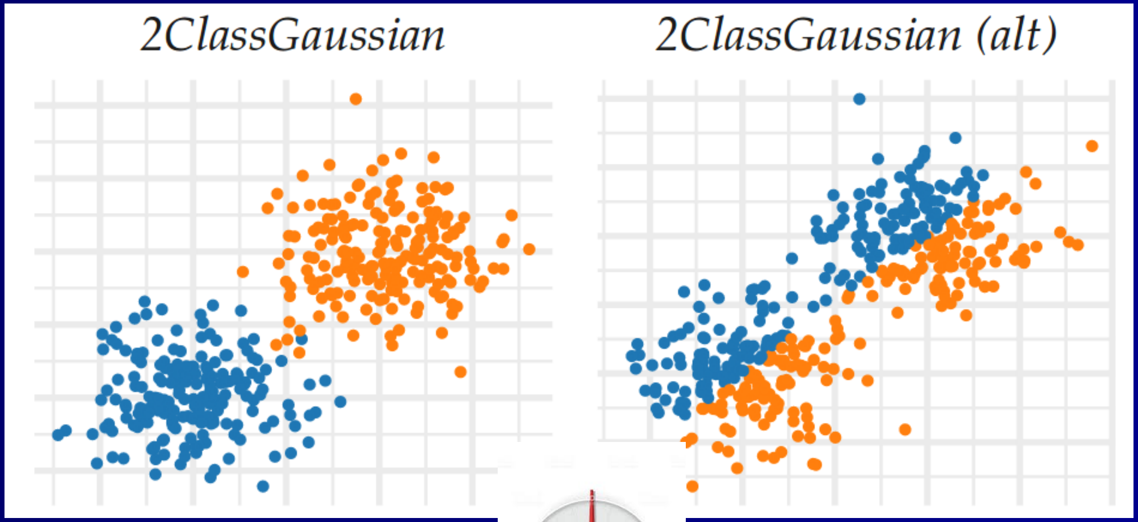
# SSL
# – LEARNING WITH ASSUMPTIONS –

- Adding unlabeled data → no guarantee to improve SL
- Does p(x) contain info about p(y|x)?
- Essential for SSL success



# SSL – LEARNING WITH ASSUMPTIONS –

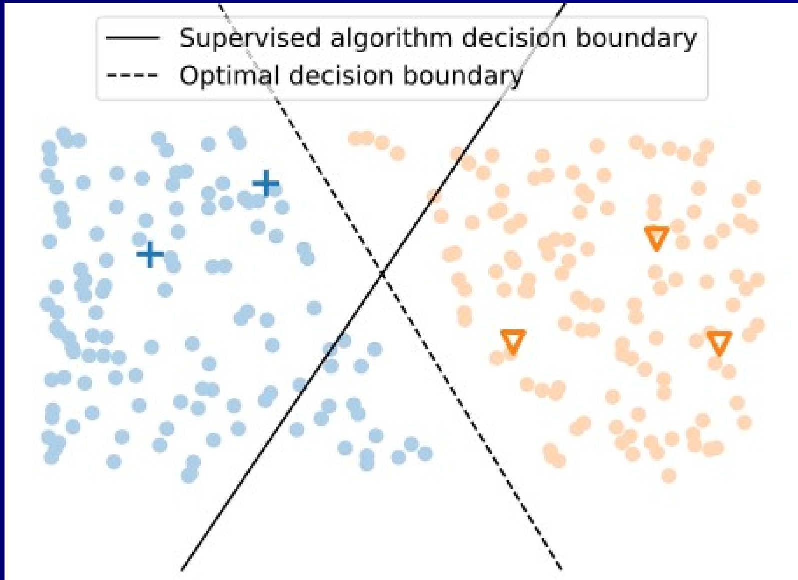
- Adding unlabeled data → no guarantee to improve SL
- Keypoint → Does *p(x)* contain info about *p(class|x)*?
- Essential for SSL success




# SSL
# – SMOOTHNESS ASSUMPTION –

- Two close points → should have same labels
- Transitively applied

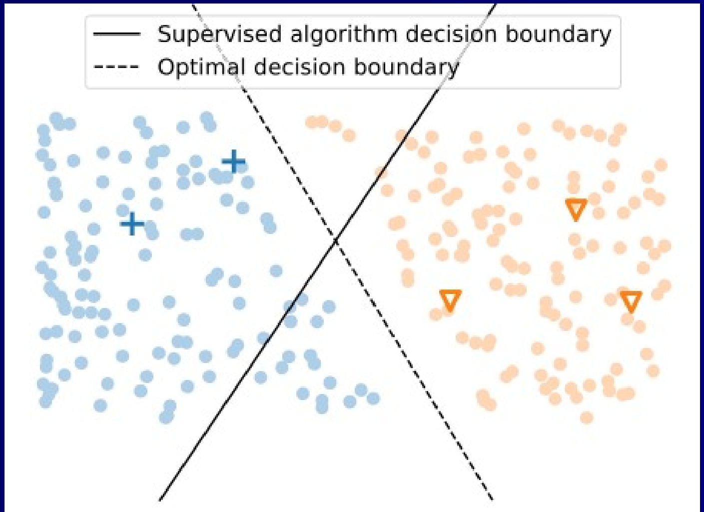
$$x_1 \sim x_2 + x_2 \sim x_3 \quad \rightarrow \quad x_1 \sim x_3$$



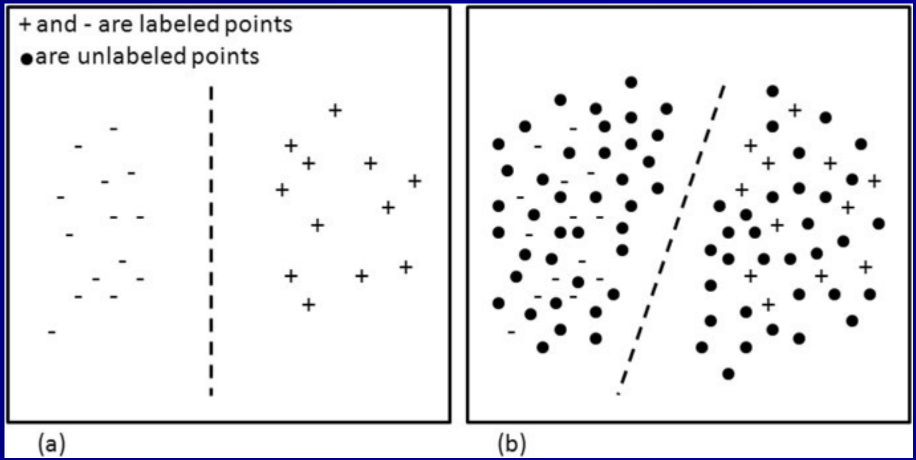

**(a)** Smoothness and low-density assumptions.

# SSL
# – LOW-DENSITY ASSUMPTION –

- Boundary → not cross high-density regions



(a) Smoothness and low-density assumptions.



