Ensembles Based on Random Projections to Improve the Accuracy of Clustering Algorithms

Alberto Bertoni and Giorgio Valentini

DSI, Dipartimento di Scienze dell' Informazione,
Università degli Studi di Milano,
Via Comelico 39, 20135 Milano, Italia
{bertoni, valentini}@dsi.unimi.it

Abstract. We present an algorithmic scheme for unsupervised cluster ensembles, based on randomized projections between metric spaces, by which a substantial dimensionality reduction is obtained. Multiple clusterings are performed on random subspaces, approximately preserving the distances between the projected data, and then they are combined using a pairwise similarity matrix; in this way the accuracy of each "base" clustering is maintained, and the diversity between them is improved. The proposed approach is effective for clustering problems characterized by high dimensional data, as shown by our preliminary experimental results.

1 Introduction

Supervised multi-classifiers systems characterized the early development of ensemble methods [1, 2]. Recently this approach has been extended to unsupervised clustering problems [3, 4].

In a previous work we proposed stability measures that make use of random projections to assess cluster reliability [5], extending a previous approach [6] based on an unsupervised version of the random subspace method [7].

In this paper we adopt the same approach to develop cluster ensembles based on random projections. Unfortunately, a deterministic projection of the data into relatively low dimensional spaces may introduce relevant distortions, and, as a consequence, the clustering in the projected space may results consistently different from the clustering in the original space. For these reasons we propose to perform multiple clusterings on randomly chosen projected subspaces, approximately preserving the distances between the examples, and then combining them to generate the final "consensus" clustering.

The next section introduces basic concepts about randomized embeddings between metric spaces. Sect. 3 presents the *Randomized embedding clustering (RE-Clust)* ensemble algorithm, and Sect. 4 show the results of the application of the ensemble method to high dimensional synthetic data. The discussion of the results and the outgoing developments of the present work end the paper.

B. Apolloni et al. (Eds.): WIRN/NAIS 2005, LNCS 3931, pp. 31–37, 2006.

2 Randomized Embeddings

2.1 Randomized Embeddings with Low Distortion

Dimensionality reduction may be obtained by mapping points from a high to a low-dimensional space: $\mu : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, with $d' < d$, approximately preserving some characteristics, i.e. the distances between points In this way, algorithms whose results depend only on the distances $||x_i - x_j||$ could be applied to the compressed data $\mu(X)$, giving the same results, as in the original input space. In this context randomized embeddings with low distortion represent a key concept. A *randomized embedding* between $\mathbb{R}^d$ and $\mathbb{R}^{d'}$ with distortion $1+\epsilon$, $(0 < \epsilon \leq 1/2)$ and failure probability P is a distribution probability on the linear mapping $\mu : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, such that, for every pair $p, q \in \mathbb{R}^d$, the following property holds with probability $\geq 1 - P$:

$$\frac{1}{1+\epsilon} \leq \frac{||\mu(p) - \mu(q)||}{||p - q||} \leq 1 + \epsilon \tag{1}$$

The main result on randomized embedding is due to Johnson and Lindenstrauss [8], who proved the following:

Johnson-Lindenstrauss (JL) lemma: Given a set S with $|S| = n$ there exists a $1 + \epsilon$-distortion embedding into $\mathbb{R}^{d'}$ with $d' = c \ \log n/\epsilon^2$, where c is a suitable constant.

The embedding exhibited in [8] consists in random projections from $\mathbb{R}^d$ into $\mathbb{R}^{d'}$, represented by matrices $d' \times d$ with random orthonormal vectors. Similar results may be obtained by using simpler embeddings [9], represented through random $d' \times d$ matrices $P = 1/\sqrt{d'}(r_{ij})$, where r_{ij} are random variables such that:

$$E[r_{ij}] = 0, \qquad Var[r_{ij}] = 1$$

For sake of simplicity, we call random projections even this kind of embeddings.

2.2 Random Projections

Examples of randomized maps, represented trough $d' \times d$ matrices P such that the columns of the "compressed" data set $D_P = PD$ have approximately the same distance are:

1. *Plus-Minus-One (PMO)*: random projections: represented by matrices $P = 1/\sqrt{d'}(r_{ij})$, where r_{ij} are uniformly chosen in $\{-1, 1\}$, such that $Prob(r_{ij} = 1) = Prob(r_{ij} = -1) = 1/2$. In this case the *JL lemma* holds with $c \simeq 4$.
2. *Random Subspace (RS)* [7]: represented by $d' \times d$ matrices $P = \sqrt{d/d'}(r_{ij})$, where r_{ij} are uniformly chosen with entries in $\{0, 1\}$, and with exactly one "1" per row and at most one "1" per column. Even if *RS* subspaces can be quickly computed, the do not satisfy the *JL lemma*.

