Image Classification with Semi-supervised Cluster Kernels

Devi G
CS14M011

Ditty Mathew
CS13D018

Abstract

The problem of image classification is suited for cluster kernel assumption because similar images have similar objects appearing in them at different places. In this project, the idea of cluster kernel combined with a semi-supervised approach of classifier training is used for image classification and the performance of the classifier on two different datasets is analysed. The performance is also compared with a non kernel classifier both in a supervised and semi-supervised setting.

1 Introduction

The term *Image classification* refers to the labeling of images into one of a number of predefined categories. Although it is not a very difficult task for humans, it has proved to be a difficult problem for machines. The problem is complicated more by the high cost involved in obtaining class labels in the case of supervised training. Semi-supervised learning(SSL) is a machine learning paradigm that lies between Supervised and Unsupervised techniques. Due to the relatively high cost involved in obtaining labels through manual annotation, the semi-supervised approach of combining a small set of labeled data with a larger unlabeled set at the time of training has gained significant attention. This has been shown to increase the accuracy and robustness of class predictions.

Most of the semi-supervised learning algorithms implicitly make use of the assumption that two objects in the same class are likely to be closer to each other. A cluster kernel is one that makes this assumption hold good in a high dimensional space, i.e., the induced distance is small for points in the same cluster and high for points in different clusters. In this project our objective is to classify images using semi-supervised cluster kernels.

There are different variants of training in semi-supervised learning. In the self training based semi-supervised approach, the more promising predictions of unlabeled data is added to the labeled set and process is repeated until all the unlabeled data labels gets predicted. In the generative model based SSL, the class conditional density is estimated using labeled and unlabeled data information. Graph based methods propagates the label information to its neighbors until a global steady state is achieved.

In this work we use a semi-supervised SVM based on cluster kernels. The key of cluster kernels is to model the marginal data structure leaving the base algorithm unchanged. Here the SVM is trained with the linear combination of two kernels - base kernel which is the standard SVM kernel gram matrix using only the labeled data and likelihood kernel which encodes similarities between labeled and unlabeled examples.

The performance of the semi-supervised approach with cluster kernel is compared with that of semi-supervised and supervised GMM. This report is organized as follows. Section 2 defines the problem. Section 3 presents the cluster kernel SVM approach and Section 4 explains semi-supervised GMM method used. Section 5 illustrates the experimental setup and the datasets considered and in Section 6 experimental results are discussed.

2 Problem Definition

The task is to classify the images into predefined categories by using semi-supervised approach. Semi-supervised learning is a partially supervised learning. We have a small set of labeled examples $D^L = \{\vec{x}_n, t_n\}_{n=1}^{N_L}$ and large unlabeled examples $D^U = \{\vec{x}_n\}_{n=N_L}^{N_L+N_U=N}$ where $N_U >> N_L$. The classification model has to be build by using information from labeled and unlabeled examples.

Main challenges in this task are

1. low number of labeled examples
2. noise

Here we are comparing the performance of non-kernel method such as semi-supervised GMM with semi-supervised cluster kernel method.

3 Approach

Kernel Methods Kernel methods embed the data observed in the input space X into a higher dimensional space, the feature space H, where the data is more likely to be linearly separable. Therefore, it is more possible to build an efficient linear classifier in the higher dimensional space H which will be nonlinearly related to the input space. $\mathbf{\Phi}$ is the mapping function from input space to kernel feature space, i.e $\mathbf{\Phi} : X \rightarrow H$. Since the computation of mapping function $\mathbf{\Phi}$ is computationally expensive for high dimensional space, kernel methods compute the similarity between training samples using pairwise inner products between mapped samples. This similarity matrix is called kernel gram matrix, i.e $\mathbf{K}(\mathbf{x_i}, \mathbf{x_j}) =< \mathbf{\Phi}(x_i), \mathbf{\Phi}(x_j) >$

Linear combination of two Kernels Let $\mathbf{K}_1$ and $\mathbf{K}_2$ be two Mercer's kernel on SxS. Then the linear combination $\mathbf{K}_1$ and $\mathbf{K}_2$, i.e $\mathbf{K}(x, y) = \mathbf{K}_1(x, y) + \mathbf{K}_2(x, y)$ is also a valid Mercer's kernel on SxS. We use this property in the semi-supervised model.

SVM kernel deformed with Cluster Kernel The underlying idea of deforming cluster kernel is to modify the SVM kernel function $\mathbf{K}$ to find the distribution of labeled and unlabeled data. The modified kernel function is the linear combination of a base kernel and a likelihood kernel. We used the standard SVM RBF kernel which uses only labeled data information as the base kernel. Then deform the base kernel with a likelihood kernel which encodes the relationship between labeled and unlabeled examples. Such kernel can be obtained by running several instances of a clustering algorithm over the unlabeled samples and it is therefore named as cluster kernel. We used k-means as the clustering algorithm and considered N bag of k-means to run it several times with same value of K. The cluster kernel defined by bag of clusters is called bagged cluster kernel.