# SSL
# – MANIFOLD ASSUMPTION –

- Manifold definition

In machine learning problems where the data can be represented in Euclidean space, the observed data points in the high-dimensional input space $\mathbb{R}^d$ are usually concentrated along lower-dimensional substructures. These substructures are known as *manifolds*: topological spaces that are locally Euclidean. For instance, when we consider a 3-dimensional input space where all points lie on the surface of a sphere, the data can be said to lie on a 2-dimensional

- Data dispersed in high-dimensions

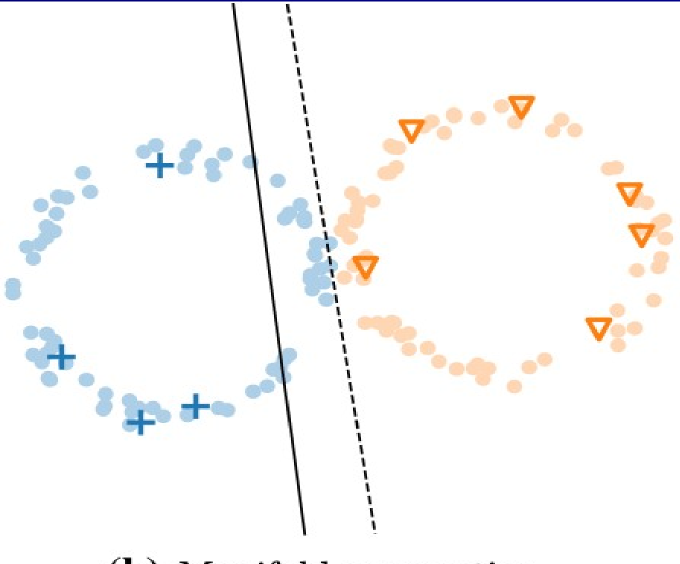
  BUT
- Concentrated in lower-dimensional structures

  → called "manifolds"

# SSL
# – MANIFOLD ASSUMPTION –

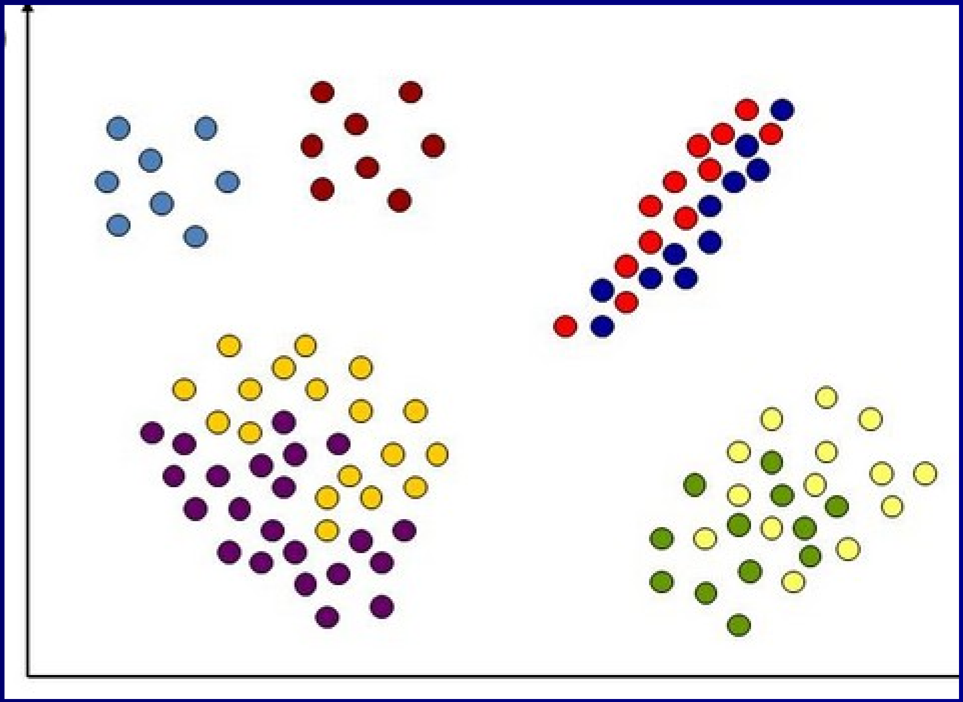
- Assumption → manifold structures exist in data
- Assumption → Points on the same manifold → same label

- Learn process → in low-dimension space


(b) Manifold assumption.

# SSL
# – CLUSTER ASSUMPTION –

- Does “generalize” previous assumptions? Yes...
- If points no meaningfully clustered → SSL no possible
- Does *p(x)* contain info about *p(class|x)*?



# SSL
# – MAIN TECHNIQUES –

- Self-training
- Co-training
- Clustering + Labeling
- Graph-based methods
- Gaussian model → EM for parameter learning

- instance $\mathbf{x}$, label $y$
- learner $f : \mathcal{X} \mapsto \mathcal{Y}$
- labeled data $(X_l, Y_l) = \{(x_{1:l}, y_{1:l})\}$
- unlabeled data $X_u = \{\mathbf{x}_{l+1:l+u}\}$, available during training. Usually $l \ll u$. Let $n = l + u$
- test data $\{(x_{n+1\ldots}, y_{n+1\ldots})\}$, not available during training

# SSL
# SELF -TRAINING

Scholar

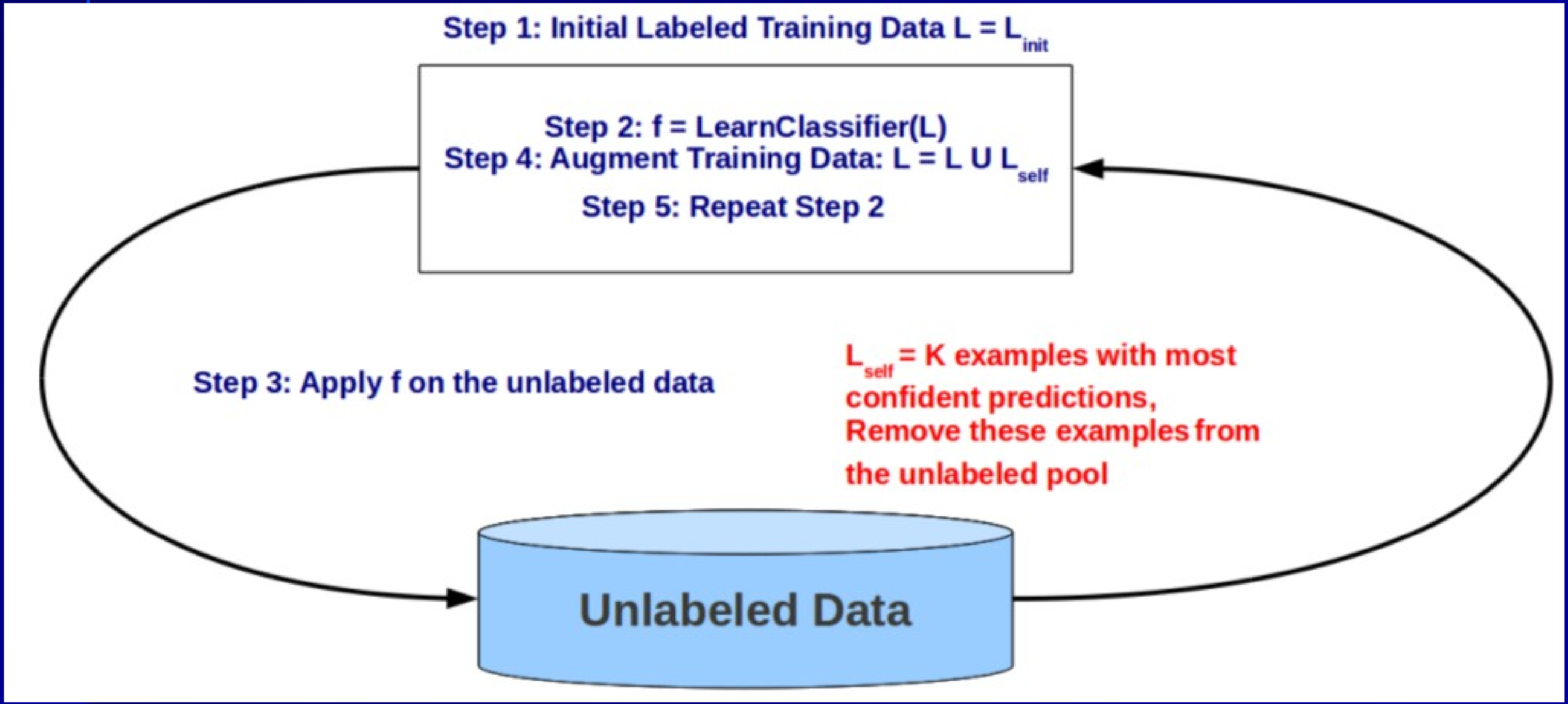
[HTML] **Text classification** method based on **self-training** and LDA topic models

