ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

Patch Affinity Propagation

Xibin Zhu and Barbara Hammer *

Bielefeld University - CITEC Centre of Excellence
D-33594 Bielefeld - Germany

Abstract. Affinity propagation constitutes an exemplar based clustering technique which reliably optimizes the quantization error given a matrix of pairwise data dissimilarities by means of the max-sum algorithm for factor graphs. Albeit very efficient for sparse matrices, it displays squared complexity in the worst case, hence it is not suited as high throughput method due to time and memory constraints. We propose an extension of affinity propagation to patch clustering such that data are treated in chunks of fixed size with limited memory requirements and linear time. We test the suitability of the approach for two biomedical applications.

1 Introduction

The increasing size and complexity of modern data sets turns data mining techniques to indispensable tools when inspecting biomedical data. Clustering constitutes one of the most important tasks to allow experts to rapidly get an overview about essential structure. Thereby, it is very important that clusters can directly be presented to experts in reasonable form. Prototype based techniques offer one very striking possibility for this purpose since clusters are represented by prototypes in the original data space; hence they can directly be inspected in the same way as given data. A variety of excellent prototype based techniques exist such as the self-organizing map or neural gas [11, 8].

Dedicated data formats and detailed information about the data structures often cause the need for problem specific similarities or dissimilarities instead of the standard Euclidean norm. Examples include DNA sequences or protein structures, biological networks, mass spectra, medical images, database entries including different types of data, etc. In consequence, standard prototype based clustering which expects data to be represented as Euclidean vectors cannot be applied to such settings. As an alternative, data can be represented in terms of pairwise similarities or dissimilarities given by dedicated distance measures such as alignments or kernels. A number of clustering techniques which rely on similarities or dissimilarities only have been proposed such as standard hierarchical methods, or extensions of prototype based techniques such as pairwise data clustering with deterministic annealing, median clustering, and relational or kernel variants of neural gas, the self-organizing map, and generative topographic mapping [7, 3, 5, 6]. Median or exemplar based clusterings restrict prototype locations to data positions. This has two benefits: standard evaluation measures as provided e.g. by the quantization error are still well defined in such settings. In addition, a direct inspection of the prototypes is possible by means of the corresponding data points. However, due to the discrete space of the prototypes, some effort has to be done to arrive at suitable optimization techniques.

*This work has been supported by the DFG under grant number HA2719/4-1 and by the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

Recently, a very promising method has been proposed [4]: Affinity propagation (AP) reformulates the quantization error such that it can be formalized as a factor graph, for which powerful optimization techniques such as the max-sum algorithm are readily available. Original AP requires all similarities to be given a priori (missing ones are interpreted as minimum similarity) and it scales, in the worst case, quadratic in the number of data points. This fact makes the algorithm unsuitable for large data sets or high throughput analysis. This scaling behavior is shared by many algorithms which rely on pairwise similarities or dissimilarities since the corresponding dissimilarity matrix has squared size.

A number of approximations to get around this problem have recently been proposed. On the one hand, kernel approaches can often be speeded up to linear techniques by means of the Nyström approximation [13]. However, a good result requires a representative set of examples, further, an integration into AP is not possible. As an alternative, patch processing has been proposed in the context of neural gas and relational neural gas [1, 6] which processes data subsequently in patches of fixed size, thereby integrating compressed information of all already seen data in terms of the prototypes. This method has the advantage that it works very well also if data display a trend, which is often the case for streaming data or very large data sets. In this contribution, we extend this technique to AP and we test its suitability in two examples from the biomedical domain.

2 Affinity Propagation

Affinity propagation (AP) constitutes an exemplar based clustering method. Assume data points v_i are characterized in terms of pairwise similarities $s_{ij} = s(v_i, v_j)$. Prototypes w_j are chosen as exemplars, i.e. $w_j \in \{v_i \mid i = 1, \ldots, N\}$. Every prototype defines its receptive field

$$R(j) = \{v_i \parallel s(v_i, w_j) > s(v_i, w_{j'}) \text{ for } j' \neq j\}$$

The goal is to find prototypes such that the quantization factor

$$E_{\text{qe}} := \sum_j \sum_{v_i \in R(j)} s(v_i, w_j)$$

is optimized. AP reformulates this objective in the following form:

$$E_{\text{qe-ap}} := \sum_i s(v_i, w_{I(i)}) + \sum_i \delta_i(I)$$

where the assignment function I assigns a prototype given by an exemplar $v_{I(i)}$ to every data point v_i, and $\delta_i(I)$ punishes invalid assignments, i.e. situations where $I(j) = i$ but $I(i) \neq i$. This leads to the punishment ∞ in case of invalid assignments and $\delta_i(I)$ is 0, otherwise. By taking the assignment according to the receptive fields, we obtain the quantization factor. This alternative formulation, however, has the benefit that it is no longer necessary to specify the number of clusters, rather the number is determined by the self-similarities of the data $s(v_i, v_i)$ which indicate in how far the point v_i is available as a prototype. Typically, these self-similarities are set to the medium of the given similarities.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