Bagged cluster kernel is defined by counting the occurrences of two samples in the same cluster over several runs of an unsupervised algorithm. Two samples that fall several times into the same cluster denote a strong similarity. The semi-supervised bagged kernel SVM(Bag-SVM) algorithm is as follows,

1. Compute the base SVM kernel $\mathbf{K}_{SVM}$, eg: using RBF kernel
2. Run N times preferred clustering algorithm with different initialization and same number of clusters
3. Build a bagged kernel $\mathbf{K}_{bag}$ based on the fraction of times two points assigned to the same cluster.

$$\mathbf{K}_{bag}(x_i, x_j) = \frac{1}{N} \sum_{p=1}^{N} [c_p(x_i) = c_p(x_j)] \tag{1}$$

where $c_p(x_i) = c_p(x_j)$ returns '1' if samples x_i and x_j belong to the same cluster in the p^{th} step

4. Define the final cluster kernel $\mathbf{K}_C$ as the linear combination of standard SVM kernel and bagged kernel

$$\mathbf{K}_C(x_i, x_j) = \alpha \mathbf{K}_{SVM}(x_i, x_j) + (1 - \alpha)\mathbf{K}_{bag}(x_i, x_j) \quad (2)$$

5. Train the SVM using $\mathbf{K}_C$

Since k-means results vary on each run, Step results will give different results. Step 3 is a valid kernel because it is the inner product in kN-dimensional space $\mathbf{\Phi} =< [c_p(x_i) = v] : p = 1 \dots N; v = 1 \dots k >$ and the linear combination of these two valid kernels is again a valid kernel.

4 Non-Kernel Method

We used semi-supervised GMM as the non-kernel method. The self training approach has followed which is a wrapper method for semi-supervised learning. The self training based approach is briefly explained below.

1. Train a classifier using D^L where $D^L = \{\vec{x}_n, t_n\}_{n=1}^{N_L}$ and N_L corresponds to number of labeled examples
2. Assign a label for each of the unlabeled examples using training labels(labeled examples) by picking up the closest ones to the labels.
3. Find subset of unlabeled examples with most confident scores. Let N_{ui} be the subset of unlabeled examples that are confident to assign to class *i*
4. Now total number of labeled examples for class *i*= $N_i + n_{ui}$, where N_i is the number of labeled examples for class *i*
5. Repeat step 2 to 4 until all the unlabeled examples get labels

5 Experiments

5.1 Datasets

The data sets used for our experiments are Scene-15 and Caltech-101.

Scene-15 consists of fifteen natural scene categories and *Caltech-101* contains pictures of objects belonging to 101 categories with about 40 to 800 images per category.Images are represented with three common image features GIST, PHOG, and LBP[1].

The GIST feature is a well-developed feature used in scene recognition. It uses Gabor filter to extract a holistic description of the image which are highly related to the underlying scene. It will help determine what type of object the image is showing.

PHOG is the spatial Pyramid representation of HOG descriptor and has been recognised to give good performance. The essential idea behind using histogram of Oriented Gradient information is that objects in an image can be better characterized by the histogram of edge directions of local regions within an image.

Local Binary Patterns (LBP) is a powerful feature for texture classification and along with HOG, it has been shown to improve the classifier performance on some data sets.

The classes used for classification in this project are 2,3,5,6 in Scene-15 dataset and classes 1,2,4,6 in Caltech -101 data set. The features are L1-normalized and then multiplied by their dimensions (e.g. for GIST, the L1-normalized feature values are then multiplied by 20.). This is to map all feature values to comparable levels.

5.2 Kernel Methods

The kernel method that we experimented with is Bagged Cluster Kernel and a linear combination of SVM RBF Kernel and Bagged Cluster Kernel. Experiments were conducted in both transductive and inductive settings.

[1]Data obtained from http://people.ee.ethz.ch/ daid/EnPro/

We used a fixed number of labeled examples per class $N_L = \{10, 20, 50, 70, 100\}$. 200 unlabeled examples from each class in both the datasets were used for transductive learning. 100 unlabeled examples and 100 test examples were used for inductive learning.

The standard SVM is trained using RBF kernel for the labeled examples and obtained $\mathbf{K}_{SVM}$. The k-means clustering is used for constructing the bagged cluster kernel. Both labeled and unlabeled examples are used to find the cluster centers. Then labeled examples are assigned to closest cluster center and the same assignment is used for predicting class membership of unlabeled examples. Then using the class membership information $\mathbf{K}_{bag}$ is constructed. In order to analyze the statistical significance of the results the k-means is executed 50 times for the same value of k and the $\mathbf{K}_{bag}$ is updated on each run. We experimented with different values of k, i.e k= $\{5, 10, 50, 70, 100\}$. We experimented with $\mathbf{K}_{SVM}$, $\mathbf{K}_{bag}$ and linear combination of these two as kernel gram matrix for SVM separately and analyzed the results.

5.2.1 RBF Kernel

In case of Scene-15 dataset, RBF kernel is trained with gamma =0.009 which has chosen by cross-validation. For Caltech-101 dataset 4 is obtained as gamma value by cross-validation. The accuracy obtained for unlabeled data by using only RBF kernel for different labeled data size (per class) across both the dataset is shown in Figure 1.