M Pavlinek, V Podgorelec - Expert Systems with Applications, 2017 - Elsevier

… The contributions of this **study** are as follows … Since too many mislabeled instances can have a negative effect on the further **learning** process, especially in early … In **self-training** it often turns out that the most reliable instances are classified predominantly only in certain categories …

Cited by 91 Related articles

# SSL
# SELF -TRAINING



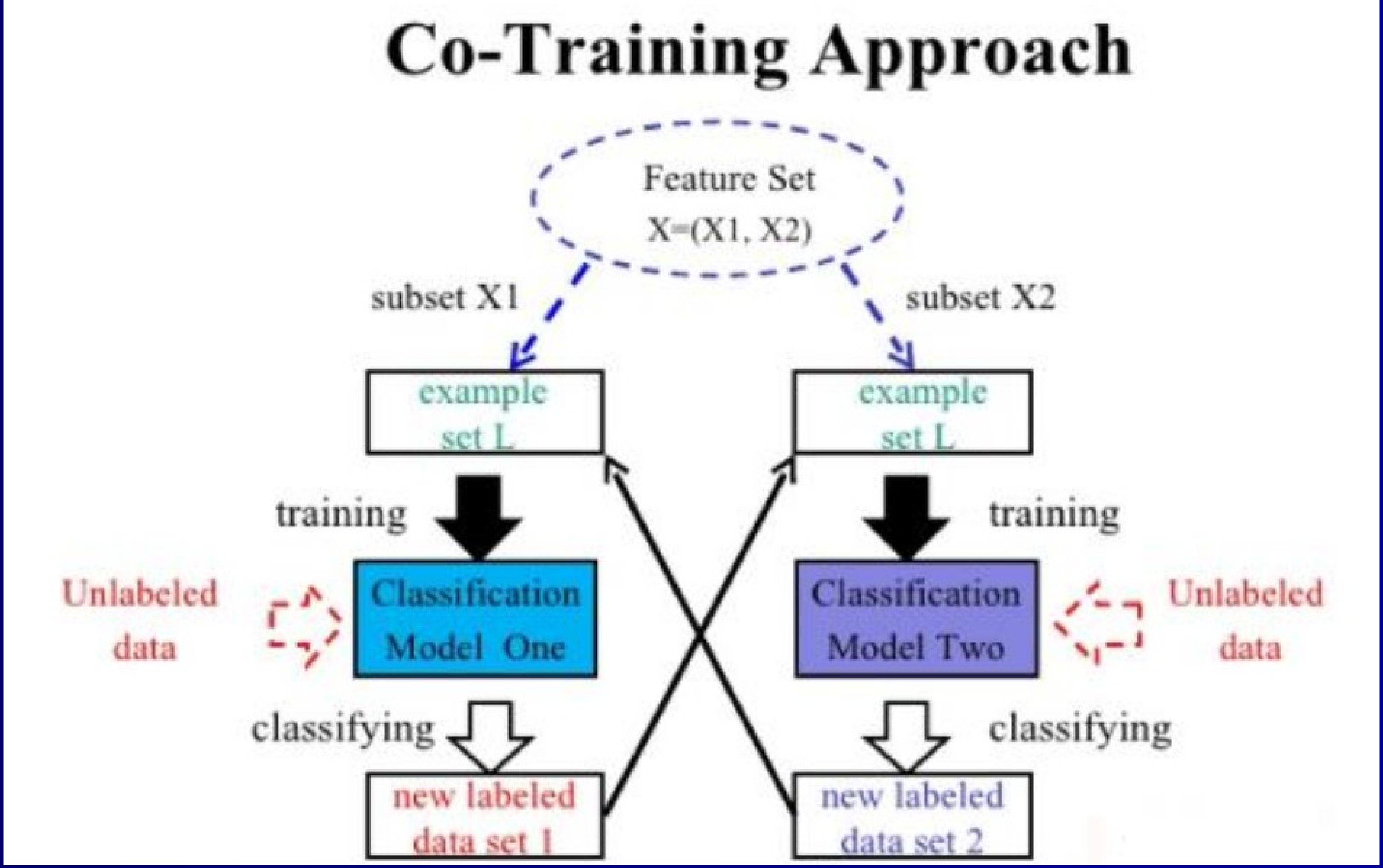
# SSL
# CO -TRAINING

[PDF] Email classification with co-training

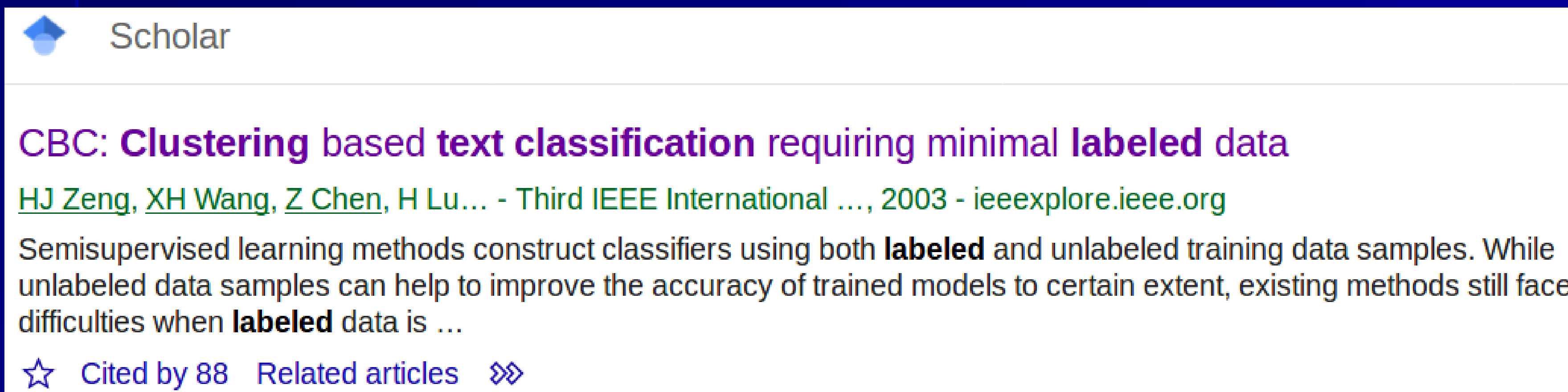
S Kiritchenko, S Matwin - Proceedings of the 2001 conference of the …, 2001 - Citeseer

The main problems in text classification are lack of labeled data, as well as the cost of labeling the unlabeled data. We address these problems by exploring co-training-an algorithm that uses unlabeled data along with a few labeled examples to boost the performance of a classifier. We experiment with co-training on the email domain. Our results show that the performance of co-training depends on the learning algorithm it uses. In particular, Support Vector Machines significantly outperforms Naive Bayes on email …

