

# Co-clustering Preliminary Experiments

Università degli studi di Napoli “Federico II”

Via Cinthia, 80126, Napoli, Italy

Vincenzo Russo  
(vincenzo.russo@neminis.org)

June, 28 2007

## **Abstract**

In this document we present the results of some preliminary experiments using Bregman Co-clustering. The experiments were conducted both over synthetic data sets and over real-world data sets taken from UCI Repository of Machine Learning Database.

# Contents

<b>1</b>	<b>Hardware</b>	<b>3</b>
<b>2</b>	<b>Software</b>	<b>3</b>
<b>3</b>	<b>Clustering evaluation</b>	<b>3</b>
<b>4</b>	<b>Real-world 1: The Iris Plant Database</b>	<b>4</b>
4.1	The Data Set . . . . .	4
4.2	Setups . . . . .	4
4.3	Minimum Sum Squared Residue Co-clustering I . . . . .	5
4.3.1	Conclusion . . . . .	5
4.4	Minimum Sum Squared Residue Co-clustering II . . . . .	6
4.4.1	Conclusion . . . . .	7
4.5	Information Theoretic Co-clustering . . . . .	7
4.5.1	Conclusion . . . . .	8
<b>5</b>	<b>Real-world 2: The Mushrooms Database</b>	<b>8</b>
5.1	The Data Set . . . . .	8
5.1.1	Preprocessing the data-set . . . . .	8
5.2	Setups . . . . .	9
5.3	Minimum Sum Squared Residue Co-clustering I . . . . .	9
5.3.1	Points with missing values . . . . .	9
5.4	Minimum Sum Squared Residue Co-clustering II . . . . .	9
5.4.1	Points with missing values . . . . .	10
5.5	Information Theoretic Co-clustering . . . . .	10
5.5.1	Points with missing values . . . . .	10
<b>6</b>	<b>Synthetic 1: Syndeca 02</b>	<b>10</b>
6.1	The Data Set . . . . .	10
6.2	Setups . . . . .	11
6.3	Minimum Sum Squared Residue Co-clustering I . . . . .	11
6.4	Minimum Sum Squared Residue Co-clustering II . . . . .	11
6.5	Information Theoretic Co-clustering . . . . .	11
6.6	Conclusion . . . . .	12

# 1 Hardware

The hardware configuration used for tests:

- CPU: PowerPC G4 at 1.5GHz
- RAM: 768 MBytes
- OS: Mac OS X

# 2 Software

The software [CGS04] used for Bregman Co-clustering experiments is developed at Data Mining Laboratory (The University of Texas at Austin) and it implements three instances of the Bregman Co-clustering scheme [BDG<sup>+</sup>04b, BDG<sup>+</sup>04a]:

- *Information Theoretic Co-clustering (ITCC)* [DMM03, DG03]
- *Minimum Sum Squared Residue Co-clustering I (MSSRCCI)* [CDGS04]
- *Minimum Sum Squared Residue Co-clustering II (MSSRCCII)* [CDGS04]

For each co-clustering algorithm we have also five different ways for updating the cluster centroids: local search and four version of batch update. We only use the former in the tests proposed here, because it allows us to avoid empty clusters; this behavior is useful in the case we know the exact number of classes.

We also have different ways to prepare the initial co-clustering, but for the moment we use the random way only.

Several other options are provided from the software, but we don't take them all in account for these preliminary experiments.

# 3 Clustering evaluation

Due the nature of these preliminary tests, where for each data-set used we know also the original classification, we adopt the classic instruments used for evaluating classifications:

- Precision/Recall/ $F_1$  in the case of binary classification
- Macroaveraging in the case of multi-class

Furthermore, we also use the *accuracy* metric, although sometimes it is not an appropriate measure [MRS07, par. 8.3]; this is why it is often used in several papers evaluating their own clustering techniques.

## 4 Real-world 1: The Iris Plant Database

The *Iris Data Set* [Fis36] is the best known database to be found in the pattern recognition literature. The data set contains three classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are *not* linearly separable from each other.

