Convex Feature Clustering and Selection With Class Label Information

Daniel Andrade s-andrade@cj.jp.nec.com
Security Research Laboratories, NEC
1753, Shimonumabe, Nakahara-ku, Kawasaki, Japan
Kenji Fukumizu fukumizu@ism.ac.jp
The Institute of Statistical Mathematics
10-3 Midoricho, Tachikawa, Tokyo 190-8562 Japan
Okajima Yuzuru y-okajima@bu.jp.nec.com
Security Research Laboratories, NEC
1753, Shimonumabe, Nakahara-ku, Kawasaki, Japan

Abstract

Clustering, like feature selection for classification, is an important step to understand and interpret the data. However, clustering of features is often performed independently of the classification step, which can lead to undesirable clustering results. Therefore, we propose a method that jointly clusters and selects features that are relevant for classification. We formulate the problem as a convex optimization problem which uses both, the similarity information between features, and the information from labeled samples. In order to solve the convex problem, we propose an ADMM algorithm with an expression of the primal-variable optimization step that allows for the precalculation of computational expensive terms. Our experiments on synthetic data shows that our proposed method can effectively use the class label information to recover the correct clustering of features. Furthermore, we confirm that our proposed algorithm with gradient descent precalculations scales well with the number of features.

1 Introduction

Clustering is often the first step to understand the data. However, depending on the application, the desired clustering can be quite different.

As a solution, [She(2010), Bondell and Reich(2008)] proposed to incorporate additional real-valued information about data samples into the clustering process. Their methods combine linear regression with an l1-penalty of the parameters' pairwise differences in [She(2010)], and with parameters' pairwise maximum in [Bondell and Reich(2008)], respectively. However, their methods cannot be applied to classification.

Here, we propose to incorporate samples with class label information into the clustering process. As an application, consider, for example, text classification, where we are interested in identifying meaningful word clusters that are relevant for the classification task. Furthermore, we assume that for each feature (word), we have additional information given, for example, in the form of feature embeddings (word embeddings). We emphasis that the class label information is available for samples (text documents) not for features, and therefore our method can be considered as a kind of *distant*-supervised clustering.[1]

Our proposed method formulates the task as a convex optimization problem combining the multi-class logistic regression loss with a sum of group lasso penalties. In contrast to recent convex clustering methods [Chi and Lange(2015), Hocking et al.(2011)Hocking, Joulin, Bach, and Vert, Wang et al.(2017)Wang, Zhang, Sun, and Fang], solving with ADMM does not lead to an analytic solution of the primal variable optimization problem. However, we show that the objective function and its gradient can be efficiently calculated by separating constant and non-constant terms for subsequent gradient descent steps. Two experiments on synthetic data confirm the ability of our proposed method to recover the correct clustering, and show that the proposed method scales well with the number of features.

[1] In order to distinguish it from clustering methods that directly place constraints on the clustered objects like e.g. in [Wang et al.(2014)Wang, Qian, and Davidson].

OPTML 2017: 10th NIPS Workshop on *Optimization for Machine Learning* (NIPS 2017).

2 Proposed Method

Let $B \in \mathbb{R}^{c \times d}$, where c is the number of classes, and d is the number of features. $B_{y,\cdot}$ is the weight vector for class y, and $B_{\cdot,i}$ are the class weights for feature i. Furthermore, $\boldsymbol{\beta}_0 \in \mathbb{R}^c$ contains the intercepts. We assume a multi-class logistic regression classifier defined by

$$f(y|\mathbf{x}, B, \boldsymbol{\beta}_0) = \frac{exp(B_{y,\cdot}\mathbf{x} + \boldsymbol{\beta}_0(y))}{\sum_{y'} exp(B_{y',\cdot}\mathbf{x} + \boldsymbol{\beta}_0(y'))} .$$

We propose the following formulation for jointly classifying samples $\boldsymbol{x}_s$, and selecting and clustering all features:

$$\underset{B,\boldsymbol{\beta}_0}{\text{minimize}} \quad -\sum_{s=1}^{n} \log f(y_s|\mathbf{x}_s, B, \boldsymbol{\beta}_0) + \nu \sum_{i_1<i_2} S_{i_1 i_2} ||B_{\cdot,i_1} - B_{\cdot,i_2}||_2 + \gamma \sum_i ||B_{\cdot,i}||_2 . \tag{1}$$

The second term is a group lasso penalty on the class weights for any pair of two features i_1 and i_2. We assume that a feature similarity matrix $S \in \mathbb{R}^{d \times d}$ is given a-priori, where $S_{i_1 i_2}$ defines the similarity between feature i_1 and i_2. Note that the penalty is large for similar features, and therefore encourages $B_{\cdot,i_1}$ and $B_{\cdot,i_2}$ to be equal. The last term encourages zero columns in B, and thus performs feature selection.

