Introduction to Dimensionality Reduction

Jonathan Rosenblatt
Ben Gurion University

January 15, 2017

Contents

1 Enter the King: Principal Component Analysis **1**

2 Preliminaries **5**

3 Latent Variable Generative Approaches **6**

4 Purely Algorithmic Approaches **8**

5 Dealing with the high-dimension **12**

Example 1 (BMI)**.** Consider the heights and weights of a sample of individuals. The data may seemingly reside in 2 dimensions but given the height, we have a pretty good guess of a person's weight, and vice versa. We can thus state that heights and weights are not really two dimensional, but roughly lay on a 1 dimensional subspace of $\mathbb{R}^2$.

Example 2 (g-factor)**.** Consider the correctness of the answers to a questionnaire with p questions. The data may seemingly reside in a p dimensional space, but assuming there is a thing as "skill", then given the correctness of a person's reply to a subset of questions, we have a good idea how he scores on the rest. Put differently, we don't really need a 200 question questionnaire– 100 is more than enough. If skill is indeed a one dimensional quality, then the questionnaire data should organize around a single line in the p dimensional cube.

Example 3 (Blind signal separation)**.** Consider n microphones recording an individual. The digitized recording consists of p samples. Are the recordings really a shapeless cloud of n points in $\mathbb{R}^p$? Since they all record the same sound, one would expect the n p-dimensional points to arrange around the source sound bit: a point in $\mathbb{R}^p$. If microphones have different distances to the source, volumes may differ. We would thus expect the n points to arrange about a line that ends at the source.

1 Enter the King: Principal Component Analysis

Principal Component Analysis (PCA) is such a basic technique, it has been rediscovered and renamed independently in many fields. It can be found under the names of *Discrete Karhunen–Loève Transform; Hotteling Transform; Proper Orthogonal Decomposition; Eckart–Young Theorem; Schmidt–Mirsky Theorem; Empirical Orthogonal Functions; Empirical Eigenfunction Decomposition; Empirical Component Analysis; Quasi-Harmonic Modes; Spectral Decomposition; Empirical*

Modal Analysis, and possibly more[1]. The many names are quite interesting as they offer an insight into the different problems that led to PCA's (re)discovery.

Return to the BMI problem in Exampl 1. Assume you now wish to give each individual a "size score", that is a **linear** combination of height and weight: PCA does just that. It returns the linear combination that has the largest variability, i.e., the combination which best distinguishes between individuals.

The variance maximizing motivation above was the one that guided Hotelling [1933]. But 30 years before him, Pearson [1901] derived the same procedure with a different motivation in mind. Pearson was also trying to give each individual a score. He did not care about variance maximization, however. He simply wanted a small set of coordinates in some (linear) space that approximates the original data well. As it turns out, the best linear-space approximation of X is also the variance maximizing one. More precisely: the *sequence* of $1, \ldots, p$ dimensional linear spaces that best approximate X in squared distance, is exactly the sequence of $1, \ldots, p$ dimensional scores, that best separate between the n samples. Pearson and Hotelling (among others) thus arrived to the exact same solution, with different motivations.

1.1 Mathematics of PCA

We now present the derivation of PCA from the two different motivations.

1.1.1 Variance Maximizing View of PCA

Proof. The sketch of the proof is the following: we will first show that the weight vector that maximizes the variance of the score is the eigenvector that corresponds to the first principal component. We will do so for the *population* covariance, Σ, and wrap up by plugging its empirical counterpart, $X'X$ (assuming a centered X).