### 4.1 The Data Set

The data-set was taken from *UCI Repository of Machine Learning Database* [NHBM98]:

- Number of instances: 150
- Number of attributes: 4
- Number of classes: 3
- Classes: Iris Setosa, Iris Versicolour, Iris Virginica
- Class distribution: 50 items per class (the first 50 items in the first class, and so on...)
- Peculiarity: 1 class is linearly separable from the other 2, but the other 2 are not linearly separable from each other

### 4.2 Setups

We performed the tests with all three available instances of the Bregman co-clustering. In each case we used the following two different setups:

1. Co-clusters requested: 3
  - Row clusters requested: 3
  - Column clusters requested: 1
2. Co-clusters requested: 6

- Row clusters requested: 3
- Column clusters requested: 2

In the first setup the co-clustering algorithm acts as a classic one-way clustering algorithm, because requesting a single column cluster is the same as taking all the features in account, without perform any clustering on them.

In the second setup we request two column clusters. We recall that clustering also along the columns (features) allows us to decrease the dimensionality, to deal with sparse data and, if we need, to find the interplay between objects and features clusters. As we can see in the following sections, we can sometimes increase the accuracy of the algorithm decreasing the dimensionality of the data-set.

### 4.3 Minimum Sum Squared Residue Co-clustering I

For the **first setup** we obtain a *macro-averaging* value of 87.649% from the results in the Table 1. The *accuracy* is 88%.

For the **second setup** we obtain a *macro-averaging* value of 87.099% from the results in the Table 2<sup>1</sup>. The *accuracy* is 87.333%.

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	92.593%	100%	96.154%
2	90%	72%	80%
3	82.143%	92%	86.792%

Table 1: The precision/recall/f1 values obtained in the first setup for each class of the Iris data-set, using MSRCCI

#### 4.3.1 Conclusion

We have performed two tests with the same co-clustering instance (*MSSR-CCI*): in the first one no column clusters was requested, in the second one two column clusters did. The contextual feature clustering happened in the second setup has improved the precision in the classification of the objects

<sup>1</sup>Since we have requested two column clusters, we have in this case two co-clusters for each row-cluster we are interested into. However, in this case each co-cluster has the same row elements for each row-cluster, so we have have simpler precision/recall/f1 table like in the first setup.

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	98.039%	100%	99.01%
2	89.744%	70%	78.652%
3	76.667%	92%	83.636%

Table 2: The precision/recall/f1 values obtained in the second setup for each class of the Iris data-set, using MSRCCI

belonging the first class, but as drawback we have a worst separation between the two non linearly separable classes.

#### 4.4 Minimum Sum Squared Residue Co-clustering II

For the **first setup** we obtain a *macro-averaging* value of 88.329% from the results in the Table 3. The *accuracy* is 88.667%.

For the **second setup** we obtain a *macro-averaging* value of 75.641% from the results in the Table 4<sup>2</sup>. The *accuracy* is 78%.

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	100%	100%	100%
2	100%	66%	79.518%
3	74.627%	100%	85.47%

Table 3: The precision/recall/f1 values obtained in the first setup for each class of the Iris data-set, using MSSRCCI

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	100%	98%	98.99%
2	94.737%	36%	52.174%
3	60.976%	100%	75.758%

Table 4: The precision/recall/f1 values obtained in the second setup for each class of the Iris data-set, using MSSRCCI

---

<sup>2</sup>Also in this case each co-cluster has the same row elements for each row-cluster, so we have have simpler precision/recall/f1 table like in the first setup.

#### 4.4.1 Conclusion

We have performed two tests with the same co-clustering instance (*MSS-RCCII*). The MSSRCCII compares better with MSSRCCI only in the first setup, while in the second setup it dramatically loses in clustering quality. Furthermore, the variance between the first and the second setup results is larger in the case of the MSSRCCII than in the case we use MSSRCCI.

