Neural Networks for Efficient Nonlinear Online Clustering

Yanis Bahroun[1], Eugénie Hunsicker[2], and Andrea Soltoggio[1]

[1] Department of Computer Science, Loughborough University
[2] Department of Mathematics, Loughborough University
Loughborough, Leicestershire, United Kingdom
{y.bahroun}@lboro.ac.uk

To appear in: Proceedings of the 24th International Conference On Neural Information Processing (ICONIP 2017), Guangzhou, China, November 14-18, 2017.

Abstract. Unsupervised learning techniques, such as clustering and sparse coding, have been adapted for use with data sets exhibiting nonlinear relationships through the use of kernel machines. These techniques often require an explicit computation of the kernel matrix, which becomes expensive as the number of inputs grows, making it unsuitable for efficient online learning. This paper proposes an algorithm and a neural architecture for online approximated nonlinear kernel clustering using any shift-invariant kernel. The novel model outperforms traditional low-rank kernel approximation based clustering methods, it also requires significantly lower memory requirements than those of popular kernel k-means while showing competitive performance on large data sets.

Keywords: Nonlinear kernel, Clustering, Hebbian learning, Neural networks.

1 Introduction

Most existing high-performing neural networks rely on offline learning and perform poorly in online learning from streamed data. Biological systems, on the contrary, learn from continuous streams of data providing inspiration principles on how to accomplish this task efficiently. Two bio-inspired principles that can be implemented into artificial neural networks are synaptic plasticity [1], hypothesized to be a key factor for human learning and memory, and sparse coding [2, 3] stating that the brain encodes the sensory inputs within the smallest number of active neurons. These two principles can be modeled in machine learning using Oja's [1] and Sanger's [4] rules. These rules are inspired by the Hebbian principle, which states that connections between two units, e.g., neurons, are strengthened when simultaneously activated, and can be implemented by feed-forward and lateral inhibitory connections as shown in [5]. The continuous update dynamic of Hebbian learning makes this rule suitable for learning from a continuous stream of data. The system learns from one input at a time with memory requirements that are independent of the number of samples.

To achieve good clustering and classification performance with data that are not linearly separable in their original Euclidean coordinates, offline systems have largely employed kernel methods. However, such methods come with a large computational cost when the data size increases, especially in the case of unbounded streams of data.

To address this problem, an online linear kernel clustering and sparse coding method was recently proposed in [6]. That method used Hebbian/anti-Hebbian learning rules derived from a cost-function minimization based on the kernel associated with the inner-product on Euclidean spaces, which is therefore restricted to linearly separable data sets.

The primary innovation of this paper is the introduction of nonlinear online kernel clustering by means of a Hebbian/anti-Hebbian neural network. This is implemented through the use of Random Fourier Features and Classical MultiDimensional Scaling (CMDS). The proposed model has been evaluated against existing online k-means on both artificial and real nonlinear publicly available data sets and has been benchmarked against a set of offline kernel methods. The results demonstrate that the proposed model achieves for the first time efficient online kernel clustering using a neural network trained by Hebbian/anti-Hebbian rules.

2 Background and Related Work

The Hebbian/anti-Hebbian learning rules implemented in the proposed model derive from a generalization of CMDS, originally used for low-dimensional embedding of data [7]. The formulation of CMDS is given as follows: for a set of inputs $x^t \in \mathbb{R}^n$ for $t \in \{1, \dots, T\}$, the concatenation of the inputs defines an input matrix $X \in \mathbb{R}^{n \times T}$. The output matrix Y of embeddings is an element of $\mathbb{R}^{m \times T}$ where $m < n$ for low-dimensional embedding. The objective function of CMDS is:

$$Y^* = \underset{Y \in \mathcal{C}}{\arg\min} \|X'X - Y'Y\|_F^2 \quad . \tag{1}$$

where F is the Frobenius norm, $X'X$ is the Gram matrix of the inputs that combines the information of similarity and norm of the vectors, and the space $\mathcal{C}$ encodes the constraints, which depends on the problem to solve. This has been generalized to sparse coding [6] using a non-negativity constraint on the output matrix, $Y \in \mathbb{R}_+^{m \times T}$, called Non-negative CMDS (NCMDS).

