ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Curvilinear Components Analysis and Bregman Divergences

Jigang Sun, Malcolm Crowe and Colin Fyfe
Applied Computational Intelligence Research Unit,
The University of the West of Scotland.
Email: Jigang.Sun,Malcolm.Crowe,Colin.Fyfe@uws.ac.uk

Abstract. Curvilinear Component Analysis (CCA) is an interesting flavour of multidimensional scaling. In this paper one version of CCA is proved to be related to the mapping found by a specific Bregman divergence and its stress function is redefined based on this insight, and its parameter (the neighbourhood radius) is explained.

1 Introduction

Multidimensional scaling (MDS) [2] is an exploratory data investigation method in which each data sample is represented by a latent point in a lower dimensional space such that the layout of the latent points best represents the relative layout of the original data points i.e. the distances between latent points best mirrors the distances between the original data points. Usually the latent space is two dimensional so that they can be investigated by eye by researchers. In this paper we will investigate a variety of MDS known as Curvilinear Component Analysis (CCA). The notation used in this paper is as follows. X stands for original data set, Y stands for the configuration of X in the latent space. D stands for the distance matrix in data space and L stands for the distance matrix in latent space. X_i stands for data point i in data space and Y_i stands for projected point in latent space. D_{ij} stands for the interpoint distance between X_i and X_j; L_{ij} stands for the interpoint distance between Y_i and Y_j. $i, j = 1, 2, \cdots, N$.

In section 2 CCA is briefly reviewed and a discrepancy between stress function definition and corresponding algorithm is noted; in section 3 a new stress function that is consistent with the algorithm is proposed; finally in section 4 the advantages of the new stress function and the 'neighbourhood radius' parameter are explained.

2 Curvilinear Component Analysis

Curvilinear Component Analysis(CCA) [3] is good at unfolding strongly nonlinear or even closed structures, which means it allows stretching of long distances. There are three versions of CCA, the first two versions were proposed in [3] where both stress function and optimisation algorithm are described, the third one is an enhanced version proposed in [4] for noisy data set. Due to limited space, in this paper only one early version in [3] is discussed. The stress function

to be minimised is defined as

$$E_{CCA}(Y) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(L_{ij}-D_{ij})^2 w(L_{ij},\lambda) \quad (1)$$

The weight function $w(L_{ij},\lambda)$ has as argument the interpoint distance in latent space rather than data space as with most MDS algorithms. There are two versions

- global weighting with $w(L_{ij},\lambda) = e^{-\frac{L_{ij}}{\lambda}}$
- local weighting which uses a step function with $w(L_{ij},\lambda) = 1$ if $L_{ij} < \lambda$, and 0 otherwise.

λ is called the 'neighbourhood radius', which decreases with time during the simulation.

The optimisation method is a kind of stochastic gradient descent. Firstly the whole stress is broken into parts, (1) is rewritten as $E_{CCA}(Y) = \sum_{i=1}^{N} E^i_{CCA}(Y)$. The value of i is randomly selected and the data point X_i is fixed. Then updating rule for Y_j is

$$\mathbf{Y_j} \leftarrow \mathbf{Y_j} - \alpha\frac{\partial E^i_{CCA}(Y)}{\partial \mathbf{Y_j}} = \frac{\partial E^i_{CCA}(Y)}{\partial L_{ij}}\frac{\partial L_{ij}}{\partial \mathbf{Y_j}} \quad (2)$$

where

$$\frac{\partial E^i_{CCA}(Y)}{\partial L_{ij}} = (L_{ij}-D_{ij})w(L_{ij},\lambda) + \frac{(L_{ij}-D_{ij})^2}{2}\frac{\partial w(L_{ij},\lambda)}{\partial L_{ij}} \quad (3)$$

, $\frac{\partial L_{ij}}{\partial \mathbf{Y_j}} = \frac{\mathbf{Y_j}-\mathbf{Y_i}}{L_{ij}}$, α is a decreasing factor.

Although the expression (3) is not complicated, the authors still consider a 'quantized version' [3] of the weight function, in which $\frac{\partial w(L_{ij},\lambda)}{\partial L_{ij}} = 0$, the second term of (3) is discarded to get

$$\frac{\partial E^i_{CCA}(Y)}{\partial L_{ij}} = (L_{ij}-D_{ij})w(L_{ij},\lambda) \quad (4)$$

and the updating rule (2) is derived as $\mathbf{Y_j} \leftarrow \mathbf{Y_j} - \alpha\frac{(L_{ij}-D_{ij})}{L_{ij}}w(L_{ij},\lambda)(\mathbf{Y_j}-\mathbf{Y_i})$. From the above we can see that each time one data point is randomly chosen and is fixed, all the other points are relocated with respect to this fixed point. When one point is relocated, only the value of the fixed point i and the current point j affect relocation of j, and since i is constant at this stage, the stress function is actually a p dimensional function in point j where p is the dimensionality of the latent space. If $p = 2$, during the optimisation process the stress function is treated as two dimensional.

