Centering SVDD for Unsupervised Feature Representation in Object Classification

Dong Wang and Xiaoyang Tan [*]

Department of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics,
#29 Yudao Street, Nanjing 210016, P.R.China
x.tan@nuaa.edu.cn

Abstract. Learning good feature representation from unlabeled data has attracted researchers great attention recently. Among others, K-means clustering algorithm is popularly used to map the input data into a feature representation, by finding the nearest centroid for each input point. However, this ignores the density information of each cluster completely and the resulting representation may be too terse. In this paper, we proposed a SVDD (Support Vector Data Description) based method to address these issues. The key idea of our method is to use SVDD to measure the density of each cluster resulted from K-Means clustering, based on which a robust feature representation can be derived. For this purpose, we add a new constraint to the original SVDD objective function to make the model align better with the data. In addition, we show that our modified SVDD can be solved very efficiently as a linear programming problem, instead of as a quadratic one. The effectiveness and feasibility of the proposed method is verified on two object classification databases with promising results.

Keywords: Feature learning, K-means, Support Vector Data Description(SVDD), C-SVDD, object classification

1 Introduction

Learning good feature representation from unlabeled data is the key to make progress in recognition and classification tasks, and has attracted great attention and interest from both academia and industry recently [1]. Deep learning method which aims to learn multiple layers of abstract representations from data has gained much success and has become a popular way for representation learning. In this method layers of representation is usually obtained by greedily training one layer at a time on the lower level [2], [3], [4], using an unsupervised learning algorithm. In this sense, the performance of single-layer learning has an big effect on the final representation. Neural network based single-layer methods, such as

[*] This work was supported by the National Science Foundation of China (61073112, 61035003,61373060), Jiangsu Science Foundation (BK2012793), Qing Lan Project, and Research Fund for the Doctoral Program (RFDP) (20123218110033).

autoencoder [5] and RBM (Restricted Boltzmann Manchine,[6]), are widely used for this but they have the disadvantages that the models are usually very complex and have many parameters to adjust. In addition, many parameters involved are need to be set through cross-validation, which is very time-consuming.

That is why a simple and fast method is preferred for unsupervised feature learning. Among others K-means clustering algorithm is commonly used to map the input data into a feature representation. The simplest way for this is to map each data point to its nearest cluster center and use it as the feature to describe the data. There is only one parameter involved in the K-means based method, i.e., the number of clusters, hence the model is very simple and fast. Coates et al. [7] shows that the K-means based encoder achieves the best performance compared with sparse autoencoder, sparse RBM and GMM (Guassian Mixture Model) under some circumstances. Despite of the success, the above K-means based feature representation scheme is not perfect from the aspect of the richness of information it conveys. Actually, such a representation is too terse, and does not take the non-uniform distribution of cluster size into account. Intuitively, those clusters containing more data are likely to be part of the features with higher influential power, compared to the smaller ones.

In this paper, we proposed a SVDD (Support Vector Data Description, [8], [9]) based method to address these issues. The key idea of our method is to use SVDD to measure the density of each cluster resulted from K-means clustering, based on which more robust feature representation could be built. Actually the K-means algorithm lacks a robust definition of the size of its clusters, since the nearest center principle is not robust against the noise or outliers common in real world applications. We advocate that the SVDD could be a good way to address this issue. Actually SVDD is a widely used tool to find a minimal a closed spherical boundary to include all the data belong to target class and therefore, given a cluster of data, we are expecting SVDD to generate a ball containing the all normal data excepting outliers. Performing this procedure on all the clusters of K-means, we will finally get K SVDD balls on which our representation can be built. In addition, considering that a bigger ball is more influential than smaller ones, we use the distance from the data to each ball's surface instead of the center as the feature.

One problem of our model comes from the instability of SVDD's center, due to the fact that its position is mainly determined by the support vectors on the boundary and the noise in the data may deviate the center far from the mode (c.f., Fig. 3(left)). This makes the SVDD ball not be consistent with the data's distribution when used for feature representation. To address this, we add a new constraint to the original SVDD objective function to make the model align better with the data. In addition, we show that our modified SVDD can be solved very efficiently as a linear programming problem, instead of as a quadratic one. Experiments on the AR face dataset and CIFAR-10 object database show that it is robust, efficient, and when combined with K-means, it provides a much richer representation for the input data and thus improves the performance of object classification.

2 Preliminaries

2.1 Unsupervised Feature Learning

The overall pipeline of the feature representation is as follows. For a given image, a set of patches are first sampled at the positions of a regular grid [7]. By mapping those patches to their nearest cluster centers, a set of feature maps could be obtained. Then one can pooling on these and reshape them into a vector, which yields the final feature representation for the input image. It is worthy mentioning that there is a small difference between the above method and others such as the CNN network [10], [11], i.e., instead of using a learnt filtering bank for convolution, the K-means centers are used as references for feature mapping. In other words, the cluster centers play the same role as the filtering bank in CNN network but its way for feature mapping is different from the latter.