A solution to the optimization problem for online NCMDS was introduced in [6], which led to a neural implementation and Hebbian learning rules for this method. The model in [6], based on Eq.1, however, has a linear structure encoded in the inner-product term $X'X$, which fails to capture the often nonlinear structure of real world data.

A way to address this problem can be found in kernel methods since they allow algorithms to be applied to implicit high-dimensional nonlinear feature spaces. The matrix $X'X$, in Eq.1, with i, j^{th} entries given $\langle x^i, x^j \rangle_{\mathbb{R}^n}$, can be replaced with any nonlinear kernel matrix $K := K(x^i, x^j)$:

$$Y^* = \underset{Y \in \mathbb{R}_+^{m \times T}}{\arg\min} \|K - Y'Y\|_F^2. \tag{2}$$

A version of Kernel MDS was suggested in [8], but the approach is for offline training only and does not perform clustering or sparse coding but dimensionality reduction.

2.1 Random Fourier Features

The increase in performance using kernel methods comes at a large computational cost when the number of samples increases. Two main approaches address this problem: 1) data dependent procedures based on a low-rank approximation of the kernel matrix, e.g. the Nyström method [9]; 2) data independent procedures based on integral representations of the kernel function, e.g., Random Fourier Features (RFF) [10]. This second approach is used in this study to approximate shift-invariant kernels such as the Gaussian kernel, where $K(x^i, x^j) := e^{\frac{-\|x^i - x^j\|^2}{2\sigma^2}}$.

Assume that K is a continuous positive-definite, shift-invariant kernel, i.e., $K(x, y) = k(x - y)$ with x and y vectors of $\mathbb{R}^n$, and is scaled such that $k(0_{\mathbb{R}^n}) = 1$. Because K is positive semi-definite, Aronszajn's theorem [11] implies that there exist a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathbb{R}^n \to \mathcal{H}$ such that for any x and $y \in \mathbb{R}^n$, then

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}. \tag{3}$$

For general kernels, explicitly defining Φ and $\mathcal{H}$ can prove challenging. However, when K is shift-invariant and scaled as above, Bochner's theorem [12] states that the Fourier transform of k, $\hat{k}$, is a probability density function in the Fourier dual space, in this case again $\mathbb{R}^n$, with the property that for $w \in \mathbb{R}^n$:

$$K(x, y) = \mathbb{E}_{\hat{k}}[f(w, x)' f(w, y)] \text{ where } f(w, x) = (cos(w'x), sin(w'x)). \tag{4}$$

Thus, K may be approximated by averaging over d Fourier components $\{w_1, \ldots, w_d\}$ sampled from the distribution $\hat{k}$ to obtain an embedding of the point x into $\mathbb{R}^{2d}$:

$$\phi(x)' := \frac{1}{\sqrt{d}}(cos(w_1'x), \ldots, cos(w_d'x), sin(w_1'x), \ldots, sin(w_d'x)). \tag{5}$$

In particular, when K is the Gaussian kernel defined above, the Fourier transform $\hat{k}$ is also a Gaussian of variance $1/\sigma^2$. Using these results, the kernel matrix K can be approximated by

$$\tilde{K} = \Phi_X' \Phi_X, \quad \text{where} \quad \Phi_X = \{\phi(x^1), \ldots, \phi(x^T)\}. \tag{6}$$

The authors of [13] proved that $\forall \delta \in (0, 1)$, with probability $1 - \delta$,

$$\|K - \tilde{K}\|_F \leq \frac{2\ln(2/\delta)}{d} + \sqrt{\frac{2\ln(2/\delta)}{d}} = O(\frac{1}{\sqrt{d}}), \tag{7}$$

proving that the convergence is uniform and not data dependent.

2.2 Kernel NCMDS and Kernel K-means

When an orthogonality constraint $YY' = \mathbb{I}$ is added to Eq.2, the algorithm is equivalent to a kernel k-means clustering method. When this constraint is relaxed to non-negativity, we obtain the Symmetric Non-negative Matrix Factorization [14] (SNMF) performing sparse coding, which is a soft-clustering task. Such a model was developed in [6] and evaluated in [15] but was limited to the linear kernel clustering and sparse coding using $K = X'X$. This motivates the choice the kernel NCMDS as a viable nonlinear clustering and sparse coding method using nonlinear kernels.

