Non-redundant Spectral Dimensionality Reduction Supplementary Material

Yochai Blau and Tomer Michaeli

Technion–Israel Institute of Technology, Haifa, Israel
{yochai@campus,tomer.m@ee}.technion.ac.il

1 Optimization Problem (4) for LEM and DFM

In the case of LEM and DFM, the objective $\boldsymbol{f}_i^T \boldsymbol{K} \boldsymbol{f}_i$ is optimized s.t. the constraints

$$\begin{aligned} &\boldsymbol{f}_i^T \boldsymbol{D} \boldsymbol{f}_j = \delta_{i,j} \ , \\ &\mathbf{1}^T \boldsymbol{D} \boldsymbol{f}_i = 0 \ , \end{aligned}$$

where $\boldsymbol{D}$ is a diagonal matrix with entries $[\boldsymbol{D}]_{i,i} = \sum_j [\boldsymbol{K}]_{i,j}$. Similarly to (5), these constraints can be interpreted as restrictions on the *weighted* means and *weighted* correlations between the projections. Namely,

$$\begin{aligned} &E[f_i(X)d(X)] = 0 \ , \\ &E[f_i(X)f_j(X)d(X)] = 0 \ , \end{aligned}$$

where $d(x) = \int k(x,y)dy$. Accordingly, we define non-redundancy in this case as zero weighted correlation between each projection and *any function* of the previous projections. That is, for each i, we would like to ensure that

$$\mathbb{E}[f_i(X) g(f_{i-1}(X), \cdots, f_1(X)) d(X)] = 0$$

for every function g (analogously to (8)). This is equivalent to the requirement that

$$\mathbb{E}[f_i(X)d(X) | f_{i-1}(X), \cdots, f_1(X)] = 0 \ ,$$

which we approximate at the data points by $\boldsymbol{P}_i \boldsymbol{D} \boldsymbol{f}_i = \mathbf{0}$.

By denoting $\hat{\boldsymbol{f}}_i = \boldsymbol{D}^{\frac{1}{2}} \boldsymbol{f}_i$, $\hat{\boldsymbol{K}} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{K} \boldsymbol{D}^{-\frac{1}{2}}$ and $\hat{\boldsymbol{P}}_i = \boldsymbol{P}_i \boldsymbol{D}^{\frac{1}{2}}$, the corresponding optimization problem becomes

$$\begin{aligned} \min_{\hat{\boldsymbol{f}}_i}/\max \quad & \hat{\boldsymbol{f}}_i^T \hat{\boldsymbol{K}} \hat{\boldsymbol{f}}_i \\ \text{s.t.} \quad & \hat{\boldsymbol{f}}_i^T \hat{\boldsymbol{f}}_i = 1 \\ & \hat{\boldsymbol{P}}_i \hat{\boldsymbol{f}}_i = \mathbf{0}, \quad \forall i > 1 \ , \end{aligned}$$

where min/max corresponds to LEM/DFM respectively. Therefore, to obtain a non-redundant version of these algorithms, we simply apply Algorithm 1 with $\hat{\boldsymbol{K}}$ and $\hat{\boldsymbol{P}}_i$ rather than with $\boldsymbol{K}$ and $\boldsymbol{P}_i$, and then multiply the extracted projections by $\boldsymbol{D}^{-\frac{1}{2}}$ from the left. Notice that to form the regression matrix $\boldsymbol{P}_i$, we use $\{\boldsymbol{f}_j\}$, and not $\{\hat{\boldsymbol{f}}_j\}$.

2 Quality of the approximations in Sect. 4

Equation (9) approximates Eq. (6) by restricting the conditional expectation only at the discrete set of N data points. First, notice that we are only interested in the projections at these N points, so that it is unnecessary to constrain the conditional expectation elsewhere. Furthermore, not only is it sufficient to restrict at these points, but we could actually restrict at far fewer points without suffering a significant degradation in performance. This is shown for the MNIST experiment (see Sect. 5.3) in Table 2 below, where the value of p indicates the ratio of the points at which the conditional expectation was constrained (the points were chosen randomly).

Table 2. MNIST experiment classification errors [%].

15K examples, all labeled							
# of proj.	LEM	ours					
		$p = 1$	$p = 0.9$	$p = 0.7$	$p = 0.5$	$p = 0.3$	$p = 0.1$
3	17.6	12.0	12.1	12.0	12.3	12.0	12.0
5	8.8	7.6	7.5	7.4	7.6	7.5	7.3
7	6.9	6.0	6.1	6.0	5.9	6.2	5.9
9	6.5	5.6	5.6	5.6	5.8	5.9	5.8
11	6.0	5.0	5.4	5.3	5.6	6.0	5.7

Equation (10) approximates Eq. (9) using the Nadaraya-Watson regressor, which converges as $N^{-\frac{4}{(i-1)+4}}$ when computing the ith projection. Notice that the regression is performed in the low-dimensional *projection* space and not the high-dimensional data space which leads to an accurate approximation when the number of projections is moderate.

