

Efficient Unsupervised Mining from Noisy Data Sets: Application to Clustering Co-occurrence Data

Hiroshi Mamitsuka*

Abstract

We propose a new data mining method for efficient unsupervised learning from noisy data sets. Our proposed method uses a learning algorithm for a stochastic (probabilistic) model as a component subroutine and repeats two steps: selectively sampling examples and running its subroutine using the selected examples. In sampling examples, we utilize the predictions, for each example in a given training data set, calculated by all previously trained models. We empirically evaluate the effectiveness of the proposed method by applying it to the problem of clustering co-occurrence data. The performance of our method was compared with those of a random-and-iterative sampling strategy and of a method using multiple copies of our component learning algorithm, which has been widely used for clustering co-occurrence data. Our results show that the performance of our method compares favorably against the two other methods, in terms of the final prediction accuracies achieved. From the experiments on synthetic data sets, we found that the advantage of our method becomes more pronounced for larger noise levels. The effectiveness of our method was confirmed by the experiments on actual protein-protein interaction data sets.

1 Introduction

We propose a new data mining method for efficient unsupervised learning from noisy data sets. The data set we consider here contains experimental noises or errors (pseudo-positive examples) at a certain high ratio, and each example in the set is unlabeled. One typical example is a data set of protein-protein interactions (PPIs) that has been increasingly obtained by modern high-throughput experimental techniques [4]. The problem is how we precisely perform unsupervised learning from such a data set. For the problem, we propose a novel method that repeats a procedure consisting of two steps: selectively sampling a set of examples, in which each example is expected to be a positive (non-noise) example, and training a component unsupervised learning algorithm with the set. Assuming a stochastic (probabilistic) model as a component learner of our iterative unsupervised procedure, we use the predictions, for each example, made by all previously obtained learners to judge whether each example is a positive or pseudo-positive example (noise).

In the literature of data sampling, typical sampling

methods (e.g. [6]) proposed to date perform accumulating repeatedly selected examples. They store the obtained examples, and once an example is stored, we have to continue using it as a training example even if it is a pseudo-positive example. In addition, there is an inevitable upper limit in the number of examples that can be stored. Our approach stores repeatedly obtained learners (or prediction results) instead of the selected examples, and training examples to build the learner are refreshed in each repetition. This property of our methodology is strongly related to an approach known as ‘Sequential multi-subset learning with model-guided instance selection,’ as described, for example, by Provost and Kolluri in their review article [8]. A number of methods proposed to date in the literature belong to this category, including Boosting [2]. We emphasize that our method is proposed for unsupervised learning, and to the best of our knowledge, no unsupervised learning method has been proposed yet in this category.

The purpose of this paper is to empirically examine how well our method works in the current context of efficient unsupervised mining from noisy data sets and to characterize under what conditions it works well. In our experiments, we focus on co-occurrence data sets, which includes a PPI data set as well as a variety of real-world data sets, such as co-occurred words in texts and purchasing records of product pairs, etc.

In the experiment on synthetic data sets, for low noise levels, we found no statistically significant difference between our method and a random strategy, in terms of the accuracy of discriminating noises. When we raised the noise level, we found that the statistically significant level by which our method out-performed the other methods became drastically larger. We further used a real PPI data set, which is inevitably noisy. The result shows that our method provides a larger average log-likelihood for test examples than those of the two other methods, demonstrating its effectiveness on a noisy real-world application.

2 The Learning/Mining Methods

2.1 Proposed Method

We describe the mining/learning method we propose, which we call Ssul,

*Institute for Chemical Research, Kyoto University, Gokasho, Uji 611-0011 Japan, E-mail: mami@kuicr.kyoto-u.ac.jp

Input: A set of given examples: S
Component learning algorithm: A
Number of iterations: T
Number of times running A at each iteration: J
Number of examples selected at each iteration: N
Number of noise candidates at each iteration: N_n
Initialization: 0. Randomly choose N examples as S_0
For $t = 0, \dots, T - 1$
 For $i = 1, \dots, J$
 1. Run A on S_t with random initial parameter values and obtain a trained model.
 2. For each $x \in S$, compute its likelihood $L_t^i(x)$ using the trained model.
 3. For each $x \in S$, compute average $\bar{L}(x)$ by $\frac{\sum_{i=1}^J L_t^i(x)}{J \times (t+1)}$.
 4. Select $\langle x_1^*, \dots, x_{N_n}^* \rangle$, whose $\bar{L}(x_1^*), \dots, \bar{L}(x_{N_n}^*)$ are smaller than those of others, and remove them from S as $S'_{t+1} = \text{remove}(S, \langle x_1^*, \dots, x_{N_n}^* \rangle)$.
 5. Randomly choose N examples from S'_{t+1} as S_{t+1} .
Output: Output all trained component models.