3 Online kernel NCMDS Using Random Fourier Features

The method proposed here extends the applicability of a Hebbian/anti-Hebbian neural network proposed in [6] to nonlinear kernel methods, where data relationships are not described by distances in Euclidean spaces but by similarity values in an implicit high-dimensional space to which data is nonlinearly mapped.

We propose an algorithm and a neural architecture to perform online kernel NCMDS for any shift-invariant kernel. They bypass the difficulty of storing a similarity matrix by approximating the kernel with RFFs, and using a set of online learning rules that are Hebbian/anti-Hebbian in that they only depend on pre- and post- synaptic activations. The approximate kernel matrix $\hat{K}$ defined in Eq.6 replaces K in Eq.2 . Thus, the offline optimisation problem is defined as:

$$\min_{\tilde{Y}\geq\mathbb{R}_+^{m\times T}} \|\tilde{K}-\tilde{Y}'\tilde{Y}\|_F = \min_{\tilde{Y}\geq\mathbb{R}_+^{m\times T}} \|\Phi_X'\Phi_X-\tilde{Y}'\tilde{Y}\|_F. \tag{8}$$

If Y^* is an optimal solution of the exact kernel NCDMS such that ε is the distance from K to the subspace spanned by $Y'Y$s, and $\tilde{Y}^*$ an optimal solution of the approximated problem (8), such that $\tilde{\varepsilon}$ is the distance from $\tilde{K}$ to the subspace spanned by $\tilde{Y}'\tilde{Y}$s, then with probability $1-\delta$,

$$\begin{aligned}\|Y^{*'}Y^*-\tilde{Y}^{*'}\tilde{Y}^*\|_F^2 &\leq \|K-Y^{*'}Y^*\|_F^2+\|\hat{K}-\tilde{Y}^{*'}\tilde{Y}^*\|_F^2+\|K-\tilde{K}\|_F^2\\ &\leq \varepsilon+\tilde{\varepsilon}+\frac{2\ln(2/\delta)}{d}+\sqrt{\frac{2\ln(2/\delta)}{d}}.\end{aligned} \tag{9}$$

Thus, the solution of the original problem is approximated by minimising the approximated problem. The quality of the approximation depends on the number of RFFs. In order to ensure the convergence of the model, an improvement to this theoretical bound should find $\tilde{\varepsilon}$ as a function of (ε, d). However, in practice, the quality of the solution depends also largely on the implementation and in fact performs well in practice as will be discussed in Section 4.

3.1 Online Kernel NCMDS

Using the method in [6], Eq. 8 can be solved online as explained in the following. For every new input x^T presented, the model must find an optimal vector $(y^T)^*$ based only on information about K for the first T inputs, $K_T := K(x^i, x^j), i, j \leq T$ and on the previous determined vector $y^1, \ldots, y^{T-1}$. The problem can be formulated as follow:

$$(y^T)^* = \arg\min_{y^T\geq 0} \|K_T - Y'Y\|_F. \tag{10}$$

Note in particular that $y^1, \ldots, y^{T-1}$ are not updated at the T^{th} step and each y^T is based only on $x^1, \ldots, x^T$ and not on the full $\{x\}$, which is unbounded in the case of streamed data. A standard development of the Frobenius norm gives the following equation:

$$(y^T)^* = \arg\min_{y^T\geq 0} \sum_{t=1}^{T}\sum_{s=1}^{T}(K(x^s,x^t)-\langle y^s, y^t\rangle_{\mathbb{R}^m})^2. \tag{11}$$

Then $\forall s,t \in \{1,\dots,T\}\times\{1,\dots,T\}$ using the approximation Eq.7 we obtain

$$(K(x^s,x^t)-\langle y^s,y^t\rangle_{\mathbb{R}^{2d}})^2 \approx \left[\begin{array}{c} -2\phi(x^s)'\ \phi(x^t)y^{s'}\ y^t + \\ (\langle y^s,y^t\rangle_{\mathbb{R}^m})^2+(\langle\phi(x^s),\phi(x^t)\rangle_{\mathbb{R}^{2d}})^2 \end{array}\right] \quad . \tag{12}$$