# SSL
# CO -TRAINING



# SSL
# – CLUSTERING + LABELING –



Scholar

CBC: **Clustering** based **text classification** requiring minimal **labeled** data

HJ Zeng, XH Wang, Z Chen, H Lu… - Third IEEE International …, 2003 - ieeexplore.ieee.org

Semisupervised learning methods construct classifiers using both **labeled** and unlabeled training data samples. While unlabeled data samples can help to improve the accuracy of trained models to certain extent, existing methods still face difficulties when **labeled** data is …

Cited by 88 Related articles

# SSL
# – CLUSTERING + LABELING –

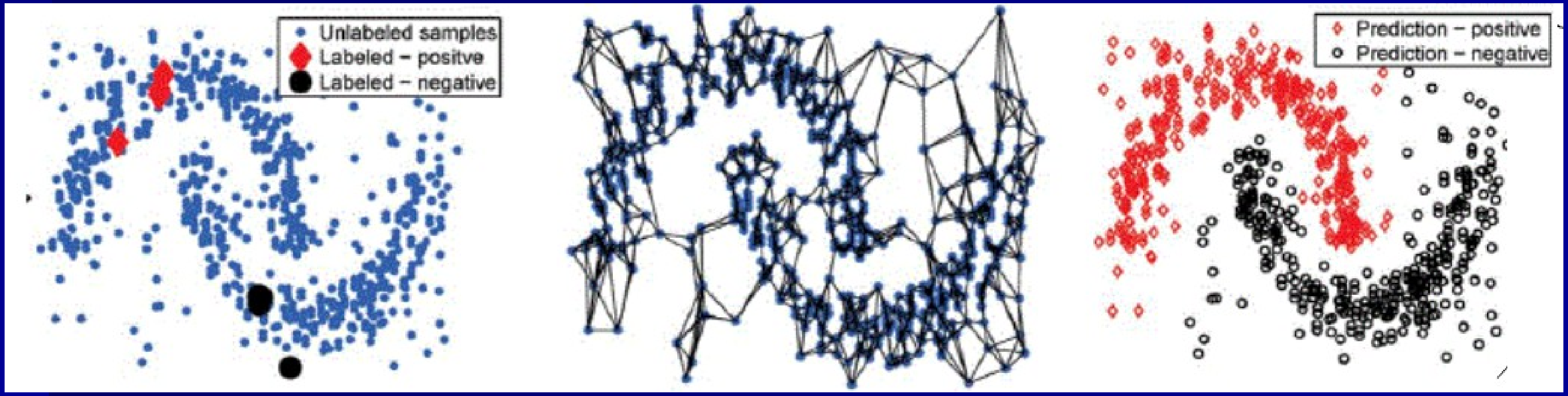
**Input**: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l), \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}$,
a clustering algorithm $\mathcal{A}$, a supervised learning algorithm $\mathcal{L}$

1. Cluster $\mathbf{x}_1, \ldots, \mathbf{x}_{l+u}$ using $\mathcal{A}$.
2. For each cluster, let $S$ be the labeled instances in it:
3. Learn a supervised predictor from $S$: $f_S = \mathcal{L}(S)$.
4. Apply $f_S$ to all unlabeled instances in this cluster.

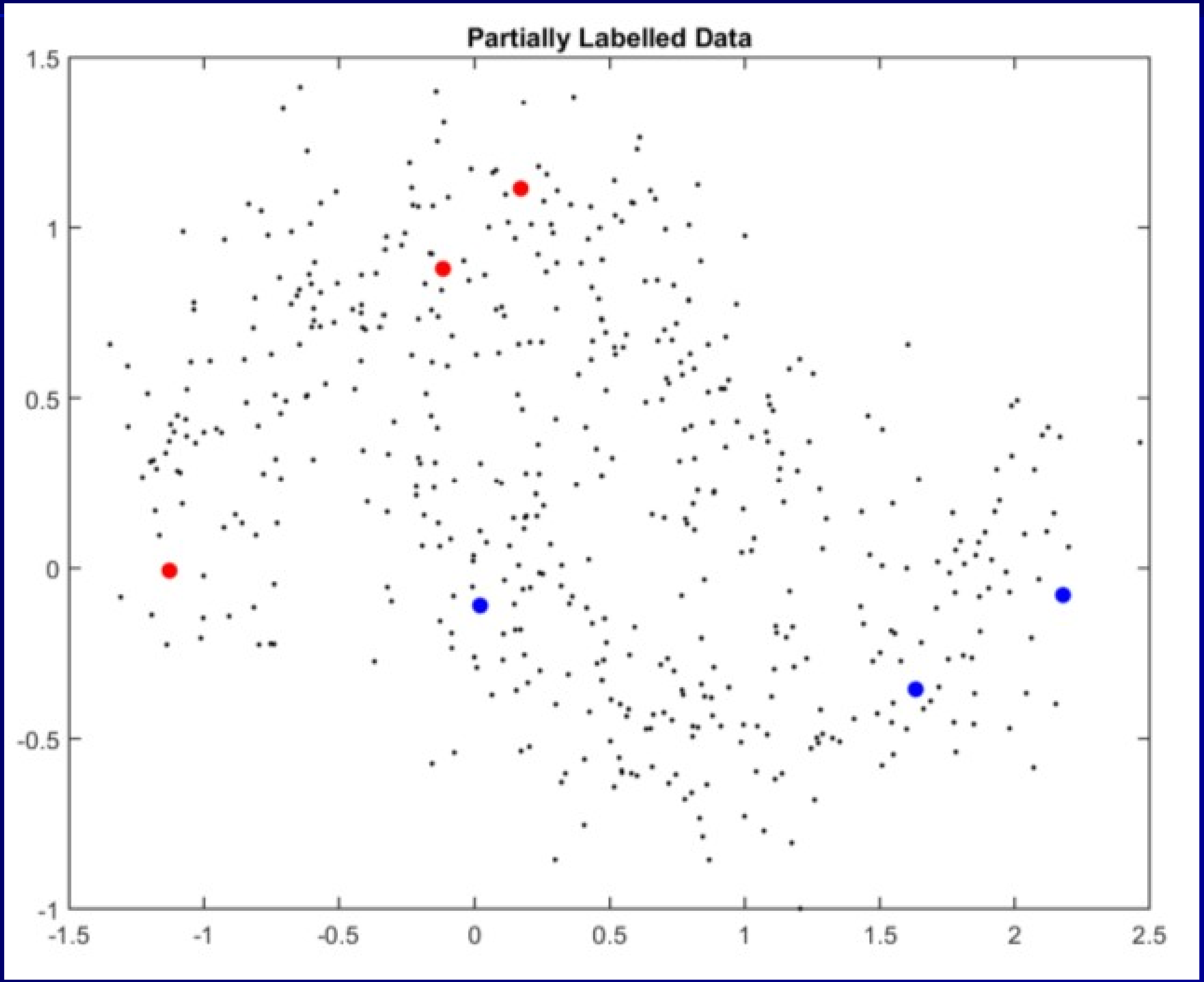
**Output**: labels on unlabeled data $y_{l+1}, \ldots, y_{l+u}$.

