A NOVEL METHOD FOR SUFFICIENT DIMENSION REDUCTION ON GAUSSIAN MIXTURE MODELS

Daniele Bracale, Felipe Maia Polo, Kevin Wibisono

University of Michigan, Ahn Arbor

1. INTRODUCTION

In this project, we introduce a novel method for sufficient dimension reduction (SDR) (Suzuki and Sugiyama, 2010, Ghojogh et al., 2021), in which we model the joint density of the features and labels $(\mathbf{X}, Y)$ as a Gaussian mixture model (GMM) and learn the sufficient statistic via mutual information maximization on the Stiefel manifold. Due to the intractability of obtaining a closed-form solution, we employ some constrained optimization algorithms and evaluate their performance on both real and artificial data. We also compare our method with other existing SDR methods. Our code is available in the Python package `sdr`.

2. SDR: PROBLEM SETTING

Let $(\mathbf{X}, Y)$, with range in $\mathbb{R}^q \times \mathbb{R}$, represent a pair of features and labels, where both $\mathbf{X}$ and Y are assumed to be continuous random vectors or variables. The goal of sufficient dimension reduction (SDR) is to find a proper transformation $T : \mathbb{R}^q \to \mathbb{R}^p$, $p < q$, such that $T(\mathbf{X})$ is a sufficient statistic for Y. That is, we want T such that

$$Y \perp\!\!\!\perp \mathbf{X} | T(\mathbf{X}).$$

In practice, it is common to constrain T to be a linear transformation represented by a matrix member of the Stiefel manifold

$$\mathbb{S}_p^q := \left\{ \mathrm{A} \in \mathbb{R}^{p\times q} : \mathrm{A}\mathrm{A}^\top = \mathrm{I}_p \right\},$$

i.e. we consider the case where $T(\mathbf{X}) = \mathrm{A}\mathbf{X}, \mathrm{A} \in \mathbb{S}_p^q$. In order to proceed, we state Proposition 1 (Cover and Thomas, 2012; Chapter 2):

Proposition 1. *Assuming $\mathrm{A} \in \mathbb{R}^{p\times q}$, then $Y \perp\!\!\!\perp \mathbf{X}|\mathrm{A}\mathbf{X}$ holds if and only if $I(Y, \mathbf{X}) = I(Y, \mathrm{A}\mathbf{X})$, where I denotes the mutual information:*

$$I(Y, \mathbf{X}) = \int p_{\mathbf{X},Y}(\mathbf{x}, y) \log \left(\frac{p_{\mathbf{X},Y}(\mathbf{x}, y)}{p_{\mathbf{X}}(\mathbf{x}) p_Y(y)} \right) d\mathbf{x} dy.$$

In general, $I(Y, \mathbf{X}) \geq I(Y, \mathrm{A}\mathbf{X})$.

Special thanks to the cats of Instagram, who have helped us stay sane during graduate school

According to Proposition 1, if there exists an $\mathrm{A}^* \in \mathbb{S}_p^q$ such that $Y \perp\!\!\!\perp \mathbf{X}|\mathrm{A}^*\mathbf{X}$, then

$$\mathrm{A}^* \in \arg\min_{A \in \mathbb{S}_p^q} f(\mathrm{A}), \tag{1}$$

where $f(\mathrm{A}) := -I(Y, \mathrm{A}\mathbf{X})$ (note however that such an A^* may not exist for a given p). Since f is not a convex function of A, there can be multiple minimizers. Moreover, as f is continuous and $\mathbb{S}_p^q$ is compact, this problem is well defined by the Weierstrass extreme value theorem. Note that the compactness of $\mathbb{S}_p^q$ follows from the fact that it is a continuous pre-image of a compact set, i.e., $\mathbb{S}_p^q = g^{-1}(\{\mathrm{I}_p\})$ where $g(\mathrm{A}) = \mathrm{A}\mathrm{A}^\top$.

With this problem in mind, we pose the following questions:

i) In the special case where $(\mathbf{X}, Y)$ is multivariate Gaussian, how could we find a closed-form solution for A^*?

ii) In other cases, how could we find an approximate solution for A^*?

We will tackle the first question in section 3, and the second question in sections 4 and 5.

3. SDR ON GAUSSIAN MODELS

In the following proposition, we prove that when the joint distribution of $(\mathbf{X}, Y)$ is multivariate Gaussian, the matrices $\mathrm{A} \in \mathbb{R}^{p\times q}$ such that $Y \perp\!\!\!\perp \mathbf{X}|\mathrm{A}\mathbf{X}$ satisfy a certain equation, effectively reducing problem (1) from minimizing an objective function to solving an equation. Moreover, when $p = 1$, the solutions can be written explicitly.

