Spectral Clustering with a Convex Regularizer on Millions of Images

Maxwell D. Collins[1], Ji Liu[2], Jia Xu[1],
Lopamudra Mukherjee[3], and Vikas Singh[1]

[1] University of Wisconsin–Madison, USA
[2] University of Rochester, USA
[3] University of Wisconsin–Whitewater, USA
mcollins@cs.wisc.edu, jliu@cs.rochester.edu, jiaxu@cs.wisc.edu,
mukherjl@uww.edu, vsingh@biostat.wisc.edu

Abstract. This paper focuses on efficient algorithms for single and multi-view spectral clustering with a convex regularization term for very large scale image datasets. In computer vision applications, multiple views denote distinct image-derived feature representations that inform the clustering. Separately, the regularization encodes high level advice such as tags or user interaction in identifying similar objects across examples. Depending on the specific task, schemes to exploit such information may lead to a smooth or non-smooth regularization function. We present stochastic gradient descent methods for optimizing spectral clustering objectives with such convex regularizers for datasets with up to a hundred million examples. We prove that under mild conditions the local convergence rate is $O(1/\sqrt{T})$ where T is the number of iterations; further, our analysis shows that the convergence improves linearly by increasing the number of threads. We give extensive experimental results on a range of vision datasets demonstrating the algorithm's empirical behavior.

1 Introduction

The need to *process* and make sense of the large number of images on the internet — for search, categorization, and ranking, motivates problems that are fundamental to vision and machine learning research today. To facilitate such inference tasks, the image is first expressed in terms of its response to a large set of specialized filters pertaining to texture, distinct object categories and appearance, among others. This information may be further complemented by textual cues that co-occur with the images, image tags, or hyperlinks to the image. With these representations in hand, the goal is to leverage all views simultaneously and derive a solution that best explains the given set of examples in the context of the inference objective of interest.

Clustering serves as an important exploratory tool for categorizing sets of images into semantically meaningful concepts. Mixture modeling and k-means remain traditional workhorses for this task and provide estimates of the parameters of an explicit model for the data. Spectral objectives, which are a focus

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 282–298, 2014.

of this paper, instead analyze the eigen structure (or spectrum) of a matrix derived from the pairwise similarities of nodes, and are especially useful when the cluster distributions correspond to more complex shapes [35]. Despite these advantages, spectral clustering is expensive for larger datasets. The most common optimization is to sparsify [7] or subsample [13,24] the matrix of similarities between examples. Even with these optimizations, solving the spectral clustering problem for very large datasets is expensive — partly due to the need for an eigen-decomposition of big matrices. These issues clearly intensify when operating with multiple views of the data, and if we seek to incorporate side advice such as tags. To address these limitations, there is significant recent interest in making spectral clustering efficient in the large dataset setting — with user interaction [8], 'activization' schemes [17], random projections [28], Spielman-Teng's near linear Laplacian solver [16], and parallelized versions of the Lanczos solver [7]. These solutions are highly effective for the standard spectral clustering objective and several also come with nice guarantees.

These advantages notwithstanding, the algorithms above can rarely be used in an off the shelf manner to address and exploit the specific needs and characteristics of the vision application above. First, few of these formulations support multiple views natively. Second, it is not straightforward to adapt the key optimization schemes to run in a distributed manner over tens of cores — this is essential if the system is expected to work efficiently on massive datasets and on distributed platforms such as CloudCV [2]. Finally, incorporating weak (or distant) supervision, user interaction and/or auxiliary domain specific priors beyond must-link/cannot-link constraints is challenging. Such side information is ubiquitous in most real world datasets and seems like a desirable feature for a system deployed in practice.

Motivated by these core issues, our primary **goal** is to develop stochastic methods, with a focus on spectral clustering for both single and multi-view data, that satisfy three criteria: **a)** Allow scaling to very large datasets ($\sim$ 100M) in a distributed manner; **b)** Show provably good convergence behavior; **c)** Offer the ability to incorporate high level priors (e.g., images share 'tags', user interaction) as a regularization term. The **contribution** of this paper is to provide an optimization scheme that meets these theoretical and practical considerations.

Related Work. The preceding section covered several relevant results on scaling spectral clustering to large datasets. Therefore, here we review related work on multi-view models. To our knowledge, among the earliest methods for multi-view clustering is a paper by Bickel & Scheffer [4] where the authors emulate co-training for clustering with two views (e.g., the webpage and in-coming hyperlinks). More recently, [5] studied multi-view clustering for images. Using a kernel CCA over *two* views, they showed that the clustering of images is facilitated by the textual description that comes with the image data. Subsequently, [6] described a nice theoretical analysis of the scenario where we infer an underlying mixture model (i.e., mixing weights), given independently drawn samples from the mixture. The authors showed that the low-dimensional subspace spanned by the means of the component distributions can be identified when the views

are conditionally uncorrelated. Strategies based on Non-negative Matrix Factorization [25], Co-training [19], Linked Matrix Factorization [30] and Random Walks [36] have also been proposed. The multi-view problem was investigated for a spectral clustering objective by [20]. This paper, which is the most closely related to ours, uses alternating maximization to co-regularize the clustering across views by requiring that the hypotheses learned from different views of the set of examples agree with each other. Like [20], our proposed approach will also operate with multiple views of the data. However, for scalability reasons, we will not impose inter-view consistency or differentially weight the given views, though our model easily permits this extension. We will instead adopt a simpler objective that considers all views to be equally informative, but still remains competitive with the more sophisticated strategies above in experiments.