3 Randomized Embedding Cluster Ensembles

Consider a data set $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i \in \mathbb{R}^d$, $(1 \leq i \leq n)$; a subset $A \subseteq \{1, 2, \ldots, n\}$ univocally individuates a subset of examples $\{x_j | j \in A\} \subseteq X$. The data set X may be represented as a $d \times n$ matrix D, where columns correspond to the examples, and rows correspond to the "components" of the examples $x \in X$. A *k-clustering* C of X is a list $C =< A_1, A_2, \ldots, A_k >$, with $A_i \subseteq \{1, 2, \ldots, n\}$ and such that $\bigcup A_i = \{1, \ldots, n\}$. A *clustering algorithm* $\mathcal{C}$ is a procedure that, having as input a data set X and an integer k, outputs a k-clustering C of X: $\mathcal{C}(X, k) =< A_1, A_2, \ldots, A_k >$.

The main ideas behind the proposed cluster ensemble algorithm *RE-Clust* (acronym for Randomized Embedding Clustering) are based on data compression, and generation and combination of multiple "base" clusterings. Indeed at first data are randomly projected from the original to lower dimensional subspaces, using projections described in Sect 2.2 in order to approximately preserve the distances between the examples. Then multiple clusterings are performed on multiple instances of the projected data, and a similarity matrix between pairs of examples is used to combine the multiple clusterings.

The high level pseudo-code of the ensemble algorithm scheme is the following:

RE-Clust **Algorithm:**

`Input`:

- a data set $X = \{x_1, x_2, \ldots, x_n\}$, represented by a $d \times n$ D matrix.
- an integer k (number of clusters)
- a real $\epsilon > 0$ (distortion level)
- an integer c (number of clusterings)
- two clustering algorithms $\mathcal{C}$ and $\mathcal{C}_{com}$
- a procedure that realizes a randomized map μ

```
begin algorithm
    (1) $d' = 2 \cdot \left(\frac{2 \log n + \log c}{\epsilon^2}\right)$
    (2) For each $i, j \in \{1, \ldots, n\}$ do $M_{ij} = 0$
    (3) Repeat for $t = 1$ to $c$
        (4) $P_t =$ Generate_projection_matrix $(d, d')$
        (5) $D_t = P_t \cdot D$
        (6) $< C_1^{(t)}, C_2^{(t)}, \ldots, C_k^{(t)} >= \mathcal{C}(D_t, k)$
        (7) For each $i, j \in \{1, \ldots, n\}$
            $M_{ij}^{(t)} = \frac{1}{k} \sum_{s=1}^{k} I(i \in C_s^{(t)}) \cdot I(j \in C_s^{(t)})$
    end repeat
    (8)$M = \frac{\sum_{t=1}^{c} M^{(t)}}{c}$
    (9) $< A_1, A_2, \ldots, A_k >= \mathcal{C}_{com}(M, k)$
end algorithm.
```

`Output`:

- the final clustering $C =< A_1, A_2, \ldots, A_k >$

In the first step of the algorithm, given a distortion level ϵ, the dimension d' for the compressed data is computed according to the *JL lemma*.

At each iteration of the main repeat loop (step 3-7), the procedure `Generate_projection_matrix` outputs a projection matrix P_t according to the randomized embedding μ, and a projected data set $D_t = P_t \cdot D$ is generated; the corresponding clustering $< C_1^{(t)}, C_2^{(t)}, \ldots, C_k^{(t)} >$ is computed by calling $\mathcal{C}$, and a $M^{(t)}$ similarity matrix is built. The *similarity matrix* $M^{(t)}$ associated to a clustering $C =< C_1^{(t)}, C_2^{(t)}, \ldots, C_k^{(t)} >$ is a $n \times n$ matrix such that:

$$M_{ij}^{(t)} = \frac{1}{k} \sum_{s=1}^{k} I(i \in C_s^{(t)}) \cdot I(j \in C_s^{(t)}) \tag{2}$$

where I is is the characteristic function of the set C_s. After step (8), M_{ij} denotes the frequency by which the examples i and j occur in the same cluster across multiple clusterings. The final clustering is performed by applying the clustering algorithm $\mathcal{C}_{com}$ to the main similarity matrix M. Choosing different random projections we may generate different *RE-Clust* ensembles (e.g. *PMO* and *RS* cluster ensembles).