Proposition 2. *Let $\mathbf{X} \sim \mathcal{N}_q(\mathbf{0}, \Sigma_x)$, $Y \sim \mathcal{N}_1(0, \sigma_Y^2)$, and $\mathbf{Z} = (\mathbf{X}, Y) \sim \mathcal{N}_{q+1}(\mathbf{0}, \Sigma)$, where*

$$\Sigma = \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \sigma_y^2 \end{bmatrix}.$$

We have:

1) $Y \perp\!\!\!\perp \mathbf{X}|\mathrm{A}\mathbf{X}$ (where $\mathrm{A} \in \mathbb{R}^{p\times q}$, not necessarily in $\mathbb{S}_p^q$) if and only if

$$\Sigma_{xy} = \Sigma_x \mathrm{A}^\top (\mathrm{A}\Sigma_x\mathrm{A}^\top)^{-1}\mathrm{A}\Sigma_{xy}$$

and

$$\Sigma_{yx} = \Sigma_{yx}\mathrm{A}^\top(\mathrm{A}\Sigma_x\mathrm{A}^\top)^{-1}\mathrm{A}\Sigma_x.$$

2) $f(\mathrm{A}) = \frac{1}{2}\log\left(1 - \frac{\Sigma_{yx}\mathrm{A}^\top(\mathrm{A}\Sigma_x\mathrm{A}^\top)^{-1}\mathrm{A}\Sigma_{xy}}{\sigma_y^2}\right)$.

3) $f(\mathrm{A}) \geq \frac{1}{2}\log\left(\frac{|\Sigma|}{|\Sigma_x|\sigma_y^2}\right)$.

4) *When* $p = 1$, A *reduces to a* q*-dimensional vector, say* $\mathrm{A} = \beta^\top$. *In this case,* $Y \perp\!\!\!\perp \mathbf{X}|\beta^\top\mathbf{X}$ *if and only if* $\beta = \lambda\Sigma_x^{-1}\Sigma_{xy}$ *(the OLS solution) for some* $\lambda \neq 0$.

It is worth noting that for $p > 1$, we still do not have a closed-form solution for A^*. Moreover, the assumption that the joint distribution $(\mathbf{X}, Y)$ is multivariate Gaussian can be too restrictive. In the following sections, we introduce a more general model setting and propose some techniques for estimating the solutions to problem (1).

4. SDR ON GAUSSIAN MIXTURE MODELS

The motivation for using Gaussian mixture models to represent the joint distribution of $(\mathbf{X}, Y)$ is the fact that any probability density function with a compact support can be estimated by a finite mixture of Gaussian distributions in the supremum norm (Lemma 1 of Sandberg (2001)). However, when we have more than one component (which is most likely the case), there is no closed-form solution for problem (1) because the mutual information is an integral that is analytically intractable. Thus, constrained optimization techniques can be employed to produce good approximate solutions.

Now, we describe our problem setting in more detail. Suppose $\mathbf{Z} = (\mathbf{X}, Y)^\top$ is modeled by a Gaussian mixture model, i.e.,

$$\mathbf{Z} \sim \sum_{j=1}^m \pi_j \mathcal{N}_{q+1}(\mu_j, \Sigma_j),$$

where $\sum_{j=1}^m \pi_j = 1$ and $\pi_j \geq 0$. For a fixed matrix A, write $\mathbf{K} = \mathrm{A}\mathbf{X} \in \mathbb{R}^p$. Observe that

$$\begin{bmatrix} \mathbf{K} \\ Y \end{bmatrix} = \begin{bmatrix} \mathrm{A}\mathbf{X} \\ Y \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{A} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}}_{=\mathrm{B}} \underbrace{\begin{bmatrix} \mathbf{X} \\ Y \end{bmatrix}}_{=\mathbf{Z}} = \mathrm{B}\mathbf{Z}.$$

We have

$$\begin{aligned} f(\mathrm{A}) &= -I(Y, \mathbf{K}) \\ &\doteq -\mathbb{E}_{\mathbf{Z}}\left[\log p_{\mathbf{K},Y}(\mathrm{B}\mathbf{Z}) - \log p_{\mathbf{K}}(\mathrm{A}\mathrm{E}_{\mathbf{x}}\mathbf{Z})\right], \end{aligned} \qquad (2)$$

where $\mathrm{E}_{\mathbf{x}} = [I_{q\times q}\ \mathbf{0}_q] \in \mathbb{R}^{q\times(q+1)}$. It is possible to show that

$$p_{\mathbf{K},Y}(\mathrm{B}\mathbf{Z}) = \sum_{j=1}^m \pi_j \mathcal{N}_{p+1}\left(\mathrm{B}\mathbf{Z}|\mathrm{B}\mu_j, \mathrm{B}\Sigma_j\mathrm{B}^\top\right)$$

and

$$p_{\mathbf{K}}(\mathrm{A}\mathrm{E}_{\mathbf{x}}\mathbf{Z}) = \sum_{j=1}^m \pi_j \mathcal{N}_p\left(\mathrm{A}\mathrm{E}_{\mathbf{x}}\mathbf{Z}|\mathrm{A}\mathrm{E}_{\mathbf{x}}\mu_j, \mathrm{A}\mathrm{E}_{\mathbf{x}}\Sigma_j\mathrm{E}_{\mathbf{x}}^\top\mathrm{A}^\top\right).$$

In practice, since the expectation given by expression (2) is intractable, we can approximate it and its gradient using Monte Carlo integration when needed.