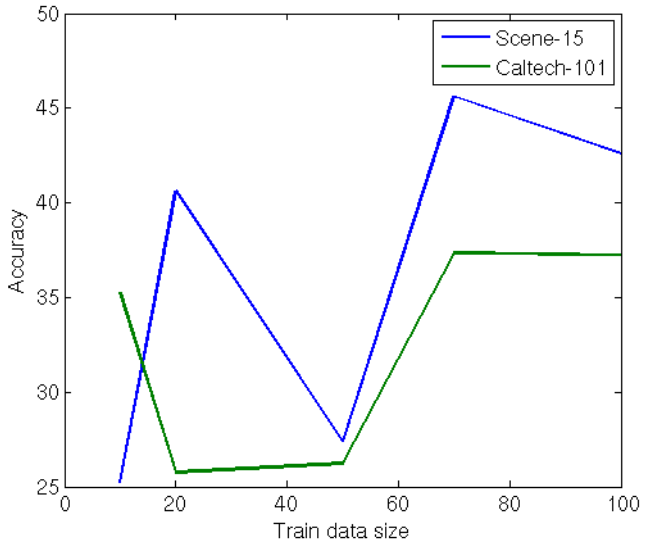


Figure 1: Plot for labeled data size vs Accuracy(unlabeled data) for RBF kernel

The confusion matrix obtained for best accuracy for both the dataset are shown in Table 2a and 2b.

class	1	2	3	4
1	52	48	46	7
2	2	0	0	0
3	7	66	38	1
4	139	86	116	192

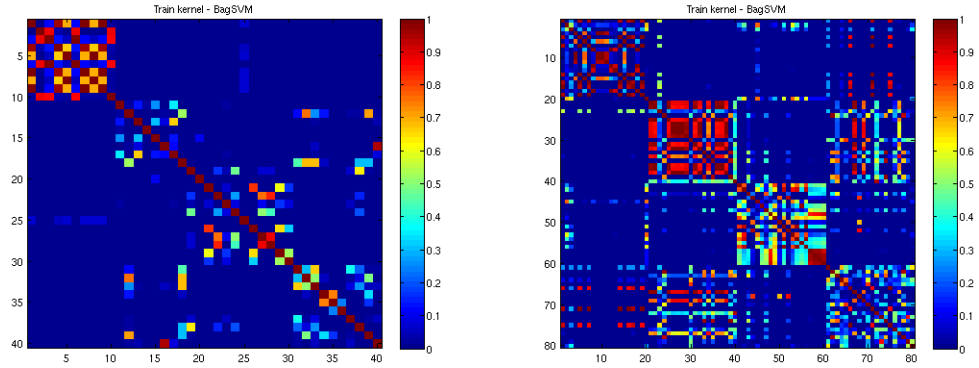
(a) Confusion matrix for the best Accuracy(train size=70) for Caltech-101

class	1	2	3	4
1	163	0	63	12
2	6	43	39	122
3	23	20	93	40
4	8	137	5	26

(b) Confusion matrix for the best Accuracy(train size=70) for Scene-15

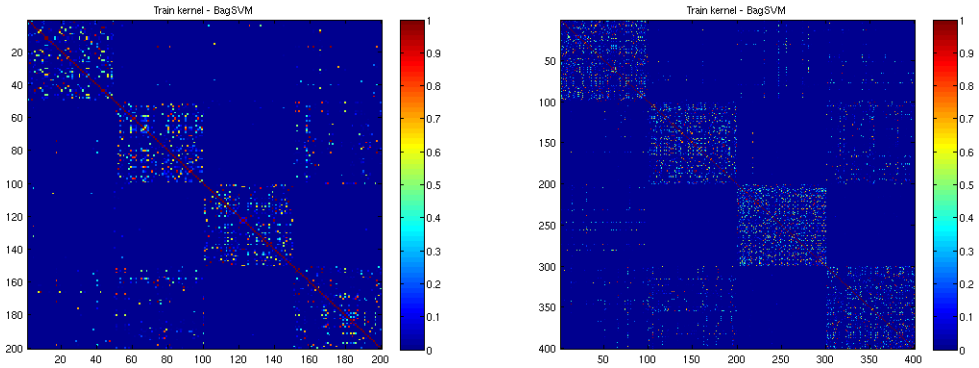
5.2.2 Bagged Cluster Kernel

50 bags of k-means has used to construct the bagged kernel. The kernel gram matrices obtained for labeled data for different size for Scene-15 dataset are shown in Figure 3.



(a) 10 labeled samples per class

(b) 20 labeled samples per class



(c) 50 labeled samples per class

(d) 100 labeled samples per class

Figure 3: Kernel gram matrix for labeled data using bagged kernel for Scene-15 dataset

The transductive accuracy obtained for unlabeled data across various labeled data size and k parameter of k-means are shown in Table 1 and the inductive accuracy obtained for test data of different labeled data size are sown in Table 2.

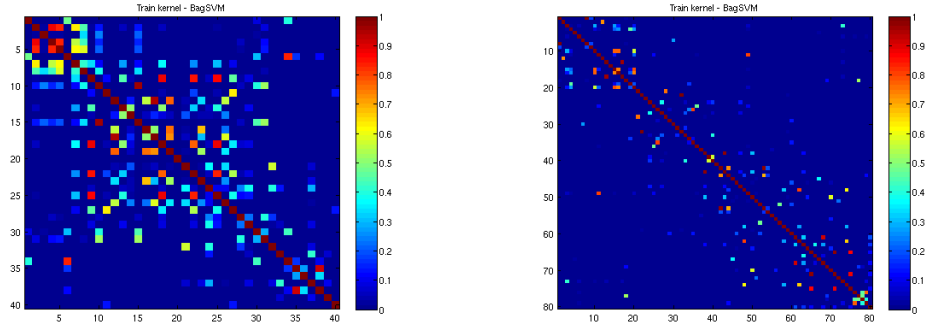
Labeled data size	K=5	K=10	K=50	K=70	K=100
10	75.6250	**76.2500**	**76.2500**	66.0000	52.5000
20	78.1250	**82.6250**	79.7500	78.2500	71.3750
50	77.0000	80.5000	83.0000	**84.5000**	81.2500
70	79.3750	80.2500	82.6250	**84.6250**	83.8750
100	78.0000	82.6250	83.0000	**85.1250**	84.6250