After replacing the kernel by its approximation in the online kernel NCMDS (Eq.10), one can prove as in [6] that the components of the optimal vector can be found using coordinate descent:

$$(y_i^T)^* = \max\left(W_i^T\phi(x^T) - M_i^T y^T, 0\right), \tag{13}$$

$$\text{with} \quad W_{ij}^T = \frac{\sum_{t=1}^{T-1} y_i^t \phi(x^t)_j}{\sum_{t=1}^{T-1}(y_i^t)^2} \quad ; \quad M_{ij}^T = \frac{\sum_{t=1}^{T-1} y_i^t y_j^t}{\sum_{t=1}^{T-1}(y_i^t)^2}\mathbf{1}_{i\neq j} \quad \forall i \in \{1,\dots,m\}. \tag{14}$$

W^T and M^T can be found using recursive formulations:

$$W_{ij}^T = W_{ij}^{T-1} + \left(y_i^{T-1}\left(\phi(x^{T-1})_j - W_{ij}^{T-1}y_i^{T-1}\right)\Big/\hat{Y}_i^T\right) \tag{15}$$

$$M_{ij\neq i}^T = M_{ij}^{T-1} + \left(y_i^{T-1}\left(y_j^{T-1} - M_{ij}^{T-1}y_i^{T-1}\right)\Big/\hat{Y}_i^T\right) \tag{16}$$

$$\hat{Y}_i^T = \hat{Y}_i^{T-1} + (y_i^{T-1})^2 \quad . \tag{17}$$

The matrices W^T and M^T are sequentially updated using only the relationship between $\phi(x^{T-1})$ and y^{T-1}, which are analogous to pre- and post-synaptic activities, thus satisfying the Hebbian principle. The model presented here can be interpreted as a two-layer feed-forward neural network with lateral synaptic connections. Each new element, x^T, presented to the model is first transformed by passing through a neural network composed of d nodes, associated with the synaptic weights $\{w_1,\dots,w_d\}$ and with cosine and sine activation functions, generating $\phi(x^T)$. Secondly, an output y^T is generated after competition between neurons in a second layer according to Eq.13, where the weight matrices W^T and M^T can be interpreted respectively as feed-forward synaptic and lateral synaptic inhibitory connections (Fig.1). The schema in Fig.1 represents the two-layer neural network derived from the minimization of Eq.10.

One advantage of the approach presented here is that, although specifically relying on RFF, the results would hold for any inter-distance that may be approximated by inner-products [16]. Also, unlike spectral clustering methods that require eigenvalue decompositions to obtain hard clusters, this model finds cluster membership according to the output neuron with the largest activation for that input. In particular, the number of clusters is determined by the number of output neurons, m.

4 Results

In this section, the algorithm is tested on clustering tasks on synthetic and real data sets. The clustering accuracy of the models is evaluated in terms of the Normalized Mutual Information (NMI) between the estimated clusters and the true clusters.

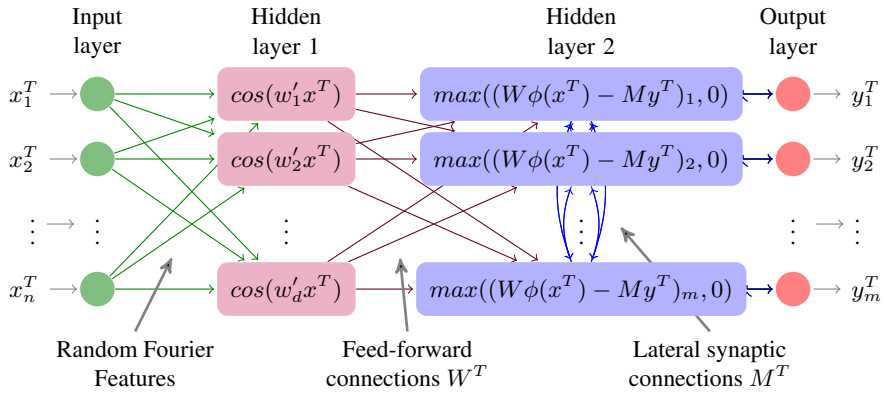
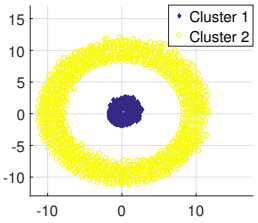


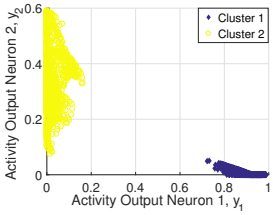
Fig. 1: Two-layer neural network performing online kernel clustering (Eq.10) for clusters defined by neurons in the output layer.