In real situations, the values of the parameters of the Gaussian mixture model, i.e., π_j, μ_j and Σ_j for every $j \in [m]$, may not be available. These parameters can be estimated in the first step using the expectation-maximization (EM) algorithm. We can then define

$$\hat{f}(\mathrm{A}) = -\mathbb{E}_{\hat{\mathbf{Z}}}\left[\log p_{\mathbf{K},Y}(B\hat{\mathbf{Z}}) - \log p_{\mathbf{K}}(\mathrm{A}\mathrm{E}_{\mathbf{x}}\hat{\mathbf{Z}})\right],$$

where $\hat{\mathbf{Z}} \sim \sum_{j=1}^m \hat{\pi}_j\mathcal{N}_{q+1}(\hat{\mu}_j, \hat{\Sigma}_j)$. In the subsequent discussion, we write $f(\mathrm{A})$ to represent either $f(\mathrm{A})$ or $\hat{f}(\mathrm{A})$.

5. CONSTRAINED OPTIMIZATION ON THE STIEFEL MANIFOLD

In this section, we summarize three methods for constrained optimization on the Stiefel manifold to solve problem (1), and propose an adaptive method which allows us to pick an optimal p according to some criterion which will be elaborated below.

Method 1: Natural gradient

The natural gradient method was first introduced by Amari (1998), and was used by Suzuki and Sugiyama (2010) in their SDR paper. Assuming that the tangent space of the manifold is equipped by the Euclidean inner product, i.e., $\langle Z_1, Z_2\rangle = \frac{1}{2}\mathrm{tr}(Z_1^\top Z_2)$, the steepest descent is given by the natural gradient $\nabla f(\mathrm{A}) = \frac{1}{2}\left(G^\top - \mathrm{A}G\mathrm{A}\right)$, where

$$G = \left[\frac{\partial f}{\partial \mathrm{A}_{i,j}}\right]^\top_{i\in[p],j\in[q]} \in \mathbb{R}^{q\times p}.$$

The geodetic from A to the direction of the natural gradient $\nabla f(\mathrm{A})$ over $\mathbb{S}_p^q$ can be expressed as

$$\mathrm{A}_{t+1} := \mathrm{A}_t \exp\left(-\eta_t\left(\mathrm{A}_t^\top G^\top - G\mathrm{A}_t\right)\right).$$

Here, exp denotes the matrix exponential and $\eta_t > 0$ denotes the step size for iteration t, which can be either fixed or chosen using the Armijo's rule.

Method 2: Cayley transformation

Wen and Yin (2013) proposed an alternative method for the same task (summarized in Tagare (2011)). There are two main differences between their approach and Suzuki and Sugiyama (2010)'s: (1) the former constructs the search curve via a Cayley transformation instead of the geodetic; (2) and the former the canonical inner product, i.e., $\langle Z_1, Z_2\rangle = \mathrm{tr}\left(Z_1^\top\left(I - \frac{1}{2}\mathrm{A}^\top\mathrm{A}\right)Z_2\right)$, instead of the Euclidean inner product.

Let $\mathrm{A} \in \mathbb{R}^{p\times q}$ be a point in the Stiefel manifold. It can be shown that the action of the differential on the tangent space of the Stiefel manifold is represented by $\nabla f(\mathrm{A}) =$

$\mathrm{A}(\mathrm{A}^\top G^\top - G\mathrm{A})$. The main idea of this algorithm is the construction of the descent curve. In particular, consider the curve

$$Y(\eta) = \mathrm{A}\left(I + \frac{\eta}{2}W\right)\left(I - \frac{\eta}{2}W\right)^{-1},$$

where $W \in \mathbb{R}^{q\times q}$ satisfies $W^\top = -W$. This curve enjoys some nice properties: (1) $Y(\eta)Y(\eta)^\top = \mathrm{I}_p$, i.e., $Y(\eta) \in \mathbb{S}_p^q$; (2) $Y'(0) = \mathrm{A}W$; and (3) setting $W := G\mathrm{A} - \mathrm{A}^\top G^\top$ gives a descent curve for f. The transformation matrix $\left(I + \frac{\eta}{2}W\right)\left(I - \frac{\eta}{2}W\right)^{-1}$ is known as the Cayley transformation. In order to optimize for A given an initial value A_0, one needs to search along $Y_t(\eta_t) = \mathrm{A}_t\left(I + \frac{\eta_t}{2}\left(G\mathrm{A}_t - \mathrm{A}_t^\top G^\top\right)\right)\left(I - \frac{\eta_t}{2}\left(G\mathrm{A}_t - \mathrm{A}_t^\top G^\top\right)\right)^{-1}$, where $\eta_t > 0$ can be either fixed or chosen using the Armijo's rule.