2.2 K-means for Feature Learning

K-means is a data clustering algorithm to divide data into a set of K clusters, with Euclidean distance as similarity measure. It aims to minimize the sum of distance between all data to their corresponding centers. Let X=$\{x_i\}$,i=1,...,n be the set of n d-dimensional points, $C = \{c_k\}$, k=1,...,K be the K clusters. Let μ_k be the mean of the cluster c_k. The objective function is defined as: $J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2$.

As mentioned in the previous section, each cluster would be used to produce a feature mapping. So if we have K clusters, the dimension of the resulting feature representation will be K as well. The simplest way for feature mapping is the so-called "hard coding" method, i.e., simply setting the winner cluster center on while all the others off, as follows,

$$f_k(x) = \begin{cases} 1 & \text{if } k = argmin_j \|c_j - x\|_2^2 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The resulting K-dimensional vector f can be thought as the MAP estimate of the input point x given the K-means model. However it is too sparse and is often not representative of the full posterior mass. A better summary is the following "soft coding":

$$f_k(x) = max\{0, \mu(z) - z_k\} \tag{2}$$

where $z_k = \|x - c_k\|_2$, and $\mu(z)$ is the mean of the elements of z. This activation function outputs 0 for the feature f_k that have an above average distance to the centroid c_k. This model leads to a less sparse representation (roughly half of the features are found to be 0 in our experiments), but as shown in the experimental section, it significantly improves the classification performance.

However, this method does not take the characteristics of each cluster into consideration. Actually, the number of data point in each cluster is usually different, so is the distribution of data points in each cluster. We believe that these

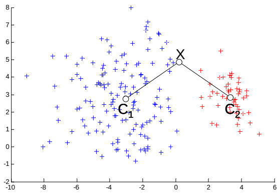


Fig. 1. Illustration of the unequal cluster effect.

differences would make a difference in feature representation as well. However, the aforementioned K-means feature mapping scheme completely ignores these and only use the position of center for coding. As shown in Fig. 1, although the data point x has the same distance to the centers $C1$ and $C2$ of two clusters, it should be assigned a higher score on $C1$ than on $C2$ since the former cluster $C1$ is much bigger than the latter. In practice such unequal clusters are not uncommon and the K-means method by itself can not reliably grasp the size of its clusters due to the existence of outliers. To this end, we propose an SVDD based method to describe the density and distribution of each cluster and use this for more robust feature representation.

3 THE PROPOSED METHOD

3.1 Using SVDD Ball to Cover Unequal Clusters

Assume that a data set contains N data objects, $\{x_i\}$, $i = 1, ..., n$ and a ball is described by its center a and the radius R. The goal of SVDD (Support Vector Data Description, [8]) is to find a closed spherical boundary around the given data points. In order to avoid the influence of outliers, SVDD actually faces the tradeoff between two conflicting goals, i.e., minimizing the radius while covering as many data points as possible.

The SVDD method can be understood as a type of one-class SVM and its boundary is solely determined by support vectors points. SVDD allows us to summarize a group of data points in a nice and robust way. Hence it is natural to use SVDD ball to model each cluster from K-means, thereby combining the strength of both models. In particular, for a given data point we first compute its distance h_k to the surface of each SVDD ball C_k, and then use the following soft coding method for feature representation similar to E.q.(2): $f_k(x) = max\{0, g(z) - h_k\}$, where $g(z) = \mu(z) - \mu(R)$ and $\mu(R)$ is the mean of radius R of balls, while $h_k = |z_k - R_k|$ is the distance from the point to the surface of the SVDD ball.

Shown in Fig. 2 for a data point x, C_i, i=1,2 respectively are the centroids of two SVDD balls with R_i, i=1,2 being their the radius respectively, and $h_i = |z_i - R_i|$ is the distance from x to the surface of $i - th$ ball. Since the distances from x to $C1$ and C_2 are equal, x will get the same scores on the two ball with the K-means scheme (c.f., E.q.(2)). However, if we take the density and size

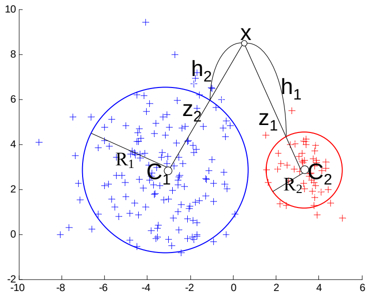


Fig. 2. Using the SVDD ball to cover the clusters of K-means.

