Fast approximated relational and kernel clustering

Frank-Michael Schleif and Xibin Zhu and Andrej Gisbrecht and Barbara Hammer
CITEC centre of excellence, Bielefeld University, 33615 Bielefeld - Germany
fschleif@techfak.uni-bielefeld.de

Abstract

The large amount of digital data requests for scalable tools like efficient clustering algorithms. Many algorithms for large data sets request linear separability in an Euclidean space. Kernel approaches can capture the non-linear structure but do not scale well for large data sets. Alternatively, data are often represented implicitly by dissimilarities like for protein sequences, whose methods also often do not scale to large problems. We propose a single algorithm for both type of data, based on a batch approximation of relational soft competitive learning, termed fast generic soft-competitive learning. The algorithm has linear computational and memory requirements and performs favorable to traditional techniques [1].

1. Introduction

The amount of digital data doubles roughly every 20 months. Hence automatic tools to deal with large data become indispensable to extract relevant information and clustering is one of the standard techniques. At the same time, dedicated data formats such as XML, graph structures, etc., are more frequent, leading to large dissimilarity data sets. Classical cluster methods, like k-means process Euclidean data only. Also kernel approaches like kernel-k-means or kernel soft competitive learning (KSCL) [14] are often limited, due to the lack of a valid kernel. Alternatives are given e.g. by the relational soft competitive learning [9] and have recently been extended to large scale problems [8]. Relational soft competitive learning (RSCL) is an extension of soft-competitive learning (SCL) [12]. It replaces the Euclidean distance function of a data point $\mathbf{v}$ to a cluster representant or prototype $\mathbf{w}$ by an implicit representation which refers to the dissimilarity matrix D only, where $d_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\|^2$ denotes the underlying dissimilarities induced by an arbitrary symmetric bilinear form. Obviously, D has squared complexity. Employing the full dissimilarity matrix can be avoided by using the Nyström approximation as shown by the authors earlier in [8]. RSCL has been considered so far only for dissimilarity data, but this restriction is not necessary. Since each similarity matrix, provided by a kernel, can be easily transformed to a dissimilarity matrix, RSCL can be effectively used for both types of data. For similarity data this leads to a *novel* very fast *batch* approach of the *online* KSCL similarly efficient as approximate kernel k-means [5] (a-KKM) [2]. We demonstrate how to derive the new algorithm for both types of data.

In section 2 we outline the related work on large scale clustering and the relational soft competitive learning as well as its kernel counterpart. We present the generic fast approximate soft competitive learning (FG-SCL) in Section 3. Section 4 summarized the results of our empirical studies.

2 Related work

2.1 Large scale clustering

Many methods have been proposed to cluster large data sets [10]. Single pass approaches reduce the time necessary to cluster the data. Sampling methods reduce the computation time by clustering on a small random data subset. Core set approaches [2] define cluster centers based on exemplars using non-random, geometric techniques. Also efficient data structures like trees have been employed and more recently parallelization techniques are used [1]. Most of the existing methods for large scale clustering are based on the Euclidean distance and fail for data which are not linearly separable. Kernel methods have been introduced to many clustering algorithms to overcome this limitation but do not scale for large data [6]. Only few attempts have been made to obtain scalable kernel clustering approaches. In [11] a core-set based extension of kernel-k-means

[1] Matlab-Code is available at `http://www.cit-ec.de/tcs`

[2] Idea in [5] is theoretically equiv. to our earlier work cited in [8]

is proposed, but the convergence of this method is not guaranteed and in [5] an efficient approximate kernel-k-means algorithm was proposed, employing the Nyström technique [15]. The representation by similarities, like for kernel methods is often not accessible for domain specific (dis-)similarity measures, since the underlying measure may not be a metric and not imply a valid kernel. The obtained dissimilarity data can be clustered by dissimilarity or relational clustering techniques, like relational soft competitive learning [9] or affinity propagation (AP) [7] [3] Most of the current techniques do not scale for large data. Subsequently we derive the fast approximate soft competitive learning, which is a novel efficient batch clustering for large similarity and dissimilarity data.