Table 1: Transductive accuracy for different K across different labeled data size for Scene-15 dataset using only Bagged Kernel

Labeled data size	K=5	K=10	K=50	K=70	K=100
10	**80.7500**	75.7500	72.2500	65.5000	59.5000
20	77.7500	78.7500	**80.2500**	68.7500	68.7500
50	78.5000	81.2500	**85.0000**	80.7500	81.2500
70	78.2500	79.2500	80.2500	81.0000	**81.2500**
100	79.0000	79.7500	82.0000	85.2500	**87.5000**

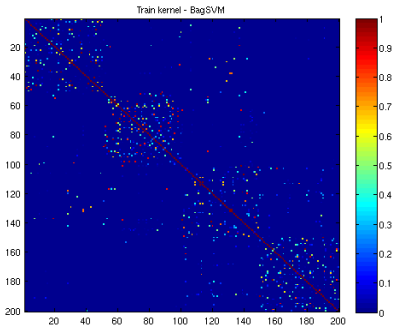
Table 2: Inductive accuracy for different K across different labeled data size for Scene-15 dataset using only Bagged Kernel

The kernel gram matrices obtained for labeled data for different size for Caltech-101 dataset are shown in Figure 4.

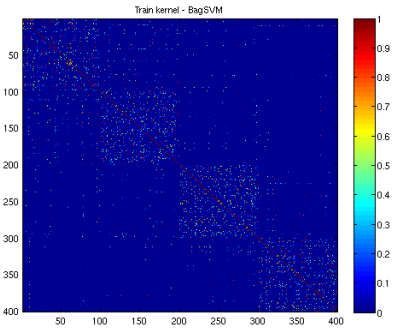


(a) 10 labeled samples per class

(b) 20 labeled samples per class



(c) 50 labeled samples per class



(d) 100 labeled samples per class

Figure 4: Kernel gram matrix for labeled data using bagged kernel for Caltech-101 dataset

The transductive accuracy obtained for unlabeled data across various labeled data size and k parameter of k-means are shown in Table 3. The inductive accuracy obtained for test data for different labeled data size are shown in Table 4

Labeled data size	K=5	K=10	K=50	K=70	K=100
10	57.1250	63.0000	**71.5000**	66.5000	54.3750
20	63.2500	70.6250	75.7500	**76.8750**	70.3750
50	63.1250	73.0000	79.0000	81.0000	**82.1250**
70	65.1250	71.1250	78.3750	79.6250	**81.8750**
100	60.3750	73.2500	75.2500	82.3750	**83.2500**

Table 3: Transductive accuracy for different K across different labeled data size for Caltech-101 dataset using only Bagged Kernel

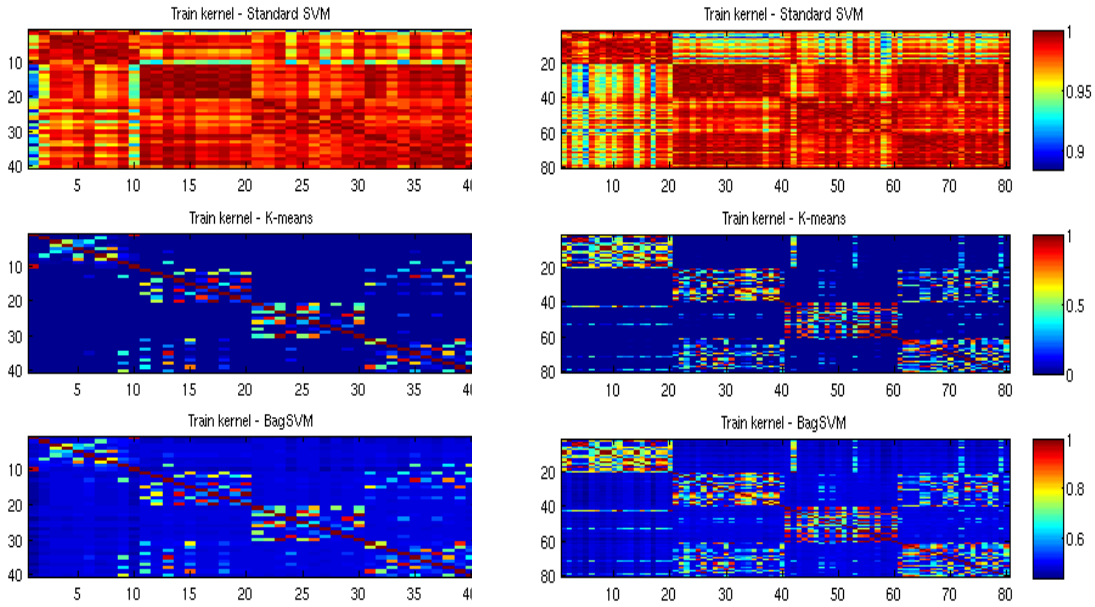
Labeled data size	K=5	K=10	K=50	K=70	K=100
10	55.7500	**63.0000**	60.5000	56.2500	47.7500
20	67.5000	70.2500	**74.2500**	73.7500	72.2500
50	63.0000	72.5000	75.2500	**80.2500**	79.5000
70	65.0000	73.7500	79.7500	85.2500	**85.5000**
100	68.0000	75.2500	78.0000	82.5000	**85.7500**

Table 4: Inductive accuracy for different K across different labeled data size for Caltech-101 dataset using only Bagged Kernel

5.2.3 SVM RBF Kernel deformed with Bagged Cluster Kernel

The analysis of linear combination of RBF kernel and bagged kernel across various labeled data size are described as follows. The α value for the linear combination used is 0.5.