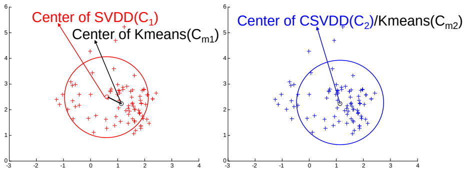


Fig. 3. Illustration of the difference between SVDD and C-SVDD, where the left ball ($C1$) is from SVDD while the right (C2) is from C-SVDD. Note that the center of C-SVDD ball aligns better with the high density region of the data points. The C_m marks the center of K-means.

of the clusters into accounts, the score from $C2$ should be higher and that is exactly our method does.

3.2 The C-SVDD Model

Although SVDD ball provides a robust way to describe the cluster of data, one unwelcome property of the ball is that it may not align well with the distribution of data points in that cluster. As illustrated in Fig. 3 (left), although the SVDD ball covers the cluster $C1$ well, its center is biased to the region with low density. This should be avoided since it actually gives suboptimal estimates on the distribution of the cluster of data.

To address this issue, inspired by the observation that the centers of K-means are always located at the corresponding mode of their local density, we propose to shift the SVDD ball to the centroid of the data such that it may fit better with the distribution of the data in a cluster. Our new objective function is then formulated as follows,

$$\begin{aligned} & min_{R,\xi_i}\ R^2 + C\sum_{i=1}^{N}\xi_i \\ & s.t.\ \|x_i - a\|^2 \leq R^2 + \xi_i \\ & \quad a = \frac{1}{N}\sum_{i=1}^{N} x_i \\ & \quad \xi_i \geq 0 \end{aligned} \tag{3}$$

where $\|.\|$ is the L_2-norm and ξ_i is the slack variable to the ith sample x_i. With Lagrange multipliers $\alpha_i \geq 0$ and $\alpha_j \geq 0$ according to KKT Conditions, one has the following dual function:

$$\begin{aligned} &max \sum_i \alpha_i \langle x_i, x_i \rangle - \frac{2}{N} \sum_i \sum_j \alpha_i \langle x_i, x_j \rangle \\ &s.t. \sum_i \alpha_i = 1 \ , \ \alpha_i \in [0, C] \ , \ i = 1, ..., N \end{aligned} \tag{4}$$

Eq.(4) can be rewritten as:

$$\begin{aligned} &min \ \frac{2}{N} \alpha^T He - \alpha^T F \\ &s.t. \ \alpha^T e = 1 \ , \ \alpha_i \in [0, C] \ , \ i = 1, ..., N \end{aligned} \tag{5}$$

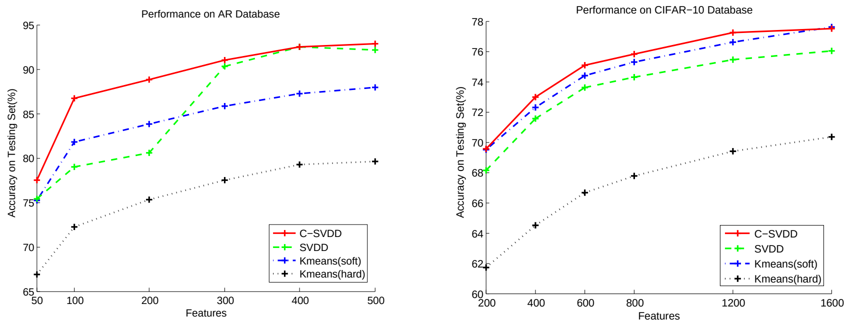
where $H = (\langle x_i, x_j \rangle))_{N \times N}$, $F = (\langle x_i, x_i \rangle)_{N \times 1}$, $e = (1, 1, ..., 1)^T$. It is worthy mentioning that this objective function is linear to α, and thus can be solve efficiently with a linear programming algorithm.

Since the model is centered towards the mode of the distribution of the data points in a cluster, we named our method as C-SVDD (centered-SVDD). Figure.3 shows the difference between SVDD and C-SVDD, where the left result is from SVDD and the right from C-SVDD. We can see that our new model aligns better with the density of the data points, as expected.

4 EXPERIMENTS AND ANALYSIS

To investigate whether the proposed method can produce good feature representation. We conducted a series of experiments on the AR face database [12] and the CIFAR-10 object dataset [13], on each of which, we compared our method (C-SVDD with K-means) with other three types of feature mapping strategies, i.e., K-means(hard), K-means(soft) and SVDD (combined with K-means). All the images in use undergone whitening preprocessing before being sampled for feature mapping [7].

