ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Fast Support Vector Clustering

Tung Pham[1], Trung Le[2], Thai Hoang Le[1], Dat Tran[3]
1-Faculty of Information Technology, VNUHCM - University of Science, Vietnam
2-Faculty of Information Technology, HCMc University of Pedagogy, Vietnam
3-Faculty of Education Science Technology and Maths, University of Canberra, Australia

Abstract. Support-based clustering has recently drawn plenty of attention because of its applications in solving the difficult and diverse clustering or outlier detection problem. Support-based clustering method undergoes two phases: finding the domain of novelty and doing clustering assignment. To find the domain of novelty, the training time given by the current solvers is typically quadratic in the training size. It precludes the usage of support-based clustering method for the large-scale datasets. In this paper, we propose applying Stochastic Gradient Descent framework to the first phase of support-based clustering for finding the domain of novelty in form of a half-space and a new strategy to do the clustering assignment. We validate our proposed method on the well-known datasets for clustering to show that the proposed method offers a comparable clustering quality to Support Vector Clustering while being faster than this method.

1 Introduction

Cluster analysis is a fundamental problem in pattern recognition where objects are categorized into groups or clusters based on pairwise similarity between those objects such that two criteria, homogeneity and separation, are gained [13]. Two challenges in the task of cluster analysis are 1) to deal with complicated data with nested or hierarchy structures inside; and 2) automatically detect the number of clusters. Recently, support-based clustering for example Support Vector Clustering (SVC) [1] has drawn a significant research concern because of its applications in solving the difficult and diverse clustering or outlier detection problem [1, 14, 10, 3, 6, 8, 7]. Support-based clustering methods have two main advantages comparing with other clustering methods: 1) ability to generate the clustering boundaries with arbitrary shapes and automatically discover the number of clusters; and 2) capability to handle well the outliers.

Support-based clustering methods always undergo two phases. In the first phase, the domain of novelty, e.g., optimal hypersphere [1] or hyperplane [11], is found in the feature space. The domain of novelty when mapped back to the input space will become a set of contours tightly enclosing data which can be interpreted as cluster boundaries. However, this set of contours does not specify how to assign a data sample to its cluster. In addition, the computational complexity of the current solvers [5, 4] to find out the domain of novelty is often quadratic. Such a computational complexity impedes the usage of support-based clustering methods for the real-world datasets. In the second phase, namely clustering assignment, based on the geometry information carried in the resultant set of contours harvested from the first phase, data samples are appointed to their clusters. Several works have been proposed for improving cluster assignment procedure [14, 10, 3, 8, 6].

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Recently, stochastic gradient descent (SGD) frameworks [12] have emerged as building blocks to develop the learning methods for efficiently handling the large-scale dataset. The main advantages of SGD-based algorithm consist of the following: 1) it is very fast; 2) it has ability to run in online mode; and 3) it does not require to load the entire dataset to the main memory in training. In this paper, we conjoin the advantages of SGD with support-based clustering. Concretely, we propose to use the optimal hyperplane as the domain of novelty. The margin, i.e., the distance from the origin to the optimal hyperplane, is maximized to make the contours enclosing the data as tightly as possible. We subsequently apply the SGD framework proposed in [12] to the first phase of support-based clustering for achieving the domain of novelty. Finally, we propose a new strategy for clustering assignment where each data sample in the extended decision boundary has its own trajectory to converge to an equilibrium point and clustering assignment is then reduced to the same task for those equilibrium points. Our clustering assignment strategy distinguishes from the existing works of [8, 9, 6] in the way to find the trajectory with a start and the initial set of data samples that need to do a trajectory for finding the corresponding equilibrium point. The experiments established on the real-world datasets show that our proposed method produces the comparable clustering quality with other support-based clustering methods while simultaneously achieving the computational speedup.

2 Stochastic Gradient Descent Large Margin One-class Support Vector Machine

2.1 Large Margin One-class Support Vector Machine

Given the dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, to define the domain of novelty, we construct an optimal hyperplane that can separate the data samples and the origin such that the margin, i.e., the distance from the origin to the hyperplane, is maximized. The optimization problem is formulated as

$$\max_{\mathbf{w},\rho} \left(|\rho| / \|\mathbf{w}\|^2 \right)$$
$$\text{s.t.:} y_i \left(\mathbf{w}^\mathsf{T} \phi(x_i) - \rho \right) \geq 0,\ i = 1, \ldots, N;\ \mathbf{w}^\mathsf{T}\mathbf{0} - \rho = -\rho < 0$$

where ϕ is a transformation from the input space to the feature space and $\mathbf{w}^\mathsf{T}\phi(x) - \rho = 0$ is equation of the hyperplane.