2.2 Soft Competitive Learning

In contrast to regular k-means, soft competitive learning (SCL) [12] extends the quantization error to incorporate data induced neighborhood cooperation:

$$E_{\mathrm{SCL}} := \sum_{ij} h_\sigma(r_{ij}) d(\mathbf{v}_i, \mathbf{w}_j) \tag{1}$$

where $h_\sigma(t) = \exp(-t/\sigma)$ exponentially scales the neighborhood range, and r_{ij} denotes the rank of prototype $\mathbf{w}_j$ with respect to $\mathbf{v}_i$, i.e. the number of prototypes $\mathbf{w}_k$ with $k \neq j$ which are closer to $\mathbf{v}_i$ as measured by the Euclidean distance d. SCL optimizes Eq. (1) by means of a stochastic gradient descent, annealing the neighborhood range σ during training such that, in the limit, the standard quantization error is approximated [12]. The iterative adaptation rule is

$$\mathbf{w}_j := \mathbf{w_j} + \eta \cdot h_\sigma(r_{ij})(\mathbf{v}_i - \mathbf{w}_j) \tag{2}$$

where η denotes the learning rate. There exists a faster (euclidean) batch optimization scheme as introduced in [4] which in turn optimizes prototype locations and assignments similar to an EM scheme, i.e. it consecutively computes

$$\mathbf{w_j} := \sum_i h_\sigma(r_{ij}) \mathbf{v}_i / \sum_i h_\sigma(r_{ij}) \tag{3}$$

with r_{ij} based on $d(\mathbf{v}_i, \mathbf{w}_j)$.

2.3 Kernelized Soft Competitive Learning

An extension of SCL to generic similarity measures, employing kernels is used in *online* Kernelized SCL (KSCL) [14][4]. KSCL optimizes the same cost function as SCL but with the Euclidean distance substituted by a kernel induced distance. Since the feature space is unknown, prototypes are expressed implicitly as linear combination of feature vectors $\mathbf{w}_i = \sum_{l=1}^{n} \alpha_{i,l} \phi(\mathbf{v}_l)$, $\alpha_i \in \mathbb{R}^n$ is the corresponding coefficient vector. Distance in feature space for $\phi(\mathbf{v}_j)$ and $\mathbf{w}_i$ is computed as:

$$\begin{aligned} d(\phi(\mathbf{v}_j), \mathbf{w}_i) &= \|\phi(\mathbf{v}_j) - \sum_{l=1}^{n} \alpha_{i,l} \phi(\mathbf{v}_l)\|^2 \\ &= k_{j,j} - 2 \cdot \sum_{l=1}^{n} k_{j,l} \alpha_{i,l} \\ &\quad + \sum_{s,t=1}^{n} k_{s,t} \alpha_{i,s} \alpha_{i,t} \end{aligned} \tag{4}$$

The update rules of SCL can be modified by substituting the Euclidean distance by the formula (4), with the kernel $k_{i,j} = k(v_i, v_j)$ and taking derivatives with respect to the coefficients $\alpha_{i,l}$ (details see [14]).

2.4 Relational Soft Competitive Learning

Relational soft competitive learning (RSCL) as introduced in [9] assumes that a symmetric dissimilarity matrix D with entries d_{ij} describing pairwise dissimilarities of data is available. In principle, it is very similar to KSCL. There are two differences: RSCL is based on dissimilarities rather than similarities, and it solves the resulting cost function using a *batch* optimization with quadratic convergence as compared to a stochastic gradient descent.

As shown in [13], there always exists a so-called pseudo-Euclidean embedding of a given set of points characterized by pairwise symmetric dissimilarities by means of a mapping Φ, i.e. a real vector space and a symmetric bilinear form (with probably negative eigenvalues) such that the dissimilarities are obtained by means of this bilinear form. As before, prototypes are restricted to linear combinations

$$\mathbf{w}_j = \sum_l \alpha_{jl} \Phi(\mathbf{v}_l) \text{ with } \sum_l \alpha_{jl} = 1 \tag{5}$$

Dissimilarities are computed as:

$$d(\Phi(\mathbf{v}_i), \mathbf{w}_j) = [D^t \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j \tag{6}$$

where $[\cdot]_i$ refers to component i of the vector. This allows a direct transfer of batch SCL to general dissimilarities by the following iterations derived from (3)

$$\alpha_{jl} := h_\sigma(r_{jl}) / \sum_l h_\sigma(r_{jl}) \tag{7}$$

[3] AP does not rely on Euclidean similarities and can use double centered dissimilarity matrices as shown in [9]