The final clustering of the features can be found by grouping two features i_1 and i_2 together if $B_{\cdot,i_1}$ and $B_{\cdot,i_2}$ are equal.

2.1 Optimization using ADMM and Precalcuations

First, we re-formulate the problem in Equation (1) as

$$\begin{aligned}
\underset{B,Z,\boldsymbol{\beta}_0}{\text{minimize}} \quad & -\sum_{s=1}^{n} \log f(y_s|\mathbf{x}_s, B, \boldsymbol{\beta}_0) + \nu \sum_{i=1}^{d} \sum_{j=i+1}^{d} S_{i,j} ||\mathbf{z}_{ij}||_2 + \gamma \sum_{i=1}^{d} ||\mathbf{z}_i||_2 \\
& \text{subject to} \\
& \forall i < j : \mathbf{z}_{ij} = \mathbf{b}_i - \mathbf{b}_j , \\
& \forall i : \mathbf{z}_i = \mathbf{b}_i .
\end{aligned}$$

where we defined $\mathbf{b}_j := B_{\cdot,j} \in \mathbb{R}^c$.

Using ADMM this can be optimized with the following sequence:

$$B^{k+1} := \underset{B,\boldsymbol{\beta}_0}{\arg\min} \quad -\sum_{s=1}^{n} \log f(y_s|\mathbf{x}_s, B, \boldsymbol{\beta}_0) + \frac{\rho}{2} \sum_{i=1}^{d} \sum_{j=i+1}^{d} ||\mathbf{z}_{ij}^k - \mathbf{b}_i + \mathbf{b}_j + \mathbf{u}_{ij}^k||_2^2 + \frac{\rho}{2} \sum_{i=1}^{d} ||\mathbf{z}_i^k - \mathbf{b}_i + \mathbf{u}_i^k||_2^2$$

$$\mathbf{z}_{ij}^{k+1} := \underset{\mathbf{z}_{ij}}{\arg\min} \quad \nu S_{i,j} ||\mathbf{z}_{ij}||_2 + \frac{\rho}{2} ||\mathbf{z}_{ij} - \mathbf{b}_i^{k+1} + \mathbf{b}_j^{k+1} + \mathbf{u}_{ij}^k||_2^2$$

$$\mathbf{z}_i^{k+1} := \underset{\mathbf{z}_i}{\arg\min} \quad \gamma ||\mathbf{z}_i||_2 + \frac{\rho}{2} ||\mathbf{z}_i - \mathbf{b}_i^{k+1} + \mathbf{u}_i^k||_2^2$$

$$\forall i < j :$$

$$\mathbf{u}_{ij}^{k+1} := \mathbf{z}_{ij}^{k+1} - \mathbf{b}_i^{k+1} + \mathbf{b}_j^{k+1} + \mathbf{u}_{ij}^k ,$$

$$\forall i \in \{1, .., d\} :$$

$$\mathbf{u}_i^{k+1} := \mathbf{z}_i^{k+1} - \mathbf{b}_i^{k+1} + \mathbf{u}_i^k ,$$

where k denotes the current iteration, and $\mathbf{u}_{ij}$ and $\mathbf{u}_i$ are the scaled dual variable for $\mathbf{z}_{ij}$ and $\mathbf{z}_i$, respectively.

In the following, we describe how each of the above optimization problems can be solved. The update of B can be solved with gradient descent. Let us define

$$g(B) := -\sum_{s=1}^{n} \log f(y_s|\mathbf{x}_s, B, \boldsymbol{\beta}_0) + \frac{\rho}{2} \sum_{i=1}^{d} \sum_{j=i+1}^{d} ||\mathbf{z}_{ij}^k - \mathbf{b}_i + \mathbf{b}_j + \mathbf{u}_{ij}^k||_2^2 + \frac{\rho}{2} \sum_{i=1}^{d} ||\mathbf{z}_i^k - \mathbf{b}_i + \mathbf{u}_i^k||_2^2 .$$