For Scene-15 dataset, with 10 labeled samples per class the kernel gram matrices obtained are shown in Figure 5. And for labeled data size per class ={20,50,100}, the best kernel gram matrices obtained are shown in Figures 5 and 6.



(a) 10 labeled samples per class

(b) 20 labeled samples per class

Figure 5: Kernel gram matrices using RBF + Bagged kernel for Scene-15 data set

The accuracy of unlabeled data is compared across different labeled data size per class using the linear combination of RBF kernel and bagged kernel. The accuracy is depicted in Figure 7. The plot is between the number of clusters used in k-means and accuracy. We can observe that the performance is increasing as the training data size increases.

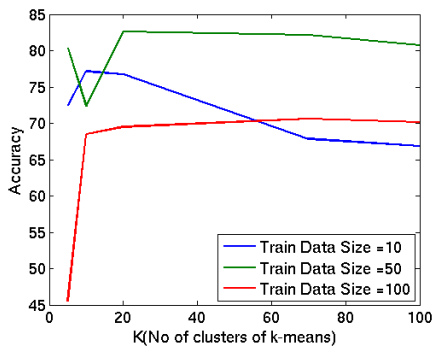
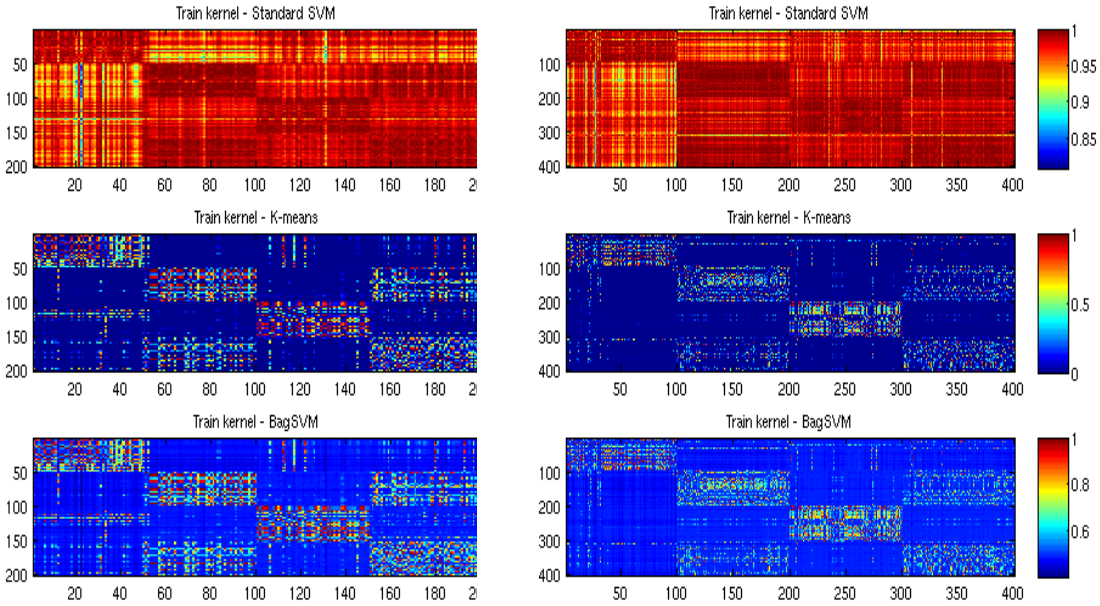


Figure 7: Plot of accuracy for different K across different training data size for Scene-15 dataset



(a) 50 labeled samples per class

(b) 100 labeled samples per class

Figure 6: Kernel gram matrices using RBF + Bagged kernel for Scene-15 data set

The Table 5 shows the transductive accuracy values of unlabeled data obtained for different values of K and labeled data size. The inductive accuracy obtained for test data samples for different labeled samples are shown in Table 6

Labeled data size	K=5	K=10	K=50	K=70	K=100
10	72.5000	77.1250	**76.7500**	67.8750	66.8750
20	79.1250	79.7500	**82.3750**	65.0000	71.1250
50	80.3750	72.3750	**82.6250**	82.1250	80.7500
70	80.2500	81.6250	**81.8750**	78.7500	69.5000
100	45.5000	68.5000	69.5000	**70.6250**	70.1250