In the case where $q > 2p$, an application of the Sherman-Morrison-Woodbury formula yields a computational speed-up. In particular, it can be shown that

$$\mathrm{A}_t\left(I + \frac{\eta_t}{2}\left(G\mathrm{A}_t - \mathrm{A}_t^\top G^\top\right)\right)\left(I - \frac{\eta_t}{2}\left(G\mathrm{A}_t - \mathrm{A}_t^\top G^\top\right)\right)^{-1}$$
$$= \mathrm{A}_t - \eta_t \mathrm{A}_t V\left(I_{2p} + \frac{\eta_t}{2}U^\top V\right)^{-1} U^\top,$$

where $U = [G, \mathrm{A}_t^\top] \in \mathbb{R}^{q\times 2p}$ and $V = [\mathrm{A}_t^\top, -G] \in \mathbb{R}^{q\times 2p}$. Note that the original equation involves inverting a $q\times q$ matrix, while the modified equation involves inverting a $2p \times 2p$ matrix.

Possible extensions of method 2

A large number of extensions of Wen and Yin (2013) algorithm are present in the literature. For example, Li et al. (2020) replaced the matrix inversion in the Cayley transformation using a fixed-point iterative estimation which only involves matrix multiplications, and combined the Cayley transformation and momentum updates into an implicit vector transform.

Method 3: Projected gradient descent

We start by mentioning the following proposition:

Proposition 3. *Let* $\mathrm{M} \in \mathbb{R}^{p\times q}$ *be a rank-p matrix, and* $\mathrm{M}^T = UDV^\top$ *be the singular value decomposition of its transpose, where* $U \in \mathbb{R}^{q\times q}$, $D \in \mathbb{R}^{q\times p}$ *and* $V \in \mathbb{R}^{p\times p}$. *We have*

$$V\mathrm{I}_{\mathbf{p}\times q}U^\top \in \underset{A\in\mathbb{R}^{p\times q}: AA^\top = I_p}{\arg\min} \|\mathrm{M} - A\|_F.$$

That is, $V I_{\mathbf{p}\times q}U^\top$ *is the projection of* M *onto the Stiefel manifold.*

Using the result in Proposition 3, a scaled gradient projection method for Stiefel manifold constrained optimization proposed by Oviedo et al. (2021) can be implemented.

An adaptive way of choosing the optimal p

We propose a method for choosing the optimal p based on the ratio between $I(Y, \mathrm{A}^*\mathbf{X})$ and $I(Y, \mathbf{X})$, where A^* is the matrix A upon convergence of the algorithm, and the mutual information is calculated via Monte Carlo integration using the estimated Gaussian mixture parameters. Concretely, we perform binary search to obtain the smallest p such that the ratio is greater than some pre-specified threshold.

6. PSEUDOCODE OF OUR METHODS

In the following section, we present the pseudocode of our methods, all of which are implemented in the package `sdr`. Our algorithm is called CATO (ConstrAined sTiefel Optimization), and its adaptive version is called CATTO (Constrained AdapTive sTiefel Optimization). For each algorithm, we implement the optimization part using the natural gradient method (CATON and CATTON), Cayley transformation method (CATOC and CATTOC), and Cayley transformation method with the Armijo's rule (CATOCA and CATTOCA).

Each algorithm starts with the estimation of the number of Gaussian mixture components, as well as the mean and covariance for each component. The number of mixture components is chosen from a list of pre-specified value according to the AIC or BIC. Algorithm 1 provides a pseudocode for this routine.

Algorithm 1 Estimating m and π_j, μ_j, Σ_j for $j \in [m]$

```
Require: $M$ (a list of possible $m$), criterion (AIC or BIC), data $(\mathbf{X}_i, y_i)_{i\in[N]}$
  for $m \in M$ do
    fit a GMM with $m$ components
    take note of the criterion value
  end for
  $m_c \leftarrow$ the $m$ corresponding to the minimum criterion value
  return $m_c, \hat\pi_j, \hat\mu_j$ and $\hat\Sigma_j$ for $j \in [m_c]$
```

After the number of components and the mixture parameters are estimated, we are ready to present CATON (Algorithm 2).

Algorithm 2 CATON

```
Require: $p$ (output dimension), $\mathrm{A}_0 \in \mathbb{S}_p^q$ (initial value), $\eta$ (step size)
  $t \leftarrow 0$
  while not converged do
    $G \leftarrow \left[\frac{\partial f}{\partial(\mathrm{A}_t)_{i,j}}\right]^\top_{i\in[p],j\in[q]}$
    $\mathrm{A}_{t+1} \leftarrow \mathrm{A}_t \exp\left(-\eta\left(\mathrm{A}_t^\top G^\top - G\mathrm{A}_t\right)\right)$
    $t \leftarrow t+1$
  end while
  return $A_t$
```

Some important details:

1. The convergence is defined through early stopping. Concretely, we stop our algorithm when the value of the loss does not improve after Z_1 iterations or when Z_2 iterations have been performed in total, whichever is earlier (for some $Z_1, Z_2 \in \mathbb{Z}^+$).

2. The loss is calculated via Monte Carlo integration, where we specify an option whether or not to generate a fresh sample from the estimated mixture model for each iteration.

3. We use Pytorch's automatic differentiation to compute G.

4. In our implementation, A_0 is the p top eigenvectors of the sample covariance matrix of the data $(\mathbf{X}_i)_{i\in[N]}$ (i.e., the PCA solution).

It is important to note the details above also apply to all other methods as well. We now present CATOC (Algorithm 3).