However for the global version, $\frac{\partial w(L_{ij},\lambda)}{\partial L_{ij}} = -\frac{1}{\lambda}e^{-\frac{L_{ij}}{\lambda}}$, the solution to the simplified version is that $L_{ij} = +\infty$, which is obviously not a solution we desire.

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

When the simplification is made, the weight function is in fact changed and thus the stress function is accordingly different but neither the new weight function nor the new stress function is given in [3].

The local weighting version receives more attention in the literature, such as in [6] its stress function is redefined as well as the updating rule. In this paper we only focus on the global weighting and we take the weight function $w(L_{ij}, \lambda)$ as the global function in the rest of the paper. Accordingly the stress function (1) is called Basic CCA

$$E_{Basic}(Y) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(L_{ij} - D_{ij})^2 e^{-\frac{L_{ij}}{\lambda}} \quad (5)$$

We now show that the new stress is a sum of Bregman Divergences.

3 Right Bregman divergences and CCA

Consider a strictly convex function $F : S \to \Re$ defined on a convex set $S \subset \Re^d$. A Bregman divergence [1] between two points, $\mathbf{p}$ and $\mathbf{q} \in S$, is defined to be

$$d_F(\mathbf{p}, \mathbf{q}) = F(\mathbf{p}) - F(\mathbf{q}) - \langle(\mathbf{p} - \mathbf{q}), \nabla F(\mathbf{q})\rangle \quad (6)$$

where the angled brackets indicate an inner product and $\nabla F(\mathbf{q})$ is the derivative of F evaluated at $\mathbf{q}$. This can be viewed as the difference between $F(\mathbf{p})$ and its truncated Taylor series expansion around $\mathbf{q}$. Thus it can be used to 'measure' the convexity of F.

When F is in one variable, the divergence (6) is expressed as $d_F(p, q) = F(p) - F(q) - \frac{dF}{dq}(p - q)$. We have in [7], used the residuals of the Taylor series $d_F(p, q) = \frac{d^2F}{dq^2}\frac{(p-q)^2}{2!} + \frac{d^3F}{dq^3}\frac{(p-q)^3}{3!} + \frac{d^4F}{dq^4}\frac{(p-q)^4}{4!} + \cdots$ to investigate Sammon mappings [5]with left Bregman divergences.

Bregman divergences have the following useful properties: $d_F(\mathbf{p}, \mathbf{q}) \geq 0$ and $d_F(\mathbf{p}, \mathbf{q}) = 0 \longleftrightarrow \mathbf{p} = \mathbf{q}$ and $d_F(\mathbf{p}, \mathbf{q}) \neq d_F(\mathbf{q}, \mathbf{p})$ except in special cases such as $F(.) = \| \, . \, \|^2$, the Euclidean norm.

In [7], we show how e.g. the Sammon mapping and other metric multidimensional scaling can be generalised with *left* Bregman divergences. We now link CCA with *right* Bregman divergences to get the real stress function. Let us select a base convex function, $F(x) = e^{-\frac{x}{\lambda}}$. Then

$$\begin{aligned} & E_{Real}(Y) \\ = \; & \lambda^2 \sum_{i=1}^{N}\sum_{j=1}^{N} d_F(D_{ij}, L_{ij}) = \lambda^2 \sum_{i=1}^{N}\sum_{j=1}^{N} (F(D_{ij}) - F(L_{ij}) - (D_{ij} - L_{ij})\nabla F(L_{ij})) \\ = \; & \lambda^2 \sum_{i=1}^{N}\sum_{j=1}^{N} \left(e^{-\frac{D_{ij}}{\lambda}} - e^{-\frac{L_{ij}}{\lambda}} + (D_{ij} - L_{ij})\frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda} \right) = \sum_{i=1}^{N}\sum_{j=1}^{N} T_{Real}(L_{ij}) \end{aligned} \quad (7)$$

Defining $E^i_{Real}(Y) = \sum_{j=1}^{N} T_{Real}(L_{ij})$, then

$$\frac{\partial E^i_{Real}(Y)}{\partial L_{ij}} = (L_{ij} - D_{ij})e^{-\frac{L_{ij}}{\lambda}} = (L_{ij} - D_{ij})w(L_{ij}, \lambda) \quad (8)$$

We can now equate (4) with (8). The stress function (7) in fact should be $E_{Real} + C, C$ a constant that does not affect mapping result so we set $C = 0$. Thus (7) is the real stress function that creates (4). λ^2 is only for qualitative comparison purposes and has no effect on the mapping result.

