Spectral Clustering with a Convex Regularizer on Millions of Images

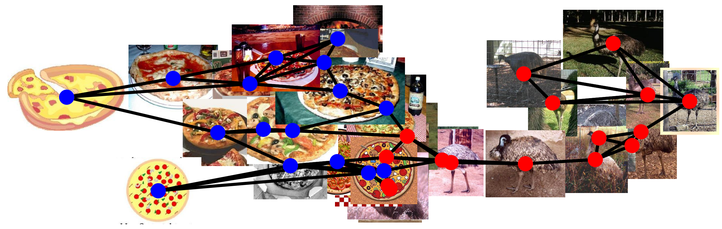
Maxwell D. Collins[1] Ji Liu[2] Jia Xu[1] Lopamudra Mukherjee[3] Vikas Singh[1]

[1]University of Wisconsin-Madison [2]University of Rochester [3]University of Wisconsin-Whitewater

http://pages.cs.wisc.edu/~mcollins/pubs/eccv2014.html

Spectral Image Clustering

Problem: Given a sparse pairwise similarity graph over images, find assignment that best groups similar images together.



Priors and Regularization

Can be used to express priors (must-link, tags, transductive learning). Incorporates *side information* and extrinsic properties of the clustering problem.

Analysis is general w.r.t. many possible regularization functions g.
g any convex, possibly non-smooth, function.
One Example: Given some known groups C of similar images:

$$g(V) := \sum_{\text{groups C}} \sqrt{\frac{1}{|C|}\sum_{t\in C} d(v_t, \bar{v}_C)^2}$$

Optimization Model

Clustering is posed as an optimization problem.

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

$L^{(u)}$ is Laplacian for graph from "view" u.

Rows of the optimal V are *quantized* to produce clusters.

Optimization on the Stiefel Manifold

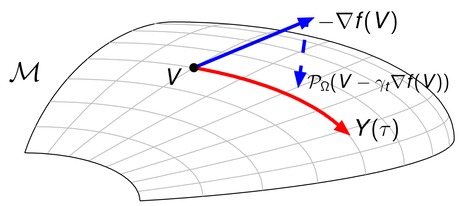
Stiefel Manifold: Manifold of orthonormal matrices.

$$S_{n,p} = \{V \in \mathbb{R}^{n\times p} \mid V^T V = I_p\}$$

For any skew-symmetric $W \in \mathbb{R}^{n\times n}$ and

$$Y(\tau) = \left(I + \frac{\tau}{2}W\right)^{-1}\left(I - \frac{\tau}{2}W\right)V$$

then $Y(\tau)^T Y(\tau) = V^T V$ for all τ.



Parallelization

Contribution: Techniques to parallelize Stiefel manifold optimization, focusing on large-scale image clustering.

1. Stochastic gradient/coordinate descent. (Used in both algorithms below)
2. Parallelizable feasible methods. (Algorithm 2)

Stochastic Gradient

Any uniformly sampled subset of the terms in

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

will in expectation be equal to the gradient.

Equivalent to sampling matrix $\hat{L}_t$ s.t. $\mathbb{E}(\hat{L}_t) = L$.

Simplest way to use this is projected stochastic gradient:

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

Convergence Theorem: Let V^* be a convergent point of the sequence $\{V_t\}$. Suppose $\{V_t\}$ is contained in a small ball with radius $\delta > 0$. Denote $f(V^*)$ as f^*. If $\mathcal{P}_\Omega$ is a nonexpansive projection on this ball, we have **upper bounds on the expected suboptimality w.r.t. the convergent point**.
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 $\mathbb{E}\left(f(\bar{V}_T)\right) - f^* \le (\phi + \phi^{-1})\frac{\delta}{2}\Upsilon$.
ii) If the step size is chosen as $\gamma_t = \theta_t \frac{f(V_t)-f^*}{(M+N)^2+\sigma^2}$, then $\mathbb{E}(f(\tilde{V}_T)) - f^* \le \frac{\delta}{\sqrt{\theta_{\min}}}\Upsilon$ where $\tilde{V}_T = \frac{1}{T}\sum_{t=1}^T V_t$, $\theta_t \in (0,2)$ and $\theta_{\min} = \min_t 1 - (\theta_t - 1)^2$.