Figure 1: Algorithm: Selective sampling for unsupervised learning (Ssul).

standing for ‘selective sampling for unsupervised learning.’ The procedure uses an arbitrary learning algorithm of a stochastic model as a subroutine and works roughly as follows (the pseudocode is shown in Fig. 1).

At an initialization stage, it randomly samples N examples from the whole given data set as an initial data set to train a component unsupervised learning algorithm (line 0). At each stage of the iteration, it first repeats the following two steps: 1) It applies a component learning algorithm to the current training data set with different initial parameter values and obtains a trained stochastic model (line 1). 2) It computes the likelihood for each training example x using the model (line 2). It then selects N_n examples that are regarded as pseudo-positive examples, i.e. noises, from the models and produces a new data set that does not contain the examples. More precisely, for each example, it averages the likelihoods previously obtained (line 3) and selects N_n examples whose computed likelihoods are the smallest. It removes these examples from the whole training data set (line 4) and randomly chooses N examples from the remaining examples as a new training data set (line 5). This procedure is repeated T times. The final output is given by all component stochastic models obtained through the iteration. The prediction of the likelihood for an arbitrary given test example is done by averaging over the likelihoods predicted by all models given as the final output.

2.2 Hofmann’s Aspect Model

2.2.1 Model Representation Here we briefly explain Hofmann’s aspect model (AM for short) [3], which we used as a component model in our experiments. AM is a latent variable model that performs soft clustering, i.e. probabilistically grouping similar examples.

To describe AM formally, we need some notation. We denote a variable by a capitalized letter, e.g. X , and the value of a corresponding variable by that same letter in lower case, e.g. x . Now, let U and V be random observable variables taking on values u_1, \dots, u_L and v_1, \dots, v_M , respectively. Let Z be a discrete-valued latent variable taking on values z_1, \dots, z_K , each of which corresponds to a cluster.

An AM for U and V with K clusters has the form:

$$p(u, v; \theta) = \sum_k p(z_k; \theta) p(u|z_k; \theta) p(v|z_k; \theta)$$

2.2.2 Learning the Model We now describe a method for learning the parameters of AM when the number of clusters K and training data D are given. A possible criterion for this purpose is maximum likelihood (ML), in which parameter values are obtained to maximize the likelihood (log-likelihood) of the training data: $\theta^{ML} = \arg \max_{\theta} \log p(D; \theta)$, where $\log p(D; \theta) = \sum_i \sum_j n(i, j) \log \sum_k p(z_k; \theta) p(u_i|z_k; \theta) p(v_j|z_k; \theta)$, and $n(i, j)$ is the number of pairs of two values u_i and v_j in a given training data set. We employ a general scheme, called EM (Expectation-Maximization) algorithm [1], to obtain the ML parameter estimates of AM. The algorithm starts with initial parameter values and iterates both an expectation step (E-step) and a maximization step (M-step) alternately until a certain convergence criterion is satisfied.

In E-step, we estimate the latent value using the complete data log-likelihood:

$$p(z_k|u_i, v_j; \theta) = \frac{p(z_k; \theta) p(u_i|z_k; \theta) p(v_j|z_k; \theta)}{\sum_k p(z_k; \theta) p(u_i|z_k; \theta) p(v_j|z_k; \theta)}$$

In M-step, we take a corresponding summation over $p(z_k|u_i, v_j; \theta)$:

$$\theta_{z_k} \propto \sum_i \sum_j n(i, j) p(z_k|u_i, v_j; \theta_{old})$$

$$\theta_{u_i|z_k} \propto \sum_j n(i, j) p(z_k|u_i, v_j; \theta_{old})$$

$$\theta_{v_j|z_k} \propto \sum_i n(i, j) p(z_k|u_i, v_j; \theta_{old})$$

3 Empirical Evaluation

We empirically evaluated the performance of the proposed methodology using synthetic and real data sets. The data set we used is typically referred to as a co-occurrence data set, each record of which is a pair of

Table 1: Data summary.

Data set	# classes	# 1st att.	# 2nd att.	# train	# test
SyM	4	1,000	1,000	40k	40k
SyL	4	5,000	5,000	400k	40k
Y2H	-	1,888	2,803	5,470	1,745 (381)

Table 2: Summary of parameter settings in our method.

