Subspace Clustering via Graph Regularized Sparse Coding

Alvaro Velasquez and Nasim Souly
Center for Research in Computer Vision
University of Central Florida

***Abstract*—Sparse coding has gained popularity and interest due to the benefits of dealing with sparse data, mainly space and time efficiencies. It presents itself as an optimization problem with penalties to ensure sparsity. While this approach has been studied in the literature, it has rarely been explored within the confines of clustering data. It is our belief that graph-regularized sparse coding can lead to a data space with discriminative features that can be clustered so that these clusters can be applied to various areas of interest in the field, such as video segmentation and tracking. In this paper, we propose a framework for clustering via graph-regularized sparse coding and apply this framework to video segmentation and image patch correspondence.**

I. Introduction

Sparse coding and graph regularization have been explored in the literature, but they are rarely utilized together. Even more rare is their usage for clustering data. Clustering can be applied anywhere that the structure or the feature space of some data needs to be grouped into meaningful sets; therefore, its applications abound in many different areas and it is because of this that there is interest in developing accurate clustering methods. It is our belief that graph-regularized sparse coding can make clustering more robust to outliers by eliminating noise in the original data while maintaining more discriminative features.

Sparse coding typically presents itself as a Least Squares estimate minimization problem of the form $min_s||X - BS||_2^2$ with a penalty on the parameter being minimized. In theory, this penalty would be the L_0 penalty, which is a constraint on the quantity of non-zero elements; unfortunately, solving for the L_0 penalty is an NP-Hard problem, and so an L_1 penalty is used as an approximation for practical purposes. Hence, the sparse coding problem can be presented as follows:

$$min_s||X - BS||_2^2 + \lambda \sum_{i=1}^{k} ||s_i||_1.$$

In the above approximation formulation, X denotes the m-by-k data matrix that is being approximated, B denotes an m-by-n overcomplete dictionary such that n $>$ k, and S is the minimization parameter corresponding to the n-by-k sparse coefficient matrix. Clearly, the product B*S attempts to approximate X and the penalty on the L_1-norm of S ensures sparsity. Ideally, the coefficient matrix will be sufficiently sparse to be efficiently computed by convex optimization [1].

As with any approximation, there is some error introduced when compared to an exact solution. A few causes of concern are apparent in practical sparse coding applications; one of these concerns is the substitution of the traditional L_0 penalty for an L_1 penalty. Another concern arises from the fact that, through sparsity, we may be leaving out relevant information in the feature space that could lead to inaccuracies in the final representation of the data. Fortunately, this is not as big of a problem as it seems since even random features contain enough information to recover the sparse representation of the data [2]. Moreover, making sure that the dimension of the feature space is sufficiently large is much more significant.

II. Related Works

Sparse coding and graph regularization have seen a surge of interest thanks to the demand for efficient solutions when dealing with large data and the fact that a sparse coding solution can be efficiently computed when dealing with sufficiently sparse data. Methods have been proposed for face recognition via sparse representations [2], clustering of sparse data [3], and graph-regularized matrix factorization [4], among other things.

It is worth noting that there are many Least Squares estimate models that can be applied to a sparse coding framework, where the optimal model selection varies depending on application and the non-differentiability of the objective function. [5] does a good job in surveying eight of the most popular methods and addressing their respective strengths and weaknesses. Throughout the rest of this paper, we use a LASSO formulation in our framework; LASSO denotes the RSS (Residual Sum of Squared errors, also known as the Least Squares estimate) problem with L1 regularization, which has received much attention in its methods of optimization, some of which are explored in detail in [6] and [7].