# SSL
# – GRAPH-BASED METHODS –

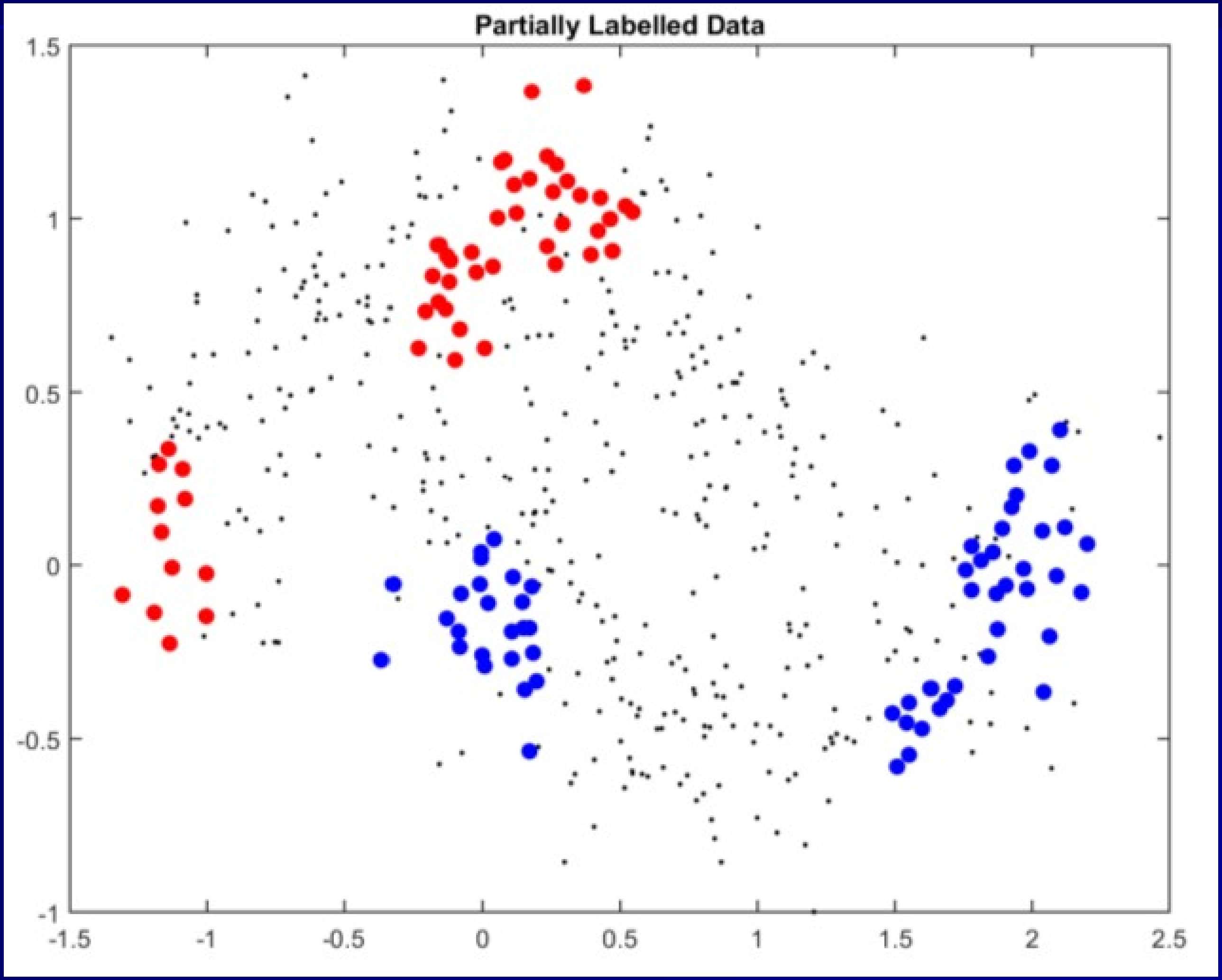
- Construct a graph with
  - nodes → instances
  - arcs → connecting "close" instances
- Propagate labels from labeled → to unlabeled
- {Smoothness + low-density region} assumptions