### 4.5 Information Theoretic Co-clustering

The results in the case of the Information Theoretic Co-clustering instance are quite interesting. It works bad in the **first setup**, where we obtain a *macro-averaging* value of 18.48% from the results in the Table 5 and an *accuracy* value of 32%. On the contrary, ITCC works extremely fine in the **second setup**, where we obtain a *macro-averaging* value of 96.658% from the results in the Table 6<sup>3</sup> and an *accuracy* value of 96.667%.

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	32.624%	92%	48.168%
2	0%	0%	n.d.
3	40%	4%	7.272%

Table 5: The precision/recall/f1 values obtained in the first setup for each class of the Iris data-set, using ITCC

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	100%	100%	100%
2	100%	90%	94.736%
3	90.909%	100%	95.238%

Table 6: The precision/recall/f1 values obtained in the second setup for each class of the Iris data-set, using ITCC

---

<sup>3</sup>Also in this case each co-cluster has the same row elements for each row-cluster, so we have have simpler precision/recall/f1 table like in the first setup.

### 4.5.1 Conclusion

We have performed two tests with the same co-clustering instance (*ITCC*). In the first setup *ITCC* behaved worse than other two co-clustering algorithms and its results were really poor. Indeed, in the second setup *ITCC* showed the best clustering quality.

## 5 Real-world 2: The Mushrooms Database

This data set [Sch87] includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the *Agaricus* and *Lepiota* Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.

### 5.1 The Data Set

The data-set was taken from *UCI Repository of Machine Learning Database*:

- Number of instances: 8124
- Number of attributes: 22
- Number of classes: 2
- Classes: Edible Mushrooms, Non-edible Mushrooms (Poisonous, unknown edibility)
- Class distribution: 4208 in the first class (51.8%), 3916 (48.2%)
- Peculiarity: 2480 missing values for the 12th feature

#### 5.1.1 Preprocessing the data-set

The original data-set taken from the UCI Repository included in the first column a label indicating the original class of each mushroom. Furthermore the attribute values are in character format. So, we have first removed the first column. Successively, we have processed all feature values with the `ord` function, which assigns an unique numeric value to each character.

Finally, the original missing value character (a question mark “?”) was turned in a value out of the range (i. e. the zero value) of `ord` function codomain. This is correct because it is the same thing the Bregman co-clustering does when the input data matrix contains missing values. We do

it by hand for the convenience of maintaining the input matrix in a dense format instead of using the sparse format.

## 5.2 Setups

As we already done in previous sections, we performed the tests with all three available instances of the Bregman co-clustering. The only setup we used in all three tests is the following:

1. Co-clusters requested: 2
  - Row clusters requested: 2
  - Column clusters requested: 1

Setups with more than one column clusters gave all the same percentage of misclassification, regardless the co-clustering instance used.

## 5.3 Minimum Sum Squared Residue Co-clustering I

We obtain:

- Precision: 78.934%
- Recall: 81.654%
- $F_1$ : 80.27%
- Accuracy: 79.21%.

### 5.3.1 Points with missing values

There are 2480 points which report a missing value for the 12th feature. The 71% of those have been right classified.

## 5.4 Minimum Sum Squared Residue Co-clustering II

We obtain:

- Precision: 61.8%
- Recall: 82.89%
- $F_1$ : 70.808%
- Accuracy: 64.599%.

### 5.4.1 Points with missing values

There are 2480 points which report a missing value for the 12th feature. The 71% of those have been right classified.

## 5.5 Information Theoretic Co-clustering

We obtain:

- Precision: 73.274%
- Recall: 17.657%
- $F_1$ : 28.456%
- Accuracy: 54.013%.

### 5.5.1 Points with missing values

There are 2480 points which report a missing value for the 12th feature. The 73% of those have been right classified.

## 6 Synthetic 1: Syndeca 02

The first experiment on synthetic data was performed on a data-set created with *SynDECA* [VV05].