3 Hyper-parameter Analysis

Three hyper-parameters need to be set when employing Algorithm 1: the kernel smoother bandwidth h, the threshold for truncating the singular values of $\boldsymbol{P}_i$, and the number of nearest neighbors (NN) computed in each row of the $\boldsymbol{P}_i$ matrices (see Sect. 4). Table 3 below shows the effect of these parameters on the classification errors in the MNIST experiment (see Sect. 5.3).

Table 3. MNIST experiment classification errors [%].

15K examples, all labeled											
# of proj.	α					SV_{th}			NNs		
	0.6	0.5	0.4	0.3	0.2	5%	3%	1%	$10K$	$7.5K$	$5K$
3	14.6	12.0	**11.6**	24.2	27.3	18.7	**12.0**	12.3	**12.0**	20	19.6
5	**7.2**	7.6	7.8	7.5	8.1	8.4	**7.6**	8.2	**7.6**	7.8	10.9
7	**5.8**	6.0	5.9	6.4	6.8	6.2	**6.0**	6.5	**6.0**	6.9	7.7
9	**5.6**	**5.6**	5.7	6.0	6.5	5.7	**5.6**	5.8	**5.6**	6.2	6.2
11	5.4	**5.0**	5.4	6.0	6.0	5.4	**5.0**	5.4	**5.0**	6.1	6.0

The kernel smoother bandwidth h is set adaptively for each projection by $h = \alpha(\sum_{j=1}^{i-1} \frac{1}{N}\|\boldsymbol{f}_j\|^2)^{1/2}$, where the parameter $\alpha \in [0.1, 0.6]$. Tuning α in this range has a mild impact on the classification errors and tuning is needed to achieve optimal results. This tuning can be done with a tune set for classification/ regression etc. and manually for visualization tasks.

The threshold for truncating the singular values of $\boldsymbol{P}_i$ is set as a percentage of the maximal singular value. A high threshold will result in a bad approximation of $\boldsymbol{P}_i$, while a low threshold will reduce the feasible set of solutions. Our analysis shows that 3% performs well in the MNIST experiment (3% worked well in the other experiments as well). Taking a slightly lower threshold has a minor effect, while taking a larger threshold degrades the classification results.

For optimal results, the number of NNs used for constructing $\boldsymbol{P}_i$ should be maximal (i.e. the number of training examples). However, as the number of training examples increases the memory resources may not be sufficient to store $\boldsymbol{P}_i$. The results show the degradation in classification error as the number of NNs decreases, indicating that it is always desired to use as many NNs as the memory resources allow.

4 Baseline comparisons for the artificial head image experiment (Sect. 5.1)

We compare our results in the artificial head images experiment with two baseline methods: PCA and ICA. As seen in Fig. 9, both methods fail to provide a representation which faithfully reveals the two underlying parameters controlling the dataset: the horizontal and vertical angles. The first *and* second projections in both methods are correlated with the horizontal angle, while the vertical angle is not captured. This affects the error in reconstructing the images from the two-dimensional embeddings: the mean PSNR is 18.3/17.6 with PCA/ICA (the PSNR with our non-redundant LEM/LTSA was 19.2/19.9), clearly showing that these baseline methods are inferior in this task.

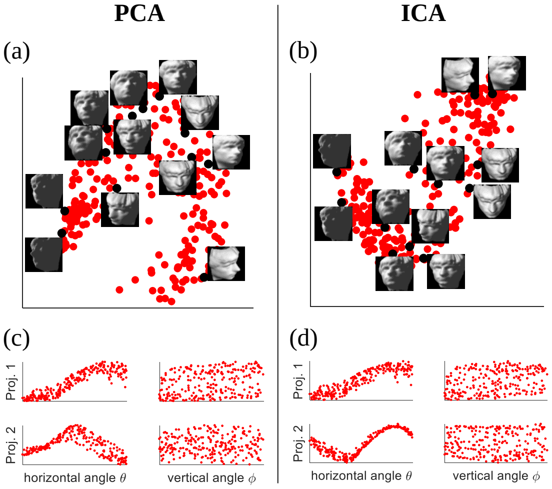

Fig. 9. The two-dimensional embeddings of the artificial head images obtained with (a) PCA, and (b) ICA. (c),(d) The first two projections of the head images vs. the horizontal and vertical angles (θ, ϕ) of the heads. The two projections extracted by these baseline methods are *both* correlated only with the horizontal angle θ.