Joint Sparsity-based Representation and Analysis of Unconstrained Activities - Supplementary Material

Raghuraman Gopalan
Video and Multimedia Technologies Research Dept., AT&T Labs-Research, Middletown, NJ 07748 USA
raghuram@research.att.com

1. Recovering Jointly Sparse Atoms

In this section we provide a brief summary of algorithmic details from [1] for the three joint sparse models (JSM-1 to 3) that we used to recover atoms C_i and I_i constituting the intermediate representation $J(V_i)$ of the video V_i.

1.1. JSM-1

In this case both the common component and innovative components are sparse. So we perform ℓ_1 minimization (since ℓ_0 optimization is NP-hard) to the signal ensemble to recover C_i and I_i under sparsity constraints. More details are available from Section 5.1 in [1].

1.2. JSM-2

In this case there is no common component but the innovations have shared supports. We perform recovery using the DCS-SOMP (Distributed Compressive Sensing - Simultaneous Orthogonal Matching Pursuit) algorithm from Section 5.2 in [1]. It is a greedy iterative variant of the Orthogonal Matching Pursuit (OMP) that estimates one innovative element at each step.

1.3. JSM-3

We used Alternating Common and Innovation Estimation (ACIE) from Section 5.3 of [1] for the case where common component is not sparse while innovations are sparse. One central aspect of this algorithm is to decouple the effect of estimated basis for innovations from the task of estimating the common component.

2. Karcher Mean and Related Computations on the Grassmannian

In Algo 1, 2 and 3 we present algorithmic computations on the Grassmann manifold $\mathcal{G}_{n,d}$ that were utilized in Sec 2.3.2 (Event clustering) and Sec 2.3.1 (Event classification, intrinsic) in the main paper. The extrinsic method for performing event classification using Grassmann kernel discriminant analysis (Sec 2.3.1) is given in Algorithm 4.

1. Given a set of N points $\{S_i\}$ on the manifold.
2. Let $\bar{\mu}_0$ be an initial estimate of the Karcher mean, usually obtained by picking one element of $\{S_i\}$ at random. Set j = 0.
3. For each $i = 1, .., k$, compute the inverse exponential map (Algorithm 2) ν_i of S_i about the current estimate of the mean, i.e. $\nu_i = \exp^{-1}_{\bar{\mu}_j}(S_i)$.
4. Compute the average tangent vector $\bar{\nu} = \frac{1}{k}\sum_{i=1}^{k} \nu_i$.
5. If $\|\bar{\nu}\|$ is small, then stop. Else, move $\bar{\mu}_j$ in the average tangent direction using $\bar{\mu}_{j+1} = \exp_{\bar{\mu}_j}(\epsilon\bar{\nu})$, where $\epsilon > 0$ is small step size, typically 0.5, and $\exp_{\bar{\mu}_j}$ is the exponential map (Algorithm 3) at $\bar{\mu}_j$.
6. Set $j = j + 1$ and return to Step 3. Continue till $\bar{\mu}_j$ does not change anymore or till maximum iterations are exceeded.

Algorithm 1: Algorithm to compute the sample Karcher mean [2].

Given two points S_1 and S_2 on the Grassmannian $\mathbb{G}_{n,d}$.

- Compute the $n \times n$ orthogonal completion Q of S_1.
- Compute the thin CS decomposition of $Q^T S_2$ given by $Q^T S_2 = \begin{pmatrix} X_C \\ Y_C \end{pmatrix} = \begin{pmatrix} V_1 & 0 \\ 0 & \tilde{V}_2 \end{pmatrix} \begin{pmatrix} \Gamma(1) \\ -\Sigma(1) \end{pmatrix} V^T$
- Compute $\{\theta_i\}$ which are given by the arccos and arcsine of the diagonal elements of Γ and Σ respectively, i.e. $\gamma_i = \cos(\theta_i)$, $\sigma_i = \sin(\theta_i)$. Form the diagonal matrix Θ with θ's as diagonal elements.
- Compute $A = \tilde{V}_2 \Theta V_1^T$.