III. Motivation

Applying graph-regularized sparse coding and traditional sparse coding to image patch tracking in the SegTrack dataset resulted in very different accuracies, demonstrating that graph-regularization complements sparse coding and offsets the loss of accuracy that results from the sparse coding approximation. It is worth noting that the reconstruction error and sparsity rate of both methods is nearly identical, and so frame-to-frame patch matching is not based on Euclidean distance, but rather on a Max-Voting scheme. We start by computing a naive dictionary consisting of all the non-overlaping 18-by-18 image patches in the current frame. Each image patch vector x_i corresponds to an 18-by-18 image patch of interest. The 32 patches

of interest $[x_i^{(1)}]_{i=1}^{32}$ were chosen manually in the first frame. For each subsequent frame j, we attempt to find the image patches $[x_i^{(j)}]_{i=1}^{32}$ that correspond to the patches $[x_i^{(j-1)}]_{i=1}^{32}$ in the previous frame in a fully unsupervised manner. We define correspondence as the index α that maximizes the sum $\sum_l \sum_\alpha b_{l\alpha} s_{\alpha k}$, which is a Max-Voting method. The algorithm is as follows:

Input: Video frames Ω, initial patches $[x_i^{(1)}]_{i=1}^{n}$, and sparsity constraint T_0.
Output: $[x_i^{(j)}]_{i=1}^{32}$ for each frame j.

q $\leftarrow$ dim($x_1^{(1)}$, 1)
for t = 2 to m **do**
  X $\leftarrow$ $[x_1^{(t-1)}, x_2^{(t-1)}, ..., x_n^{(t-1)}]$
  B $\leftarrow$ Non-overlapping q-by-q patches in $\Omega^{(t)}$
  S $\leftarrow$ $min_S ||X - BS||_2^2 + \alpha Tr(S^T L S) + \beta \sum_{i=1}^{m} ||s_i||_1$
  L $\leftarrow$ $D - W$; where $W_{ij} = I_{x_i \in N(x_j)} * e^{-||x_i - x_j||^2/\sigma}$ and $D_{ij} = \sum_j W_{ij}$ whenever i = j and 0 otherwise
  for k = 1 to dim(B, 2) **do**
    for s = 1 to n **do**
      if k = $argmax_\alpha \sum_l \sum_\alpha b_{l\alpha} s_{\alpha k}$ **then**
        $x_s^{(t)} \leftarrow b_k$
      end if
    end for
  end for
end for

Note that the algorithm above is nearly identical when using graph regularization and when using traditional sparse coding; simply remove the graph regularization term $\alpha Tr(S^T L S)$ to maintain the traditional sparse coding framework. As part of the graph regularization term, the Laplacian matrix L of the data must be computed and, as a result, so must the similarity and degree matrices. We utilize a Heat Kernel metric for the similarity matrix. The results clearly show that adding graph regularization to a sparse coding formulation makes it more robust to outliers, as can be seen in figures 1 and 2. The results are quantified in the graphs seen in figures 3 and 4.

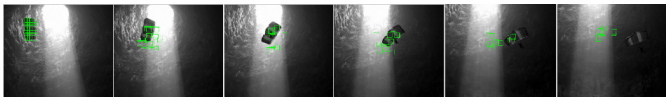


Fig. 1. Tracking results without graph regularization (Frames 1, 11, 21, 31, 41, 51).

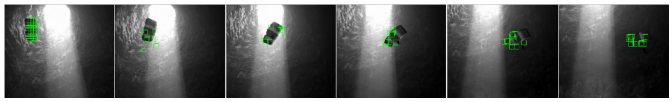


Fig. 2. Tracking results with regularization (Frames 1, 11, 21, 31, 41, 51).

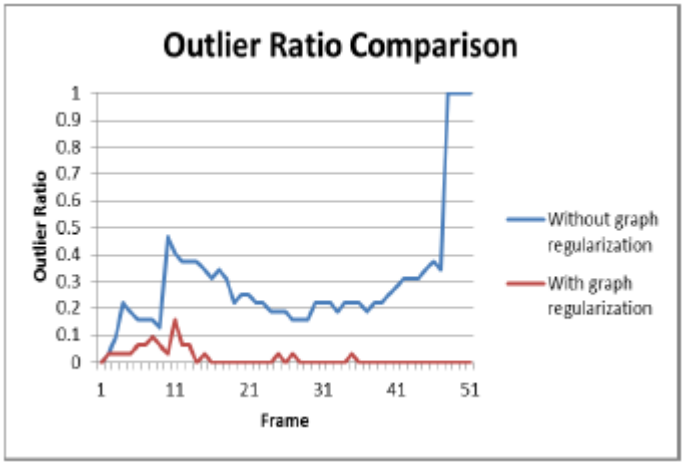


Fig. 3. Outlier ratios.