2 Multi-View Spectral Clustering Model

Assume we are given l views of a dataset, $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, consisting of examples to be grouped into p clusters. Classically, for a single view, spectral clustering achieves this task by finding the p minimum eigenvectors of a Laplacian matrix $L \in \mathbb{R}^{n \times n}$, which encodes an appropriate graph over the examples. Typically, the eigenvectors are found via iterative methods such as Lanczos and its variations [22] that allow for implementations that can exploit the underlying sparsity of L. Spectral clustering may be viewed as a minimization of the trace of $V^T L V$ over the set $S_{n,p} \subseteq \mathbb{R}^{n \times p}$ of orthonormal $n \times p$ matrices V. $S_{n,p}$ is known as the *Stiefel manifold*. At the optimum, the columns of V must span the same subspace as the eigenvectors of the p least eigenvalues of L. To extend this to the multiview case, where we have a Laplacian denoted as $L^{(u)}$ for the u^{th} view of the data, one possibility is to penalize the Frobenius norm of variations between the V-representations of each view-pair [20]. Here, the number of additional terms grows quadratically with the number of views. Instead, the 'centroid' based formalization in [20] enforces the view-specific eigenvectors to be similar by requiring that they lie close to a common centroid. Similar to this intuition, we look for a common V that balances the solution over all the views. This translates into the following model,

$$\min_{V \in \mathbb{R}^{n \times p}} h(V) := \sum_u \operatorname{tr}(V^T L^{(u)} V) \quad \text{s.t.} \quad V^T V = I \tag{1}$$

The simple formulation above offers important scalability benefits. However, it has the limitation of uniformly weighting the views — which is mathematically equivalent to summing up the view-wise Laplacians. This raises two issues: (a) Experimentally, is the unweighted sum of features much worse than multi-view methods that impose consistency across V's for each pair of views? (b) If not, is it attractive to pre-compute the combined Laplacian and then run a single view spectral clustering on it? We will present results in Section 5 to show that the objective in (1) is empirically competitive with multi-view approaches

in [20] (note that analogously, feature concatenation remains a powerful baseline for MKL methods [14]). Regarding the second issue, rather than sum up the Laplacians beforehand, we will work with the views separately. Besides being a natural point at which to decompose the objective for stochastic gradient descent (as described later), dividing the Laplacians between separate computational nodes has useful performance advantages. For instance, in a distributed optimization setting, one does not need to copy these matrices (that exceed 50GB for large datasets) across each participating core. The entries of these matrices can also be lazily computed, with nearest neighbor lookups performed lazily to save unneeded work.

Incorporating Group Priors. Separate from Laplacians, there is typically a great deal of side information available suggesting (with varying degrees of confidence) that certain subsets of examples are likely to belong to the same class. Must-link constraints are tedious to deploy via user supervision for a large set of examples — instead, one may impose this prior indirectly. For example, if a set of images share five or more tags and the data source is somewhat reputable, it yields valuable group level advice to regularize (1) and complements the information extracted from the image.

Assume that we have a group prior information about examples where a group is defined as $C = \{v_1, v_2, \cdots, v_{|C|}\}$ where each v_j is a row of V corresponding to an example. To encode similarity in how their respective representations in V behave, we have a group concentration term, which measures the distance of each example (in the group) to the group's center $\bar{v}$:

$$g_C(V) = \sqrt{\frac{1}{|C|} \sum_{t=1}^{|C|} d(v_t, \bar{v})^2}, \tag{2}$$

where $\bar{v} = \frac{1}{|C|} \sum_{t=1}^{|C|} v_t$ and $d(\cdot, \cdot)$ is a suitable distance function. This regularization essentially measures intra-group distances, and we obtain a multi-view spectral clustering problem with a group prior:

$$\min_{V \in \mathbb{R}^{n \times p}} \quad f(V) := h(V) + g(V) \quad \text{s.t.} \quad V^T V = I, \tag{3}$$

where $g(V) = \lambda \sum_{\forall C} g_C(V)$ is a convex real-valued function that is the sum of the concentration terms for all groups (e.g. tags) in the dataset, with hyperparameter weight λ. We should point out that (2) is merely a simple example to make the following presentation concrete. Our subsequent analysis of this problem allows for non-smooth g, and group norms such as $\ell_{2,1}$ and others may be used based on specific needs. Some priors can be subsumed into the Laplacians whereas others can not, we make no assumptions on this. We denote the overall objective by $f(V)$.

3 Stochastic Gradient Descent Procedure

Our optimization scheme seeks to distribute the problem in such a way that at any given step, one only needs to consider a subset of the examples. This is done

using the method of *stochastic gradient descent* [9]. It is easy to see that the objective for $h(V)$ can be expressed as

$$\sum_u \sum_{ij} L_{ij}^{(u)} \langle V_{i\cdot}, V_{j\cdot} \rangle = \sum_u \sum_{i \sim j} w_{ij}^{(u)} \|V_{i\cdot} - V_{j\cdot}\|_2^2$$

where the inner sum is over non-zero entries of the Laplacian matrix for the u^{th} view in the first instance and edges of the corresponding graph with weights $w_{ij}^{(u)}$ in the second expression. Each term of the sum can be considered a sub-function of the objective, and so we can descend along the gradient of a randomly selected subset of the terms. Depending on the sampling strategy (discussed shortly), we want the resulting descent direction, *in expectation*, to be equivalent to an ordinary gradient descent on the full objective. Later, this will provide convergence guarantees.

At iteration t, one only obtains $\hat{L}_t$ that is a sample of L satisfying $\mathbb{E}(\hat{L}_t) = L$. To keep the notations and the presentation simple, we write our results and sampling strategy in the context of a single Laplacian L. For the following, we assume a simple procedure that uniformly selects edges from the graph, though our analysis also applies to additional sampling strategies discussed in Section 3.2, including the multi-view case. The stochastic gradient descent algorithm can then be applied to the entire objective of (3), resulting in the following update:

$$V_{t+1} = \mathcal{P}_\Omega(V_t - \gamma_t(2\hat{L}_t V_t + \partial g(V_t))), \tag{4}$$

where $\Omega := S_{n,p}$, γ_t the stepsize, and $\mathcal{P}_\Omega$ is a projection onto the feasible set.