Algorithm 2: Numerical computation of the velocity matrix: The inverse exponential map [3].

References

[1] D. Baron, M. Duarte, M. Wakin, S. Sarvotham, and R. Baraniuk. Distributed compressive sensing. *arXiv preprint arXiv:0901.3403*, 2009. 1

[2] Y. Chikuse. *Statistics on special manifolds*. Springer Verlag, 2003. 1

- Given a point on the Grassmann manifold S_1 and a tangent vector $B = \begin{pmatrix} 0 & A^T \\ -A & 0 \end{pmatrix}$.
- Compute the $n \times n$ orthogonal completion Q of S_1.
- Compute the compact SVD of the direction matrix $A = \tilde{V}_2 \Theta V_1$.
- Compute the diagonal matrices $\Gamma(t')$ and $\Sigma(t')$ such that $\gamma_i(t') = \cos(t'\theta_i)$ and $\sigma_i(t') = \sin(t'\theta_i)$, where θ's are the diagonal elements of Θ.
- Compute $\Psi(t') = Q \begin{pmatrix} V_1 \Gamma(t') \\ -\tilde{V}_2 \Sigma(t') \end{pmatrix}$, for various values of $t' \in [0, 1]$.

Algorithm 3: Algorithm for computing the exponential map, and sampling along the geodesic [3].

Given intermediate representations $J(V_i)$'s corresponding to training videos with one of m activity labels, orthonormalize their columns to obtain $\bar{J}(V_i)$'s. Similarly obtain $\bar{J}(\tilde{V}_i)$'s from unlabeled test videos $\tilde{V}_i$'s.
Training:

- Compute the matrix $[K_{train}]_{ij} = k_P(\bar{J}(V_i), \bar{J}(V_j))$ for all $\bar{J}(V_i), \bar{J}(V_j)$ in the training set, where k_P is the projection kernel defined earlier.
- Solve $\max_\gamma L(\gamma)$ by eigen-decomposition (1), with $K^\star = K_{train}$.
- Compute the $(m$-1)-dimensional coefficients, $F_{train} = \gamma^T K_{train}$

Testing:

- Compute the matrix $[K_{test}]_{ij} = k_P(\bar{J}(V_i), \bar{J}(\tilde{V}_j))$ for all $\bar{J}(V_i)$ in training, and $\bar{J}(\tilde{V}_j)$ in testing.
- Compute $(m$-1)-dimensional coefficients, $F_{test} = \gamma^T K_{test}$ by solving for (1) with $K^\star = K_{test}$.
- Perform 1-nearest neighbor classification (f_2) from the Euclidean distance between F_{train} and F_{test}.

The Rayleigh quotient $L(\gamma)$ is given by,

$$L(\gamma) = \max_\gamma \frac{\gamma^T K^\star (\bar{Y} - 1_B 1_B^T / B) K^\star \gamma}{\gamma^T (K^\star (I_B - \bar{Y}) K^\star + \sigma^2 I_B) \gamma} \quad (1)$$

where $K^\star$ is the Gram matrix (K_{train} or K_{test}), 1_B is a uniform vector $[1...1]^T$ of length B corresponding to the number of training videos, $\bar{Y}$ is the block-diagonal matrix whose y^{th} block ($y = 1$ *to* m) is the uniform matrix $1_{B_y} 1_{B_y}^T / B_y$, B_y is the number of training videos in y^{th} activity class, and $\sigma^2 I_B$ is a regularizer to make computations stable.

Algorithm 4: Grassmann Kernel Discriminant Analysis [4].

[3] K. Gallivan, A. Srivastava, X. Liu, and P. Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *Workshop on Statistical Signal Processing*, pages 315–318, Feb 2003. 1, 2

[4] J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pages 376–383. ACM, 2008. 2