4.1 Artificial Data Set

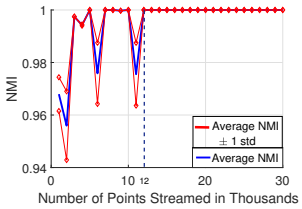
We evaluate the proposed model on the following toy example: let us consider $x^t \in \mathbb{R}^2$ such that with probability 0.5, $x^t \in C_1 = \{x \in \mathbb{R}^2, \|x\|^2 = R_1^2 + \varepsilon_1\}$, and probability 0.5, $x^t \in C_2 = \{x \in \mathbb{R}^2, \|x\|^2 = R_2^2 + \varepsilon_2\}$, where C_1 and C_2 represent two concentric circles, which are also the two clusters of interest.



(a) The two clusters in the input space colored according to the estimation of the model.



(b) Activity of the output neurons for each data points.



(c) NMI as a function of the number of points streamed (in thousands).

Fig. 2: Evaluation of the model on a toy example of two concentric circles composed of 500 points each. (a) The clusters estimated in the input space, (b) the activity of the output neurons, (c) the average NMI over 20 trials.

The simulation proves that the model learns to separate the two nonlinear 1-dimensional manifolds that are C_1 and C_2 (Fig2a and Fig2b), which is not possible if the linear kernel is used. The activities of the output neurons are recorded in Fig.2b, showing a clear separation between the two clusters in terms of neuronal activities. The

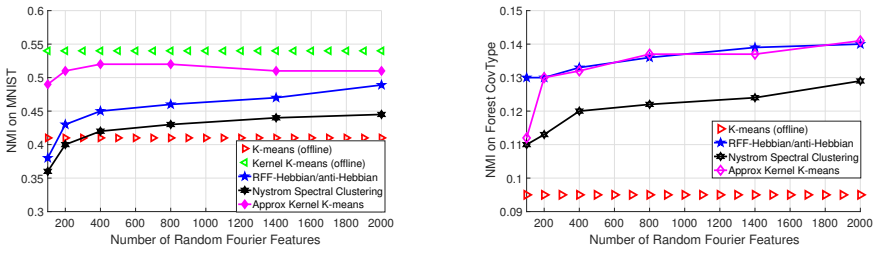
average NMI over 20 trials is recorded in Fig.2c showing that the model effectively clusters the data with high probability after 12,000 points are streamed.

4.2 Empirical Analysis: Large Scale Clustering

In this section, the model is evaluated against four models: the kernel k-means [17], which is an offline method requiring to compute and store the entire kernel matrix, the approximate kernel k-means algorithm [18, 13], the Nyström approximation-based spectral clustering algorithm [9], and the k-means algorithm in terms of their NMI.

Two public domain data sets are used: the MNIST [19] and the Forest Cover Type [20]. The *MNIST* [19] contains 60,000 training images and 10,000 test images from 10 classes. In this experiment, training and test images are combined into one data set. The *Forest Cover Type* [20] contains 581,012 data points from 7 classes. The number of output neurons m is fixed and equal to the true number of classes in the data set. The number of RFFs, d, is set to vary from 100 to 2000. In the results below we present the results obtained for the optimal parameters of the Gaussian kernel.

Fig.3a shows that the proposed model largely outperforms the standard k-means clustering technique, which is the only other model that also admits an online version, and the Nyström-based spectral clustering model [9]. The best performances on the MNIST are obtained by the kernel k-means algorithm and approximated kernel k-means [18, 13]. The latter can be efficiently implemented on large scale data sets or streams of data but relies on storing a sub-sample of the input data: in this case at least a 100 data points to perform batch training. Thus, the RFF Hebbian/anti-Hebbian method achieves the best fully online performance. As emphasized earlier, the kernel k-means could not be applied to the Forest Cover Type because of its large size, which would require computing and storing a kernel matrix of size $581,012 \times 581,012$ largely exceeding the memory of any standard computer. In this data set, the proposed model reached similar performances as the approximated kernel k-means, outperforming the Nyström-based method and the standard k-means.



