Feature Combination for Multiclass Object Classification with Clustered Dataset

Peng Wang
COSC 174 - Machine Learning and Statistical Data Analysis
Spring 2012

Problem Statement

In multiclass object classification, a key element is to design a robust identification of relevant class specification in presence of intra-class variations. However, this is a difficult problem due to high variability of visual appearance within each class. One possible approach is to adaptively combine a set of diverse and complementary features-such as features based on color, shape-in order to discriminate each class best from all other classes [1].

Gehler and Nowzin proposed a booting approach that is to optimize jointly over a linear combination of some real valued functions (v_f). 5 fold CV is used to select the best hyper parameters for each v_f individually. Intuitively, randomly split the training dataset may not yield the best training result of v_f. Torralba and Efros [2, 3] provided a comparison study using a set of popular datasets to evaluate relative data bias, cross-dataset generalization, etc. The study showed an insight that we might be able to improve training by carefully collect the datasets.

Inspired by Torralba and Efros's research, I proposed a variation of feature combination algorithm that implements data classification before the hyper parameters selection in the Gehler and Nowzin's approach. We anticipate that pre-data classification will yield a better training result in hyper parameters selection, which will ultimately improve the overall model performance in terms of accuracy.

Methods

My proposed model is a combination of feature combination for multiclass object classification *and* a clustered dataset, each of which has an associated weighting of

stylistic features. I will address the methods for dealing with each aspect of the project individually.

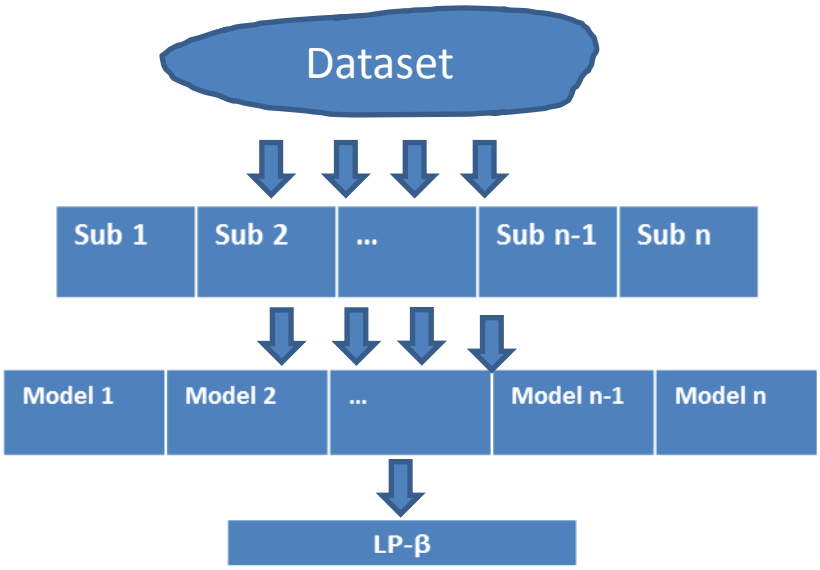

Figure 1. Model schematic

Proposed algorithm

1. Choose F features, cluster dataset to K subsets by spectral clustering algorithm [8]
2. Train C classifiers for F features on K*F subsets. The real valued output of K*F SVMs:

$$V_{fk}(x) = K_f(x)^T * \alpha_{fk} + b_{fk}$$

Where: Km: kernel matrix for feature f, f=1,…F, k=1,…K.

3. Mixing coefficients β_{fk} are learned the following multiclass extension of LP-boost

$$\min_{\beta,\xi,\rho} \quad -\rho + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i$$

s.t.

$$\sum_{f=1}^{F}\sum_{k=1}^{K}\beta_{fk} * V_{fk,c(i)}(x_i) - \max_{c(j)\neq c(i)}\sum_{f=1}^{F}\sum_{k=1}^{K}\beta_{fk} * V_{fk,c(j)}(x_i) + \xi_i \geq \rho$$

$i = 1, \ldots, N$

$$\sum_{f=1}^{F}\sum_{k=1}^{K}\beta_{fk} = 1, \quad \beta_{fk} \geq 0$$

Where: c(i) is the classifier of class i

An important problem when analyzing relational data between features is clustering, finding sets data sets that are "more similar" to each other in the dataset. Here, we implement the spectral clustering algorithm by Andrew NG et al for data clustering [11].

1. **Spectral clustering**

   Three futures are selected for clustering: color, shape and texture. Kernel matrix is defined as: Km=exp(gamma^-1*d(xi, xj)) [1]. Where, gamma is the mean of the pairwise distances, and d is the pairwise distance matrix.