Note that stochastic optimization on the Grassmannian and Stiefel manifolds has been considered in the context of GROUSE [1] and related work [34], and is not novel to this work specifically. In particular, [1] considers rank-one updates of the orthogonal solution matrix V on incomplete portions of the data.

3.1 Convergence of Stochastic Gradient

Generally, it is difficult to assess the convergence rate for non-convex optimization, but in our case the convergence can be obtained easily by properly choosing the stochastic gradient such that the objective decreases monotonically, for example, full gradient and (block) coordinate gradient. Therefore, based on the convergence "assumption," the following result shows that under some mild conditions, the *local* convergence rate is $O(1/\sqrt{T})$, where T is the number of iterations. We provide a brief outline of the proof in this section. Note that the convergence rate analysis is not only useful as a performance measure but helps provide the optimal sampling strategy for our optimization method and also shows how the framework will behave with parallelization across additional cores. To our knowledge, this is the first result of this kind for spectral clustering with regularization. Let $\Delta_t = \hat{L}_t - L$ where $\hat{L}_t$ is the sampled Laplacian at the t^{th} iteration and $L := \sum_u L^{(u)}$. We first define:

$$\sigma^2 := \max_{V^T V = I, t} \mathbb{E}(\|\Delta_t V\|_F^2); \quad M := \max_{V^T V = I} \|LV\|_F; \quad N := \max_{V^T V = I} \|\partial g(V)\|_F.$$

Notice that M and N are constants decided by L and g respectively, while σ^2 directly depends on the sampling strategy. For convenience, we define a function Υ as $\Upsilon(M, N, \sigma^2, T) \coloneqq \sqrt{((M+N)^2+\sigma^2)/T}$. Our convergence result states:

Theorem 1. *Let V^* be a convergent point of the sequence $\{V_t\}$ generated from (4). Suppose $\{V_t\}$ is contained in a small ball with radius $\delta > 0$. Denote $f(V^*)$ as f^*, and let ϕ be a positive value. If $\mathcal{P}_\Omega\left(V_t - \gamma_t(\hat{L}_t V_t + \partial g(V_t))\right)$ is a nonexpansive projection on this ball, we have:*
***i)** If the stepsize is chosen as $\gamma_t = \frac{\phi\delta}{\sqrt{((M+N)^2+\sigma^2)T}}$ and $\bar{V}_T = (\sum_{t=1}^T \gamma_t)^{-1} \sum_{t=1}^T \gamma_t V_t$, then $\mathbb{E}\left(f(\bar{V}_T)\right) - f^* \leq (\phi + \phi^{-1})\frac{\delta}{2}\Upsilon$.*
***ii)** If the step size is chosen as $\gamma_t = \theta_t \frac{f(V_t)-f^*}{(M+N)^2+\sigma^2}$, then $\mathbb{E}(f(\tilde{V}_T)) - f^* \leq \frac{\delta}{\sqrt{\theta_{\min}}}\Upsilon$ where $\tilde{V}_T = \frac{1}{T}\sum_{t=1}^T V_t$, $\theta_t \in (0,2)$ and $\theta_{\min} = \min_t 1 - (\theta_t - 1)^2$.*

From Theorem 1, it is clear that independent of how the stepsize is chosen, the local convergence rate is essentially bounded by $\Upsilon \in O(1/\sqrt{T})$. Theorem 1 is proved in the extended version of this paper. Next, we further investigate the behavior of σ^2, and introduce sampling strategies based on nodes and edges.

Similar convergence can also be achieved by the partial stochastic gradient projection method, that is,

$$V_{t+1} = \mathcal{P}_\Omega(V_t - \gamma_t \partial_{[t]} f(V_t)) \tag{5}$$

where $\partial f(V_t) \coloneqq LV_t + \partial g(V_t)$ is the subgradient of $f(V)$ at V_t and $\partial_{[t]} f(V)$ is a vector with the same size as $\partial f(V)$ taking the same values on the set $[t]$ and setting the rest as 0. More details are provided in the supplemental material.

3.2 Sampling

To meet the requirement of stochastic gradient, the randomly generated $\hat{L}_t$ should satisfy $\mathbb{E}(\hat{L}_t) = L$. The following discusses a sampling strategy that only uniformly samples the nonzero elements in L. Note that nonzero elements in L correspond to edges in the graph. Define $\bar{L}_{ij} \in \mathbb{R}^{n\times n}$ to be an extended matrix with L_{ij} at the ij^{th} element and zeros at the rest. We generate the stochastic gradient at the current iteration as $\hat{L}_t = \frac{\|L\|_0}{|\mathcal{E}|}\sum_{ij\in\mathcal{E}} \bar{L}_{ij}$ where $\mathcal{E}$ is the set of randomly selected edges.