4 Experimental Results

In this section we present some preliminary experimental results with the *RE-Clust* ensemble algorithm. The Ward's hierarchical agglomerative clustering algorithm [10] has been applied as "base" clustering algorithm.

4.1 Experimental Environment

Synthetic Data Generation. We experimented with 2 different sample generators, whose samples are distributed according to different mixtures of high dimensional gaussian distributions.

Sample1 is a generator for 5000-dimensional data sets composed by 3 clusters. The elements of each cluster are distributed according to a spherical gaussian with standard deviation equal to 3. The first cluster is centered in $\mathbf{0}$, that is a 5000-dimensional vector with all zeros. The other two clusters are centered in $0.5\mathbf{e}$ and $-0.5\mathbf{e}$, where $\mathbf{e}$ is a vector with all 1.

Sample2 is a a generator for 6000-dimensional data sets composed by 5 clusters of data normally distributed. The diagonal of the covariance matrix for all the classes has its element equal to 1 (first 1000 elements) and equal to 2 (last 5000 elements). The first 1000 variables of the five clusters are respectively centered in $\mathbf{0}$, $\mathbf{e}$, $-\mathbf{e}$, $5\mathbf{e}$, $-5\mathbf{e}$. The remaining 5000 variables are centered in 0 for all clusters.

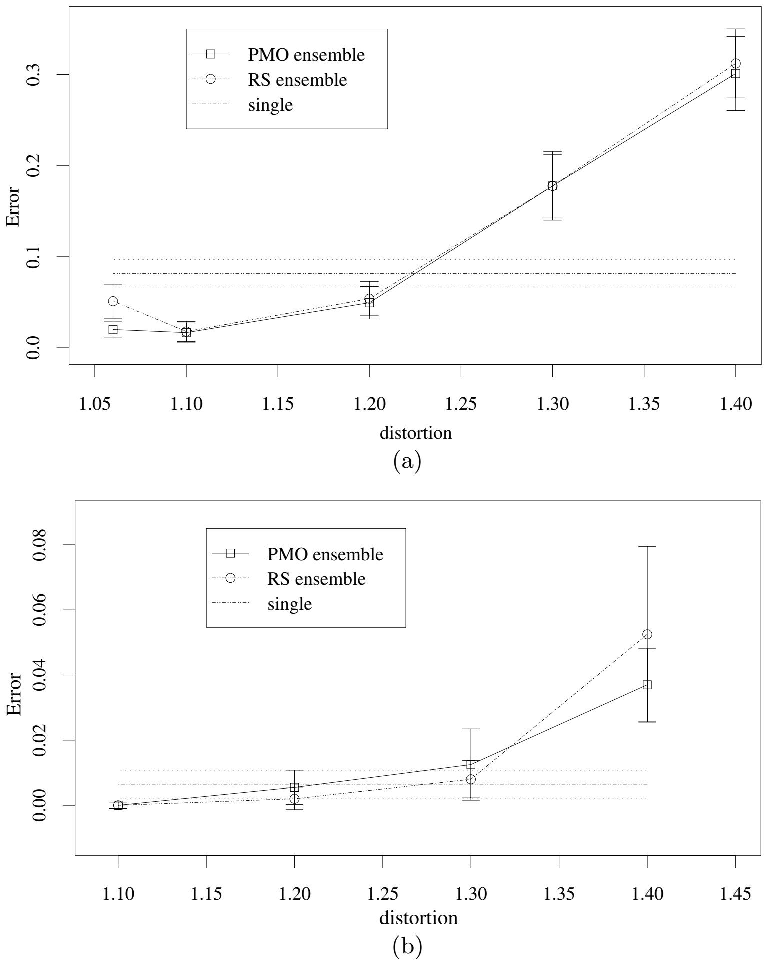
For each generator, we considered 30 different random samples each respectively composed by $60, 100$ examples (that is, 20 examples per class).

Experimental Setup. We compared classical single hierarchical clustering algorithm with our ensemble approach considering *PMO* and *RS* random projections (Sect. 2.2). We used 30 different realizations for each synthetic data set, using each time 20 clusterings for both *PMO* and *RS* ensembles. For each *PMO* and *RS* ensemble we experimented with different distortions, corresponding to $\epsilon \in [0.06, 0.5]$.