   Affinity matrices for color shape and texture:

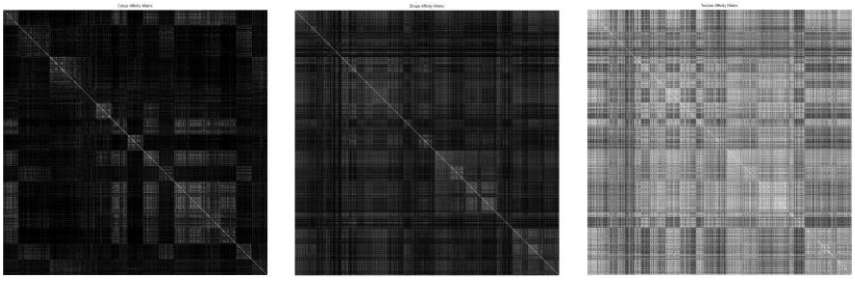

Figure 2. Affinity matrices

   However, there is a open issue in the spectral clustering: how to estimate the number of groups? For feature color, it is possible for me to manually label the

data. But it is much harder for me to label the data based on texture and shape. Therefore, it is worth studying a new method to automatically determine the number of groups. Lihi Zelnik-Manor and Pietro Perona [12] proposed a self-tunning spectral clustering algorithm, which will automatically give the number of groups.

2. Self-tuning spectral clustering

Zelnik-Manor modified the Ng-Jordan-Weiss algorithm by substituting the locally scaled affinity matrix instead of affinity matrix. The resulting algorithm automatically defined a scheme to set the scale parameter, which is used in affinity matrix calculation. By analyzing the eigenvectors, Zelnik-Manor proposed a new method to determine the number of groups automatically.

	Color	Shape	Texture
Best number of groups	3	4	6

However, in practice, less number of groups works best for our project. Therefore, number of groups in our project is set at k=2.

3. Training for SVMs

The SVMs are trained on clustered dataset. The algorithm I choose to learn SVMs is L1-norm soft margin. The label of dataset y(i)s have been mapped into {-1 1}. “quadprog” in Matlab is used to optimize the primal problem:

$$arg\min_{\alpha,b\xi} \frac{1}{2}\|\alpha\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$s.t.\ \ y^{(i)}\left(\alpha^T x^{(i)} + b\right) > 1 - \xi_{i,} \quad \forall i$$
$$\xi_i \geq 0, \ \ \forall i$$

4. LP-beta

In our project, the function V_f is a C dimensional space $V_f(x)$->R^c. The c'th output at sub dataset k of V_f is denote by $V_{fk.}$ The mixing coefficients β_{fk} are learned by the multiclass extension of LP-beta. Where, β_{fk} is a single vector for all classes. The β_{fk} defines a combination that works well for all classes jointly. "linprog" from Mosek is used to optimize the linear combination problem.

Dataset

In consistent with Gehler and Nowzin's study, I will apply the method to two data sets: Oxford flowers [4] and Caltech datasets [5, 6]. I might choose some other datasets for preliminary testing.

Figure 3. Example images from Oxford flowers dataset

The dataset consists of 17 different types of flowers with 80 images per category. The dataset comes with three pre-computed distance matrices [4]. I split the dataset into three subsets: test (17x20 images), train (17x40) and validation (17x20 images).

Results and discussion

In this section I present results on the Oxford flowers dataset. 7 experiments have been conducted for comparison study. Training accuracies for all experiment are above 99%. Since the training accuracy is not very informative across the experiments, I will not report it for each experiment individually.

1. Experiment 1

Experiment 1 serves as a baseline case here for comparison study later. We do not perform any data clustering. We simply train SVMs on three feature dataset for each class. Then, SVMs trained on the dataset are linearly combined by the β_{fk} optimized through LP-beta.

	Validation	Test
Accuracy	0.7382	0.7265

2. Experiment 2

   For the second experiment, spectral clustering algorithm is applied to three feature dataset. k, here, is 2 for each feature. SVMs are trained on clustered sub datasets. The final classifier is a linearly combination of SVMs trained in previous step.

	Validation	Test
Accuracy	0.6088	0.6471

   However, the accuracies on validation and test sets are worse comparing to the results from the baseline case. The reason for the worse results is caused by some poor performance SVMs trained on clustered sub dataset. Some clustered sub datasets only have few positive data points. As a consequence, the SVMs trained on these datasets have poor accuracy in validation and test sets.

3. Experiment 3

   Given the results in experiments 2, it is intuitively that we need to change the data clustering strategy. Instead of clustering the data on three feature dataset, we clustered the dataset within each class. Firstly, we define the subset data for each class for three features respectively. Then, we train the SVMs on each subset data within each class, and linearly combine the SVMs.

	Validation	Test
Accuracy	0.6794	0.6882

No surprise, the accuracy has been improved. However, the results from experiment 3 are still worse than the one from experiment 1. A intuitively guess for the worse results is caused by the SVMs for experiment 3 are trained on the subset data, which has less positive label data than experiment 1. In LP-beta step, it requires strong SVMs, which normally can be obtained from training on dataset with more positive label data.