Starting with the first principal component. For a random p-vector, $\mathbf{x}$ denote $\Sigma := \mathbf{Cov}[\mathbf{x}]$, so that for a fixed p-vector v: $\mathbf{Cov}[v'\mathbf{x}] = v'\Sigma v$. Finding a linear combination of $\mathbf{x}$ that best separates individuals, means maximizing $\mathbf{Cov}[v'x]$ w.r.t. to v. Clearly, $\mathbf{Cov}[v'x]$ may explode if any v is allowed. It is most convenient, mathematically, to constrain the l_2 norm: $\|v\|_2^2 = 1$. Maximizing under a constraint, using Lagrange-Multipliers:

$$argmax_v\{v'\Sigma v - \lambda(\|v\|_2^2 - 1)\}. \tag{1}$$

Differentiating w.r.t v and equating zero:

$$(\Sigma - \lambda I)v = 0. \tag{2}$$

We thus see that any of the p eigenvalue-eigenvector pairs of Σ is a local extremum. Which of them to pick? To find a *global* maximum we return to the original problem, and use the fact that the only solutions to (1) are eingenvalue-eigenvector pairs:

$$argmax_{v:\|v\|_2^2=1}\{v'\Sigma v\} = argmax_\lambda\{v'\lambda v\} \tag{3}$$

so that the global maximum is obtained with the largest eigenvalue λ. Put differently, the weight vector that returns the score that best separates individuals, is the eigenvector of Σ with the largest eigenvalue.

The second principal component can be found by solving the same problem, with the additional constraint of v_2 orthogonal to v_1, and so on, until v_p.

The last missing ingredient is that instead of the true covariance between the features, Σ, we use the (centered) empirical covariance $X'X$. □

[1] `http://en.wikipedia.org/wiki/Principal_component_analysis`

Remark 1. Readers familiar with matrix norms will recognize that the above is exactly the derivation of the spectral norm of Σ, and the variance of the first score, is exactly the spectral norm[2] of Σ.

1.1.2 Linear-Space approximation view

In here, we try to find a series of $\mathcal{M}_q; q = 1, \ldots, p$, such that $\mathcal{M}_q$ is a *linear* subspace of dimension q which well approximates X in the Frobenius norm. The problem to solve is

$$argmin_{A \in \mathbb{R}^{q \times p}, S \in \mathbb{R}^{n \times q}} \{\|X - SV\|_{Frob}\} \tag{4}$$

and $\mathcal{M}_q$ is the span of S, and $V'X = Xv_1, \ldots, Xv_q$ are the q PCs.

1.1.3 Bi Plot

The *Bi-Plot* shows the two PCs, $(PC_1, PC_2) := (Xv_1, Xv_2)$, of the original data points. These scores are known as the *Principal Componets* (PCs). The contribution of each original variable to each PC, $V = (v_1, \ldots, v_q)$, is called the *Loadings*. The bi-plot also shows v_1, v_2, i.e., the contribution of each of the original variables to each of the PCs. See example in Figure 1.

Principal Components

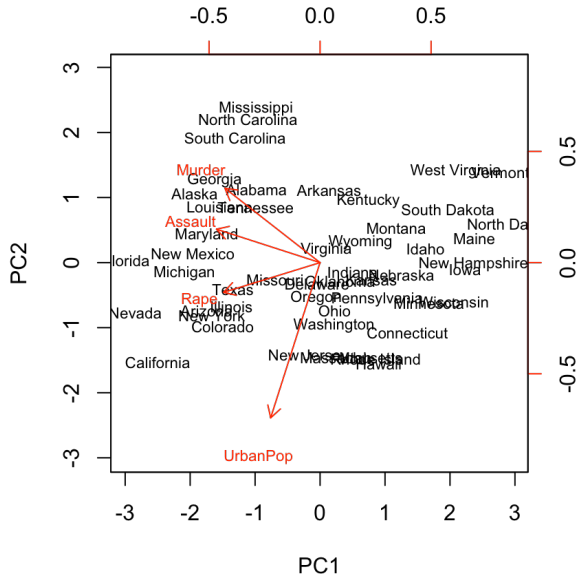

Figure 1: **BiPlot**. Arrest type data for USA states. Data includes urban population size, number of rape related arrests, assault related, and murder related ($p = 4$). Each city is presented against its two first PCs. Arrows encode the loadings. They show that PC1 encodes a general crime level, as it is the average of all type of crimes. PC2 measures the level of urbanization, as it is dominated by the UrbanPopulation variable.
Source: `https://goo.gl/85qtKv`

1.1.4 Scree Plot

[TODO]

[2] A.k.a. *induced norm* l_2 *norm*, *operator norm*.