Algorithm 3 CATOC

```
Require: $p$ (output dimension), $\mathrm{A}_0 \in \mathbb{S}_p^q$ (initial value), $\eta$ (step size)
  $t \leftarrow 0$
  while not converged do
    $G \leftarrow \left[\frac{\partial f}{\partial(\mathrm{A}_t)_{i,j}}\right]^\top_{i\in[p],j\in[q]}$
    if $q \leq 2p$ then
      $W_t = G\mathrm{A}_t - \mathrm{A}_t^\top G^\top$
      $\mathrm{A}_{t+1} \leftarrow \mathrm{A}_t\left(I + \frac{\eta}{2}W_t\right)\left(I - \frac{\eta}{2}W_t\right)^{-1}$
    else
      $U \leftarrow [G, \mathrm{A}_t^\top]$
      $V \leftarrow [\mathrm{A}_t^\top, -G]$
      $\mathrm{A}_{t+1} \leftarrow \mathrm{A}_t - \eta\mathrm{A}_t V\left(I_{2p} + \frac{\eta}{2}U^\top V\right)^{-1}U^\top$
    end if
    $t \leftarrow t+1$
  end while
  return $A_t$
```

CATOCA is similar to CATOC, but with the Armijo's rule implemented to pick η_t dynamically for each iteration. More detail can be found in Tagare (2011). Lastly, we present the pseudocode for our adaptive algorithm (Algorithm 4).

Algorithm 4 CATTON/CATTOC/CATTOCA

```
Require: $lr$ (loss ratio)
  low $\leftarrow 0$
  high $\leftarrow q$
  $arr_{res} \leftarrow [None] \times q$
  while low $!=$ high do
    mid $= \lfloor \frac{\text{low}+\text{high}}{2} \rfloor$
    run CATON/CATOC/CATOCA with $p =$ mid $+ 1$
    get $\mathrm{A}^*$ (matrix A upon convergence)
    $arr_{res}$[mid] $\leftarrow \mathrm{A}^*$
    if $I(Y, \mathrm{A}^*\mathbf{X})/I(Y, \mathbf{X}) \leq lr$ then
      low $\leftarrow$ mid $+ 1$
    else
      high $=$ mid
    end if
  end while
  return low $+ 1$ (the optimal $p$), $arr_{res}$[low] (the corresponding $\mathrm{A}^*$)
```

7. EXPERIMENTS IN THE GAUSSIAN CASE

7.1. When $p = 1$

Consider $Z = (\mathbf{X}, Y) \sim \mathcal{N}_{q+1}(\mathbf{0}, \Sigma)$ where

$$\Sigma = \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

with $q = 3$, $\sigma_y^2 = 2$,

$$\Sigma_x = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad \Sigma_{xy} = \Sigma_{yx}^\top = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}.$$

Suppose $p = 1$. Starting from $\mathrm{A}_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, we run

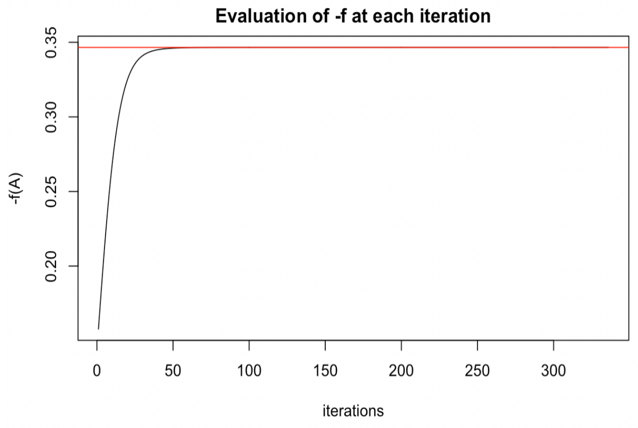
$$\mathrm{A}_{t+1} := \mathrm{A}_t \exp\left(-\eta_t\left(\mathrm{A}_t^\top G^\top - G\mathrm{A}_t\right)\right)$$

with $\eta_t = 0.1$ until $\|\partial f\|_F^2 < 10^{-8}$. We get

$$\mathrm{A}^* = \begin{bmatrix} 0.707 & 0 & 0.707 \end{bmatrix},$$

which implies Y is independent of X given $0.707(X_1 + X_3)$. Moreover, this is consistent with the equation in Proposition 2 point 4), i.e., A^* is effectively the normalized vector $\beta = \Sigma_x^{-1}\Sigma_{xy}$. It is important to note that A^* is a global minimum; indeed, as we can see in the Figure 1, the algorithm converges to the corresponding lower bound $\frac{1}{2}\log\left(\frac{|\Sigma|}{|\Sigma_x|\sigma_y^2}\right)$ (explained in Proposition 1 and Proposition 2 point 3), denoted by the red line. The case when $p = 2$ is presented in the appendix.