It occurs that the margin is invariant if we scale $(\mathbf{w}, \rho)$ by a factor k. Hence without loss of generality, we can assume that $\rho = 1$. Using the same derivation as in standard Support Vector Machine (SVM) [2], we yield the following optimization problem in primal.

$$\min_{\mathbf{w}} \left(J(\mathbf{w}) \triangleq \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{N} \sum_{i=1}^{N} \max \left\{ 0, 1 - \mathbf{w}^\mathsf{T} \phi(x_i) \right\} \right) \tag{1}$$

2.2 SGD-based Solution In Primal

To efficiently solve the optimization in Eq. (1), we use stochastic gradient descent method. At t^{th} round, we sample the data point x_{n_t} from the dataset $\mathcal{D}$.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Let us define $g_t(\mathbf{w}) \triangleq \frac{1}{2}\|\mathbf{w}\|^2 + C\max\left\{0, 1 - \mathbf{w}^\mathsf{T}\phi(x_{n_t})\right\}$. It is obvious that $g_t(\mathbf{w})$ is $1-$strongly convex w.r.t. $f(\mathbf{w}) \triangleq \frac{1}{2}\|\mathbf{w}\|^2$ over the feature space and $f(\mathbf{w})$ is 1-strongly convex w.r.t. the norm $\|.\|_2$ over the feature space as well.

The sub-gradient is $\lambda_t = \mathbf{w}_t - C1_{[\mathbf{w}_t^\mathsf{T}\phi(x_{n_t})<1]}\phi(x_{n_t}) \in \partial g_t(\mathbf{w}_t)$, where $1_A(.)$ is the indicator function. Therefore, the update rule is

$$\mathbf{w}_{t+1} = \nabla f^*(\nabla f(\mathbf{w}_t) - \eta_t\lambda_t) = \left(1 - \frac{1}{t}\right)\mathbf{w}_t + \frac{C}{t}1_{[\mathbf{w}_t^\mathsf{T}\phi(x_{n_t})<1]}\phi(x_{n_t}) \quad (2)$$

Algorithm 1 is proposed to find the optimal hyperplane which defines the domain of novelty. At each round, one data sample is uniformly sampled from the training set and the update rule in Eq. (2) is applied to determine the next hyperplane, i.e., $\mathbf{w}_{t+1}$. Finally, the last hyperplane, i.e., $\mathbf{w}_{T+1}$ is outputted as the optimal hyperplane. According to the theory displayed in the next section, we can randomly output any intermediate hyperplane and the approximately accurate solution is still warranted in a long-term training. Nonetheless, in Algorithm 1, we make use of the last hyperplane as output to exploit as much as possible the information accumulated through the iterations. It is worthwhile to note that in Algorithm 1, we store $\mathbf{w}_t$ as $\mathbf{w}_t = \sum_i \alpha_i\phi(x_i)$.

Algorithm 1 Algorithm for solving SGD-LMSVC in primal

$\mathbf{w}_1 = \mathbf{0}$
for $t = 1$ **to** T **do**
  Sampling n_t from $[N] = \{1, 2, ..., N\}$.
  $\mathbf{w}_{t+1} = \left(1 - \frac{1}{t}\right)\mathbf{w}_t + \frac{C}{t}1_{[\mathbf{w}_t^\mathsf{T}\phi(x_{n_t})<1]}\phi(x_{n_t})$.
endfor
Output: $\mathbf{w}_{T+1}$

2.3 Convergent Rate Guarantee

In this section, we show the convergent rate guarantee of Algorithm 1. We use the framework for regularized loss minimization proposed in [12]. Our main proof is based on Theorem 1 in [12], which is for completeness we restate here.

Theorem 1. *Let f be 1-strongly convex function w.r.t. $\|.\|$ over S. Assume that for all t, g_t is σ-*strongly *convex w.r.t. f. Additionally, let L be a scalar such that $\frac{1}{2}\|\lambda_t\|_*^2 \leq L$ for all t. Then, for all $u \in S$ the following bound holds for all $T \geq 1$, $\sum_{t=1}^T g_t(\mathbf{w}_t) - \sum_{t=1}^T g_t(u) \leq \frac{L}{\sigma}(1 + log(T))$.*

Before investigating the convergent rate of Algorithm 1, we assume that data are bounded in the feature space, i.e., $\|\phi(x)\| \leq R$ for all $x \in \mathcal{X}$. The convergent rate $\left(\mathrm{O}\left(\frac{\log(T)}{T}\right)\right)$ of Algorithm 1 is guaranteed through the following theorem.