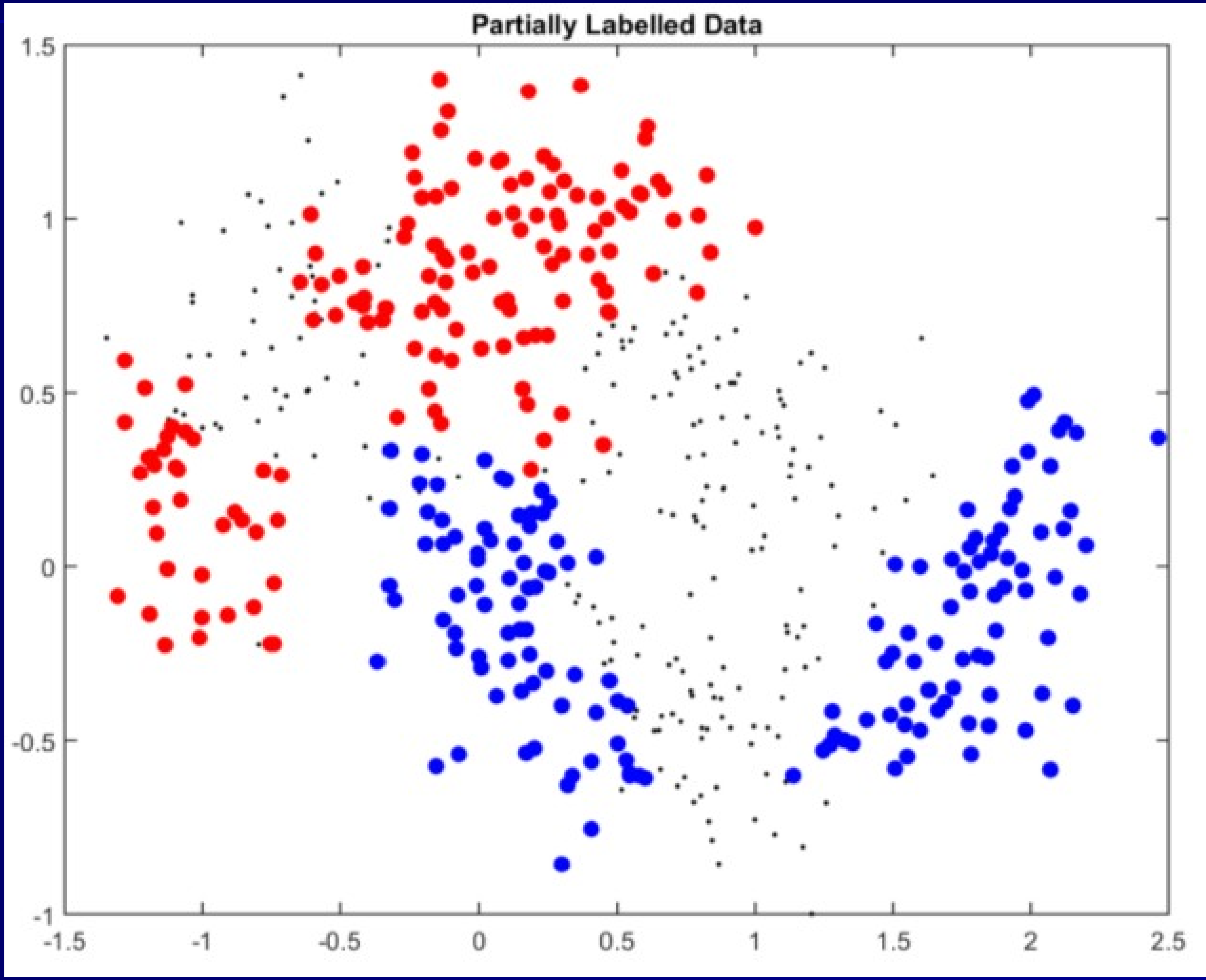
# SSL
# – GRAPH-BASED METHODS –



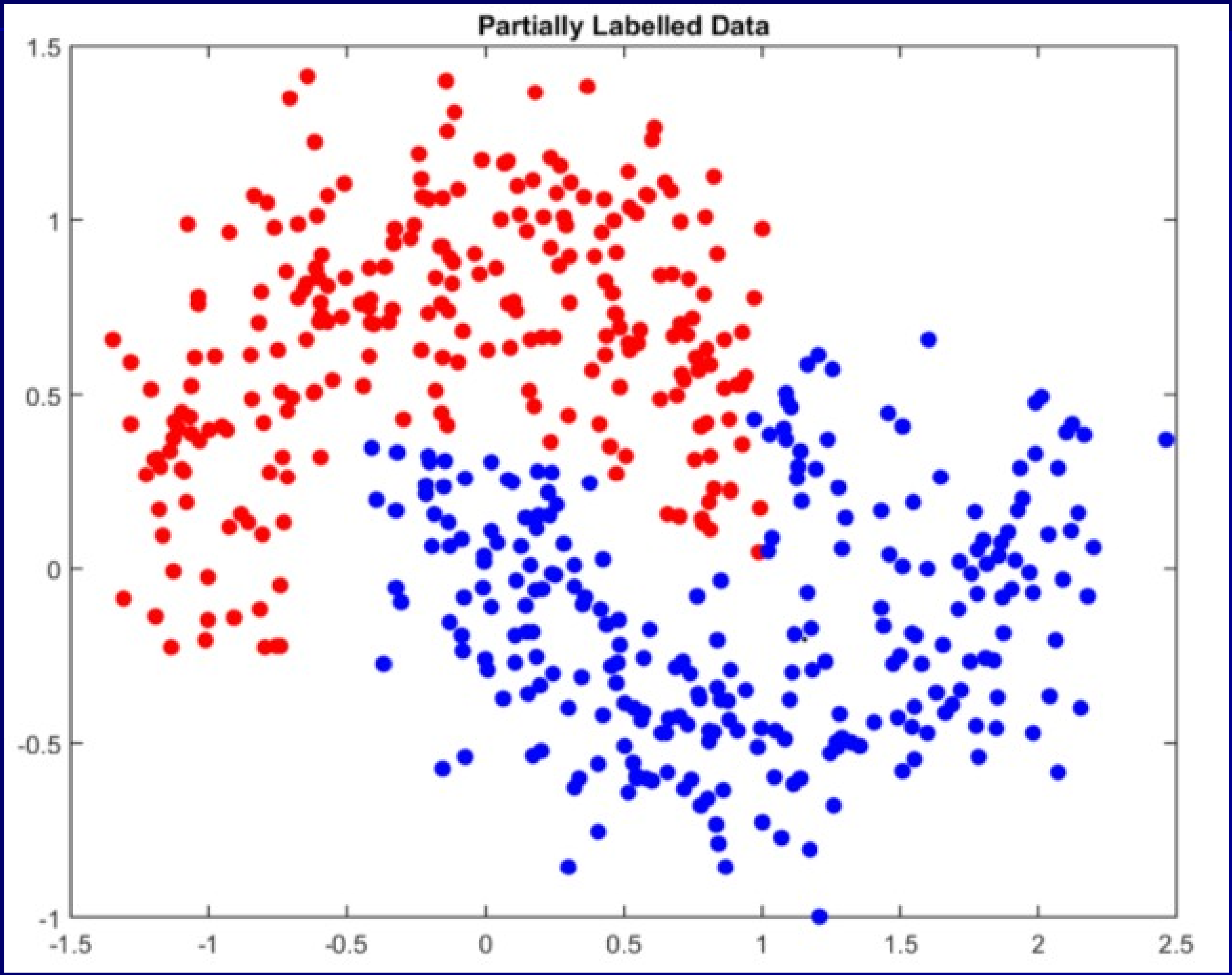
# SSL
# – GRAPH-BASED METHODS –



# SSL
# – GRAPH-BASED METHODS –



# SSL
# – GRAPH-BASED METHODS –



# SSL
# – GRAPH-BASED METHODS –

Scholar

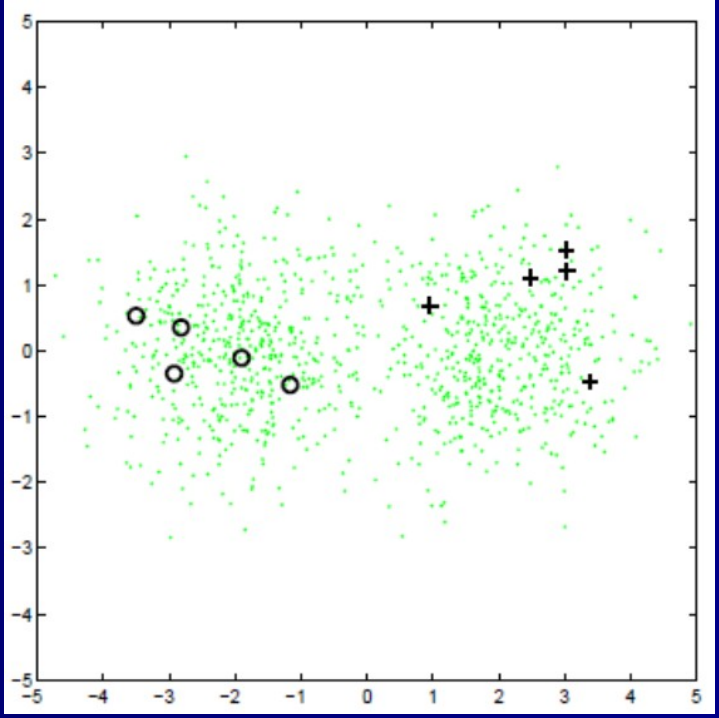
[PDF] Seeing stars when there aren't many stars: **Graph-based semi-supervised** learning for sentiment **categorization**

AB Goldberg, X Zhu - … first workshop on **graph based** methods for natural …, 2006 - aclweb.org

… Unlike tra- ditional **text categorization based** on topics, senti … Pang and Lee showed that **supervised** machine learn- ing techniques (**classification** and regression) work well for rating … We demonstrate that the answer is 'Yes.' Our approach is **graph-based semi-supervised** learning …

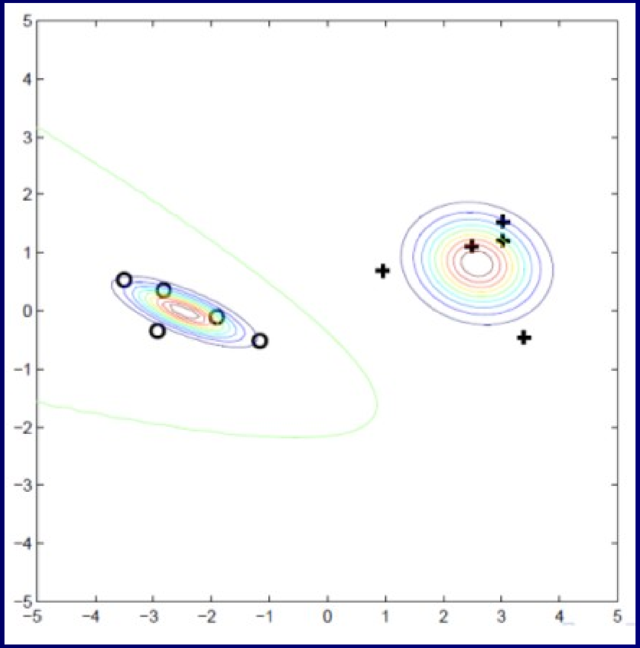
# SSL
# – GAUSSIAN MODEL –

- Assuming Gaussian mixture model for labeled data
- Assumption → needed to be correct!



