Background on Martin distance

Jules Gottraux

November 2020

Classification using subspace angles

As the first classification algorithm, we will use a nearest neighbor classifier between the transition matrix of the models fitted from the training set and the model of the test video. This algorithm will, as distances, use the squared Martin and Frobenius distance on the subspace angles $\theta_1, \theta_2, .., \theta_{2n}$. Let models $\mathscr{M}_1$ and $\mathscr{M}_2$ be $\mathscr{M}_i = (A_i, C_i), i \in \{1, 2\}$, then if $\|A_i\|_2 < 1$ the subspace angles can directly be computed from the eigenvalues of matrix P^{ij}:

$$P^{ij} = P_{ii}^{-1} P_{ij} P_{jj}^{-1} P_{ji} \tag{1}$$

with P_{ij} the solution to:

$$P_{ij} = A_i^T P_{ij} A_j + C_i^T C_j \iff (A_i^T)^{-1} P_{ij} + P_{ij}(-A_j) = (A_i^T)^{-1} C_i^T C_j$$

Which is a Sylvester equation, i.e. $A_s X + X B_s = C_s$ of parameters:

$$A_s = (A_i^T)^{-1}$$
$$B_s = (-A_j)$$
$$C_s = (A_i^T)^{-1} C_i^T C_j$$

Once we have the matrices P^{ij} and P^{ji} with their corresponding eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$ we define:

$$\theta_k = \begin{cases} \cos^{-1}(\sqrt{\lambda_i^{ij}}), & \text{if } k \in \{1, ..., n\} \\ \cos^{-1}(\sqrt{\lambda_i^{ji}}), & \text{if } k \in \{n+1, ..., 2n\} \end{cases}$$

For the model definition $\mathscr{M}_i = (A_i, C_i), i \in 1, 2$, if model i does not uses a linear mapping but rather a neural network, we set the corresponding $C_i = I_{R \times P}$. Then from the subspace angles, the distances can be computed as follows:

$$d_M^2(\mathscr{M}_1, \mathscr{M}_2) = -\log \prod_{i=1}^{2n} \cos(\theta_i)^2$$

$$d_F^2(\mathscr{M}_1, \mathscr{M}_2) = 2 \sum_{i=1}^{2n} \sin(\theta_i)^2$$

We can also use a kernel SVM algorithm to classify the models from those distances. If we have S samples points (corresponding of models $\mathscr{M}_s$ fitted on video fragments) the associated kernel K_{RBF} can be defined as:

$$K_{RBF,ij} = e^{-\gamma d_X^2(\mathscr{M},\mathscr{M})}, \quad \gamma > 0, \quad X \in M, F$$

Enforcing small intra-cluster distance

When learning the transition matrices A_i, we want to cluster matrices from the same activity $g \in G$ together. Meaning that if we have two videos i and j from activity g_i and g_j we want $d_X^2(\mathscr{M},\mathscr{M})$ to be small if $g_i = g_j$ and high if $g_i \neq g_j$. We can enforce that be minimizing not only the prediction error but also this intra-cluster distance. So once we have obtained the points $\mathscr{M} = (A_i, C_i)$ for all video fragment we can fix the projection C_i and continue to learn A_i by minimizing the following loss. For each activity $g \in G$, with $i, j \in \{k :$ video k has activity $g\}$:

$$\mathscr{L} = \alpha \sum_i \|X_i^+ - A_i X_i^-\|_F^2 + (1-\alpha) \sum_{i \neq j} d_X^2(\mathscr{M}_i, \mathscr{M}_j) \tag{2}$$

Then at test time for a given video fragment, we fit the projection and the matrix using loss (**??**). Then, two approaches are possible, we stop here and classify as before using k-nearest neighbors or SVM. Or we can fix the projection C_t and continue learning the matrix A_t. To continue the learning of A_t, we will have $|G|$ models: $\mathscr{M}_t^g = (A_t^g, C_t)$. For each activity g we have training samples: $\mathscr{M}_i = (A_i, C_i), i \in \mathscr{I} = \{i :$ video i is in activity $g\}$ and we minimize:

$$\mathscr{L}_g = \alpha \|X_t^+ - A_t^g X_t^-\|_F^2 + \frac{(1-\alpha)}{|\mathscr{I}|} \sum_i d_X^2(\mathscr{M}_i, \mathscr{M}_t^g) \tag{3}$$

Then, the out of those $|G|$ models, we predict the gesture of the model that gives the "best" approximation.

Derivative of the Martin distance

Let $\mathscr{M} = (\boldsymbol{C}, \boldsymbol{A})$ and $\mathscr{M}_i = (\boldsymbol{C}_i, \boldsymbol{A}_i)$ be the parameters of two dynamical systems. We want to find the transition matrix $\boldsymbol{A}$ of model $\boldsymbol{A}$ that minimizes the Martin distance:

$$\min_{\boldsymbol{A}} \; d_M^2(\mathscr{M}, \mathscr{M}_i). \tag{4}$$

A natural strategy to minimize over $\boldsymbol{A}$ would be to use a gradient descent method. For that purpose, we need the derivative of the Martin distance w.r.t. matrix $\boldsymbol{A}$. Assuming that $\boldsymbol{C}$ has orthonormal columns (*i.e.,* $\boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{I}$), it can be shown that derivative w.r.t. the jk-th entry of $\boldsymbol{A}$ is given by:

$$\frac{\partial d_M^2(\cdot)}{\partial A_{jk}} = \operatorname{Tr}\left((\boldsymbol{X}^{-1})^T \frac{\partial \boldsymbol{X}}{\partial A_{jk}}\right) - 2\operatorname{Tr}\left((\boldsymbol{X}_i^{\dagger})^T \frac{\partial \boldsymbol{X}_i}{\partial A_{jk}}\right), \tag{5}$$

where $.^\dagger$ denote pseudo-inverse and where

$$\boldsymbol{X} = \boldsymbol{A}^T \boldsymbol{X} \boldsymbol{A} + \boldsymbol{I} \tag{6}$$

$$\boldsymbol{X}_i = \boldsymbol{A}_i^T \boldsymbol{X}_i \boldsymbol{A}_i + \boldsymbol{I} \tag{7}$$

$$\frac{\partial \boldsymbol{X}}{\partial A_{jk}} = \boldsymbol{A}^T \frac{\partial \boldsymbol{X}}{\partial A_{jk}} \boldsymbol{A} + \boldsymbol{J}_{jk}^T \boldsymbol{X} \boldsymbol{A} + \boldsymbol{A}^T \boldsymbol{X} \boldsymbol{J}_{jk} \tag{8}$$

$$\frac{\partial \boldsymbol{X}_i}{\partial A_{jk}} = \boldsymbol{A}^T \frac{\partial \boldsymbol{X}_i}{\partial A_{jk}} \boldsymbol{A}_i + \boldsymbol{J}_{jk}^T \boldsymbol{X}_i \boldsymbol{A}_i, \tag{9}$$

with $\boldsymbol{J}_{jk}$ being the single-entry matrix (*i.e.,* all zeros except 1 at the jk-th entry). Note that the above relations correspond to the solutions of discrete Lyapunov or Sylvester equations.

Note: for computation purposes, we can rewrite the derivative of martin:

$$\frac{\partial d_M^2(\cdot)}{\partial A_{jk}} = \sum_{j,k} (X^{-1} \circ \frac{\partial \boldsymbol{X}}{\partial A_{jk}}) - 2 \sum_{j,k} (X_i^\dagger \circ \frac{\partial \boldsymbol{X}_i}{\partial A_{jk}}) \tag{10}$$