Fig. 1. Plot of $-f$ per iteration: $p = 1$



8. EXPERIMENTS WITH GMM

In the next two experiments, our objective is to compare our method with other well-known dimensional reduction methods: Sliced Inverse Regression (SIR) (Li, 1991), Sliced Average Variance Estimation (SAVE) Shao et al. (2007), and Principal Components Analysis (Bishop and Nasrabadi, 2006). The first two alternative methods are also methods used for sufficient dimension reduction.

8.1. Visualizing the transformation

In the first experiment, we visualize and compare the matrix used for dimension reduction among the four different methods. First, we generate data using the data generating process proposed by Friedman (1991) and implemented in Scikit-Learn (Pedregosa et al., 2011)[1]. Using that implementation, we create a dataset containing 15 independent features which only the first 5 are related to the label variable. Given that, a good supervised dimension method will be capable of distinguishing the good from bad predictors. In Figure 2, we plot the absolute values of the matrix obtained by each method and transformed by the Varimax rotation for the sake of visualization. To fit our own method, we use CATOC (Algorithm 3). From Figure 2, we can see that both GMM SDR (ours) and SAVE are the best at selecting the most important variables, i.e., the first 5 variables.

8.2. Experimenting with real data

In this second experiment, we use 8 different datasets[2] to evaluate the different methods for dimension reduction. We standardize each column in every dataset. After splitting each dataset into training and test sets, we fit the dimension reduction methods in the training part, reducing from the original number of features, which can vary from 5 to 32, to 1 feature. Then we estimate the Spearman correlation of the remaining feature with the target. In Table 1, we compare the estimated Spearman correlation achieved by each method on the test set. To fit our own dimension reduction method, we use CATOCA and use the SIR matrix as the starting guess (instead of the PCA estimate).

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_friedman1.html

[2] From www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html and https://archive.ics.uci.edu/ml/datasets.php.

Fig. 2. Visualizing the matrix used for dimension reduction comparing the four different methods. We create a dataset containing 15 independent features which only the first 5 are related to the label variable. We plot the absolute values of the matrix obtained by each method and transformed by the Varimax rotation for the sake of visualization. We can see that both GMM SDR (ours) and SAVE are the best at selecting the most important variables.

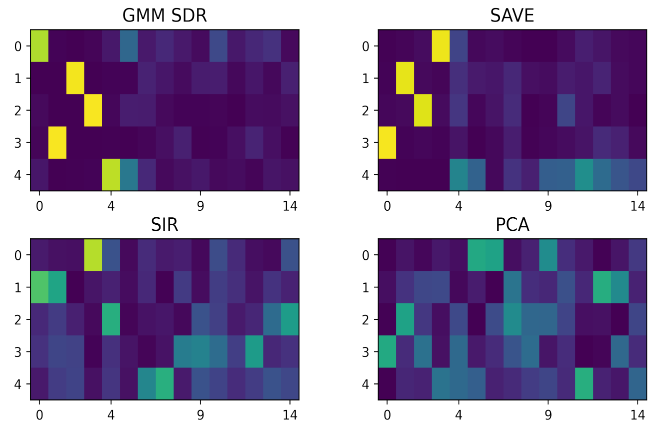


Table 1. In this second experiment, we reduce the number of features to 1 feature and then estimate the Spearman correlation of the remaining feature with the target.

Dataset	GMM SDR	SAVE	SIR	PCA
abalone	0.52	0.75	0.15	0.66
bank32nh	0.07	0.77	0.26	0.14
cal housing	0.02	0.84	0.07	0.15
cpu act	0.05	0.92	0.56	0.88
delta ailerons	0.74	0.82	0.82	0.58
elevators	0.59	0.06	0.76	0.47
fried delve	0.55	0.86	0.86	0.01
puma32H	0.02	0.41	0.40	0.06

References

S.-I. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2012.

J. H. Friedman. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991.

B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley. Sufficient dimension reduction for high-dimensional regression and low-dimensional embedding: Tutorial and survey. *arXiv preprint arXiv:2110.09620*, 2021.

J. Li, L. Fuxin, and S. Todorovic. Efficient riemannian optimization on the stiefel manifold via the cayley transform. *arXiv preprint arXiv:2002.01113*, 2020.

K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414): 316–327, 1991.

H. Oviedo, O. Dalmau, and H. Lara. Two adaptive scaled gradient projection methods for stiefel manifold constrained optimization. *Numerical Algorithms*, 87(3):1107–1127, 2021.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12: 2825–2830, 2011.

I. W. Sandberg. Gaussian radial basis functions and inner product spaces. *Circuits, Systems and Signal Processing*, 20(6):635–642, 2001.

Y. Shao, R. D. Cook, and S. Weisberg. Marginal tests with sliced average variance estimation. *Biometrika*, 94(2):285–296, 2007.

T. Suzuki and M. Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 804–811. JMLR Workshop and Conference Proceedings, 2010.

H. D. Tagare. Notes on optimization on stiefel manifolds. *Yale University, New Haven*, 2011.

Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1):397–434, 2013.