In order to simplify the following discussion, we assume that the sampling strategy chooses a fixed number of edges at each iteration. These assumptions imply that every nonzero element (edge) in L has equal probability to be chosen. Let $\lambda_i(\Delta_t^T\Delta_t)$ denote the i^{th} largest eigenvalue of $\Delta_t^T\Delta_t$. From the definition of σ^2, we have

$$\sigma^2 = \mathbb{E}\left(\max_{V^TV=I} \|\Delta_t V\|_F^2\right) = \mathbb{E}\left(\sum_{i=1}^p \lambda_i(\Delta_t^T\Delta_t)\right) \leq \mathbb{E}(\|\Delta_t\|_F^2) = \sum_{ij\in\mathcal{E}} \mathbb{E}((\Delta_t)_{ij}^2) \tag{6}$$

To estimate the upper bound of $\mathbb{E}((\Delta_t)_{ij}^2)$, we consider the sampling strategy without replacement (the replacement case can be handled similarly).

$$(\Delta_t)_{ij} = \begin{cases} \left(\frac{\|L\|_0}{|\mathcal{E}|} - 1\right) L_{ij} & \text{w. p. } \frac{|\mathcal{E}|}{\|L\|_0} \\ -L_{ij} & \text{w. p. } 1 - \frac{|\mathcal{E}|}{\|L\|_0} \end{cases} \tag{7}$$

One can easily verify that $\mathbb{E}((\Delta_t)_{ij}^2) = \left(\frac{\|L\|_0}{|\mathcal{E}|} - 1\right) L_{ij}^2$. Thus, from (6) we have $\sigma^2 \leq \left(\frac{\|L\|_0}{|\mathcal{E}|} - 1\right) \|L\|_F^2$. When the cardinality of L is large, $\|L\|_0 \gg |\mathcal{E}|$. In other words, σ^2 dominates the convergence rate. Further, M^2 is bounded by $\|L\|_F^2$, which indicates that the convergence rate is bounded by

$$\begin{aligned} & O\left(T^{-1/2}\sqrt{(\|L\|_0/|\mathcal{E}| - 1)\,\|L\|_F^2 + (M+N)^2}\right) \\ & \leq O\left(T^{-1/2}\left(\sqrt{\|L\|_0/|\mathcal{E}|}\|L\|_F + N + \sqrt{N\|L\|_F}\right)\right). \end{aligned}$$

When $\sqrt{\frac{\|L\|_0}{|\mathcal{E}|}}$ is large, the bound is dominated by

$$O\left(\sqrt{\|L\|_0/(T|\mathcal{E}|)}\|L\|_F\right) = O((T|\mathcal{E}|)^{-1/2})$$

Note that the size of $\mathcal{E}$ is proportional to the number of threads. It means that the convergence can be speeded up linearly by increasing the number of threads (on different cores or slave computers) — exactly the behavior one hopes to achieve in the ideal situation. In addition, this linear speedup property is also achieved by the *partial* stochastic gradient projection method.

This edge sampling strategy can be easily extended to the setting where multiple separable views live in a distributed environment. The basic change here is that one needs to sample edges across all views, satisfying the condition $\mathbb{E}(\hat{L}_t) = \sum_u L^{(u)}$. This condition is true for a sampling strategy that first chooses a single u with probability proportional to the number of edges in $L^{(u)}$ from which edges are sampled identically to the single-view case. Similar linear speedup properties can be obtained; the result above carries through with essentially mechanical changes. Besides sampling edges, one may sample nodes (nodes correspond to the coordinates of V) to generate the stochastic gradient. However, when sampling nodes, the probability of sampling a given node must be weighted by its degree in order to achieve the same consistency conditions.

Projection vs Manifold Optimization. In order to realize the full benefits of parallelizing the optimization across multiple threads, we propose the manifold optimization method of Section 4. This does not satisfy the conditions of a non-expansive projection P_Ω in Theorem 1. Rather, it has the properties of a block coordinate descent method and does not leave the feasible region. While the manifold optimization has weaker convergence guarantees, it avoids the *projection* step, which requires synchronization between the parallel threads. See Fig. 1 side-by-side pseudocode showing the distinction.

Require: $f : \mathbb{R}^{n\times p} \to \mathbb{R}$, $V_0 \in S_{n,p}$
  for $t = 1, \ldots, T$ **do**
    Pick some u
    Sample $\hat{L}_t$ from $L^{(u)}$'s (see Section 3.2)
    Get subgradient $d \in 2\hat{L}_t V_t + \partial g(V_t)$
    Pick step size γ_t
    Take step in $\mathbb{R}^{n\times p}$: $V'_{t+1} \leftarrow V_t - \gamma_t d$
    Project onto feasible set:
      $V_{t+1} \leftarrow \mathcal{P}_{S_{n,p}}(V'_{t+1})$
  end for

Require: $f : S_{n,p} \to \mathbb{R}$, $V_0 \in S_{n,p}$
  for $t = 1, \ldots, T$ **do**
    Select $\mathcal{K} \subseteq \{1, \ldots, n\}$
    Take *descent curve* $Y(\tau)$ in $S_{n,p}$ s.t.
      $Y(0) = V_t$
      $\frac{d(f\circ Y)}{d\tau}\Big|_{\tau=0} \leq 0$
      $(Y(\tau))_{ij} = (V_t)_{ij} \quad \forall \tau, i \notin \mathcal{K}$
    Pick step size τ_t
    $V_{t+1} \leftarrow Y(\tau_t)$
  end for

Fig. 1. Comparison on projection (discussed in Section 3) and projection-free manifold (see Section 4) algorithms for solving optimization problems over the Stiefel manifold $S_{n,p}$. When done in parallel, multiple processors may perform independent iterations on different choices of $\hat{L}_t$ and $\mathcal{K}$.

4 Projection-Free Manifold Optimization Procedure

Ideally, we want to be able to split up the problem into subsets of examples, while *also* producing iterates that satisfy the constraints $V^T V = I$. Say we have a subset $\mathcal{K}$ of k row indices, corresponding to rows of V (the submatrix corresponding to these rows is denoted by $V_{\mathcal{K}\cdot} \in \mathbb{R}^{k\times p}$). We are given a feasible iterate V, and seek to compute the next iterate W such that it *also* lies in the Stiefel manifold $S_{n,p}$ and is thus feasible for the problem in (3), *and* W only differs from V in the rows selected by $\mathcal{K}$. This means that any number of parallel computational units, asynchronously modifying mutually disjoint subsets of the rows of V, will still produce feasible iterates.