### 6.1 The Data Set

- Number of instances: 1000
- Number of attributes: 10
- Number of classes: 5
- Class distribution: 327 in the first class, 134 in the second one, 162 in the third one, 132 in the fourth one, 133 in the fifth one.
- Peculiarity: only 888 points are distributed over the five classes. Remaining 112 points are noise (outliers).
- Peculiarity: the various clusters have different shapes, like circle, rectangular, ellipse, random and square.

## 6.2 Setups

As we already done in previous sections, we performed the tests with all three available instances of the Bregman co-clustering. The only setup we used in all three tests is the following:

1. Co-clusters requested: 2
  - Row clusters requested: 2
  - Column clusters requested: 1

## 6.3 Minimum Sum Squared Residue Co-clustering I

In this test we achieve a macro-averaging value of 57.237% from the results in the Table 7 and an accuracy value of 71.171%.

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	96.176%	100%	98.05%
2	36.676%	95.522%	53.002%
3	2.326%	1.235%	1.614%
4	80.488%	100%	89.19%
5	70.492%	32.331%	44.33%

Table 7: The precision/recall/f1 values obtained in the first setup for each class of the Syndeca 02 data-set, using MSRCCI

## 6.4 Minimum Sum Squared Residue Co-clustering II

In this test we achieve a macro-averaging value of 94.17% from the results in the Table 8 and an accuracy value of 100%.

The MSSRCCII behaves well in this case. The “strange” discrepancy between the accuracy and the macro-averaging is due to the noise points; in fact, the MSSRCCII correctly classifies all 888 non-noise points, but the noise points have been turned in false-positive classifications in the clustering results, so the macro-averaging value cannot be 100%.

## 6.5 Information Theoretic Co-clustering

The ITCC results are really poor, with a macro-averaging value of 13.107% and an accuracy value of 33.221%.

Class ( <i>vs Rest</i> )	Precision	Recall	$F_1$
1	87.668%	100%	93.428%
2	90.541%	100%	95.036%
3	95.294%	100%	97.59%
4	81.481%	100%	89.796%
5	90.476%	100%	95%

Table 8: The precision/recall/f1 values obtained in the first setup for each class of the Syndeca 02 data-set, using MSRCCII

## 6.6 Conclusion

The peculiarity of this data-set consist of 112 point of noise. Such points are outliers, but the Bregman co-clustering are unable to identify them and than exclude them from the clustering process (or from the clustering results). However, we are interested in robustness of the algorithms, i. e. how much an algorithm perform well despite the noise points. The more robust algorithm in this case is the MSSRCCII, the second type of the Euclidean Distance based Bregman co-clustering instance, which classify correctly all 888 non-noise points, despite the “foreign” points.

## References

- [BDG<sup>+</sup>04a] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD)*, pages 509–514, August 2004. 2
- [BDG<sup>+</sup>04b] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. Technical report, UTCS TR04-24, UT, Austin, 2004. 2
- [CDGS04] H. Cho, I. Dhillon, Y. Guan, and S. Sra. Minimum sum squared residue co-clustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 114–125, April 2004. 2
- [CGS04] Hyuk Cho, Yuqiang Guan, and Suvrit Sra. Co-cluster (v 1.1). Bregman co-clustering software, 2004. 2
- [DG03] I. S. Dhillon and Yuqiang Guan. Information theoretic clustering of sparse co-occurrence data. Technical report tr-03-39, The University of Texas at Austin, Department of Computer Sciences, September 2003. 2
- [DMM03] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003)*, pages 89–98, 2003. 2
- [Fis36] Ronald Aylmer Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936. 4
- [MRS07] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2007. 3
- [NHBM98] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz. UCI repository of machine learning databases, 1998. 4.1
- [Sch87] Jeffrey Curtis Schlimmer. *Concept acquisition through representational adjustment*. PhD thesis, Department of Information and Computer Science, University of California, Irvine, 1987. 5

- [VV05] Jhansi Rani Vennam and Soujanya Vadapalli. Syndeca: A tool to generate synthetic datasets for evaluation of clustering algorithms. In *11th International Conference on Management of Data (COMAD 2005)*, Goa, India, January 2005. 6