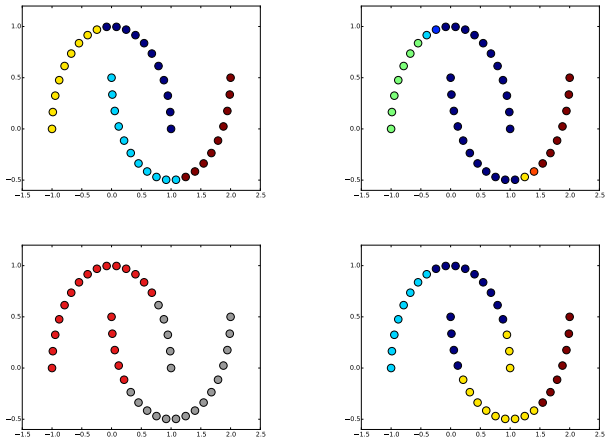

Figure 1: Shows the 40 features in their 2 dimensional embedding space. Left-upper plot: desired clustering result and output of proposed method, whereas bright yellow (cluster 1) and bright blue (cluster 2) features are associated with class A, and dark blue (cluster 3) and dark brown (cluster 4) features are associated with class B. Right upper plot: best clustering result achieved by convex clustering method [Chi and Lange(2015)]. Lower half: k-means with 2 and 4 clusters, respectively.

Due to the second term, the calculation of $g(B)$ is in $O(d^2)$, which is prohibitively expensive for repeated calculation in gradient descent. However, the double sum in the second term can be expressed as follows:[2]

$$\sum_{i=1}^{d}\sum_{j=i+1}^{d}||\mathbf{z}_{ij}^k-\mathbf{b}_i+\mathbf{b}_j+\mathbf{u}_{ij}^k||_2^2 = q_{all}-2\sum_{i=1}^{d}\mathbf{b}_i^T\mathbf{q}\uparrow_i+2\sum_{j=1}^{d}\mathbf{b}_j^T\mathbf{q}\downarrow_j+d\sum_{i=1}^{d}\mathbf{b}_i^T\mathbf{b}_i-\bar{\mathbf{b}}^T\bar{\mathbf{b}}\,,$$

where we defined $\bar{\mathbf{b}} := \sum_{i=1}^{d}\mathbf{b}_i$, and $q_{all} := \sum_{i=1}^{d}\sum_{j=i+1}^{d}\mathbf{q}_{ij}^T\mathbf{q}_{ij}$, $\mathbf{q}\uparrow_i := \sum_{j=i+1}^{d}\mathbf{q}_{ij}$, and $\mathbf{q}\downarrow_j := \sum_{i=1}^{j-1}\mathbf{q}_{ij}$, with $\mathbf{q}_{ij} := \mathbf{z}_{ij}^k+\mathbf{u}_{ij}^k$. Since q_{all}, $\mathbf{q}\uparrow_i$ and $\mathbf{q}\downarrow_j$ can be precalculated, each repeated calculation of $g(B)$ is in $O(d)$. Furthermore, we note that

$$\nabla_{\mathbf{b}_i}g(B) = -\sum_{s=1}^{n}\nabla_{\mathbf{b}_i}\log f(y_s|\mathbf{x}_s,B,\boldsymbol{\beta}_0)+\rho\Big((d+1)\mathbf{b}_i-\bar{\mathbf{b}}+\mathbf{r}_i\Big)\,,$$

where we defined $\mathbf{r}_i := -\mathbf{z}_i^k-\mathbf{u}_i^k+\sum_{j=1}^{i-1}(\mathbf{z}_{ji}^k+\mathbf{u}_{ji}^k)-\sum_{j=i+1}^{d}(\mathbf{z}_{ij}^k+\mathbf{u}_{ij}^k)$, which again can be precalculated.

The update of $\mathbf{z}_{ij}$ and $\mathbf{z}_i^{k+1}$ can be solved analytically by using the soft-thresholding operator:

$$\mathbf{z}_{ij}^{k+1} = \Phi_{\frac{\nu S_{i,j}}{\rho}}(\mathbf{b}_i^{k+1}-\mathbf{b}_j^{k+1}-\mathbf{u}_{ij}^k)\,,\text{and}\;\;\mathbf{z}_i^{k+1} = \Phi_{\frac{\gamma}{\rho}}(\mathbf{b}_i^{k+1}-\mathbf{u}_i^k)\,,$$

where $\Phi_a(\mathbf{v})$ is the block soft-thresholding operator (see e.g. [Boyd et al.(2011)Boyd, Parikh, Chu, Peleato, and Eckstein], page 45) defined as

$$\Phi_a(\mathbf{v}) := \max\left(1-\frac{a}{||\mathbf{v}||_2},0\right)\cdot\mathbf{v}\,,\text{with}\;\Phi_a(\mathbf{0})=\mathbf{0}.$$

3 Experiments

We ran two experiments on synthetic data. The first experiment is used to compare the quality of several clustering methods with our proposed method. The second experiment tests the quality and runtime of our proposed method for joint clustering and feature selection.