# SSL
# – GAUSSIAN MODEL –

- Assuming Gaussian mixture model for labeled data
- Assumption → needed to be correct!




Model parameters: $\theta = \{w_1, w_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$

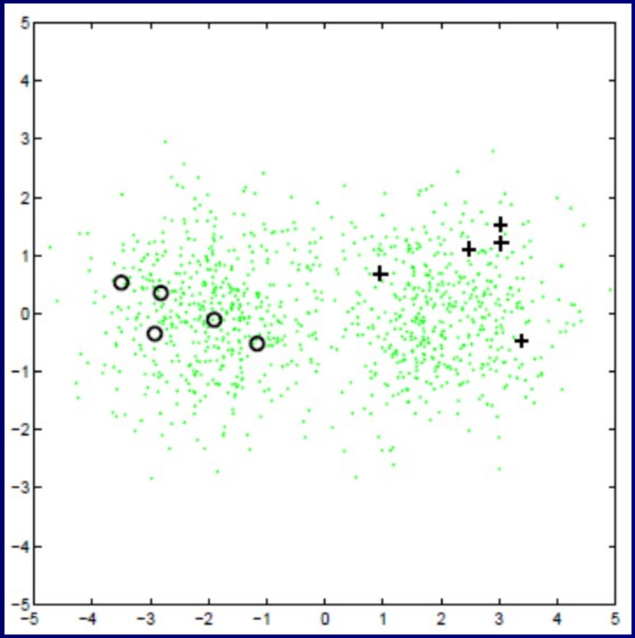
The GMM:

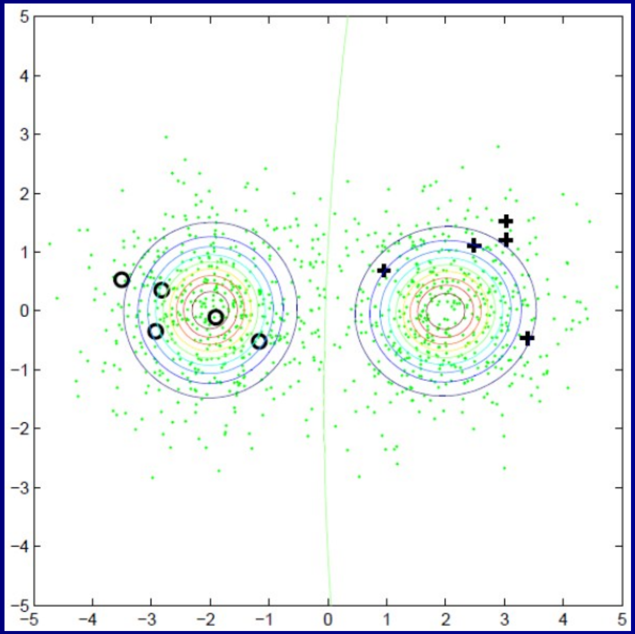
$$\begin{aligned} p(x, y|\theta) &= p(y|\theta)p(x|y,\theta) \\ &= w_y \mathcal{N}(x; \mu_y, \Sigma_y) \end{aligned}$$

Classification: $p(y|x,\theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)} \gtrless 1/2$

# SSL
# – GAUSSIAN MODEL –

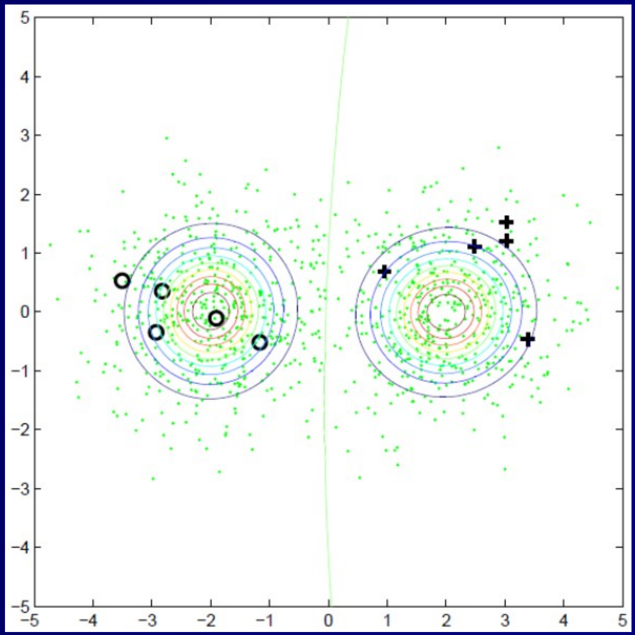
- Adding unlabeled data
- Holding Gaussian mixture assumption



# SSL
# – GAUSSIAN MODEL –

- How to calculate model parameters with unlabeled data?
- Expectation-Maximization algorithm - EM




Model parameters: $\theta = \{w_1, w_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$
The GMM:

$$\begin{aligned} p(x, y|\theta) &= p(y|\theta)p(x|y, \theta) \\ &= w_y \mathcal{N}(x; \mu_y, \Sigma_y) \end{aligned}$$

Classification: $p(y|x, \theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)} \gtrless 1/2$

# SSL
# – GAUSSIAN MODEL –

- How to calculate model parameters with unlabeled data?
- Expectation-Maximization algorithm - EM

Scholar

Text classification from labeled and unlabeled documents using EM

K Nigam, AK McCallum, S Thrun, T Mitchell - Machine learning, 2000 - Springer

This paper shows that the accuracy of learned text classifiers can be improved by augmenting a small number of labeled training documents with a large pool of unlabeled documents. This is important because in many text classification problems obtaining training labels is expensive, while large quantities of unlabeled documents are readily available.

Cited by 3684 Related articles

# SSL
# – GAUSSIAN MODEL –

- EM for parameter learning → Gaussian mixture model
- Model parameters: $\theta = \{w_1, w_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$

1. Start from MLE $\theta = \{w, \mu, \Sigma\}_{1:2}$ on $(X_l, Y_l)$,
   - $w_c$=proportion of class $c$
   - $\mu_c$=sample mean of class $c$
   - $\Sigma_c$=sample cov of class $c$

   repeat:

2. The E-step: compute the expected label $p(y|x,\theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)}$ for all $x \in X_u$
   - label $p(y=1|x,\theta)$-fraction of $x$ with class 1
   - label $p(y=2|x,\theta)$-fraction of $x$ with class 2

3. The M-step: update MLE $\theta$ with (now labeled) $X_u$

# SSL
# – GAUSSIAN MODEL –

- EM for parameter learning → Gaussian mixture model
- Model parameters: $\theta = \{w_1, w_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$

**Maximum Likelihood** from **Incomplete Data Via** the ***EM* Algorithm**
AP Dempster, NM Laird… - Journal of the Royal …, 1977 - Wiley Online Library
A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched …
Citado por 60207   Artículos relacionados   Las 67 versiones

1. Start from MLE $\theta = \{w, \mu, \Sigma\}_{1:2}$ on $(X_l, Y_l)$,
   - $w_c$=proportion of class $c$
   - $\mu_c$=sample mean of class $c$
   - $\Sigma_c$=sample cov of class $c$

   repeat:
2. The E-step: compute the expected label $p(y|x,\theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)}$ for all $x \in X_u$
   - label $p(y=1|x,\theta)$-fraction of $x$ with class 1
   - label $p(y=2|x,\theta)$-fraction of $x$ with class 2
3. The M-step: update MLE $\theta$ with (now labeled) $X_u$

# SSL
# – SOFTWARE –

## RSSL: Semi-supervised Learning in R

Jesse H. Krijthe[1,2]