9. APPENDIX

9.1. Proof of Proposition 2

Proof. We first prove 1). Notice that

$$\begin{bmatrix} \mathbf{X} \\ Y \\ \mathrm{A}\mathbf{X} \end{bmatrix} = \Psi \begin{bmatrix} \mathbf{X} \\ Y \end{bmatrix}, \quad \Psi = \begin{bmatrix} \mathrm{I}_q & \mathbf{0}_q \\ \mathbf{0}_q^\top & 1 \\ \mathrm{A} & \mathbf{0}_p \end{bmatrix}$$

with $\Psi \in \mathbb{R}^{(q+1)\times(p+q+1)}$. Thus,

$$[\mathbf{X}, Y, \mathrm{A}\mathbf{X}] \sim \mathcal{N}_{1+q+p}(\mathbf{0}, \Psi\Sigma\Psi^\top)$$

with

$$\Psi\Sigma\Psi^\top = \begin{bmatrix} \Sigma_x & \Sigma_{xy} & \Sigma_x\mathrm{A}^\top \\ \Sigma_{xy}^\top & \sigma_y^2 & \Sigma_{xy}^\top\mathrm{A}^\top \\ \mathrm{A}\Sigma_x & \mathrm{A}\Sigma_{xy} & \mathrm{A}\Sigma_x\mathrm{A}^\top \end{bmatrix}.$$

This means $Y, \mathbf{X}|\mathrm{A}\mathbf{X}$ is Gaussian with covariance matrix

$$\begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^\top & \sigma_y^2 \end{bmatrix} - \begin{bmatrix} \Sigma_x\mathrm{A}^\top \\ \Sigma_{xy}^\top\mathrm{A}^\top \end{bmatrix} (\mathrm{A}\Sigma_x\mathrm{A}^\top)^{-1} \begin{bmatrix} \mathrm{A}\Sigma_x & \mathrm{A}\Sigma_{xy} \end{bmatrix},$$

whence $Y \perp\!\!\!\perp \mathbf{X}|\mathrm{A}\mathbf{X}$ if and only if the off-diagonal block elements of the covariance matrix are zero, showing the first part of 1). The second part follows easily upon realizing that $\Sigma_{yx} = \Sigma_{xy}^\top$. Now, we prove 2). Note that some notations (e.g., $\mathrm{B}\mathbf{Z}$) are borrowed from section 4. The following lemma is well known:

Lemma 1. *If* $(\mathbf{X}_1, \mathbf{X}_2) \sim N(\mathbf{0}, \Sigma)$ *with* $\Sigma = \begin{bmatrix} \Sigma_1 & * \\ * & \Sigma_2 \end{bmatrix}$, *then* $I(\mathbf{X}_1, \mathbf{X}_2) = \frac{1}{2}\log\left(\frac{|\Sigma_1||\Sigma_2|}{|\Sigma|}\right)$.

Since $\mathbf{Z} \sim \mathcal{N}_{q+1}(\mathbf{0}, \Sigma)$ and

$$\begin{cases} (\mathrm{A}\mathbf{X}, Y) = \mathrm{B}\mathbf{Z} & \sim \mathcal{N}_{p+1}\left(\mathrm{B}\mathbf{Z}|\mathbf{0}, \mathrm{B}\Sigma\mathrm{B}^\top\right) \\ \mathrm{A}\mathbf{X} & \sim \mathcal{N}_p\left(\mathrm{A}\mathrm{E}_{\mathbf{x}}\mathbf{Z}|\mathbf{0}, \mathrm{A}\mathrm{E}_{\mathbf{x}}\Sigma\mathrm{E}_{\mathbf{x}}^\top\mathrm{A}^\top\right) \\ & = \mathcal{N}_p\left(\mathrm{A}\mathrm{E}_{\mathbf{x}}\mathbf{Z}|\mathbf{0}, \mathrm{A}\Sigma_x\mathrm{A}^\top\right) \\ Y & \sim \mathcal{N}_1\left(y|0, \sigma_y^2\right), \end{cases}$$

by Lemma 1 we get

$$\begin{aligned} f(\mathrm{A}) = -I(Y, \mathbf{K}) &= \tfrac{1}{2}\log\left(\frac{|\mathrm{B}\Sigma\mathrm{B}^\top|}{|\mathrm{A}\Sigma_x\mathrm{A}^\top|\sigma_y^2}\right) \\ &= \tfrac{1}{2}\log\left(\frac{\left|\begin{pmatrix} \mathrm{A}\Sigma_x\mathrm{A}^\top & \mathrm{A}\Sigma_{xy} \\ \Sigma_{yx}\mathrm{A}^\top & \sigma_y^2 \end{pmatrix}\right|}{|\mathrm{A}\Sigma_x\mathrm{A}^\top|\sigma_y^2}\right) \\ &= \tfrac{1}{2}\log\left(\frac{|\mathrm{A}\Sigma_x\mathrm{A}^\top|\,|\sigma_y^2 - \Sigma_{yx}\mathrm{A}^\top(\mathrm{A}\Sigma_x\mathrm{A}^\top)^{-1}\mathrm{A}\Sigma_{xy}|}{|\mathrm{A}\Sigma_x\mathrm{A}^\top|\sigma_y^2}\right) \\ &= \tfrac{1}{2}\log\left(1 - \frac{\Sigma_{yx}\mathrm{A}^\top(\mathrm{A}\Sigma_x\mathrm{A}^\top)^{-1}\mathrm{A}\Sigma_{xy}}{\sigma_y^2}\right), \end{aligned}$$