In our first experiment, we illustrate the advantage of the proposed method on a binary classification task with 40 features. Each feature is associated with a two dimensional feature embedding. The 40 features have a

[2]Due to the space constraints, we omit all derivations.

Table 1: Shows the total runtime in seconds for all $\gamma, \nu \in \{1.0, 0.1, 0.01\}$. ✓denotes that the correct clustering and feature selection could be found. CVXPY uses the convex solver of cvxpy [Diamond and Boyd(2016)]. ADMM and ADMM-Pre refer to the proposed method w/o precalculations. The number in brackets denotes the number of ADMM iterations.

	CVXPY	ADMM (100)	ADMM-Pre (100)	ADMM-Pre (200)	d
Runtime	7.31	51.03	16.92	28.02	20
Result	✓	✓	✓	✓	
Runtime	350.14	1227.62	172.34	337.57	100
Result	✓	✓	✓	✓	
Runtime	5142.36	4355.54	630.19	1188.0	200
Result	✓	not correct	not correct	✓	

Table 2: Class weights of each feature cluster. Features in the same cluster have the same class weights.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Class A	5.26	7.62	-5.42	-4.47
Class B	-6.15	-5.43	5.36	5.42

two moon shape, as shown in Figure 1. The 40 features are divided into 4 clusters, displayed in the left-upper plot of Figure 1 The class weights of each feature belonging to the same cluster are equal. The class weights for each cluster are shown in Table 2. In order to apply our method and the convex clustering method from [Chi and Lange(2015)], we create the similarity matrix S, using $S_{i_1,i_2} = exp(-0.5 \cdot ||\mathbf{e}_{i_1} - \mathbf{e}_{i_2}||)$, where $\mathbf{e}_i$ is the feature embedding of feature i. Furthermore, we generate 100 samples from each class by sampling from a multivariate normal distribution with mean vector equaling the class weight and covariance matrix S. As shown in Figure 1, neither k-means nor the method from [Chi and Lange(2015)] can find the correct clustering.

In our second experiment, we test the ability to perform jointly clustering and feature selection that are relevant for classification. The number of classes is now 3. First, we assume there are 100 features evenly split into 10 clusters. The a-priori information of features is given by the similarity matrix S defined such that features in the same cluster have similarity 0.9, and otherwise 0. Furthermore, we select 2 clusters, where each is split evenly into two clusters j_1 and j_2. The class weights are set such that cluster j_1 and cluster j_2 are associated with different classes. We call such two clusters contradicting. Furthermore, 4 clusters contain irrelevant features, and the corresponding class weights are set to 0. In summary, the features are grouped into 12 clusters, out-of which 4 clusters contain features that are a-priori similar, but associated to different classes, and, another 4 clusters are irrelevant. Finally, like in the first experiment, we sample 100 samples from each class. We find that our proposed method can perfectly identify the relevant features and groups them into the correct clusters. In contrast, any two step approach, clustering followed by classification with feature selection, will necessarily fail to identify the contradicting clusters.

Finally, in order to investigate the runtime, we repeat the second experiment for different number of features, i.e. $d \in \{20, 100, 200\}$, with $\gamma, \nu \in \{1.0, 0.1, 0.01\}$. We ran all experiments on one core Intel(R) Xeon(R) CPU with 2.60GHz. The results in Table 1 show that that the proposed ADMM with precalculations for the gradient descent step can be around 5 times faster than standard convex solvers, while achieving the same clustering solution.

4 Conclusions

We proposed a method for distant-supervised clustering with additional class label information of the samples. Our experiments on synthetic data confirmed the potential advantage of the proposed method over standard clustering methods like k-means and convex clustering. Furthermore, our experiments showed that the proposed ADMM with precalculations scales well with the number of features. Therefore, in future work, we plan to apply our method to real text data with more than 1000 features.

References

[Bondell and Reich(2008)] H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008.

[Boyd et al.(2011)Boyd, Parikh, Chu, Peleato, and Eckstein] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[Chi and Lange(2015)] E. C. Chi and K. Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.

[Diamond and Boyd(2016)] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[Hocking et al.(2011)Hocking, Joulin, Bach, and Vert] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*, page 1, 2011.

[She(2010)] Y. She. Sparse regression with exact clustering. *Electronic Journal of Statistics*, 4:1055–1096, 2010.

[Wang et al.(2017)Wang, Zhang, Sun, and Fang] B. Wang, Y. Zhang, W. W. Sun, and Y. Fang. Sparse convex clustering. *Journal of Computational and Graphical Statistics*, (just-accepted), 2017.

[Wang et al.(2014)Wang, Qian, and Davidson] X. Wang, B. Qian, and I. Davidson. On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, pages 1–30, 2014.