Taking an optimization problem over only the rows in $\mathcal{K}$, we will show this produces a subproblem that seeks a rotation of the linearly independent columns of $V_{\mathcal{K}\cdot}$. W.l.o.g., assume $V_{\mathcal{K}\cdot} = [V_{\mathcal{K}\mathcal{I}}, V_{\mathcal{K}\mathcal{I}}R]$, where $V_{\mathcal{K}\mathcal{I}} \in \mathbb{R}^{k\times|\mathcal{I}|}$ is a maximal subset of linearly independent columns of $V_{\mathcal{K}\cdot}$ and $R \in \mathbb{R}^{|\mathcal{I}|\times(p-|\mathcal{I}|)}$ is the linear mapping from $V_{\mathcal{K}\mathcal{I}}$ to the dependent columns. Let $P = {V_{\mathcal{K}\mathcal{I}}}^T V_{\mathcal{K}\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ be the matrix of inner products of these columns. We know by construction that $P \succ 0$. Taking any orthonormal $U \in S_{k,|\mathcal{I}|}$,

$$W(U) = \begin{bmatrix} UP^{1/2} & UP^{1/2}R \\ V_{\bar{\mathcal{K}},\mathcal{I}} & V_{\bar{\mathcal{K}},\bar{\mathcal{I}}} \end{bmatrix} \in \mathbb{R}^{n\times p} \tag{8}$$

assuming w.l.o.g. above that $\mathcal{K}$ selects the first $|\mathcal{K}|$ rows of the matrix. This is constructed such that those rows of V in the *complement* of $\mathcal{K}$ (denoted by $\bar{\mathcal{K}}$) are unchanged in W and the constraints are preserved:

$$\begin{aligned} {W_{\mathcal{K}\cdot}}^T W_{\mathcal{K}\cdot} &= \begin{bmatrix} P^{1/2}U^T U P^{1/2} & P^{1/2}U^T U P^{1/2}R \\ R^T P^{1/2}U^T U P^{1/2} & R^T P^{1/2}U^T U P^{1/2}R \end{bmatrix} \\ &= \begin{bmatrix} P & PR \\ R^T P & R^T PR \end{bmatrix} = V_{\mathcal{K}\cdot}^T V_{\mathcal{K}\cdot} \end{aligned}$$

so that

$$W^T W = W_{\mathcal{K}\cdot}^T W_{\mathcal{K}\cdot} + W_{\bar{\mathcal{K}}\cdot}^T W_{\bar{\mathcal{K}}\cdot} = V_{\mathcal{K}\cdot}^T V_{\mathcal{K}\cdot} + V_{\bar{\mathcal{K}}\cdot}^T V_{\bar{\mathcal{K}}\cdot} = V^T V.$$

Therefore, if V lies on the Stiefel manifold $S_{n,p}$, so must W.

The above construction successfully reduces the problem of finding a feasible iterate W to modifying a subset of the rows given an appropriately constructed matrix $U \in S_{k,|\mathcal{I}|}$. The specific choice of U is determined by moving along a curve in the smaller Stiefel manifold. The starting point is given as $U_0 = V_{\mathcal{K}\mathcal{I}} P^{-1/2}$, for which $W(U_0) = V$. To generate a *geodesic* [12] that serves as a descent curve, we can project a subgradient of $f \circ W$ onto the tangent space of the manifold $S_{k,|\mathcal{I}|}$ at U_0, and take the manifold exponential map. An analogous procedure generates curves from the Cayley transformation [33], which can be calculated more cheaply. This curve on $S_{k,|\mathcal{I}|}$ can be mapped by W to a curve on Stiefel manifold $S_{n,p}$ in the original problem. This construction produces a descent curve meeting the conditions in Fig. 1.

If we perform a line search over this descent curve such that the objective function is monotonically nonincreasing, the convergence of this algorithm is apparent. However, there is no guarantee to converge to the global solution because of the non-convexity of problem (1).

5 Experiments

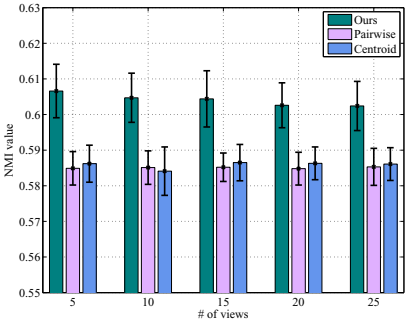
We have performed a number of experiments to evaluate our methods on several aspects: (a) performance w.r.t. to a variety of datasets with special emphasis on scalability as a function of size (b) comparison with state of the art method [20] when appropriate (c) performance when incorporating high level priors into the model. Though our focus is to show that the method is applicable for multi-view spectral clustering (with convex regularization) for larger computer vision datasets, for which few alternatives are available, we also performed some experiments on machine learning datasets as a sanity check, where we match reported results. Our vision datasets cover Caltech 101, Caltech 256, LabelMe and TinyImages. For experiments with very large datasets, we also used simulated datasets with on the order of hundreds of millions of examples. As a performance comparison measure we report on Normalized Mutual Information (NMI).

5.1 Multi-view ML Datasets

UCI Digits: The first dataset we use is the handwritten digits (0-9) data from the UCI repository. The dataset consists of 2000 examples, with six sets of features for each image from which we construct six views. These results are summarized in Figure 2b. Since our method depends on initialization, we repeat the experiments 10 times (different initializations) and report on the best NMI value obtained and standard deviations. The authors of [20] provide an initialization in their code using eigenvectors of individual views.

