Orbit Learning using Convex Optimization

Tony Jebara
Columbia University
New York, NY 10027, USA
jebara@cs.columbia.edu

Yoshua Bengio
Universite de Montreal
Montreal, H3C 3J7, Canada
bengioy@iro.umontreal.ca

Recently, learning approaches have been brought to bear on nonlinear datasets by assuming samples lie on a low-dimensional Riemannian manifold in the embedding space. One solution is to model local variations linearly [1]. A fundamental difficulty with such solutions is they cannot extrapolate or generalize far from the training data when the manifold is curved. Consequently, the manifold needs to be densely sampled; particularly at high curvature regions. To deal with this serious problem, we consider manifolds that emerge from group actions on the data as in [2]: the manifold is actually an orbit of unknown transformations. For instance, images of a 3D object from varying viewpoints can be seen as the result of multiple nonlinear operators that act upon a small set of prototype views. Although this may seem like a non convex optimization problem, we propose a formulation that is convex and produces global estimates of the group generators while faithfully approximating data with a low-dimensional orbit. This approach allows us to generalize to new parts of the space and to extrapolate the manifolds nonlinearly while still keeping a compact and regularized description of the data.

Consider a dataset of vectors $\vec{x}_1 \dots \vec{x}_T$ on a low dimensional orbit manifold constructed by group actions. Here, actions mean a matrix acting on a data vector $x_{t'}$ to map it to another x_t via the exponentiated matrix product $\vec{x}_t \approx \exp(A_{t,t'})\vec{x}_{t'}$. We may consider mappings $n = 1 \dots N$ (where $N = T^2$) between all pairs of points $t = 1 \dots T$ and $t' = 1 \dots T$. Or, we may consider a subset of the T^2 mappings, i.e. choosing a single prototype by locking $t' = 1$. Another choice is mapping points to their k nearest neighbors. Ultimately, we seek $A_{t,t'}$ matrices that faithfully reconstruct the data by minimizing the sum of reconstruction errors $E_{t,t'}(A_{t,t'}) \equiv E_n(A_n)$ for any choice of N such mappings. This is like a regression where input data has to regenerate itself as output.

In addition to low reconstruction error, an important additional constraint on the transformation matrices A_n is that they themselves form a low dimensional subspace. This means only a few axes of freedom are present from the group actions or orbits. For instance, each A_n matrix could be a linear combination of a few matrices as $A_n = \sum_{j=1}^{J} c_{n,j} V_j$. If we had the optimal A_n matrices, we could recover the V_j matrices by vectorizing the A_n matrices and performing principal components analysis. Denote the vectorized A_n matrices as $\vec{a}_n = \text{vec}(A_n)$. Also, denote the joint parameters of all A_n matrices as a. To obtain a small subspace with few eigenmatrices, we require that the $\vec{a}_n$ vectors occupy a small volume. We use the determinant of the covariance of $\vec{a}_n$ as a coarse estimator of volume. Therefore, in addition to reconstruction error, we minimize the determinant of the covariance: $|Cov(a)| = \left| N^{-1} \sum_n \vec{a}_n \vec{a}_n^T - N^{-2} \sum_{n,m} \vec{a}_n \vec{a}_m^T \right|$. This determinant cost is not quite convex but, by regularizing the covariance and penalizing its trace, we obtain the convexified cost $C(a) = \log |Cov(a) + \epsilon_1 I| + \epsilon_2 \text{tr}(Cov(a))$. Details are deferred for a later paper [3]. In addition to having A_n matrices that live in a low-dimensional subspace, we want them to also reconstruct the dataset. We could minimize the squared error reconstruction for each mapping via $\|\vec{x}_t - \exp(A_{t,t'})\vec{x}_{t'}\|^2$ yet summing these terms with $C(a)$ creates a non convex cost. Instead, we minimize slightly different terms which *are* convex $E_n(a) = \log \text{tr}\left(\exp(-A_{t,t'})\vec{x}_t \vec{x}_t^T + \exp(A_{t,t'})\vec{x}_{t'} \vec{x}_{t'}^T\right)$. For non-negative data, this error's minimum coincides with minima of squared error. Adding these surrogate error terms gives our final convex cost function $C(a) = \log |Cov(a) + \epsilon_1 I| + \epsilon_2 \text{tr}(Cov(a)) + \lambda \sum_n E_n(a)$. While minimizing the orbit's dimensionality, we minimize reconstruction error weighted by λ (like a Lagrange multiplier enforcing an amount of tolerable reconstruction error). After finding the $A_1 \dots A_N$ matrices that minimize cost, we perform PCA on them to obtain coefficients c_{nj} and eigenmatrices V_j spanning the

orbit compactly in J dimensions (where $J < D$, the original dataset's dimensionality). More importantly, we can extrapolate nonlinearly away from the manifold (i.e. complete a spherical manifold by only seeing a small piece of it).