where we remove the absolute value because the value inside is a real positive value. To prove 3), observe that Proposition 1 gives us $f(\mathrm{A}) \geq -I(Y, \mathbf{X}) \overset{\text{Lemma 1}}{=} \frac{1}{2}\log\left(\frac{|\Sigma|}{|\Sigma_x|\sigma_y^2}\right)$. To prove 4), simply note that $Y \perp\!\!\!\perp \mathbf{X} | \beta^\top \mathbf{X}$ if and only if

$$\Sigma_{xy} = \frac{\beta^\top \Sigma_{xy}}{\beta^\top \Sigma_x \beta}\Sigma_x \beta,$$

if and only if

$$\beta = \frac{\beta^\top \Sigma_x \beta}{\beta^\top \Sigma_{xy}}\Sigma_x^{-1}\Sigma_{xy},$$

where we use 1). From here, we can see that any solution β is of the form $\lambda \Sigma_x^{-1}\Sigma_{xy}$ for some $\lambda \neq 0$ provided Σ_x is invertible. However, it is easy to see that if β is a solution, so is $\xi\beta$ for any $\xi \neq 0$. Therefore, we have $Y \perp\!\!\!\perp \mathbf{X}|\beta^\top \mathbf{X}$ if and only if $\beta = \lambda\Sigma_x^{-1}\Sigma_{xy}$ for some $\lambda \neq 0$.

□

9.2. Proof of Proposition 3

Proof. Since $\|\mathrm{M} - A\|_F^2 = \operatorname{tr}((\mathrm{M} - A)^\top(\mathrm{M} - A)) = q + \operatorname{tr}(\mathrm{M}^\top \mathrm{M}) - 2\operatorname{tr}(\mathrm{M}^\top A)$, it is sufficient to maximize $\operatorname{tr}(\mathrm{M}^\top A) = \operatorname{tr}(UDV^\top A) = \operatorname{tr}(DV^\top AU) = \operatorname{tr}(DO)$, where $O = V^\top AU$. Notice that $OO^\top = V^\top AUU^\top A^\top V = \mathrm{I}_p$, which implies $0 \leq |o_{ij}| \leq 1$. Then $\operatorname{tr}(\mathrm{M}^\top A) = d_{11}o_{11} + \cdots + d_{qq}o_{qq}$ is maximized when (for example) $O = \mathrm{I}_{p\times q}$, i.e., $V^\top AU = \mathrm{I}_{p\times q}$ i.e., $A = V\mathrm{I}_{p\times q}U^\top$. □

9.3. Experiment in the Gaussian case when $p = 2$

We use the same structure as in the case when $p = 1$. Starting from $\mathrm{A}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, we run

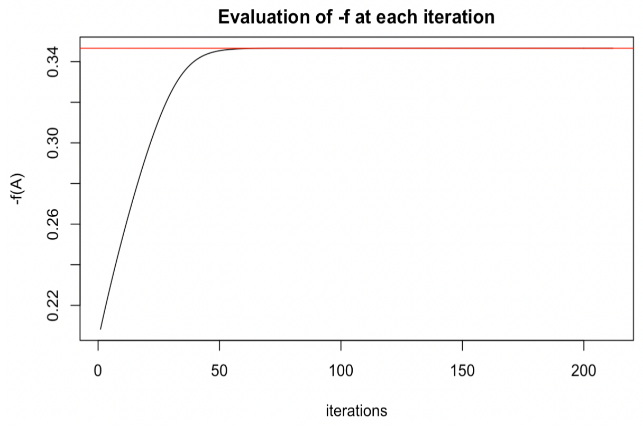
$$\mathrm{A}_{t+1} := \mathrm{A}_t \exp\left(-\eta_t\left(\mathrm{A}_t^\top G^\top - G\mathrm{A}_t\right)\right)$$

with $\eta_t = 0.1$ until $\|\partial f\|_F^2 < 10^{-8}$. We get

$$\mathrm{A}^* = \begin{bmatrix} 0.734 & 0.165 & 0.657 \\ 0.095 & 0.935 & -0.341 \end{bmatrix}.$$

Thus, Y is independent of X given $0.734X_1 + 0.165X_2 + 0.657X_3$ and $0.095X_1 + 0.935X_2 + -0.341X_3$. Also, we notice that this is consistent with the equation in Proposition 2 point 1). More importantly, A^* is also a global minimum; indeed, as we can see in Figure 3, the algorithm converges to the corresponding lower bound $\frac{1}{2}\log\left(\frac{|\Sigma|}{|\Sigma_x|\sigma_y^2}\right)$ denoted by the red line.

Fig. 3. Plot of $-f$ per iteration: $p = 2$



9.4. A picture of cute cats

Fig. 4. Cat 1 to Cat 2: "I love you"

9.5. Next steps

1. Implement and evaluate the projected GD algorithm.
2. Try using data instead of Monte-Carlo integration on the estimated GMM parameters.