Data set	N	N_n
SyM	10,000 / 30,000	4,000
SyL	30,000	40,000
Y2H	5,000	470

two discrete attributes. The pair indicates that the two attribute values co-occur and are correlated. For the data sets we used, we assumed that examples are categorized into a relatively small number of clusters. Each example in the synthetic data sets we used was generated according to the clusters we pre-defined, and we trained our component learning model using the generated examples for modeling a mixture of clusters. Here, a noise (pseudo-positive example) is an example that does not fall into any true pre-defined cluster.

The properties of the data sets we used are summarized in Table 1. The parameter settings in our experiments are summarized in Table 2. (We fixed $J = 20$.) In the tables, ‘SyM’ and ‘SyL’ indicate synthetic data sets, and ‘Y2H’ indicates a real data set. In generating synthetic data sets, we change a part of the positive examples to pseudo-positive examples, and thus we have the information on whether each example is a positive or pseudo-positive example. We generated the SyM, a medium-sized data set, to check how our method works for this size of data sets. The SyL is generated to investigate how our iterative sampling methodology works for a larger data set, which cannot be directly applied to a component learning algorithm. To test the SyL data set, we set $N = 30,000$, which is fewer than 10% of the number of all training examples.

We evaluated the performance of our method by comparing it with those of two other methods, i.e. a random-and-iterative subset sampling method (hereafter called Rand for short) and a method using a set of aspect models (hereafter called AMs for short). Rand is a strategy that, as done in Ssul, repeats the two steps of sampling a set of examples from a given database and applying a component learning algorithm to it, but it samples the example randomly. The result of AMs corresponds to that obtained after the initialization stage of Ssul and Rand. All of our experiments were run on a Linux workstation with Intel 2.4 GHz Xeon processors.

Table 3: Average final accuracies of Ssul, Rand and AMs for SyM data set, and the t values calculated between Ssul and Rand and between Ssul and AMs.

Noise level	N	Final accuracy (%)			t (vs.R)	t (vs.A)
		Ssul	Rand	AMs		
10	10k	99.96	100	98.06	2.73	27.93
	30k	100	100	99.94	1.12	4.25
20	10k	99.92	99.93	93.74	0.65	25.15
	30k	99.81	99.93	99.54	4.47	5.47
30	10k	98.70	96.40	80.72	22.25	25.72
	30k	97.66	88.21	76.30	21.25	8.63

3.1 Evaluation on Synthetic Data Sets The synthetic co-occurrence data sets we used in our experiments were generated as follows: We first define disjoint clusters, each of which has an equal size, on the space consisting of two attributes. We then randomly generate pairs (of the two attributes), each of which falls into one of the clusters. Finally, we repeat certain times to randomly pick a pair and alter an attribute value of the pair so that the pair does not fall into any true cluster. In our experiments, in order to test the performance of our method in noisy conditions, we changed the noise level (the ratio of noises to all examples) of the training examples, while we fixed the noise level of the test examples at 50%.

In evaluation, our method computes the likelihood for a test example by averaging over the likelihoods calculated by all component models obtained. We set a threshold value, and if the likelihood is larger than the threshold, the example is regarded as a positive example. The prediction accuracy is calculated as the maximum ratio of the number of positive examples to the number of all examples when we change the value of the threshold in the range of all possible likelihood values. Our evaluation was mainly done in terms of the ‘final prediction accuracy’ (on separate test data). By final prediction accuracy, we mean the accuracy level reached for data sizes large enough that the predictive performance appears to be saturating.

For each parameter setting in synthetic data sets, training and test data sets are generated separately, and a data set (of training and test data sets) is generated five times randomly. The results (final prediction accuracy) of our experiments using synthetic data sets were then averaged over the five runs.

We first show the results of our experimentation on SyM in the form of learning curves in Fig. 2. In a learning curve, a set of predictive accuracies is plotted against the total computation time (including disk access time), which is required for attaining each of the accuracies. The figure shows that no significant