1.1.5 Why did Hotelling and Pearson arrive to the same solution?

We have currently offered two motivations for PCA: (i) Find linear combinations $v_1, \ldots, v_p$ that best distinguish between observations, i.e., maximize variance. (ii) Find the linear subspaces $\mathcal{M}_1, \ldots, \mathcal{M}_p$ that best approximates the data. The reason these two problems are equivalent, Pythagoras.

Informally speaking, the data has some total variance. In analogy to the $SST = SSR + SSE$ decomposition in linear regression, the total variance of X can be decomposed into the part in $\mathcal{M}_q$, and the part orthogonal. The orthogonal part is the distance of X from $\mathcal{M}_q$. Maximizing the variance in $\mathcal{M}_q$ is thus the same as minimizing the distance from X to $\mathcal{M}_q$.

The only unresolved matter- is why the solution to the variance maximization problem is a *linear* subspace? This is simply because all the scores, are linear combinations of columns of X, thus span a linear subspace, as is sought in the linear-subspace approximation view.

1.2 How many PCs can one recover?

On the face of it, with p variables one can find p PCs. Things are not that simple however.

In the population version of the problem, i.e., when Σ, and not $X'X$ is known, the number of PCs is the number of "independent information pieces" in $\mathbf{x}$. It will trivially be p unless entries in $\mathbf{x}$ are fully multicollinear. Stating that Σ has full rank, is thus identical to stating that the variables of $\mathbf{x}$ are not fully correlated, and thus, that p PCs can be recovered.

In the empirical version of the problem, i.e., when $X'X$, and not Σ is known, things further complicate. First, for purely algebraic reasons, there may be as many non zero eigenvalues as the rank of X. Clearly, if $p > n$, variables of X have to be linearly dependent, so that X, thus $X'X$, cannot possibly be of full rank. Put differently, the number of PCs that may be computed is the same as the dimension of the rank of X.

While the algebraic problem ends when $p < n$, the statistical one only begins, and ends if $n \gg p$. This is because p eigenvalues and vectors include roughly $\mathcal{O}(p + p^2)$ parameters. If $p < n$ but $p \sim n$ then we do not have many observations per estimated parameter. In the statistical literature, this is known as a *high-dimensional* problem. In the engineering parlance, we say we have low *signal-to-noise*. For a an analytical treatment of the statistical properties of PCA, see Nadler [2008]. A purely algorithmic approach to the choice of q is also available by adopting the supervised learning view of dimensionality reduction. As can be seen from Eq.(4), PCA, like most other reduction approaches, is an Empirical Risk Minimization (ERM) problem. As such, a resampling approach is made possible by putting data aside, and looking for a minimum of the out-of-sample reconstruction error[3], as q grows from 1 to p.

Empirical Risk Minimization

1.3 PCA as a Graph Method

It turns out that we may find the sequence of best approximating linear subspaces, i.e., the PCs, without the actual measurements X, but only with a dissimilarity graph. In particular, with the $n \times n$ graph $\mathfrak{D}$ of Euclidean distances between individuals: $\mathfrak{D}_{i,j} := \|x_i - x_j\|_2$.

It should come of no surprise that we don't need the actual measurements, X, since the optimal loadings, v, only depend on the covariance Σ, or its empirical counterpart, $X'X$. It may, however, be quite surprising that given only the distances between individuals, we may not recover the covariance between variables, $X'X$, but we can recover the PCs. Put differently, to find a low dimensional $\mathcal{M}$ that approximates the data, we don't need the whole data, but rather, only the graph of distances between data points.

For proof of the above statement, we refer the reader to Friedman et al. [2001, Sec.18.5.2]

[3] Using your favorite resampling algorithm such as V-fold CV, Bootstrapping, train-test, etc.