To numerically solve this optimization problem, the function $E_{\mathrm{qe-ap}}$ can be reformulated as a factor graph which can approximately be optimized by means of the max-sum algorithm. In turn, responsibilities

$$r_{ij} = s_{ij} - \max_{j' \neq j} \{a_{ij'} + s_{ij'}\}$$

and availabilities

$$\begin{aligned} a_{ij} &= \min\{0, r_{jj} + \sum_{i' \neq i,j} \max\{0, r_{i'j}\}\} \\ a_{ii} &= \sum_{i' \neq i} \max\{0, r_{i'i}\} \end{aligned}$$

are determined leading to assignments $I(i) = \mathrm{argmax}_j(a_{ij} + r_{ij})$ [4].

3 Patch Affinity Propagation

Patch processing processes data consecutively in patches of small size m. All already visited data are compressed by means of prototypes. These serve as additional inputs to the next patch, counted with their respective multiplicities as given by the size of their receptive fields. For this purpose, an extension of the clustering method which takes into account multiple points is necessary. Since all data are taken into account this way, processing of data sets which are not i.i.d. is possible. This procedure has been proposed for vectorial neural gas in [1] and relational neural gas for dissimilarity data in [6].

We extend AP towards patch clustering, resulting in patch affinity propagation (PAP). Since AP constitutes an exemplar based clustering method, the transfer of the meta algorithm is immediate: we just repeatedly apply AP to a given patch and the exemplars as provided by the previous patch, counted with multiplicities. This meta-algorithm is denoted in Fig. 1. Obviously, the algorithm visits only a linear subset of the full similarity matrix, whereby the exact parts depend on the found exemplars. This way, a linear time and limited memory algorithm is obtained provided access to the similarities is $\mathcal{O}(1)$. In practical applications, it is often the case that the full dissimilarity matrix is not given anyway, rather data are represented in some form (such as e.g. DNA sequences in a database in bioinformatics applications), and a method to compute dissimilarities given two data points is available (such as sequence alignment for DNA sequences). PAP offers an ideal solution for such setting: Since only a subpart of the matrix is necessary, only a small fraction of the dissimilarity matrix has to be computed on the fly while performing the algorithm.

To apply patch processing to AP, we need to extend AP such that it can deal with data points which are contained in the training set more than once (multiple data points). One simple way to do so would exist in a simulation of the update equations provided by standard AP, where points are directly included in the updates according to their multiplicities. Unfortunately, this procedure is not possible, since several exactly identical data points prevent convergence of AP – the intuitive reason being that the algorithm cannot decide which point in a set of exactly identical points should become the exemplar.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

init exemplars $E = \emptyset$, patch number $p = 1$
repeat
compute patch of size m: $P_{m,m} = \{s(v_i, v_j)\} \mid p \cdot m < i, j \leq (p+1) \cdot m\}$
compute similarities patch - exemplars:
$P_{m,|E|} = \{s(v_i, v_j) \mid p \cdot m < i \leq (p+1) \cdot m, v_j \in E\}$
compute similarities of exemplars: $P_{|E|,|E|} = \{s(v_i, v_j) \mid v_i, v_j \in E\}$
set $P := \begin{pmatrix} P_{m,m} & P_{m,|E|} \\ P^t_{m,|E|} & P_{|E|,|E|} \end{pmatrix}$
init diagonal entries of P
set multiplicities according to E if $v_i \in E$
set multiplicities to 1 for $v_i \notin E$
perform patch clustering with multiplicities for P
$\rightarrow$ this yields new exemplars E
set multiplicities in E as size of the receptive fields (with multiplicities)

Fig. 1: Principled algorithm for patch clustering

Thus, we use the following observation to extend AP to multiple points: if data point v_i is contained in the data set m_i times, the cost function becomes

$$E_{\text{qe}} = \sum_{i,j:v_i \in R(j)} m_i \cdot s(v_i, w_j)$$

Obviously, this cost function is obtained if the original similarities $s(v_i, v_j)$ are substituted by $m_i \cdot s(v_i, v_j)$. Hence we simply use these products of similarities and multiplicities in the update equations for the responsibilities as computed in AP. Note that the resulting AP algorithm tries to optimize a cost function which has the same global optima as the one beforehand. The exact behavior of the resulting AP algorithm is not identical to the original AP algorithm applied for multiple points, in particular convergence is given for the former case.

The initialization of the diagonal terms should also be adapted accordingly, putting a bias towards points with large multiplicities. We achieve this by a division of the preferences along the diagonal by the respective multiplicities. Obviously, PAP converges since AP does. However, unlike AP, the full procedure can no longer be interpreted as optimization of one cost function due to the iterative approach, rather an approximation thereof. Note that it is not necessary to randomly permute data when using PAP since already seen data are always incorporated in compressed form by means of the prototypes. See [1] for a demonstration of the suitability of patch clustering for non i.i.d. data sets.