4. Experiment 4

   In experiment 3, we concluded that LP-beta tends to work better on SVMs trained on dataset with more positive label data. In experiment 4, we manually segmented train set to 17*39 and 17*1, two subsets. In this way, we at least have one SVMs trained on dataset with more positive label data. And we increase the dimension of β_{fk}.

	Validation	Test
Accuracy	0.6794	0.6412

   However, the results form experiment 4 is not as good as we expected. An analysis of a low accuracy is straight forward. We increased the fitting parameters, which demands more training data.

5. Experiment 5

   Since there is no way to obtain more training data points, I simply linearly combined the SVMs trained from experiment 1 and 3, three SVMs for each feature for each class. It gave the best accuracy for data clustering based algorithm.

	Validation	Test
Accuracy	0.7147	0.6912

6. Experiment 6

   In previous experiments, all algorithms based on clustered dataset failed to outperform the baseline case. Experiment 6, I conducted in this project is to

linear combine the best SVMs in terms of validation error from experiment 1 and experiment 3, one SVM for each feature.

	Validation	Test
Accuracy	0.7088	0.6735

However, the validation error is in terms of accuracy of prediction of class. While in LP-beta, it takes real values. The optimization result heavily relies on the quality of the SVMs. It requires SVMs that results large margin in training and validation steps.

7. Experiment 7

In this experiment, I changed my linear combination method in experiment 6. Instead of linear combine the best SVMs in terms of validation error from experiment 1 and experiment 3, I replace the classifiers in experiment 3 with linear combination of classifiers in experiment 3. New algorithm:

- Train SVMs on the dataset without clustering, you will have one classifier for each feature of each class, c_f(i), i for class 1 to 17, f for feature.
- Cluster the data within each class, k=2.
- Train SVMs within each class, you will have two classifiers for each feature of each class, c_f1(i) and c_f2(i)
- Loop through the 17 classifier, replace c_f(i) by c_f1(i)+c_f2(i). By linear combining c_f1(i) and c_f2(i), we make points further away from the hyper-plane.
- Run LP-beta, and choose the best combination of replacement

	Validation	Test
Accuracy	0.7618	0.7471

Finally, we figure out an algorithm that outperforms the state of art classifier. In this algorithm, the weak learners in experiment 1 have been replaced by strong learners, which are composed of two weak learners by linear combination from experiment 3.

Conclusion

In this project, we tested several clustering based multiclass object classification method. Experiment 1-6 failed to outperform the baseline case. Experiment 7 actually outperformed the state of art classifier. One may be improve the accuracy by carefully picking classifiers for the linearly combination step.

There several lessons I learned from this project:

- The LP-beta algorithm for multiclass object classification requires strong SVMs
- Clustering data for all classes method tends to have worse results comparing to the one with clustering data within each class
- Clustering based algorithms increase the number of fitting parameters, which demands for more training data.
- Feature combination multiclass object classification algorithm takes real value for β_{fk} learning. Therefore, the accuracy of individual classifier is not a good indicate of a strong SVM.
- Simply segmenting dataset to increase the dimension of β_{fk} does not benefit the performance of the algorithm.

Acknowledgement

I thank Professor Lorenzo Torresani for his help on this project and his teaching on Machine Learning. I also thank our TA, Weifu Wang. At last, I want to thank my classmates for helpful discussion and comments.

References

1. P. V. Gehler and S. Nowozin, "On Feature Combination Methods for Multiclass Object Classification", IEEE International Conference on Computer Vision (ICCV), 2009.
2. A. Torralba, and A. Efros, "Unbiased Look at Dataset Bias", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
3. A. Bergamo, and L. Torresani, "Exploiting weakly-labeled Web images to improve object classification: a domain adaptation approach", NIPS, 2010.

4. M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In ICCV, pages 1447-1454, 2006.
5. L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories", In CVPRW, page 178, 2004.
6. P. V. Gehler and S. Nowozin, "Let the kernel figure it out: Principled learning of pre-processing for kernel classifiers"' In CVPR, 2009.
7. L. Duan, I.W.-H. Tsang, D. Xu, and S. J. Maybank, "Domain transfer svm for video concept detection", In CVPR, 2009.
8. K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Transferring visual category models to new domains", In ECCV, 2010.
9. J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms", MULTIMEDIA -07,2007.
10. C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 1st ed.,
11. A.Y. Ng, M.I. Jordan, and Y. Weiss, On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems 13, 2001. October 2007.
12. L. Zelnik-manor and P. Perona, Self-tuning spectral clustering, Advances in Neural Information Processing Systems 17, pp. 1601-1608, 2005, (NIPS'04)