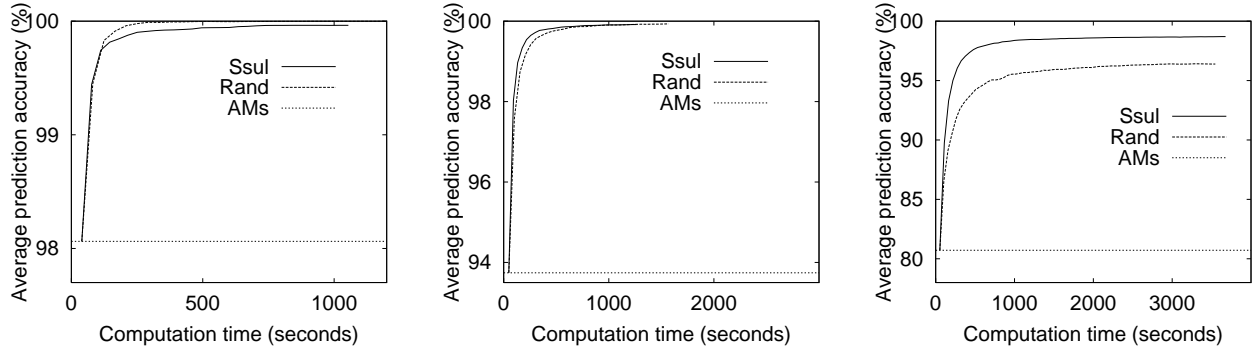


Figure 2: For SyM and $N = 10,000$, average prediction accuracies plotted against total computation time when the noise level is (a) 10, (b) 20 and (c) 30%.

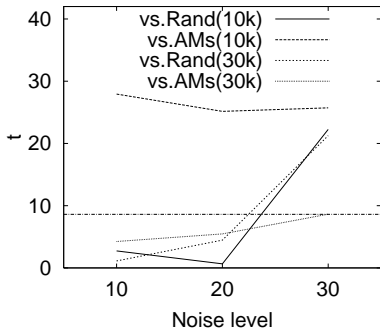


Figure 3: t values plotted against noise levels in SyM.

Table 4: Average final accuracies of Ssul, Rand and AMs for SyL data set, and the t values calculated between Ssul and Rand and between Ssul and AMs.

Noise level	Final accuracy (%)			t (vs. R)	t (vs. A)
	Ssul	Rand	AMs		
5	99.33	99.49	76.74	2.82	22.41
10	94.57	90.85	63.04	11.91	112.49
15	80.03	79.18	58.13	5.73	60.47

difference between Ssul and Rand was found at lower noise levels, i.e. 10 and 20%, though both can be favorably compared with AMs at these noise levels. However, note that once when the noise level was raised to 30%, Ssul clearly out-performed Rand.

These results are summarized in Table 3. The table shows that the final accuracies reached by the three methods for the data set and the t values of the (pairwise) mean difference significance test for the respective cases are given in the table. The t values are calculated using the following formula: $t = \frac{|ave(D)|}{\sqrt{\frac{var(D)}{n}}}$, where we let D denote the difference between the accuracies of the two methods for each data set in our five trials, $ave(X)$ the average of X , $var(X)$ the variance of X , and n the number of data sets (five in our case). For the case where $n = 5$, if t is greater than 8.610 then it is more than 99.9% statistically significant that one achieves a higher accuracy than the other.

Table 5: Final average log-likelihood of Ssul, Rand and AMs for Y2H.

Test data	# clusters (K)	Average log-likelihood		
		Ssul	Rand	AMs
All	4	-13.827	-13.910	-13.889
	10	-13.190	-13.274	-13.255
Exc	4	-14.686	-14.769	-14.754
	10	-14.387	-14.464	-14.493

As shown in Table 3, in SyM, for a noise level of 30%, the t values range from 8.63 to 25.72, although for noise levels of 10 and 20%, the t values are smaller than 8.610 in six out of eight cases. We can statistically see that the predictive performance of Ssul is much better than the two other methods for larger noise levels. This result can be visualized by the graph shown in Fig. 3. From the figure, we can say that our methodology is especially effective for noisy data sets.

In order to investigate the applicability of our method to larger data sets, we compared the performance of our method with those of the two other methods, using the SyL data set. The results are summarized in Table 4. For a noise level of 10%, Ssul is approximately 4% better than Rand, but no statistically significant difference in the performance of the two methods was found for a noise level of 15%. This was contrary to the fact obtained by the experiments on SyM, i.e. that the performance difference of the two methods becomes more pronounced for larger noise levels. This result is supposed to be due to the fact that our sampling strategy utilizes the predictions done by its component learning algorithm, and our strategy does not work if we cannot obtain good predictions by our component learning algorithm. One more item of note is that the performance difference between Rand and AMs is considerably larger than that between Ssul and Rand. This result indicates that an iterative sampling strategy is more effective for a larger data set.