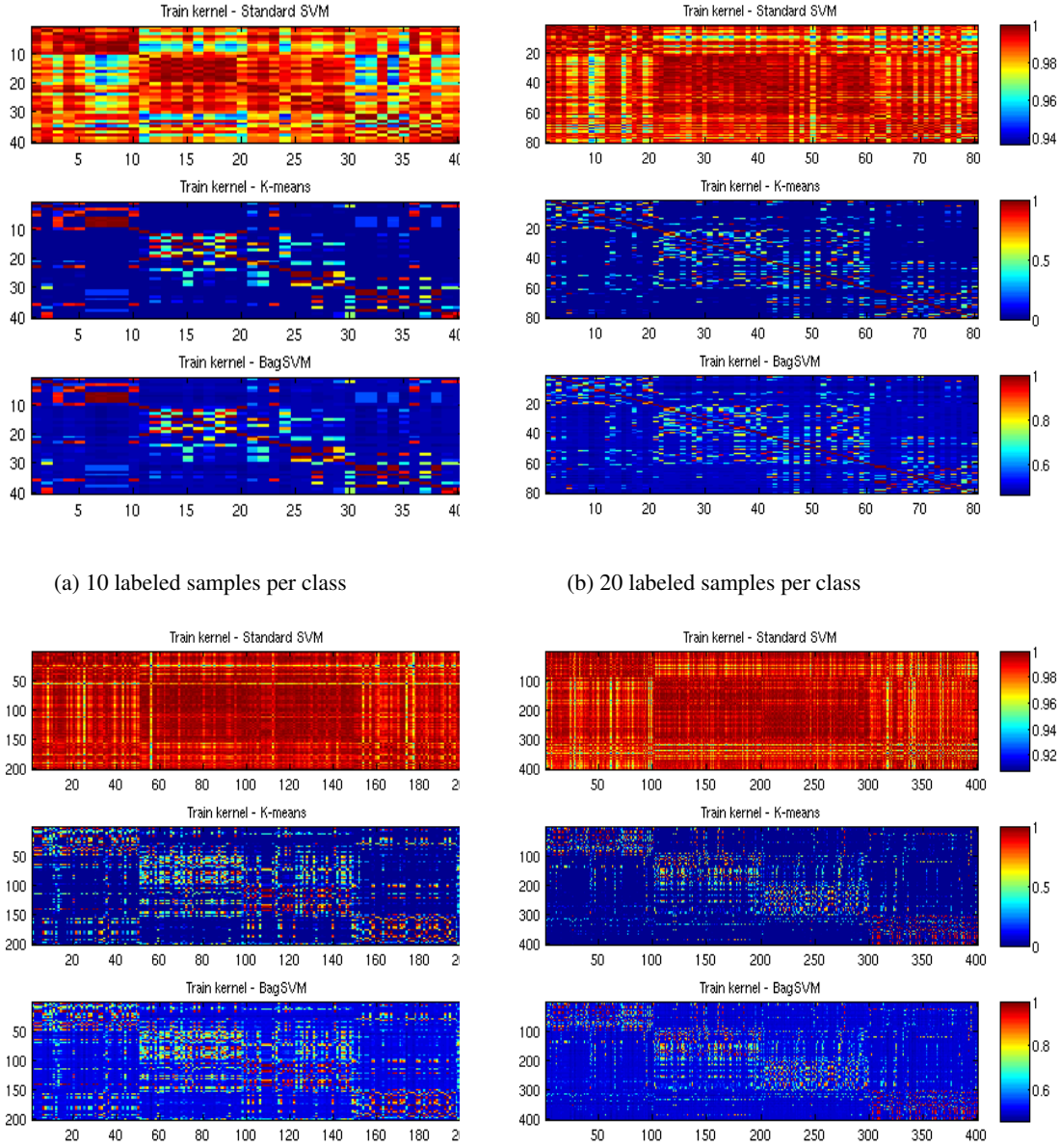
Table 5: Transductive accuracy for different K across different labeled data size for Scene-15 dataset using linear combination of RBF kernel and Bagged Kernel

Labeled data size	K=5	K=10	K=50	K=70	K=100
10	65.5000	63.7500	**79.0000**	60.2500	59.2500
20	77.5000	76.0000	**76.2500**	74.5000	69.2500
50	65.7500	75.5000	**82.0000**	73.2500	70.7500
70	62.0000	81.2500	82.0000	**86.5000**	84.7500
100	75.5000	85.5000	86.7500	86.5000	**86.7500**

Table 6: Inductive accuracy for different K across different labeled data size for Scene-15 dataset using linear combination of RBF kernel and Bagged Kernel

The combination of RBF kernel and bagged kernel is outperforming RBF kernel alone. But the performance on bagged kernel without RBF kernel is better than linear combination of both. The detailed observation is discussed in the section 6.

For Caltech-101 dataset, with per class labeled data size as 10, 20,50,100} are shown in Figure 8.



(a) 10 labeled samples per class

(b) 20 labeled samples per class

(c) 50 labeled samples per class

(d) 100 labeled samples per class

Figure 8: Kernel gram matrices using RBF+Bagged kernel for Caltech-101 data set

The accuracy comparison across different size of labeled samples per class using the linear combination of RBF kernel and bagged kernel is shown in Figure 9. The plot is between the number of clusters used in k-means and accuracy.