[1] Pattern Recognition Laboratory, Delft University of Technology
[2] Department of Molecular Epidemiology, Leiden University Medical Center
jkrijthe@gmail.com

```r
set.seed(1)
df <- generate2ClassGaussian(2000,d=2,var=0.6,expected=TRUE)

classifiers <- list("LS"=function(X,y,X_u,y_u) {
 LeastSquaresClassifier(X,y,lambda=0)},
  "Self"=function(X,y,X_u,y_u) {
    SelfLearning(X,y,X_u,LeastSquaresClassifier)})

measures <- list("Accuracy" =  measure_accuracy)

lc1 <- LearningCurveSSL(as.matrix(df[,1:2]),
                        df$Class,classifiers=classifiers, measures=measures,
                        type="fraction", test_fraction=0.5,repeats=3)
plot(lc1)

iris = read.csv("iris.csv", header=TRUE, sep=",")
lc2 = LearningCurveSSL(as.matrix(iris[1:4]), iris$variety,
                       classifiers=classifiers, measures=measures,
                       type="fraction", fracs = seq(0.1,0.8,0.1),
                       test_fraction=0.5, repeats=3)
plot(lc2)
```

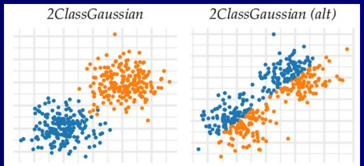



**Table 1.** Overview of classifiers available in RSSL

| Classifier | R | Interface | Port | Reference |
|---|---|---|---|---|
| **(Kernel) Least Squares Classifier** | ✓ | | | [8] |
| Implicitly Constrained | ✓ | | | [13] |
| Implicitly Constrained Projection | ✓ | | | [12] |
| Laplacian Regularized | ✓ | | | [1] |
| Updated Second Moment | ✓ | | | [23] |
| Self-learning | ✓ | | | [20] |
| Optimistic / "Expectation Maximization" | ✓ | | | [11] |
| **Linear Discriminant Analysis** | ✓ | | | [25] |
| Expectation Maximization | ✓ | | | [5] |
| Implicitly Constrained | ✓ | | | [10] |
| Maximum Constrastive Pessimistic | | | ✓ | [18] |
| Moment Constrained | ✓ | | | [17] |
| Self-learning | ✓ | | | [20] |
| **Nearest Mean Classifier** | ✓ | | | [25] |
| Expectation Maximization | ✓ | | | [5] |
| Moment Constrained | ✓ | | | [16] |
| Self-learning | ✓ | | | [20] |
| **Support Vector Machine** | ✓ | | | |
| SVMlin | | ✓ | | [24] |
| WellSVM | | | ✓ | [14] |
| S4VM | | | ✓ | [15] |
| Transductive SVM (Convex Concave Procedure) | ✓ | | | [9,3] |
| Laplacian SVM | ✓ | | | [1] |
| Self-learning | ✓ | | | [20] |
| **Logistic Regression** | ✓ | | | |
| Entropy Regularized Logistic Regression | ✓ | | | [7] |
| Self-learning | ✓ | | | [20] |
| Harmonic Energy Minimization | ✓ | | | [27] |

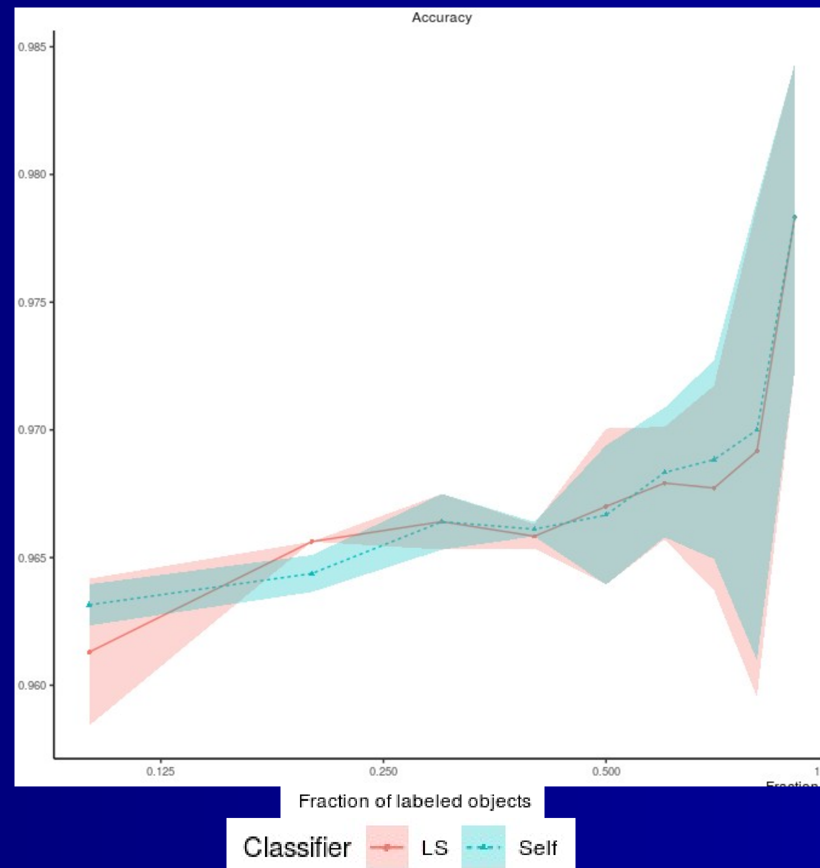
# SSL
# – SOFTWARE –

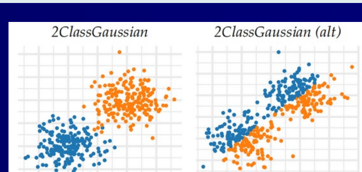
**RSSL: Semi-supervised Learning in R**

Jesse H. Krijthe[1,2]

[1] Pattern Recognition Laboratory, Delft University of Technology
[2] Department of Molecular Epidemiology, Leiden University Medical Center
jkrijthe@gmail.com

```
set.seed(1)
df <- generate2ClassGaussian(2000,d=2,var=0.6,expected=TRUE)

classifiers <- list("LS"=function(X,y,X_u,y_u) {
 LeastSquaresClassifier(X,y,lambda=0)},
  "Self"=function(X,y,X_u,y_u) {
    SelfLearning(X,y,X_u,LeastSquaresClassifier)})

measures <- list("Accuracy" =  measure_accuracy)

lc1 <- LearningCurveSSL(as.matrix(df[,1:2]),
                        df$Class,classifiers=classifiers, measures=measures,
                        type="fraction", test_fraction=0.5,repeats=3)
plot(lc1)

iris = read.csv("iris.csv", header=TRUE, sep=",")
lc2 = LearningCurveSSL(as.matrix(iris[1:4]), iris$variety,
                       classifiers=classifiers, measures=measures,
                       type="fraction", fracs = seq(0.1,0.8,0.1),
                       test_fraction=0.5, repeats=3)
plot(lc2)
```



# SSL
# – PROPOSED EXERCISE –

- RSSL package → its R-vignette [github] [arXiv]

- Choose an artificial dataset offered by the package
- Check the variety of artificial datasets and their shapes
- Functions to generate artificial datasets

- Choose and load a real dataset → spambase.csv, umic-sa.csv, epinions.csv…
- Link to datasets
- LearningCurveSSL( ) function
- Understand associated parameters
- Change parameter values and check the result
- Change base classifiers
- Choose different SSL strategies
- Type of measures-metrics offered by RSSL
- Does the SSL strategy improve SL?