This observation will later be very useful for other dimensionality reduction algorithms, which operate not on the original data points, but rather, on dissimilarity graphs.

Think about it. Is it surprising that for a low dimensional representation of the data we only need to know the distances between points? Is it surprising that the PCs are agnostic to the means of the data?

1.4 PCA as Supervised Linear Regression

There is a strong link between dimensionality reduction and supervised learning. Look at Eq.(4), does it not look very familiar to a linear regression? Indeed, PCA can be thought of as finding a small set of predictors that linearly predict X. Almost all supervised learning algorithms can thus be applied to dimensionality reduction, by viewing X as the labels, y, of some latent scores in $\mathcal{M}$.

2 Preliminaries

2.1 Terminology

Variable A.k.a. *dimension*, or *feature* in the machine learning literature, or *column* for reasons that will be obvious in the next item.

Data A.k.a. *sample*, *observations*. Will typically consist of n, p dimensional vectors. We typically denote the data as a $n \times p$ matrix X.

Manifold A space which is regular enough so that it is *locally* has all the properties of a linear space. We will denote an arbitrary manifold by $\mathcal{M}$, and by $\mathcal{M}_q$ a q dimensional[a] manifold.

Embedding Informally speaking: a "shape preserving" mapping of a space into another.

Linear Embedding An embedding done via a linear operation (thus representable by a matrix).

Generative Model Known to statisticians as the *sampling distribution*. The assumed stochastic process that generated the observed X.

[a] You are probably used to thinking of the *dimension* of linear spaces. We will not rigorously define what is the dimension of a manifold, but you may think of it as the number of free coordinates needed to navigate along the manifold.

2.2 Motivations for dimensionality reduction

Scoring Give each observation an interpretable, simple score (Hotelling's motivation).

Latent structure Recover unobservables from indirect measurements. E.g: Blind signal reconstruction, CT scan, cryo-electron microscopy, etc.

SNR Denoise measurements before further processing like clustering, supervised learning, etc.

Compression Save on RAM ,CPU, and communication when operating on a lower dimensional representation of the data.

2.3 Properties of a dimensionality reduction approaches

Here are some properties that characterize dimensionality reduction approaches. The reader is advised to try and characterize each approach along the following properties:

Generative vs. algorithmic Refers to the motivation of the approach. Is it stated as an algorithm, or stated via some generative probabilistic model. PCA is purely algorithmic.

Linear $\mathcal{M}$ vs. non-linear $\mathcal{M}$. Is the target manifold linear or not? In PCA, $\mathcal{M}$ is linear.

Linear embedding vs. non-linear embedding. Is the embedding into $\mathcal{M}$ a linear operation? In PCA, the embedding is linear, and indeed, represented by a matrix.

Learning an embedding vs. an embedding function? Will we need to apply the reduction to new data? If yes, we need to learn an *embedding function*, $g : x \mapsto g(x) \in \mathcal{M}$. If no, and we merely want to low dimensional representation of existing data, $\{g(x_i)\}_{i=1}^n$, we only need to learn an embedding. PCA learns an embedding *function*.

Euclidean vs. non-Euclidean. Many dimensionality reduction methods only need a dissimilarity (i.e. distance) graph to operate. The Euclidean distance is historically the most popular dissimilarity measure. Some approaches are agnostic to the measure used, some have Euclid hard-wired into them, and some, hard-wire themselves with some non-Euclidean norm. PCA has Euclid hard-wired.

Remark 2 (Non linear manifolds and non-linear embeddings)**.** The distinction we make between non-linear *embeddings* and non-linear manifolds is non-standard in the literature. The term *non-linear space embedding* is used in both contexts, but typically only for the type of embedding, and not the target manifold.

2.4 Dimensionality Reduction and Supervised Learning

In the machine learning literature, dimensionality reduction belongs in the *unsupervised* type of learning. Indeed, class labels are not required to reduce dimension. The problem, however, are not unrelated, as demonstrated for the PCA problem (Sec 1.4), X can be viewed as the labels, making dimensionality reduction a supervised learning problem, with *unobservable*, or *missing* features.