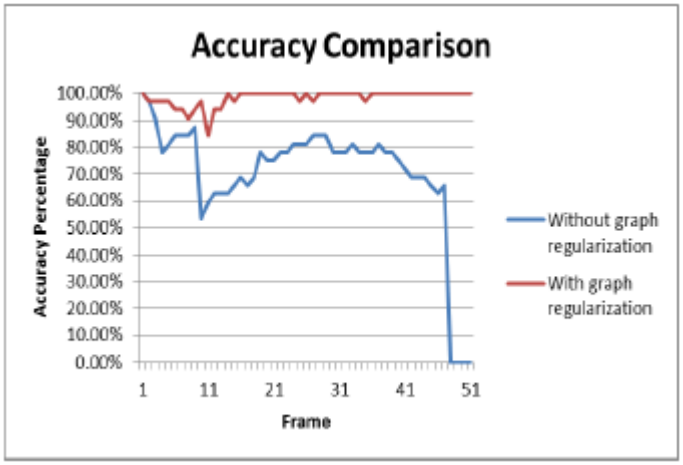


Fig. 4. Accuracies.

IV. Proposed Framework

We applied the method discussed in the previous section to segmentation. Denote by x_i the i^{th} q-by-q-by-q cuboid in the image volume Ω. An overcomplete dictionary for sparse data can be learned for the image volume by using K-SVD [8], which seeks to represent the data samples $[x_i]_i^N$ as sparse compositions by solving

$$min_{B,S} ||X - BS||_F^2 \text{ subject to } \forall i, ||x_i||_0 < T_0.$$

Let B denote the dictionary learned from this method. We can then solve the minimization problem

$$min_S ||X - BS||_2^2 + \alpha Tr(S^T L S) + \beta \sum_{i=1}^{k} ||s_i||_1$$

to obtain a sparse coefficient matrix S, where s_i is the column-wise vector that corresponds to the i^{th} cuboid in the data matrix such that $b_j * s_i$ approximates x_i. Since we

are approximating the data matrix via the product of the dictionary and the coefficient matrix, a reconstruction error ϵ is to be expected. Traditionally, we seek to minimize this reconstruction error to measure the effectiveness of the model; in doing so, we encounter the following minimization problem:

$$min_S||S||_1 \text{ s.t. } ||X - BS||_2^2 < \epsilon'$$

Where ϵ' is an acceptable reconstruction error threshold. Altogether, this makes for the following framework:

$$min_S||X - BS||_2^2 + \alpha Tr(S^T LS) + \beta \sum_{i=1}^{m} ||s_i||_1$$
$$\text{s.t. } ||X - BS||_2^2 < \epsilon'$$

Unfortunately, the model above does not fully complement the geometric interpretation of the data that is being exploited through the graph regularization term. A better model involves using a Max-Voting scheme instead of placing a constraint on the reconstruction error. This is due to the fact that graph regularlation yields sparse vectors with a weight distribution such that very few of the non-zero elements can discriminatively characterize the data. The results of using this Max-Voting scheme in tracking image patches can be seen in Section 3. A visualization of the framework can be seen below in figure 5.