Reuters Multilingual: In addition, we consider multiview spectral clustering on a natural language dataset. We subsample the dataset in a manner consistent with [20]. Since the features for this dataset are sparse and high-dimensional, we first use Latent Semantic Analysis (LSA) [15] to reduce the dimensionality.

5.2 Multi-view Vision Datasets



(a)

	Digits	Reuters
Ours	0.798(0.03)	0.312(0.01)
[20] Pairwise	0.659	0.305
[20] Centroid	0.669	0.308
Best 1-view	0.641	0.288

(b)

Fig. 2. (a) Caltech101, showing the NMI values for different choices of views for ours and [20]. (b) Comparison on UCI Digits and Reuters, with mean (and s.d.) NMI.

Caltech101: We evaluated the method on Caltech101, a popular benchmark for object categorization with 102 categories of images (101 distinct objects and background), and 30 images per category. To generate the views, we use the UCSD-MKL dataset — a collection of kernels derived from various visual features (up to 25) for Caltech101 data. We used only the training class kernels in an unsupervised setting. Kernels for 5 random splits, with each split containing information regarding 1515 images (15 images for each of the 101 categories) is provided. We report also results randomly selecting subsets of the views. In each case, we report our summaries as well as [20] by averaging across all 5 splits. The results in Figure 2a suggest that the method compares well to [20].

Caltech256: A similar but bigger dataset is Caltech256, which contains 256 object classes and more than 30000 images across all classes, We restrict our evaluations to three main features for each image: V1-like [29], SURF [3] and Region Covariance (RegCov) [32], for generating views for this data. The Spectral Hashing method in [18] was then adapted to construct the graph for the Laplacians. Note that here we cannot perform comparisons with [20] since their method requires a dense kernel construction. Because of the nature of this dataset, the V1-like view alone yields a NMI of 0.267, SURF gives 0.207, whereas RegCov performs poorly at 0.088. Contrary to the results from other datasets above, here, the multi-view performance at 0.181 is close to but worse than the best view (with two views, SURF and V1-like, multi-view NMI is 0.22). There are two primary reasons. First, the views do not seem to be uncorrelated and the

Fig. 3. Example results from LabelMe. The first row corresponds to a certain cluster from our multi-view method *without* tag prior. This cluster is best matched with the 'opencountry' category in the ground truth, but includes a subset of images that were false positives (red box) and false negatives (green box) for this particular cluster. Introducing group prior on a *separate* set of images (not shown) "propagates" and helps correctly put both the red and green blocks in the correct class. Second row (left) shows the new images that were introduced into the 'opencountry' category, as a result.

best view, V1-like, seems to dominate the others. Since there are only a few feature types, we cannot expect an improvement over the single best view. Despite these issues, the evaluations suggest that solving multi-view spectral clustering for these sizes *is* feasible, if the features are assumed to be provided.

ImageNet: We can construct a dataset with similar properties to the above following a similar procedure in [24]. We use ILSVRC 2013 [10], an updated version of the challenge set that is the basis of the dataset in [24]. ImageNet categories consist of Wordnet noun synsets, which precisely defines the object in the image. From ILSVRC 2013 [10] we select 100 categories at random, with a total of 127885 images selected. We use four views derived from Decaf [11], GIST [27], TinyImage [31], and SIFT [26] features. Each view considered separately produces NMIs of 0.198, 0.181, 0.181, and 0.184 respectively. The multiview objective combining all four produces a labeling with an NMI of 0.203.

5.3 Incorporating Group Structure

LabelMe: To evaluate the group prior effect, we used the LabelMe data [27], which includes eight outdoor scene categories: coast, forest, highways, inside city, mountain, open country, street and tall buildings. There are 2688 color images and each category contains at least 260 images. We employ three views of visual features: Gist [27], Spatial Pyramid Matching (SPM) [21], and Object Bank (OB) [23]. The group prior information comes from the text tags available in LabelMe annotations. We ask users to study the text tags and pick 19 'major' tags out of 781. With each tag, we build 19 groups, each of which is a set of images that share a single tag (like beach, tree trunks, car). We note these groups are only about 70–90% correct with respect to the ground truth. For example, 'insidecity' images and 'street' images both include the "building" tag. Our prior regularization term $g(V)$ is the sum of Frobenius norm of 19 groups. These groups covered only about 1500 of $\sim$ 2700 images.

To evaluate how such a prior incrementally improves performance, we add subsampling schemes at levels $\{0\%, 10\%, 60\%, 100\%\}$, where 100% means we use all 1500 images that have tags and 0% is standard multi-view spectral clustering. For all of our experiments with the prior, we set $\lambda = \frac{10^2 \|L\|_1}{\|L\|_0}$. Representative examples are shown in Figure 3 and demonstrate how priors on some images may in fact help correctly classify a subset of images that do *not* have this auxiliary data available. The NMI values for GIST [27], SPM [21] and OB [23] in a single view setting were 0.448, 0.419, and 0.511 respectively. The no-prior model improves the NMI to 0.561. Tag priors at the 10%, 60%, and 100% (i.e., 1500 images) incrementally improve NMI from 0.561 to 0.613, 0.633 and finally 0.679, suggesting their utility in this setting.

5.4 Jumbo-Sized Datasets

We summarize our main experiments on very large datasets here. Note that there are significant implementation issues (e.g., memory management, data structures, queries) in successfully running a system on tens of millions of examples.

TinyImages: TinyImages [31] is a set of nearly eighty million 32×32 color images collected from internet searches. The dataset is distributed along with GIST features computed on each image, which were used as the basis of our clustering. Nearest neighbors were computed using [18], from which a weighted graph with 320 million edges was constructed. The dataset includes a keyword associated with each image, for 24690 images the dataset authors evaluated the accuracy of this keyword out of which 5660 images depicted the associated keyword. This keyword is the only form of label provided with the TinyImage dataset, no ground truth is available.