4 Experiments

We test the algorithm for two data sets stemming from biomedical applications. In both cases, data are labeled such that we can evaluate the performance of the algorithms in terms of the classification error:

- The Copenhagen Chromosomes data constitute a benchmark from cytogenetics [9]. 4,200 human chromosomes from 22 classes (the autosomal chromosomes) are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings are compared using edit distance with insertion/deletion costs 4.5 [12].

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

- The vibrio data set consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra encounter approx. 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The mass spectra are preprocessed with a standard workflow using the BioTyper software [10]. As usual, mass spectra display strong functional characteristics due to the dependency of subsequent masses, such that problem adapted similarities such as described in [2, 10] are beneficial. In our case, similarities are calculated using a specific similarity measure as provided by the BioTyper software [10].

In both cases, we report the results of a ten fold cross-validation with 10 repeats. We compare to relational neural gas (RNG) and patch relational neural gas (PRNG) as presented in [6]. The space and time complexity of PAP and PRNG is comparable (linear for the time complexity and constant for the memory requirement). Implementing different optimization techniques for the clustering objective, their suitability is different depending on the structure of the data, AP being more appropriate for possibly multimodal clearly separated clusters, while NG is better suited for noisy data. The parameter choices are as follows:

- Affinity Propagation (AP): self-similarities are set by binary search such that a fixed number of exemplars is appropriately obtained, starting from the median, where the number of clusters is comparable to the number used for the alternative methods. (We accept partial deviations thereof, an exact fit of a given number of clusters often requiring additional trials.) Adaptation is done until convergence is observed.

- Patch Affinity Propagation (PAP): patch size is 100 and we use the same strategy to determine the self-similarities as AP.

- Relational Neural Gas (RNG): we use a fixed number of neurons K, and an exponential annealing schedule for the neighborhood range starting from $K/2$. 100 epochs are used for training.

- Patch Relational Neural Gas (PRNG): we use the same parameters as RNG for each patch, and patch size 100. The number of coefficients used for the k-approximation of the prototype vectors is set to $k = 10$.

We use the standard double centering to transform similarities to dissimilarities since NG requires dissimilarities rather than similarities, as AP. Prototypes are labeled using posterior labeling according to majority vote on the training data.

The results of the algorithms are displayed in Tab. 1. Since patch clustering relies on a small subset of the full information, it can be expected that it leads to information loss which depends on the data characteristics and the algorithm. Interestingly, depending on the data characteristics, this loss is not very severe such that patch clustering can be seen as a valid alternative if time and memory constraints are given, or data cannot be processed at all due to its size. For the Chromosomes data, PAP leads to a decrease of the classification accuracy by more than 10% compared to AP, while PRNG and RNG display both a quite respectable accuracy of about 90%. In contrast, AP and PAP display results close to 100% for the Vibrio data set. Here, AP displays very good results since it reliably finds exemplars which represent the well separated clusters present in the data. The same holds for the PAP approximation albeit in much less effort.

ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2011, i6doc.com publ., ISBN 978-2-87419-044-5. Available from http://www.i6doc.com/en/livre/?GCOI=28001100817300.

	Classification Accuracy	**Clusters**
Chromosome		
AP	0.895(0.006)	$K = 58$
PAP	0.755(0.008)	$K = 66$
RNG	0.910(0.004)	$K = 60$
PRNG	0.879(0.018)	$K = 60$
Vibrio		
AP	0.999(0.000)	$K = 49$
PAP	0.999(0.000)	$K = 53$
RNG	0.894(0.008)	$K = 50$
PRNG	0.887(0.028)	$K = 50$

Table 1: Results on the Chromosomes and Vibrio data set, the standard deviation is shown in parenthesis

5 Conclusions

We extended AP to patch clustering such that a linear time constant memory algorithm results, which seems particularly suited for large data sets where a comparably well cluster separation can be observed. Further tests for large size benchmarks are the subject of ongoing work.

References

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

[2] 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. *Applied and Environmental Microbiology*, 74(17):5402–5407, 2008.

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

[4] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[5] A. Gisbrecht, B. Mokbel, and B. Hammer. Relational generative topographic map. In M. Verleysen, editor, *ESANN'10*, pages 277–282. D side, 2010.

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

[7] J. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[8] T. Kohonen. *Self-Organizing Maps*. Springer, 3rd edition, 2001.

[9] C. Lundsteen, J-Phillip, and E. Granum. Quantitative analysis of 6985 digitized trypsin g-banded human metaphase chromosomes. *Clinical Genetics*, 18:355–370, 1980.

[10] T. Maier, S. Klebel, U. Renner, and M. Kostrzewa. Fast and reliable maldi-tof ms–based microorganism identification. *Nature Methods*, (3), 2006.

[11] 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.

[12] M. Neuhaus and H. Bunke. Edit distance based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006.

[13] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.