The AR face database [12] contains over 4,000 color images corresponding to 126 people's faces. Every person has 2 sessions images with 13 for each. Images are all frontal view faces with different facial expressions, illumination conditions, and occlusions. Here we use all the images from the first session for training while those in the second session for testing. All images are resized to 64×64. For training we sample 40000 patches with size 6×6 from training set, and cluster them using K-means by varying the number of clusters K. Then we do the feature mapping as described in the previous section. Note that for the normalization parameter C in SVDD and C-SVDD, If $C = 0$, the representation result of $C - SVDD$ is equal to K-means, while a larger C value means more noise is allowed to enter the ball. We use 5-cross validation to set its value from a range of {0.005, 0.01, 0.1, 1}.



(a) (b)

Fig. 4. Comparative performance of the proposed method with various K-means based encoding strategies on (a) the AR dataset and (b) the CITAR-10 dataset.

The CIFAR-10 [13] dataset is a more complicated database which consists of 60000 32x32 color images in 10 classes with 6000 images per class. There are 50000 training images and 10000 test images, and the training set is divided into five batches. We also use 5-cross validation to set the best C value for C-SVDD and SVDD with a range of $\{0.004, 0.005, 0.006, 0.008, 0.01\}$. The receptive field is 6 by 6, and 400000 patches are sampled for training.

Figure 4 gives the results. It can be seen that our C-SVDD-based representation method is the best performer on both datasets. The K-means (hard) method is the worst one as expected due to its extremely sparse representation, while replacing the hard coding with a soft one (K-means (soft)) significantly improves the performance. The figure also reveals that the scheme of simply adding SVDD ball onto the top of soft K-means does not necessarily work and may actually hurt the performance due to the bias it introduced (as explained in the previous section). However, once this problem solved, the performance is improved a lot. Another point needing to be pointed out is that when the number of features (i.e., the cluster number K in K-means) increases, the performance of all the four methods improves consistently. This indicates the importance of encoding richer information in the feature representation.

Table 1 gives the comparative performance (%) of our method with other state-of-the-art single-layer network results on the CIFAR-10 dataset. For a fair comparison, we adopted the same evaluation protocol as that in [7], and all the results except the last row are directly cited from it. It is clear that our C-SVDD method performs the best among the compared methods.

5 Conclusion

In this paper, we proposed a SVDD based feature learning algorithm that enhances the K-means ”soft” feature representation. The key idea of our method is to describe the density and distribution of each cluster from K-means with a SVDD ball for more robust feature representation. For this purpose, we presented

Table 1. Comparative performance (%) with other state-of-the-art single-layer network methods on the CIFAR-10 dataset.

Algorithm	Accuracy
Sparse auto-encoder [7]	73.4
Sparse RBM [7]	72.4
K-means (Hard) [7]	68.6
K-means (Triangle, 4000 features) [7]	79.6
C-SVDD (4000 features) (ours)	**79.8**

a new SVDD algorithm called C-SVDD that centers the SVDD ball towards the mode of local density of each cluster. Furthermore we show that the objective of C-SVDD can be solved very efficiently as a linear programming problem. Experiments on the AR and the CIFAR-10 database show that our C-SVDD based feature representation method outperforms the original K-means based scheme.

References

1. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. arXiv preprint arXiv:12065538. 2012;.
2. Le QV, Ranzato M, Monga R, Devin M, Chen K, Corrado GS, et al. Building high-level features using large scale unsupervised learning. arXiv preprint arXiv:11126209. 2011;.
3. Agarwal A, Triggs B. Hyperfeatures–multilevel local coding for visual recognition. In: Computer Vision–ECCV 2006. Springer; 2006. p. 30–43.
4. Jarrett K, Kavukcuoglu K, Ranzato M, LeCun Y. What is the best multi-stage architecture for object recognition? In: Computer Vision, 2009 IEEE 12th International Conference on. IEEE; 2009. p. 2146–2153.
5. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504–507.
6. Cueto MA, Morton J, Sturmfels B. Geometry of the restricted Boltzmann machine. Algebraic Methods in Statistics and Probability,(eds M Viana and H Wynn), AMS, Contemporary Mathematics. 2010;516:135–153.
7. Coates A, Lee H, Ng AY. An analysis of single-layer networks in unsupervised feature learning. Ann Arbor. 2010;1001:48109.
8. Tax DM, Duin RP. Support vector data description. Machine learning. 2004;54(1):45–66.
9. Xu J, Yao J, Ni L. Fault detection based on SVDD and cluster algorithm. In: Electronics, Communications and Control (ICECC), 2011 International Conference on. IEEE; 2011. p. 2050–2052.
10. Niu XX, Suen CY. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. Pattern Recognition. 2012;45(4):1318–1325.
11. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11):2278–2324.
12. Martinez AM. The AR face database. CVC Technical Report. 1998;24.
13. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto. 2009;.