Theorem 2. *Let $\mathbf{w}_t$ be defined as in Algorithm 1. For any $u \in S$, the following bounds hold for all $T \geq 1$, $\sum_{t=1}^T g_t(\mathbf{w}_t) - \sum_{t=1}^T g_t(u) \leq 2R^2C^2(1 + log(T))$.*

Theorem 3. *Assume that $n_1, n_2, ..., n_T$ are uniformly selected from $[N]$ and r is uniformly selected from $[T]$. Given $\delta \in (0; 1)$, with the probability greater than $1 - \delta$, the following holds:*

$$J(w_r) < J(w^*) + \frac{2R^2C^2(1 + \log(T))}{\delta T}$$

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

3 Clustering Assignment

After solving the optimization problem, we yield the decision function $f(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x) - 1$. To find the equilibrium points, we need to solve the equation $\nabla f(x) = 0$. To this end, we use the fixed point technique and assume that Gaussian kernel is used, i.e., $K(x, x') = e^{-\gamma \|x - x'\|^2}$. We then have

$$\frac{1}{2}\nabla f(x) = \sum_{i=1}^{N} \alpha_i (x_i - x) e^{-\gamma \|x - x_i\|^2} = 0 \rightarrow x = \frac{\sum_{i=1}^{N} \alpha_i e^{-\gamma \|x - x_i\|^2} x_i}{\sum_{i=1}^{N} \alpha_i e^{-\gamma \|x - x_i\|^2}} = P(x)$$

To find an equilibrium point, we start with the initial point $x^{(0)} \in \mathbb{R}^d$ and iterate $x^{(j+1)} = P\left(x^{(j)}\right)$. By fixed point theorem, the sequence $x^{(j)}$, which can be considered as a trajectory with start $x^{(0)}$, converges to the point $x_*^{(0)}$ satisfying $\mathrm{P}\left(x_*^{(0)}\right) = x_*^{(0)}$ or $\nabla f\left(x_*^{(0)}\right) = 0$, i.e., $x_*^{(0)}$ is an equilibrium point.

Let us denote $B_\epsilon = \{x_i : 1 \leq i \leq N \wedge |f(x_i)| \leq \epsilon\}$, namely the extended boundary for a tolerance $\epsilon > 0$. It follows that the set B_ϵ forms a strip enclosing the decision boundary $f(x) = 0$. Algorithm 2 is proposed to do clustering assignment. In Algorithm 2, the task of clustering assignment is reduced to itself for M equilibrium point. To fulfill cluster assignment for M equilibrium points, we run $m = 20$ sample point test as proposed in [1].

Algorithm 2 Clustering assignment procedure.

$E = \emptyset$.
foreach $x^{(0)}$ **in** B_ϵ **do**
  Find the equilibrium point $x_*^{(0)}$.
  if$\left(x_*^{(0)} \notin E\right)$ $E = E \cup \left\{x_*^{(0)}\right\}$
endfor
//Assume that $E = \{e_1, e_2, \ldots, e_M\}$
Do m sample point test with for E to find cluster indices for $e_1, e_2, \ldots, e_M$.
Each point $x^{(0)} \in B_\epsilon$ is assigned to the cluster of its corresponding equilibrium point $x_*^{(0)} \in E$.
Each point $x \in \mathcal{D} \backslash B_\epsilon$ is assigned to the cluster of its nearest neighbor in B_ϵ.

4 Experiments

To explicitly prove the performance of the proposed algorithm, we establish experiments on the real datasets. Clustering problem is basically unsupervised learning and there is not therefore a perfect measure to compare two given clustering algorithms. In this paper, we examine four typical clustering validity indices (CVI) including compactness, purity, rand index, and Davies-Bouldin index (DB index). The good clustering algorithm should produce the solution which has high purity, rand index, DB index and low compactness.

We perform experiment on 11 well-known datasets for clustering. These datasets are fully labeled and the CVIs like purity and rand index can be com-

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

pletely estimated. We compare the proposed method SGD-LMSVC with Support Vector Clustering (SVC) [1]. For finding the domain of novelty, SVC constructs an optimal hypersphere in the feature space which has a minimal volume and encloses all data. We use the well-known LIBSVM solver [4] to implement the first phase of SVC, i.e., finding the domain of novelty. The second phase of SGD-LMSVC and SVC, i.e., clustering assignment, is the same as described in Subsection 3 where ϵ was set to 0.01. All codes are implemented in $C\#$ and experimented on the computer with CPU 2.6GHz dual-core and 4GB RAM.

RBF kernel given by $K(x, x') = e^{-\gamma\left\|x-x'\right\|^2}$ is employed. The width of kernel γ is searched on the grid $\left\{2^{-5}, 2^{-3}, ..., 2^3, 2^5\right\}$. The trade-off parameter C is searched on the same grid. Actually, to make consistent, we set the trade-off parameter as $C \times N$ in the proposed method.