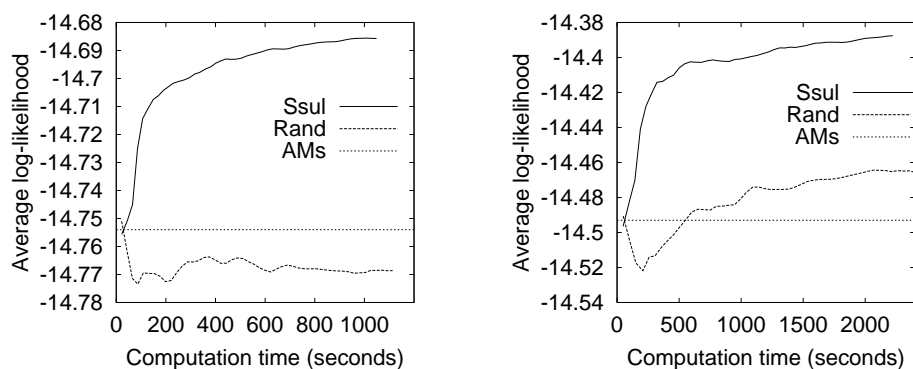


Figure 4: Average log-likelihood for Exc when the number of clusters is (a) 4 and (b) 10.

3.2 Evaluation on Real Protein-Protein Interaction Data Sets We further evaluated our methodology using an actual data set in the area of molecular biology. Concretely, we used protein-protein interaction data sets available from the KEGG database[5] as a training data set. The data sets are the results obtained from the four yeast-two-hybrid (Y2H) experiments performed by Ito et al. [4], etc. We mixed the four data sets and obtained 5,470 pairs as a training data set.

As a test data set, we used the MIPS comprehensive yeast genome database [7], which contains a physical interaction table of yeast proteins. The table includes a variety of protein-protein physical interactions, which are obtained by not only modern high-throughput experimental techniques such as Y2H but also traditional manual experiments. From the table, we selected 1,745 protein-protein interactions, each pair of which consists of one of the first attribute values and one of the second attribute values of the 5,470 training data pairs. Out of the 1,745, we selected 381 that were not detected by any Y2H experiment. We called the two datasets of containing 1,745 and 381 pairs All and Exc, respectively.

In the test data sets, since no example is labeled as to whether it is a positive or pseudo-positive example, we have to regard all examples as positive examples. Thus, we evaluated the three methods by averaging the likelihoods calculated for examples in the test data sets instead of the predictive accuracy we used in the experiments of synthetic data sets. If the average likelihood calculated by our method is larger than those of the two other methods, we can say that our method has a higher predictive ability than them. However, the MIPS test data set also may have experimental errors or noises, and then we used the average of the largest 80% of the calculated likelihoods.

We show part of the results of the above experiment (in the case of Exc) in Fig. 4. In the figure, the average log-likelihood for the Exc test data set is plotted against the total computation time. As shown in the figure,

the average log-likelihood obtained by our method is favorably compared with those of Rand and AMs.

The results of the average log-likelihoods are summarized in Table 5. The table shows that the average likelihood computed by our method is larger than those by the other methods. In particular, for Exc, which contains perfectly new examples, the prediction done by our method gives a higher average likelihood. The result indicates that when arbitrary real-world co-occurrence examples are given, our strategy is effective for distinguishing pseudo-positive (noises) from others.

4 Concluding Remarks

We have proposed a new sampling method for unsupervised learning that targets the mining from noisy data sets and have empirically shown that it works effectively for both synthetic and real data sets.

References

- [1] A. P. Dempster, N. M. Laird and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. R. Statist. Soc. B, 39 (1977), pp. 1–38.
- [2] Y. Freund and R. Shapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, J. C. S. S., 55 (1997), pp. 119–139.
- [3] T. Hofmann, *Unsupervised learning by probabilistic latent semantic analysis*, Machine Learning, 42 (2001), pp. 177–196.
- [4] T. Ito et al., *A comprehensive assessment of large-scale datasets of protein-protein interactions*, PNAS, 98(8) (2001), pp. 4569–4574.
- [5] M. Kanehisa, et al., *The KEGG databases at GenomeNet*, NAR, 30 (2002), pp. 42–46.
- [6] C. Meek et al., *The learning-curve sampling method applied to model-based clustering*, Journal of Machine Learning Research, 2 (2002), pp. 397–418.
- [7] H. Mewes et al., *MIPS: A database for genomes and protein sequences*, NAR, 30 (2002), pp. 31–34.
- [8] F. Provost and V. Kolluri, *A survey of methods for scaling up inductive algorithms*, Knowledge Discovery and Data Mining, 3 (1999), pp. 131–169.