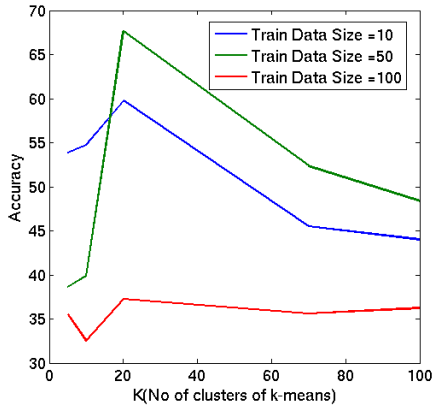


Figure 9: Plot of accuracy for different K across different labeled data size for Caltech-101 dataset

The Table 8 shows the transductive accuracy values for unlabeled data obtained for different values of K and labeled data size. The inductive accuracy values are shown in Table **??**

Labeled data size	K=5	K=10	K=50	K=70	K=100
10	53.8750	54.7500	**59.7500**	45.5000	44.0000
20	39.6250	51.1250	**59.3750**	33.3750	29.2500
50	38.6250	39.8750	**67.6250**	52.3750	48.3750
70	35.8750	35.2500	**40.8750**	34.5000	38.3750
100	35.5000	32.5000	**37.2500**	35.6250	36.2500

Table 7: Transductive accuracy for different K across different labeled data size for Caltech-101 dataset using linear combination of RBF kernel and Bagged Kernel

Labeled data size	K=5	K=10	K=50	K=70	K=100
10	45.7500	58.2500	60.0000	**62.0000**	45.0000
20	60.2500	69.0000	**71.0000**	66.7500	54.0000
50	34.0000	52.7500	74.5000	**80.5000**	80.2500
70	60.2500	71.7500	73.2500	81.2500	**82.7500**
100	61.5000	80.0000	67.0000	81.7500	**82.7500**

Table 8: Inductive accuracy for different K across different labeled data size for Caltech-101 dataset using linear combination of RBF kernel and Bagged Kernel

The performance of combination of both the kernel is better than SVM RBF kernel. But bagged kernel alone is outperforming other two.

5.3 Semi-supervised GMM

Feature vector selection Since the original feature vector is 119 dimensional, using it for training a semi-supervised classifier will require a large labeled set initially. Therefore, we have used the following two types of reduced feature vectors.

1. 20 dimensional GIST
2. 10 orthogonal dimensions obtained by applying PCA on original dataset

In both the cases, **the labeled set contains 40 samples**. In transductive setting, the unlabeled set has 260 samples. In inductive setting, the unlabeled set contains 160 samples and test set contains 100 samples.

5.3.1 Transductive setting

The confidence level threshold θ used for moving a predicted item into labeled set is in terms of the log of likelihood.

Using GIST feature vector

When $\theta >= 20$, only around 65% of the unlabeled set goes into the labeled set. Transductive accuracy is 85.71% for scene-15 and 82.92% for caltech-101 dataset.The confusion matrices for the two datasets are shown below.

class	1	2	3	4
1	95	0	0	8
2	4	151	0	27
3	0	0	97	0
4	0	31	0	77

Table 9: Dataset scene-15 with $\theta > 20$

class	1	2	3	4
1	117	20	12	5
2	22	179	6	0
3	12	22	140	8
4	2	0	4	114

Table 10: Dataset caltech-101 with $\theta > 20$

When $\theta >= 5$, 90% percent get labeled for scene-15 dataset and 95% for caltech-101 dataset. Transductive accuracy is 82.39% for scene-15 and 73.45% for caltech-101 dataset. The confusion matrices for the two datasets are shown in tables 11 and 12.

class	1	2	3	4
1	190	2	17	25
2	2	186	0	59
3	5	1	232	3
4	2	48	2	169

Table 11: Dataset scene-15 with $\theta > 5$

class	1	2	3	4
1	151	39	46	19
2	14	188	43	12
3	11	25	209	9
4	24	2	23	191

Table 12: Dataset caltech-101 with $\theta > 5$

Using PCA's reduced feature vector When we applied PCA on the 8000*119 feature dataset, the first 10 dimensions accounted for more than 95% of the variance. We projected the data on these ten dimensions and applied GMM on the reduced feature dataset. However, the performance was below that obtained using GIST features only. This is indicative of the fact that PCA is good for reduced dimensionality representation but not necessarily good for discrimination tasks.

Accuracy obtained on scene-15 dataset is 73.62% and on caltech-101 dataset is 85.23%. When the $\theta >= -15$, 71% of unlabelled got labelled in the first dataset and 83% got labeled in the second dataset.

5.3.2 Inductive setting

In the case of scene-15 dataset, confidence threshold of $\theta >= 15$ gives 82.80% transductive accuracy and 83.5% inductive accuracy. Around 64% of the unlabeled data got labeled. Increasing the confidence threshold improved the transductive accuracy but the inductive performance was poorer. Confusion matrices on unlabeled and test data are shown in tables 13 and 14.

class	1	2	3	4
1	81	0	0	14
2	0	118	0	22
3	1	0	93	0
4	3	31	0	50

Table 13: Dataset scene-15 with $\theta > 15$ on unlabeled data

class	1	2	3	4
1	90	1	1	8
2	0	77	0	23
3	3	1	94	2
4	3	24	0	73

Table 14: Dataset scene-15 with $\theta > 15$ on test data

class	1	2	3	4
1	68	9	3	0
2	12	80	1	0
3	4	9	76	4
4	0	0	5	40

Table 15: Dataset caltech-101 with $\theta > 25$ on unlabeled data

class	1	2	3	4
1	92	2	2	4
2	17	77	1	5
3	7	14	76	3
4	2	3	11	84

Table 16: Dataset scene-101 with $\theta > 25$ on test data

In the case of caltech-101 dataset, the confidence threshold of $\theta >= 25$ gives 84.88% transductive accuracy and 82.25% inductive accuracy. Only 48% of unlabeled data got labeled due to high confidence. But the performance on unseen test data was better than with models using lower confidence.Confusion matrices on unlabeled and test data are shown in tables 15 and 16.