We implemented the ensemble algorithms and the scripts used for the experiments in the *R* language (code is freely available from the authors).

4.2 Results

With *sample1* (Fig.1 (a)) for 1.10 distortion, that corresponds to projections from the original 5000 into a 3407 dimensional subspace, *RE-Clust* ensembles perform



(a)

(b)

Fig. 1. Comparison of mean errors between single hierarchical clustering, PMO and RS ensembles with different $1 + \epsilon$ distortions. For ensembles, error bars for the 99% confidence interval are represented, while for single hierarchical clustering the 99% confidence interval is represented by the dotted lines above and below the horizontal dash-dotted line. (a) *Sample1* data set (b) *sample2*.

significantly better than single clustering. Indeed *PMO* ensembles achieve a 0.017 ± 0.010 mean error over 30 different realizations from *sample1*, and *RS* ensembles a 0.018 ± 0.011 mean error against a 0.082 ± 0.015 mean error for single hierarchical clustering. Also with an estimated 1.20 distortion (with a corresponding subspace dimension equal to 852) we obtain significantly better results with both *PMO* and *RS* ensembles.

With *sample2* (Fig.1 (b)) the difference is significant only for 1.10 distortion, while for larger distortions the difference is not significant and, on the contrary, with 1.4 distortion *RE-Clust* ensembles perform worse than single clustering. This may be due both to the relatively high distortion induced by the randomized embedding and to the loss of information due to the random projection to a too low dimensional space. Anyway, with all the high dimensional synthetic data sets the *RE-Clust* ensembles achieve equal or better results with respect to a “single” hierarchical clustering approach, at least when the distortions predicted by the *JL lemma* are lower than 1.30.

5 Conclusions

Experimental results with synthetic data (Sect. 4.2) show that *RE-Clust* ensembles are effective with high dimensional data, even if we need more experiments to confirm these results.

About the reasons why *RE-Clust* outperforms single clustering, we suspect that *RE-Clust* ensembles can reduce the variance component of the error, by “averaging” between different multiple clusterings, and we are planning to perform a bias-variance analysis of the algorithm to investigate this topic, using the approach proposed in [11] for supervised ensembles.

To evaluate the performance of *RE-Clust* with other “base” clustering algorithms, we are experimenting with *Partitioning Around Medoids (PAM)* and *fuzzy-c-mean* algorithms.

Acknowledgement

The present work has been developed in the context of the *CIMAINA* Center of Excellence, and it was partially funded by the italian COFIN project *Linguaggi formali ed automi: metodi, modelli ed applicazioni.*

References

[1] Dietterich, T.: Ensemble methods in machine learning. In Kittler, J., Roli, F., eds.: Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy. Volume 1857 of Lecture Notes in Computer Science., Springer-Verlag (2000) 1–15

[2] Valentini, G., Masulli, F.: Ensembles of learning machines. In: Neural Nets WIRN-02. Volume 2486 of Lecture Notes in Computer Science. Springer-Verlag (2002) 3–19

[3] Strehl, A., Ghosh, J.: Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. Journal of Machine Learning Research **3** (2002) 583–618

[4] Hadjitodorov, S., Kuncheva, L., Todorova, L.: Moderate Diversity for Better Cluster Ensembles. Information Fusion (2005)

[5] Bertoni, A., Valentini, G.: Random projections for assessing gene expression cluster stability. In: IJCNN 2005, The IEEE-INNS International Joint Conference on Neural Networks, Montreal (2005) (in press).

[6] Smolkin, M., Gosh, D.: Cluster stability scores for microarray data in cancer studies. BMC Bioinformatics **4** (2003)

[7] Ho, T.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 832–844

[8] Johnson, W., Lindenstrauss, J.: Extensions of Lipshitz mapping into Hilbert space. In: Conference in modern analysis and probability. Volume 26 of Contemporary Mathematics., Amer. Math. Soc. (1984) 189–206

[9] Bingham, E., Mannila, H.: Random projection in dimensionality reduction: Applications to image and text data. In: Proc. of KDD 01, San Francisco, CA, USA, ACM (2001)

[10] Ward, J.: Hierarchcal grouping to optimize an objective function. J. Am. Stat. Assoc. **58** (1963) 236–244

[11] Valentini, G.: An experimental bias-variance analysis of SVM ensembles based on resampling techniques. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics **35** (2005)