Datasets	Time		Purity		Rand Index		Compactness		DB Index	
	SVC	SGD	SVC	SGD	SVC	SGD	SVC	SGD	SVC	SGD
Aggregation	31.46	**8.98**	**1.00**	**1.00**	**1.00**	**1.00**	**0.29**	**0.29**	**0.68**	0.67
Breast Cancer	19.99	**2.42**	0.98	**0.99**	0.82	**0.85**	1.26	**0.68**	**1.58**	1.38
Compound	6.85	**0.07**	**0.66**	0.62	**0.92**	0.88	0.50	**0.21**	2.45	0.86
D31	1.83	**0.42**	0.94	**0.99**	**0.88**	0.81	1.41	**0.26**	**2.33**	1.35
Flame	2.33	**0.19**	0.86	**0.87**	0.75	**0.76**	0.58	**0.44**	**1.30**	0.65
Glass	1.05	**0.02**	0.50	**0.71**	0.77	**0.91**	0.72	**0.68**	0.53	**0.56**
Iris	5.82	**0.55**	**1.00**	**1.00**	**0.97**	0.96	0.98	**0.25**	**1.95**	1.17
Jain	**0.03**	0.53	0.37	**0.46**	0.70	**0.71**	0.96	**0.36**	**1.23**	1.08
Pathbased	4.16	**0.13**	**0.60**	0.50	0.81	**0.94**	**0.18**	0.30	0.36	**0.73**
R15	1.61	**0.23**	0.88	**0.90**	**0.74**	0.71	0.61	**0.13**	**2.96**	1.42
Spiral	467.89	**0.66**	0.09	**0.33**	0.15	**0.94**	2.00	**0.17**	**1.41**	0.98

Table 1: Total times (in second) on the experimental datasets of SGD-LMSVC (SGD) and SVC.

We report the total time (cf. Table 1) including the training time to find the domain of novelty and the clustering assignment time for each competitive algorithms. Determining the number of iterations in Algorithm 1 is really a challenge. To resolve it, we use the stopping criterion $\|\mathbf{w}_{t+1} - \mathbf{w}_t\| \leq \theta = 0.01$, i.e., the next hyperplane does only a slight change.

For each CVI, we boldface the method that yields a better outcome, i.e., higher value for purity, rand index, and DB index and lower value for compactness. As shown in Table 1, our proposed SGD-LMSVC is comparable with SVC on all considering CVIs. Especially, our proposed SGD-LMSVC surpasses SVC on the compactness, purity, and rand index. Regarding the amount of time taken for clustering, as our expectation, SGD-LMSVC is much faster than SVC. The computational speedup is even around 707 times for the Spiral dataset.

5 Conclusion

In this paper, we have proposed a fast support-based clustering method, which conjoins the advantages of SGD-based method and kernel-based method. Furthermore, we have also proposed a new strategy for clustering assignment. We validate our proposed method on 11 well-known datasets for clustering. The experiment has shown that our proposed method has achieved the comparable clustering quality comparing with the baseline while being much faster.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

References

[1] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[2] B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[3] F. Camastra and A. Verri. A novel kernel method for clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):801–804, 2005.

[4] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[5] T. Joachims. Advances in kernel methods. chapter Making Large-scale Support Vector Machine Learning Practical, pages 169–184. 1999.

[6] K.-H. Jung, D. Lee, and J. Lee. Fast support-based clustering method for large-scale problems. *Pattern Recognition*, 43(5):1975–1983, 2010.

[7] T. Le, D. Tran, P. Nguyen, W. Ma, and D. Sharma. Proximity multi-sphere support vector clustering. 22(7-8):1309–1319, 2013.

[8] J. Lee and D. Lee. An improved cluster labeling method for support vector clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):461–464, March 2005.

[9] J. Lee and D. Lee. Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1869–1874, November 2006.

[10] J. H. Park, X. Ji, H. Zha, and R. Kasturi. Support vector clustering combined with spectral graph partitioning. pages 581–584, 2004.

[11] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, July 2001.

[12] S. Shalev-Shwartz and Y. Singer. Logarithmic regret algorithms for strongly convex repeated games. In *The Hebrew University*, 2007.

[13] R. Shamir and R. Sharan. Algorithmic approaches to clustering gene expression data. In *Current Topics in Computational Biology*, pages 269–300. MIT Press, 2001.

[14] J. Yang, V. Estivill-Castro, and S. K. Chalup. Support vector clustering through proximity graph modelling. In *Neural Information Processing, 2002. ICONIP'02*, volume 2, pages 898–903, 2002.