5.4 Supervised GMM

The dataset is divided in the ratio 2:1 into training and test data. Each of the 4 classes has 200 images for training and 100 images for testing. Each image feature vector is 119 dimensional concatenating the 20 dimensional GIST, 59 dimensional PHOG and 40 dimensional LBP features. Due to the large feature size and limited training samples, we have used diagonal co-variance matrix with number of GMM components per class limited to 2. We have used a small regularization constant of 0.00001.

Dataset-15 is classified with 89% test data accuracy.

Dataset-101 is classified with 93.25% test data accuracy.

6 Discussion & Observations

6.1 Scene-15 dataset

6.1.1 Comparison of Performance

Method	Labeled data Size	K(no of clusters)	Accuracy(Scene-15) % (Transductive)
Supervised GMM	200	-	89
Semi-supervised GMM(GIST)	40	-	85.71
Semi-supervised GMM(PCA)	40	-	73.62
SVM-RBF kernel	70	-	45.63
SVM-Bagged kernel	**20**	**10**	**82.63**
SVM-Bagged kernel	100	70	85.13
SVM-RBF+Bagged kernel	50	50	82.63

Observations

- Supervised GMM performs well with training data size equal to 200 labeled examples per class and gives maximum accuracy of 89%.
- Semi-supervised GMM with GIST features is performing better than with the reduced dimensions obtained by PCA. This is because PCA is not discriminative.
- For semi-supervised GMM, best performance obtained for threshold, $\theta(log(posteriorprobability)) >= 20$. High confidence helps to avoid placing noisy sample points to be moved from unlabeled to labeled set.
- RBF kernel is poorly performing with less number of labeled data compared with other kernel methods
- Bagged Kernel - From the kernel gram matrices shown in Figure 3, we can observe that matrix is becoming block diagonal from k=10 to k=50 and after that not much improvement is there. The best k observed is 50. 82% accuracy obtained with 20 labeled examples per class.
- RBF kernel + Bagged kernel - From the kernel gram matrices shown in Figures 5 we can observe that RBF kernel is fully dark for RBF kernel. This is due to few labeled examples. For $\mathbf{K}_{bag}$, we can see the blocks even-though, it is not pure block diagonal. In Figure 6, block diagonal structure is clearly visible for $\mathbf{K}_{bag}$. In Figure 7, we can observe that best accuracy is obtained for 50 labeled examples per class. The accuracies shown in Table 2a, promises the same.

6.2 Caltech-101 dataset

6.2.1 Comparison of Performance

Method	Labeled data Size	K(no of clusters)	Accuracy(Caltech-101) % (Transductive)
Supervised GMM	200	-	93.25
Semi-supervised GMM(GIST)	40	-	82.92
Semi-supervised GMM(PCA)	40	-	85.23
SVM-RBF kernel	70	-	37.38
SVM-RBF+Bagged kernel	50	50	67.63
SVM-Bagged kernel	100	100	83.25

Observations

- Supervised GMM performs well with training data size equal to 200 labeled examples per class and gives maximum accuracy of 93%.

- Semi-supervised GMM is performing well with GIST features as well as with reduced features.
- Figure 1 illustrate that accuracy obtained using RBF kernel is performing poor with less labeled data
- Bagged kernel - In Figure 4, for labeled data size =10, block diagonal of kernel gram matrix is not visible across classes. From labeled datasize=20 onwards, block diagonal is visible. The best accuracy obtained at labeled data size=100 and K =100
- RBF kernel + Bagged kernel - For label data size=10 , it is not fully block diagonal. From labeled data size =20, the block diagonal is becoming clearer. All these are illustrated in Figure 8
- Deformed kernel is performing better at k=50
- Bagged Cluster Kernel is performing better than SVM kernel deformed with bagged kernel

7 Conclusion

Bagged cluster kernel is performing similar to semi-supervised GMM in both Scene-15 and Caltech-101 dataset. And the performance is good with less labeled examples and accuracy is close to supervised GMM which uses more labeled examples. But the performance of bagged cluster kernel is not stable due to the instability of k-means. In Caltech-101, dataset SVM kernel deformed with bagged cluster kernel is performing poor, where as deformed kernel is performing close to semi-supervised GMM in Scene-15. Finally we conclude that the semi-supervised performance using cluster kernel is promising well in Scene-15 dataset.

References

[1] Jason Weston, C Leslie, Eugene Ie, D Zhou, A Elisseeff, W S Noble. (2005) Semi-supervised protein classification using cluster kernels, *Vol. 21 no. 15 2005, pages 32413247 doi:10.1093/bioinformatics/bti497*

[2] Devis Tuia & Gustavo Camps. (2009) Semisupervised Remote Sensing Image Classification With Cluster Kernels, *224 IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, VOL. 6, NO. 2*

[3] Dengxin Dai & Luc Van Gool. (2013) Ensemble Projection for Semi-supervised Image Classification, *ICCV*

[4] L. Fei-Fei, R. Fergus and P. Perona. (2004) Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*

[5] Devis Tuia & Gustavo Camps. (2011) Urban Image Classification With Semisupervised Multiscale Cluster Kernels, *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, VOL.*

[6] Olivier Chapelle and Weston, Jason and Schölkopf, Bernhard. (2003) Cluster Kernels for Semi-Supervised Learning, *Advances in Neural Information Processing Systems*