[4] No *batch* version of KSCL was proposed so far.

with r_{jl} based on $d(\Phi(\mathbf{v}_j), \mathbf{w}_l)$. This algorithm can be interpreted as soft competitive learning in pseudo Euclidean space for every symmetric dissimilarity matrix D. If negative eigenvalues are present, however, convergence is not always guaranteed, albeit can mostly be observed in practice [9].

3 Fast generic soft competitive learning

The Nyström approximation (see e.g. [15]) substitutes a given full Gram matrix $S = (S(\mathbf{v}_i, \mathbf{v}_j))_{i,j} = (s_{ij})_{ij}$ by a low rank approximation. This may lead to linear complexity of a model depending on such a matrix. For a given kernel s, by the Mercer theorem, one can find an expansion: $s(\mathbf{w}, \mathbf{v}) = \sum_{i=1}^{\infty} \lambda_i \Phi_i(\mathbf{w}) \Phi_i(\mathbf{v})$ with eigenfunctions Φ_i and eigenvalues λ_i as solutions of an integral equation. It can be approximate with the Nyström technique, by sampling according to the distribution p: $\frac{1}{m} \cdot \sum_k s(\mathbf{w}, \mathbf{v}_k) \Phi_i(\mathbf{v}_k) \approx \lambda_i \Phi_i(\mathbf{w})$. Using the matrix eigenproblem $S^{(m)} \mathbf{U}^{(m)} = \mathbf{U}^{(m)} \Lambda^{(m)}$ of the $m \times m$ Gram matrix $S^{(m)}$, approximations for the eigenfunctions can be derived:

$$\lambda_i \approx \lambda_i^{(m)}/m \qquad \Phi_i(\mathbf{w}) \approx \sqrt{m} \cdot \vec{s}_{\mathbf{w}} \mathbf{u}_i^{(m)} \tag{8}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $\mathbf{U}^{(m)}$, and $\vec{s}_{\mathbf{w}}$ refers to the vector $(s(\mathbf{v}_1, \mathbf{w}), \ldots, s(\mathbf{v}_m, \mathbf{w}))$. Thus, for a given $N \times N$ Gram matrix, m rows and respective columns are picked randomly (landmarks) and the enumeration is changed such that these are the first m rows and columns. Rows are denoted by $S_{m,N}$ and columns by $S_{N,m}$. Hence, using the approximation (8) we obtain $\tilde{S} = \sum_{i=1}^{m} 1/\lambda_i^{(m)} \cdot S_{N,m} \cdot \mathbf{u}_i^{(m)} (\mathbf{u}_i^{(m)})^t S_{m,N}$, where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem. Hence we obtain, $S_{m,m}^{-1}$ denoting the pseudoinverse $\tilde{S} = S_{N,m} S_{m,m}^{-1} S_{m,N}$ which is of complexity $\mathcal{O}(m^3 N)$ instead of $\mathcal{O}(N^2)$, i.e. it is linear if the approximation quality m is fixed.

Similarly, dissimilarities D can be approximated if D is symmetric. Being a normal matrix, D allows a diagonalization, i.e. it can be interpreted as operator $d(\mathbf{v}, \mathbf{w}) = \sum_i \lambda_i \Phi_i(\mathbf{v}), \Phi_i(\mathbf{w})$. Therefore, the same mathematical treatment as for kernels is possible, but negative eigenvalues are allowed. For RSCL, this yields the approximation of the distance computation (6)

$$d(\mathbf{v_i}, \mathbf{w_j})^2 \approx [D_{N,m}(D_{m,m}^{-1}(D_{m,N}\alpha_j))]_i \tag{9}$$
$$-\frac{1}{2} \cdot (\alpha_j^t D_{N,m}) \cdot (D_{m,m}^{-1}(D_{m,N}\alpha_j))$$

which is $\mathcal{O}(m^3 N)$. Again, the approximation is exact if the number of samples m is chosen according to the rank of D. [5]