Given our final convex cost, we implemented an update rule via variational upper bounds. Consider the following tangential upper bound on log-determinants $\log|S| \leq \mathrm{tr}(\hat{S}^{-1}S) + \log|\hat{S}| - \mathrm{tr}(\hat{S}^{-1}\hat{S})$ which achieves equality at $S = \hat{S}$. Applying this to our cost and ignoring constant terms gives

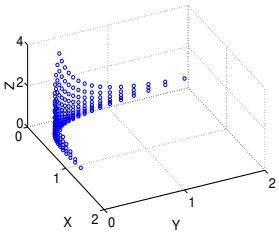
$$C(a) \quad \leq \quad N^{-1}\sum_n \vec{a}_n^T \tilde{M} \vec{a}_n^T - N^{-2}\sum_{n,m} \vec{a}_n^T \tilde{M} \vec{a}_m + \lambda \sum_n E_n(a).$$

where we define $\tilde{M} = (Cov(a) + \epsilon_1 I)^{-1} + \epsilon_2 I$ as our regularized inverse covariance. Similarly, we upper bound reconstruction error terms $E_n(a)$ with a quadratic that makes tangential contact at the old value of each mapping matrix $\tilde{A}_n$ or $\tilde{a}_n$ as follows (this holds for symmetric A_n and almost holds otherwise):

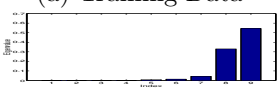
$$E_n(a) \quad \leq \quad \frac{1}{2}\|\vec{a}_n - \tilde{a}_n\|^2 + (\vec{a}_n - \tilde{a}_n)^T \vec{d}_n + \log \mathrm{tr}\left(\exp(-\tilde{A}_n)\vec{x}_t\vec{x}_t^T + \exp(\tilde{A}_n)\vec{x}_{t'}\vec{x}_{t'}^T\right)$$

$$\text{where} \quad \vec{d}_n \quad = \quad \mathrm{vec}\left(\frac{\exp(\frac{1}{2}\tilde{A}_n)\vec{x}_{t'}\vec{x}_{t'}^T\exp(\frac{1}{2}\tilde{A}_n) - \exp(-\frac{1}{2}\tilde{A}_n)\vec{x}_t\vec{x}_t^T\exp(-\frac{1}{2}\tilde{A}_n)}{\mathrm{tr}\left(\exp(-\tilde{A}_n)\vec{x}_t\vec{x}_t^T + \exp(\tilde{A}_n)\vec{x}_{t'}\vec{x}_{t'}^T\right)}\right).$$

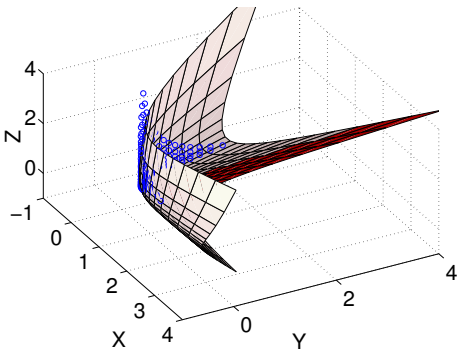
We obtain the update rule: $\vec{a}_n \leftarrow \left((2N^{-1} - 2N^{-2})\tilde{M} + \lambda I\right)^{-1}\left(\lambda\tilde{a}_n - \lambda\vec{d}_n + N^{-2}\sum_{m\neq n}\tilde{M}\vec{a}_m\right)$ by taking derivatives of the fully quadratic upper bound. This update is interleaved with periodic updates of $\tilde{M}$ and monotonically reduces the original $C(a)$ cost until its minimum. We can sparsify the A_n matrices by zeroing out some entries or enforce some custom linear constraints on them, i.e. $\vec{a}_n^T\vec{q}_{in} \geq c_{in}\forall i, n$ by iterating *quadratic programming* instead of minimizing the quadratic bound analytically. We tested our method on ≈ 200 wedged-shaped 3D samples in Figure (a). Matrices were found that reconstruct each sample from a single arbitrary prototype point. Cost decreased monotonically converging in under 100 epochs. Figure (b) shows the resulting eigenvalues of the covariance of the matrices. We computed the top 2 eigenmatrices and explored linear combinations of them to reconstruct the orbit (Figures (c) and (d)). The orbit fits the data and extrapolates it nonlinearly while keeping it strictly positive.



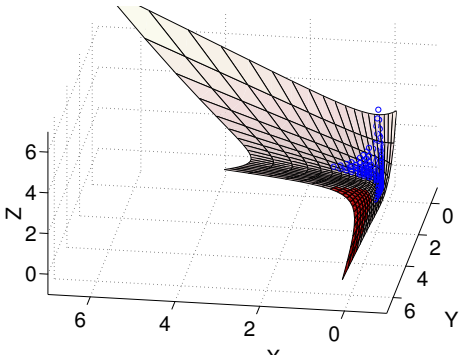
(a) Training Data



(b) Eigenvalues



(c) Learned Orbit ($\epsilon_1 = \epsilon_2 = \frac{1}{2}, \lambda = 2$)



(d) Learned Orbit from Another View.

We are investigating more ambitious datasets (i.e. images, etc.) and considering using the above in regression problems where A_n matrices map inputs x_t to their outputs y_t (instead of to other inputs).

References

[1] S. Roweis and L. Saul, Science, 290(5500):2323-2326, December 2000.
[2] R. Rao and D. Ruderman, Neural Information Processing Systems 11, pp. 810-816, NIPS, 1999.
[3] T. Jebara, Workshop on Artifical Intelligence & Statistics, AISTAT, 2003.