We split the entire TinyImages dataset into two clusters using spectral clustering with the manifold optimization method. With 34 CPU cores, the optimization averaged one iteration every 0.015 seconds. To qualitatively evaluate the clustering at a local scale, we look at how individual keywords are split between the clusters. While most keywords are split by this clustering, some keywords corresponding to more homogeneous sets of images are well separated into one cluster or the other. In Figure 4, we show a subset of the keywords sampled from both well-clustered and poorly clustered images.

ImageNet: We can also test a clustering task on the full ILSVRC2013 dataset. This full dataset has 1000 categories and 1281165 images. Since our optimization procedure considers a high-n low-p regime, we find a two-way split as in the TinyImages. The (non-normalized) MI of the two-way labelling versus the ground truth is 0.229.

Mixture Model: To assess the scalability of our optimization scheme, independent of issues related to generating a diverse set of feature descriptors and side information on a large vision dataset, we performed experiments to evaluate if we can reliably process a Gaussian Mixture Model. We ran the model on mixtures

comprising of 10^6 and 10^8 examples distributed concurrently across (up to 36) heterogeneous CPU cores. For $|\mathcal{K}| = 1024$, this setup computed iterations at a rate of one iteration every 0.016 seconds and 0.034 seconds respectively. Within 50000 iterations, the 10^6 case reaches an objective value of 0.054 with an NMI of 0.769 against the true label of which Gaussian distribution from which each point is sampled. On the 10^8 case an NMI of 0.683 was seen with the objective reduced to 2.685.

5.5 Model Characteristics

Varying $|\mathcal{K}|$ and Number of Iterations: The size of $\mathcal{K}$, the number of examples chosen in sampling in each iteration, is a key parameter in our approach. To show how this choice impacts the performance of the model, we use 5 kernels chosen from the Caltech101 experiments as our views. The kernels and the initialization are kept fixed in different runs, whereas $|\mathcal{K}|$ and the number of iterations are varied from 100 to 1000, and the objective is shown in Figure 5. The objective is progressively lower for increasing values of $|\mathcal{K}|$ and the iterations converge sooner with increasing values, as it approaches the full gradient. The

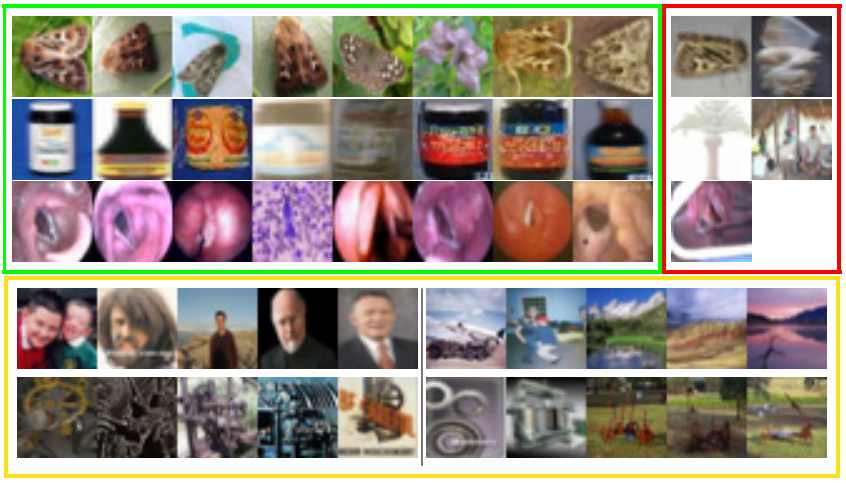


Fig. 4. Results of our method applied to the TinyImages dataset, looking at how five selected keywords (top to bottom: `antler_moth`, `cassareep`, `true_vocal_cord`, `john`, and `machinery`) are split by the clustering. To the left of the divide is a sampling of images for which spectral clustering produces a "dominant" label for this keyword, and the rightmost columns are given the "wrong" label. Green and red boxes mark these groups for the three keywords shown for which we achieved a good separation of the clusters. The keywords in the yellow box do not have an informative cluster.

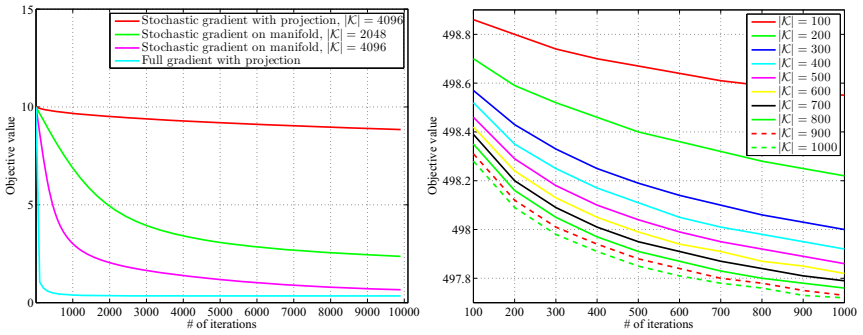


Fig. 5. (a) Plot showing the convergence rate of ordinary gradient descent with projection onto the Stiefel manifold, stochastic gradient descent with projection, and stochastic gradient descent using manifold optimization. (b) Plot showing the variation in objective with increasing iterations and different values of $|\mathcal{K}|$ from 100 to 1000. The objective value is drawn against iterations, demonstrating that quicker convergence is achieved with larger samples at the expense of increasing per-iteration cost.

rate of change in the objective as a function of iterations is similar for $|\mathcal{K}| \geq 300$, which suggests that a relatively small value should suffice.