4 Comparison between the Real CCA and the Basic CCA

The Basic CCA (5) is rewritten as $E_{Basic}(Y) = \sum_{i=1}^{N}\sum_{j=1}^{N} T_{Basic}(Y)$. T_{Basic} is the stress for distance D_{ij}. The solutions for $\frac{dT_{Basic}(Y)}{dL_{ij}} = (L_{ij} - D_{ij})e^{-\frac{L_{ij}}{\lambda}}\left(1 - \frac{L_{ij}-D_{ij}}{2\lambda}\right) = 0$ are $L_{ij} = D_{ij}$ and $L_{ij} = D_{ij} + 2\lambda$. λ determines the width and height of stress curve as depicted in Figure 1(a). When λ is reduced from 2 to 1.5 the right part of the graph becomes shorter and narrower as can be seen in the graph.

It is clear from the graph that the stress function is monotonically decreasing with respect to the latent space distances $L_{ij} \in [0, D_{ij}]$; it is increasing for $L_{ij} \in [D_{ij}, D_{ij} + 2\lambda]$; after the peak $L_{ij} = D_{ij} + 2\lambda$ the stress decreases with distance. The parameter λ has slight influence on the graph $L_{ij} \in [0, D_{ij}]$ but determines the width and height of graph for $L_{ij} > D_{ij}$. Obviously when data are configured infinitely far apart, $L_{ij} = +\infty$, the stress function is minimised to the lowest value, zero. But this does not give us a solution with which to best visualise the data. The acceptable solution must be a configuration that is a *local minimum* instead of the global minimum.

The shape of the stress function causes potential problems during optimisation using any gradient descent method. For example even if the output initialisation guarantees that $L_{ij} < D_{ij} + 2\lambda$, the mapped distances can possibly climb over the peak and become larger and larger which is a good reason for [3] to make the simplification (4). To prevent this from happening, we propose a change to the global weight version of the Basic CCA $w(L_{ij}, \lambda) = e^{-\frac{L_{ij}}{\lambda}}$ if $L_{ij} < D_{ij} + 2\lambda$, 0 if $L_{ij} \geq D_{ij} + 2\lambda$. We have shown empirically that this change has made the Basic CCA better.

The Real CCA We may alternatively express (7) as the tail of the Taylor series, given $\frac{\partial F^{(n)}(x)}{\partial x^n} = \left(\frac{-1}{\lambda}\right)^n e^{-\frac{x}{\lambda}}$ which gives
$T_{Real}(Y) = e^{-\frac{L_{ij}}{\lambda}}\frac{(D_{ij}-L_{ij})^2}{2!} - \frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda}\frac{(D_{ij}-L_{ij})^3}{3!} + \frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda^2}\frac{(D_{ij}-L_{ij})^4}{4!} + \cdots$
$= T_{Basic} - \frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda}\frac{(D_{ij}-L_{ij})^3}{3!} + \frac{e^{-\frac{L_{ij}}{\lambda}}}{\lambda^2}\frac{(D_{ij}-L_{ij})^4}{4!} + \cdots$. So we can see that (7) is an extension of (5), and (5) is an approximation of (7). Figure 1(b) compares the Basic CCA and the Real CCA by T_{Basic} and T_{Real}. For $L_{ij} < D_{ij}$, T_{Real} is slightly lower than T_{Basic}; while for $L_{ij} > D_{ij}$ T_{Real} is much greater than T_{Basic}. It is also worth noting that the graph for Real CCA is not symmetric; the left part where $L_{ij} < D_{ij}$ (projection) is steeper than the right part where $L_{ij} > D_{ij}$

(unfolding) as illustrated in Figure 1(c). The same distance error $|E_{ij}|, E_{ij} = L_{ij} - D_{ij}$, in the right part $T_{real}|_{L_{ij}>D_{ij}}$, contributes less stress than in the left $T_{real}|_{L_{ij}<D_{ij}}$. This is the reason the distances tend to be mapped longer. Like the Sammon mapping, small distances are focused on. From $T_{Real}(L_{ij}) = \lambda^2 e^{-\frac{D_{ij}}{\lambda}}(1 - e^{-\frac{E_{ij}}{\lambda}} - \frac{E_{ij}e^{-\frac{E_{ij}}{\lambda}}}{\lambda})$, for the same distance error E_{ij},we can see that the smaller the D_{ij} is, the higher the stress is, as illustrated in Figure 1(d). The neighbourhood radius λ indicates a *inflection point*. The root for $\frac{d^2T_{Real}(L_{ij})}{dL_{ij}^2} = e^{-\frac{L_{ij}}{\lambda}}(\frac{D_{ij}+\lambda-L_{ij}}{\lambda}) = 0$ is $L_{ij} = D_{ij} + \lambda$. For $L_{ij} \geq D_{ij} + \lambda$ the stress starts to grow at reducing speed until it levels off. Thus distances which are mapped longer than their original values are not seriously penalised in stress. As with the Basic CCA, λ also has serious consequences on the right part of the stress graph as illustrated in Figure 1(e). When λ decreases from 2 to 1, for $L_{ij} < D_{ij}$ the stress decreases a little; but for $L_{ij} > D_{ij}$ the stress reduces sharply, and the width of the graph is also reduced. When $\lambda = 2, D_{ij} < \lambda$, the stress for the leftmost is lower than the rightmost; it is the opposite situation for $\lambda = 1$ where $D_{ij} > \lambda$; when $D_{ij} = \lambda = 1.5$, they are equal. Figure 1(f) shows that stress for the right part reduces much faster than the left part with the decrease of λ.