For similarity data we transform the similarity matrix S to a dissimilarity matrix D using Equations from [13].

$$d(\mathbf{v_i}, \mathbf{v_j})^2 = s(\mathbf{v_i}, \mathbf{v_i}) + s(\mathbf{v_j}, \mathbf{v_j}) - 2s(\mathbf{v_i}, \mathbf{v_j})$$

This is coupled with the Nyström approximation, avoiding the full calculation of the matrix S, assuming a general kernel function is given. The approach is shown in Algorithm 1. Pseudo code for RSCL is given in [9]. Hence FG-SCL is a wrapper around a modified RSCL. It processes matrices S using Alg. 1 to obtain a Nyström approximated matrix D or directly approximates D using the prior discussed Nyström approximation. The obtained approximation matrices $D_{N,m}$ and $D_{m,m}^{-1}$ are used as an input for RSCL with the distance calculations replaced by Eq. (9). The runtime-complexity of FG-SCL is dominated by the distance calculations with $\mathcal{O}(m^3 N)$. RSCL and hence FG-SCL shows fast convergence (see [9]) due to the batch approach. The memory complexity is dominated by the $m \times N$ dis-/similarity matrix. The memory complexity of FG-SCL is $\mathcal{O}(mN)$.

Algorithm 1 Pseudocode to obtain Nyström approximated dissimilarities from similarities

```
function SIMILARITY_TO_DISSIMILARITY(S,m)
    [Landmarks ] = Select_m_of_N_RandomLandmarks(N,m)
    [S_{m×N}] = (S_{L,I})_{L∈Landmarks, I∈[1:N]}
    [Diags ] = (S_{I,I})_{I∈[1:N]}
    for every L in Landmarks and I in [1 : N] do
        d(L, I)^2 = Diags_L + Diags_I − 2 · S_{L,I};
    end for
end function
```

4 Experiments

We compare the efficiency of fast generic SCL with approximated k-means and affinity propagation using Euclidean data (MNIST, USPS, SPAM) and for the dissimilarity data (CHROMO, SWISS, BACT) in comparison to AP. The MNIST data [6] contain $70000, 784$-dimensional binary images from 10 digit classes. We used a neural kernel $k(\mathbf{v_i}, \mathbf{v_j}) = \tanh(a v_i^\top v_j + b)$ with $a = 0.0045$ and $b = 0.11$ acc. to [5](AP was not applicable for this large data set). The USPS [7] contain $11000, 256$-dimensional character feature vectors from 10 classes analyzed by an RBF kernel with $\sigma = 1e-7$. The SPAM database [8], contain $4601, 57$-dimensional

[5] Samples should be representative for the full data. Thus, the Nyström method is *not* appropriate if data display e.g. a trend or the data set is already small.

[6] http://yann.lecun.com/exdb/mnist/

[7] http://www.cs.nyu.edu/~roweis/data.html

[8] http://archive.ics.uci.edu/ml/datasets

MNIST	ACC	DQE	Time
FG-SCL	**0.82 (0.11)**	**397**	1715
A-KKM	0.80 (0.03)	449	**529**
AP	n.a.	n.a.	n.a.
USPS	ACC	DQE	Time
FG-SCL	0.82 (0.01)	**105**	442
A-KKM	0.82 (0.01)	108	**109**
AP	0.76 (0.01)	424	953
SPAM	ACC	DQE	Time
FG-SCL	**0.86 (0.04)**	12492	**5**
A-KKM	0.78 (0.12)	12364	8
AP	0.76 (0.04)	**6063**	304

Table 1: Results for the similarity data. Nyström: $m = 0.1 \cdot N$ (SPAM, USPS), $m = 0.01 \cdot N$ (MNIST)