Comparison of Projection and Manifold Optimization Techniques: We compare the manifold optimization of Section 4 and the method using projection on synthetic data. These use a single normalized Laplacian of a random graph over $n = 10^5$ points in 4 clusters. All three methods are applied to solve (1). As we can see from Figure 5, ordinary gradient descent converges in the fewest iterations due to using the entire matrix and $O(n^2)$ computations in *each* iteration. The manifold optimization method converges faster than projection in part because of heuristics used in selecting the step size. Further, the convergence rate increases when the sample size is increased. The Lanczos method (i.e., MATLAB's `eigs`) fails due to excessive memory requirements ($>$ 32GB).

6 Conclusion

We describe a scalable stochastic optimization approach for Multi-view spectral clustering with a convex regularizer. A useful feature of this approach is that at any given step, the gradient is computed only for a subset of the examples — the direct consequence is that with an increase in the number of examples, the optimization can still make progress without having to compute the full gradient at each step. We provide a detailed analysis of its convergence properties, which sheds light on how adding a large number of processors in a distributed environment will affect its performance. Finally, we discuss how high-level priors can be easily leveraged within this framework. The highly scalable implementation accompanying this paper is particularly useful in applications where would want

to effectively leverage such meta knowledge within inference, which remains difficult in alternatives based on Nyström extension. Our empirical evaluations on several ML, vision, and synthetic datasets suggest that the model is scalable and efficient, and matches the performance of other existing multi-view spectral clustering models.

Acknowledgments. We thank anonymous ECCV reviewers for extensive and helpful comments. This research is funded by grants NIH R01 AG040396, NSF CAREER award 1252725, NSF RI 1116584, NSF CGV 1219016, NSF DMS-0914524, NSF DMS-1216318, and ONR Award N00014-13-1-0129. Collins was supported by a CIBM fellowship (NLM 5T15LM007359).

References

1. Balzano, L., Nowak, R., Recht, B.: Online identification and tracking of subspaces from highly incomplete information. In: Proceedings of the Allerton Conference on Communication, Control and Computing (2010)
2. Batra, D., Agrawal, H., Banik, P., Chavali, N., Alfadda, A.: CloudCV: Large-scale distributed computer vision as a cloud service (2013), `http://www.cloudcv.org`
3. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
4. Bickel, S., Scheffer, T.: Multi-view clustering. In: Proceedings of the IEEE International Conference on Data Mining (2004)
5. Blaschko, M.B., Lampert, C.H.: Correlational spectral clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)
6. Chaudhuri, K., Kakade, S.M., Livescu, K., Sridharan, K.: Multi-view clustering via canonical correlation analysis. In: Proceedings of the International Conference on Machine Learning (2009)
7. Chen, W., Song, Y., Bai, H., Lin, C., Chang, E.Y.: Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(3), 568–586 (2011)
8. Chen, X., Cai, D.: Large scale spectral clustering with landmark-based representation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2011)
9. Darken, C., Moody, J.: Towards faster stochastic gradient search. In: Advances in Neural Information Processing Systems (1993)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2009)
11. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. ArXiv preprint ArXiv:1310.1531 (2013)
12. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications 20(2), 303–353 (1998)
13. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the Nyström method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), 214–225 (2004)

14. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: Proceedings of the IEEE International Conference on Computer Vision (2009)
15. Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (1999)
16. Khoa, N.L.D., Chawla, S.: Large scale spectral clustering using resistance distance and Spielman-Teng solvers. In: Ganascia, J.-G., Lenca, P., Petit, J.-M. (eds.) DS 2012. LNCS, vol. 7569, pp. 7–21. Springer, Heidelberg (2012)
17. Krishnamurthy, A., Balakrishnan, S., Xu, M., Singh, A.: Efficient active algorithms for hierarchical clustering. In: Proceedings of the International Conference on Machine Learning (2012)
18. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: Proceedings of the IEEE International Conference on Computer Vision (2009)
19. Kumar, A., Daumé III, H.: A co-training approach for multi-view spectral clustering. In: Proceedings of the International Conference on Machine Learning (2011)
20. Kumar, A., Rai, P., Daumé III, H.: Co-regularized multi-view spectral clustering. In: Advances in Neural Information Processing Systems (2011)
21. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2006)
22. Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK users' guide: Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods, vol. 6 (1998)
23. Li, L., Su, H., Xing, E.P., Fei-Fei, L.: Object bank: A high-level image representation for scene classification & semantic feature sparsification. In: Advances in Neural Information Processing Systems (2010)
24. Li, M., Lian, X.C., Kwok, J., Lu, B.L.: Time and space efficient spectral clustering via column sampling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2011)
25. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization. In: SIAM International Conference on Data Mining (2013)
26. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
27. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision 42(3), 145–175 (2001)
28. Sakai, T., Imiya, A.: Fast spectral clustering with random projection and sampling. In: Machine Learning and Data Mining in Pattern Recognition (2009)
29. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2005)
30. Tang, W., Lu, Z., Dhillon, I.S.: Clustering with multiple graphs. In: Proceedings of the IEEE International Conference on Data Mining (2009)
31. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(11), 1958–1970 (2008)
32. Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)
33. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Mathematical Programming, 1–38 (2012)

34. Xu, J., Ithapu, V.K., Mukherjee, L., Rehg, J.M., Singh, V.: GOSUS: Grassmannian Online Subspace Updates with Structured-sparsity. In: Proceedings of the IEEE International Conference on Computer Vision (2013)
35. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems (2004)
36. Zhou, D., Burges, C.J.C.: Spectral clustering and transductive learning with multiple views. In: Proceedings of the International Conference on Machine Learning (2007)