(a) NMI for the MNIST data set (b) NMI for the Forest CoverType data set

Fig. 3: Evaluation of the neural network against other models on the MNIST (a), and on the Forest CoverType (b).

The advantage of the model proposed is that it only requires storing the weight matrices, $\{w_1, \ldots, w_d\}, W^T$, and M^T as it is designed to be used for online learning from streamed data. Although the model does not outperform the offline kernel k-means, it requires the least amount of memory and computation while showing better performance than offline linear clustering techniques such as k-means, to which it directly compares.

5 Conclusion

This study introduced an online learning algorithm for nonlinear clustering based on kernel Non-negative Classical Multidimensional Scaling and Hebbian learning rules. The proposed model is effective for clustering data sets from non-linearly clusterable data. This kernel version introduced here to extend the Hebbian/anti-Hebbian networks [6] is shown to perform well on real world data sets such as the MNIST and the Forest Cover Type. One possible limitation in the computation of the model, and topic of further investigations, is the convergence of the hidden layer with recurrent inhibitory connections when a large number of clusters is required. Preliminary results indicate that the model is also suitable for implementing sparse coding, an important property for unsupervised learning algorithms that is worth future investigations.

References

1. Oja, E.: Neural networks, principal components, and subspaces. International journal of neural systems 1(01), 61–68 (1989)
2. Barlow, H.B.: Unsupervised learning. Neural computation 1(3), 295–311 (1989)
3. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision research 37(23), 3311–3325 (1997)
4. Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural networks 2(6), 459–473 (1989)
5. Plumbley, M.D.: A Hebbian/anti-Hebbian network which optimizes information capacity by orthonormalizing the principal subspace. In: Artificial Neural Networks, 1993., Third International Conference on. pp. 86–90. IET (1993)
6. Pehlevan, C., Chklovskii, D.B.: A Hebbian/anti-Hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features. In: 2014 48th Asilomar Conference on Signals, Systems and Computers. pp. 769–775. IEEE (2014)
7. Cox, T.F., Cox, M.A.: Multidimensional scaling. CRC press (2000)
8. Williams, C.K.: On a connection between kernel PCA and metric multidimensional scaling. In: Advances in neural information processing systems. pp. 675–681 (2001)
9. Williams, C.K., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Proceedings of the 13th International Conference on Neural Information Processing Systems. pp. 661–667. MIT press (2000)
10. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Advances in neural information processing systems. pp. 1177–1184 (2008)
11. Aronszajn, N.: Theory of reproducing kernels. Transactions of the American mathematical society 68(3), 337–404 (1950)
12. Rudin, W.: Fourier analysis on groups. Courier Dover Publications (2017)

13. Chitta, R., Jin, R., Jain, A.K.: Efficient kernel clustering using random Fourier features. In: Data Mining (ICDM), IEEE 12th International Conference on. pp. 161–170. IEEE (2012)
14. Ding, C., He, X., Simon, H.D.: On the equivalence of nonnegative matrix factorization and spectral clustering. In: Proceedings of the 2005 SIAM International Conference on Data Mining. pp. 606–610. SIAM (2005)
15. Bahroun, Y., Soltoggio, A.: Online representation learning with single and multi-layer Hebbian networks for image classification tasks. In: Proceedings of the 26th International Conference on Artificial Neural Networks, ICANN 2017. p. to appear. Springer International Publishing (2017)
16. Pennington, J., Felix, X.Y., Kumar, S.: Spherical random features for polynomial kernels. In: Advances in Neural Information Processing Systems. pp. 1846–1854 (2015)
17. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural computation 10(5), 1299–1319 (1998)
18. Chitta, R., Jin, R., Havens, T.C., Jain, A.K.: Approximate kernel k-means: Solution to large scale kernel clustering. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 895–903. ACM (2011)
19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
20. Blackard, J.A., Dean, D.J.: Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Computers and electronics in agriculture 24(3), 131–151 (1999)