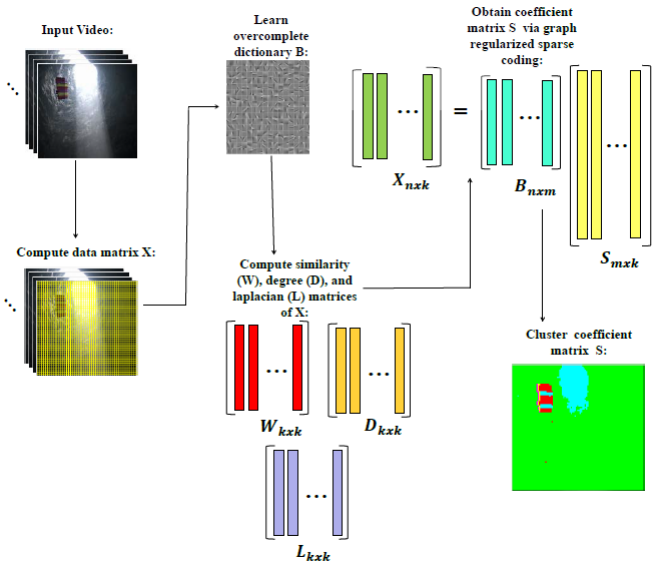


Fig. 5. Framework.

V. Algorithm

Our algorithm for video segmentation seeks to provide an accurate approximation of the different objects in a video by using cuboids. The algorithm is as follows:

Input: Image volume Ω, cuboid size q, sparsity constraint T_0, and number of clusters c.
Output: Cluster indices Θ.

for t = 1 to m **do**
  X $\leftarrow$ Non-overlapping q-by-q-by-q cuboids in Ω
  B $\leftarrow$ $min_{B,S}||X - BS||_F^2$ subject to $\forall i$, $||x_i||_0 < T_0$.
  L $\leftarrow$ $D - W$; where $W_{ij} = I_{x_i \in N(x_j)} * e^{-||x_i - x_j||^2/\sigma}$ and $D_{ij} = \sum_j W_{ij}$ whenever i = j and 0 otherwise
  S $\leftarrow$ $min_S||X - BS||_2^2 + \alpha Tr(S^T LS) + \beta \sum_{i=1}^{m} ||s_i||_1$
  $\Theta_t \leftarrow$ cluster S
  t $\leftarrow$ t + q
end for

Results of this algorithm when compared to traditional sparse coding formulations without graph regularization can be seen in figure 5.

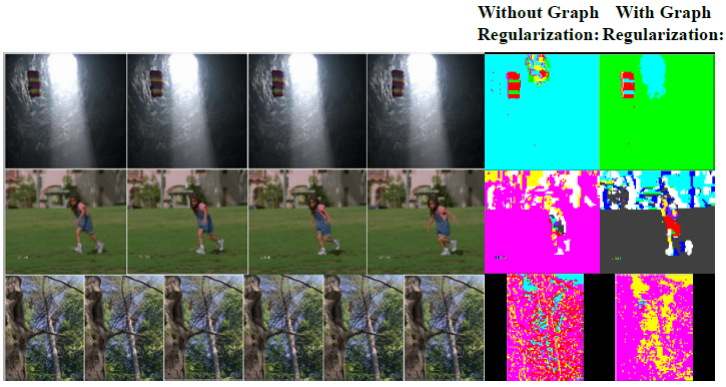


Fig. 6. Segmentation results.

VI. Conclusion and Future Work

Further testing needs to be conducted to prove the benefits of this framework as it pertains to clustering. It would be worthwhile to look into creating a novel clustering algorithm that utilizes Max-Voting as its main criterion for determining cluster centroid locations and update stages; this is because Max-Voting seems to exploit the distribution of graph-regularized data, hence it seems adequate that a Max-Voting clustering algorithm would best cluster the data in question. Furthermore, how to best represent the data matrix discriminatively is worth exploring as this plays a significant role in determining the values of the coefficient matrix and, subsequently, affects the accuracy of the clustering results.

References

[1] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[2] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.

[3] B. Saha, D. S. Pham, D. Phung, and S. Venkatesh, "Sparse subspace clustering via group sparse coding," 2013.

[4] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[5] M. Schmidt, "Least squares optimization with l1-norm regularization," *Project Report, University of British Columbia*, 2005.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[7] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[8] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: Design of dictionaries for sparse representation," *Proceedings of SPARS*, vol. 5, pp. 9–12, 2005.