Lecture 18 Oct. 30 -2006

Data Visualization

STAT 442 / 890, CM 462

Lecture: Ali Ghodsi

1 Action Respecting Embedding

Action respecting embedding [1]takes a sequence of high-dimensional data $x_1, \ldots, x_t$, along with associated discrete actions $a_1, \ldots, a_{t-1}$. The data are assumed to be in some order, where action a_i was taken between data points x_i and x_{i+1}. The final piece of input is a similarity function, $||x_i - x_j||$, defining a distance over the high-dimensional data points. For vector data, Euclidean distance is often sufficient, but other data-specific similarities can be employed.

The overall structure of the algorithm follows the same three steps of SDE :

(i) Construct a neighbourhood graph.

(ii) Solve a semidefinite program to find the maximum variance embedding subject to constraints.

(iii) Extract a low-dimensional embedding from the dominant eigenvectors of the learned kernel matrix.

ARE, though, seeks to exploit the additional information provided by the action labels

of the data. This information is mainly exploited by adding action-respecting constraints into the semidefinite program.

1.1 Action Respecting Constraints

The most important, contribution of ARE is the addition of action respecting constraints. The evaluation of learned manifolds is often subjective and usually amounts to demonstrating that the manifold corresponds to the known data generator's own underlying degrees of freedom. Action labels, even with no interpretation or implied meaning, provide more information about the underlying generation of the data. It is natural to expect that the actions correspond to some simple operator on the generator's own degrees of freedom. For example, a camera that is being panned left and then right, has actions that correspond to a simple translation in the camera's actuator space. We therefore want to constrain the learned representation so that the labeled actions correspond to simple transformations in that space. In particular, we can require all actions to be a simple rotation plus translation in the resulting low-dimensional representation.[1]

We can formalize this constraint by first observing rotation plus translation is exactly the space of *distance preserving* transformations. A transformation $\mathcal{T}$ is distance preserving and thus a rotation plus translation if and only if:

$$\forall x, x' \quad ||\mathcal{T}(x) - \mathcal{T}(x')|| = ||x - x'||.$$

Let us consider this in the context of an action-labeled data sequence. All actions must

[1]These are the subset of linear transformations that don't involve any scaling component.

be distance preserving transformations in the learned representation. Therefore, for any two data points, x_i and x_j, the same action taken at those data points must preserve their distance. Letting $\Phi(x_i)$ denote data point x_i in the learned space, We require that action a's transformation, $\mathcal{T}_a$, must satisfy:

$$\begin{aligned} \forall i, j \qquad & ||\mathcal{T}_a(\Phi(x_i)) - \mathcal{T}_a(\Phi(x_j))|| = \\ & ||\Phi(x_i) - \Phi(x_j)||. \end{aligned} \tag{1}$$

Now, if we let $a = a_i$ and consider the case where $a_j = a_i$. Then, $\mathcal{T}_a(\Phi(x_i)) = \Phi(x_{i+1})$ and $\mathcal{T}_a(\Phi(x_j)) = \Phi(x_{j+1})$, and Constraint (1) becomes:

$$||\Phi(x_{i+1}) - \Phi(x_{j+1})|| = ||\Phi(x_i) - \Phi(x_j)||. \tag{2}$$

We want to pose this not as a constraint on distances, but rather as a constraint on inner products, i.e., on the learned kernel matrix, K. We can square both sides of the equation and rewrite it in terms of K resulting in the following set of constraints:

$$\begin{aligned} \forall i, j \quad & a_i = a_j \Rightarrow \\ & K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)} = \\ & K_{ii} - 2K_{ij} + K_{jj} \end{aligned} \tag{3}$$

We can add Constraint (3) into SDE's usual constraints to arrive at the optimization and algorithm shown in Table 1. There is a slight modification to SDE's usual neighbour constraint, changing strict equality into an upper bound. This modification insures that the constraints are feasible by allowing the zero matrix to be a feasible solution. Notice that

the additional action respecting constraints are still linear in the optimization variables, K_{ij}, and so the optimization remains a semidefinite program. Since the neighbourhood graph η_{ij} is fully connected, the optimization is bounded, convex, and feasible, and therefore can be solved efficiently with various general-purpose toolboxes. The results in this paper were obtained using SeDuMi [2] in MATLAB. These results also used highly penalized slack variables in SDE's neighbourhood constraint to help improve the stability of the solution. This was recommended by Weinberger and colleagues in their original SDE work [3].

Algorithm: ARE

Construct neighbours, η, according to Equation (??).

Maximize $\mathrm{Tr}(K)$ **subject to** $K \succeq 0$: $\sum_{ij} K_{ij} = 0$,

$\forall ij \quad \eta_{ij} > 0 \vee [\eta^T \eta]_{ij} > 0 \Rightarrow$

$K_{ii} - 2K_{ij} + K_{jj} \leq ||x_i - x_j||^2$, **and**

$\forall ij \quad a_i = a_j \Rightarrow$

$K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)} =$

$K_{ii} - 2K_{ij} + K_{jj}$

Run Kernel PCA with learned kernel, K.

Table 1: ARE Algorithm.

References

[1] M. Bowling, A. Ghodsi, and D. Wilkinson. Action respecting embedding. In *International Conference on Machine Learning*, 2005.

[2] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999.

[3] K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 988–995, 2004.