5 Conclusion

This paper has investigated CCA in the context of Bregman divergences. We have shown how Bregman divergences enable us to investigate some discrepancies in the CCA method. Clearly the type of analysis performed in this paper can be also performed on other MDS methods such as the Sammon mapping but this is the subject of another paper. Clearly also there are many other optimization techniques that can be used but again space has constrained us to the stochastic optimisation algorithm.

References

[1] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[2] I. Borg and P.J.F. Groenen. *Modern multidimensional scaling*. New York: Springer, 2nd edition, 2005.

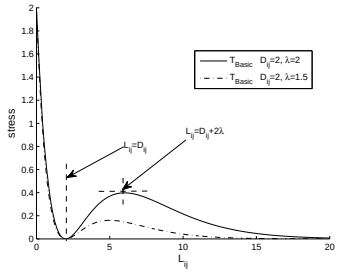
[3] Pierre Demartines and Jeanny Hérault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1), 1997.

[4] Jeanny Hérault, Claire Jausions-Picaud, and Anne Guérin-Dugué. Curvilinear component analysis for high-dimensional data representation: I. theoretical aspects and practical use in the presence of noise. In *IWANN (2)*, pages 625–634, 1999.

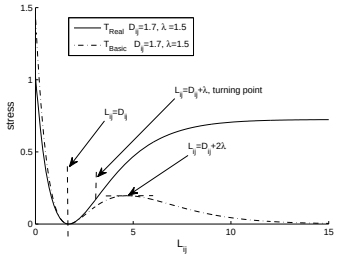
[5] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computing*, 18, 1969.

[6] John Aldo Lee and Michel Verleysen. Simbed: Similarity-based embedding. In *ICANN (2)*, pages 95–104, 2009.

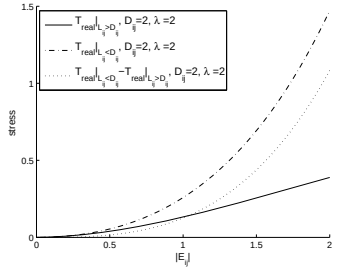
[7] Jigang Sun and Colin Fyfe. Extending metric multidimensional scaling with Bregman divergences. Technical report, University of the West of Scotland, September 2009.



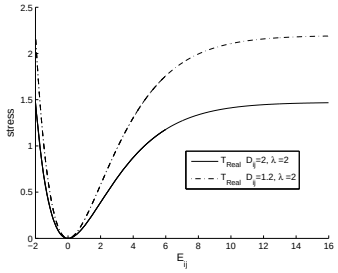
(a) Stress graph for the Basic CCA



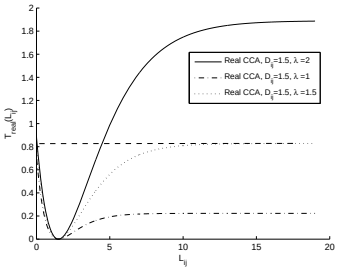
(b) stress of the Real CCA is greater than the Basic CCA when $L_{ij} > D_{ij}$.



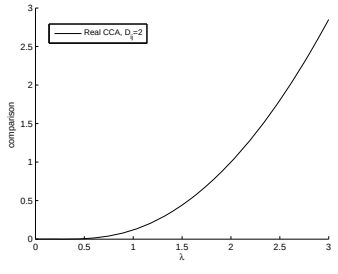
(c) Stress comparison with respect to the same distance error when projection and unfolding.



(d) Given the same λ and same distance error, the stress for $D_{ij} = 1.2$ is higher than the stress for $D_{ij} = 2$.



(e) λ influences on the right part of the stress graph.



(f) comparison= $\frac{\lim_{L_{ij}\to+\infty} T_{Real}(L_{ij})}{\lim_{L_{ij}\to 0} T_{Real}(L_{ij})}$

Fig. 1: Basic CCA vs Real CCA