Descent Curves in Parallel

Reduce the problem to optimizing over some subset of the rows or V index by $\mathcal{K} \subset \{1, ..., n\}$. Computational units have disjoint choices of $\mathcal{K}$.

$V_{\mathcal{K}\mathcal{I}}$:= Maximal subset of linearly independent colums of submatrix $V_{\mathcal{K}\cdot}$.
$V_{\mathcal{K}\cdot} = [V_{\mathcal{K}\mathcal{I}}, V_{\mathcal{K}\mathcal{I}}R]$ $\qquad P := V_{\mathcal{K}\mathcal{I}}^T V_{\mathcal{K}\mathcal{I}}$
If $V \in S_{n,p}$ and $U \in S_{|\mathcal{K}|,|\mathcal{I}|}$, then

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

Two Algorithms

Projected Stochastic Gradient (#1)

Require: $f : \mathbb{R}^{n\times p} \to \mathbb{R}$, $V_0 \in S_{n,p}$
  for $t = 1, ..., T$ **do**
    Pick some u
    Sample $\hat{L}_t$ from $L^{(u)}$'s
    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

Projection-free (#2)

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

Multiple iterations run in parallel, with sampling of $\hat{L}_t$ and $\mathcal{K}$ to avoid conflicts. Projection step requires synchronization.

Experimental Setup

Evaluating: **(a)** Performance, with special emphasis on scalability as a function of size. **(b)** Accuracy comparison with other multi-view spectral clustering. **(c)** Accuracy as a function of the use of priors/regularization.

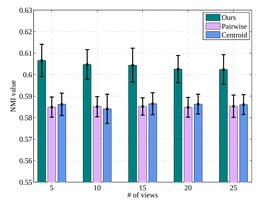
Large-scale Datasets		Features/Views
LabelMe	~2,700 images	Gist, SPM, Object Bank
Caltech101	~9,000	See UCSD-MKL
Caltech256	~30,000	V1-like, SURF, RegCov
ILSVRC 2013 (subset)	~130,000	Decaf, Gist, TinyImage, SIFT
(full)	~1,300,000	" "
TinyImages	~80,000,000	Gist
Artificial data (GMM)	up to 10^8	N/A

Naïve methods unsuitable for spectral clustering Caltech256 and larger.

Accuracy

Normalized Mutual Information (NMI) vs ground-truth was comparable to other multi-view spectral clustering models (Kumar et. al. 2011).

Caltech101



ML Datasets

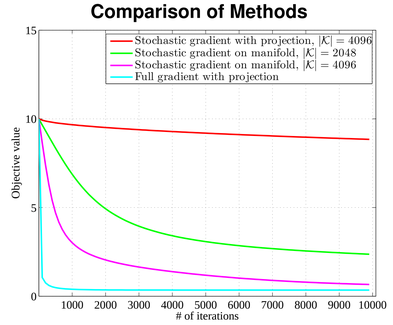
	Digits	Reuters
Ours	0.798(0.03)	0.312(0.01)
Pairwise	0.659	0.305
Centroid	0.669	0.308
Best 1-view	0.641	0.288

NMI on artificial GMM: 0.769 for 10^6 points, 0.683 for 10^8 points.

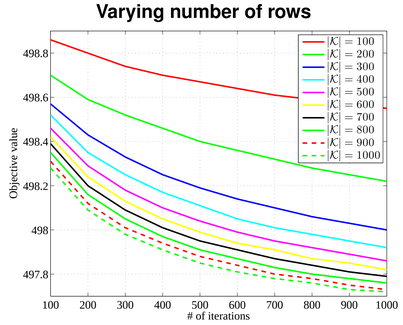
Using priors from tags on LabelMe increases NMI from 0.561 for no prior to 0.679 when using all tags.

Convergence and Performance

Comparison of Methods



Varying number of rows



Feasible methods (with a line search) show improved convergence versus projected stochastic coordinate descent.

Iterations needed for convergence depends on the number of rows used. Trade-off between number of iterations and computational cost of each.

Classical subspace iteration methods (e.g. Arnoldi's algorithm in ARPACK and MATLAB's `eigs`) limited by memory usage. >32GB used for $n = 10^5$.

European Conference on Computer Vision (ECCV)

NSF RI 1116584, NSF CAREER award 1252725, NIH R01 AG040396, NSF CGV 1219016, and others.
Collins was supported by a CIBM fellowship (NLM 5T15LM007359).