feature vectors, processed by a linear kernel to obtain the similarity matrix S. The *Copenhagen Chromosomes* data (CHROMO), consist of $4,200$ samples in 21 of gray-valued images, transformed to string distances using the edit distance (see [16]). The *SwissProt* (SWISS) data, consist of $10,988$ protein sequences, in 32 classes; dissimilarities are calculated by the Smith-Waterman algorithm (see [16]).The bacteria data (BACT), contain 2007 mass spectrometry fingerprints in 30 classes of different bacteria species, analyzed by the scoring function from [3]. For comparison with [9] we used 250 clusters for SWISS, 60 clusters for CHROMO and 30 clusters for BACT. For MNIST and USPS we used 100 clusters, each and 2 clusters for SPAM. All models are initialized randomly and trained for upto 100 iterations. Results for similarity data are shown in Tab. 1 and for dissimilarity data in Tab 2. We provide the mean dual quantization error (DQE, see [16]) , a kind of intra-cluster distance, the mean classification error by a post-labeling (majority vote) (ACC) and the mean runtime (Time) on 10 CV runs with 10 repetitions (standard deviations given in brackets). FG-SCL gives comparable good results with respect to the error-measures. Especially the quantization errors are better or similar to alternative methods. The runtime is comparable to A-KKM and better compared to AP.

5 Conclusions

We have proposed an approximation based soft-competitive learning algorithm for the analysis of similarity *and* dissimilarity data. It is efficient for large (dis-) similarity data sets. We showed that the proposed algorithm is (i) efficient in both computational and memory requirement, avoiding a full computation of the (dis-) similarity-matrix, and (ii) is able to yield similar clustering results as the a-KKM or AP. For similarity data the algorithm is equivalent to a-KKM and outperforms AP. For dissimilarity data FG-SCL is substantially faster than AP with guaranteed lower memory complexity, keeping similar good clustering results.

CHROMO	ACC	DQE	Time
FG-SCL	**0.91(0.02)**	**3971**	**186**
AP	0.90(0.00)	4711	380
SWISS	ACC	DQE	Time
FG-SCL	0.87 (0.00)	348	**2769**
AP	**0.93 (0.00)**	**337**	14162
BACT	ACC	DQE	Time
FG-SCL	**0.74 (0.02)**	**229**	**26**
AP	0.56 (0.03)	248	615

Table 2: Results for the dissimilarity data

Acknowledgment

This work was supported by the "German Science Foundation (DFG)" under grant number HA-2719/4-1. Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded by the German Excellence Initiative is gratefully acknowledged. We thank Dr. M. Kostrzewa and colleagues at Bruker Daltonik GmbH, Germany, for the BACT data and support.

References

[1] Nikolai Alex, Alexander Hasenfuss, and Barbara Hammer. Patch clustering for massive data sets. *Neurocomputing*, 72(7-9):1455–1469, 2009.

[2] Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *STOC*, pages 250–257, 2002.

[3] S. B. Barbuddhe, T. Maier, G. Schwarz, M. Kostrzewa, H. Hof, E. Domann, T. Chakraborty, and T. Hain. Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry. *Appl. a. Env. Microbio.*, 74(17):5402–5407, 2008.

[4] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.

[5] Radha Chitta et al. Approximate kernel k-means: solution to large scale kernel clustering. In Chid Apté, Joydeep Ghosh, and Padhraic Smyth, editors, *KDD*, pages 895–903. ACM, 2011.

[6] M. Filippone, F. Camastra, F. Massulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pat. Rec.*, 41:176–190, 2008.

[7] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[8] Andrej Gisbrecht, Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer. Linear time heuristics for topographic mapping of dissimilarity data. In *IDEAL'2011*, pages 25–33, 2011.

[9] B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity datasets. *Neural Computation*, 22(9):2229–2284, 2010.

[10] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31:651–666, 2010.

[11] C. Liang and D. Xiao-Ming. Scaling up kernel grower clustering method for large data sets via core-sets. *Acta Aut. Sinica*, 34(3):376–382, 2008.

[12] T. Martinetz, S. Berkovich, and K. Schulten. Neural Gas Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

[13] E. Pekalska and R. Duin. *The dissimilarity representation for pattern recognition*. World Scientific, 2005.

[14] A. Kai Qin and Ponnuthurai N. Suganthan. Kernel neural gas algorithms with application to cluster analysis. In *ICPR (4)*, pages 617–620, 2004.

[15] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS*, pages 682–688. MIT Press, 2000.

[16] X. Zhu, A. Gisbrecht, F.-M. Schleif, and B. Hammer. Approximation techniques for clustering dissimilarity data. *NeuroComputing*, 90:72–84, 2012.