3 Latent Variable Generative Approaches

All generative approaches to dimensionality reduction will include some unobserved set of variables, which we can try to recover from the observable X. The unobservable variables will typically have a lower dimension than the observables, thus, dimension is reduced. We start with the simplest case of linear Factor Analysis.

3.1 Factor Analysis (FA)

FA originates from the psychometric literature. We thus revisit the IQ (actually g-factor) Example 2:

Example 4 (g-factor[4])**.** Assume n respondents answer p quantitative questions: $x_i \in \mathbb{R}^p, i = 1, \ldots, n$. Also assume, their responses are some *linear* function $V \in \mathbb{R}^p$ of a single personality attribute, s_i. We can think of s_i as the subject's "intelligence". We thus have

$$x_i = V s_i + \varepsilon_i \tag{5}$$

[4] `https://en.wikipedia.org/wiki/G_factor_(psychometrics)`

And in matrix notation, for $q < p$ latent attributes:

$$X = SV + \varepsilon, \tag{6}$$

where V is the $q \times p$ matrix of factor loadings, and S the $n \times q$ matrix of latent personality traits. In our particular example where $q = 1$, the problem is to recover the unobservable intelligence scores, $s_1, \ldots, s_n$, from the observed answers X.

We may try to estimate SV by assuming some distribution on S and ε and apply maximum likelihood. Under standard assumptions on the distribution of S and ε, recovering S from $\widehat{SV}$ is still impossible as there are infinitely many such solutions. In the statistical parlance we say the problem is *non identifiable*, and in the applied mathematics parlance we say the problem is *ill posed*. To see this, consider an orthogonal *rotation* matrix R ($R'R = I$). For each such R: $SV = SR'RV = S^*V^*$. While both solve Eq(6), V and V^* may have very different interpretations. This is why many researchers find FA an unsatisfactory inference tool.

Remark 3 (Identifiability in PCA)**.** The non-uniqueness (non-identifiability) of the FA solution under variable rotation is never mentioned in the PCA context. Why is this? This is because the methods solve different problems. The reason the solution to PCA is well defined is that PCA does not seek a single S but rather a *sequence* of S with dimensions growing from 1 to p.

Remark 4 (Linear and non-linear embeddings)**.** In classical FA in Eq.(6) is clearly an embedding to a linear space. The one spanned by S. Under the classical probabilistic assumptions on S and ε the embedding itself is also linear, and is sometimes solved with PCA. Being a generative model, there is no restriction for the embedding to be linear, and there certainly exists sets of assumptions for which the FA embedding is non linear.

FA Terminology The FA terminology is slightly different than PCA:

- **Factors**: The unobserved attributes S. Not to be confused with the *principal components* in the context of PCA.
- **Loadings**: The V matrix; the contribution of each factor to the observed X.
- **Rotation**: An arbitrary orthogonal re-combination of the factors, S, and loadings, V, which changes the interpretation of the result.

The FA literature does offer several heuristics to "fix" the solution of the FA. These are known as *rotations*:

- **Varimax**: By far the most popular rotation. Attempts to construct factors that are similar to the original variables, thus facilitating interpretation[5].
- **Quartimax**: Seeks a minimal number of factors to explain each variable. May thus result factors that are uninterpretable, since they all rely on the same variables.
- **Equimax**: A compromise between Varimax and Quartimax.
- **Oblimin**: Relaxes the requirement of the factors to be uncorrelated, so that they may be similar to the original variables; even more so than in varimax. This facilitates the interpretability of the factors.
- **Promax**: A computationally efficient approximation of oblimin.

Remark 5 (Rotations as Bayesian Priors)**.** I always thought of the various rotations as the resulting Bayesian posterior estimates with different priors on V. I have no idea if this intuition can be made rigorous, but it works for me...

[5]This can be seen as a "soft" approach to sPCA (Sec.5.2)

3.2 Non Linear Factor Analysis

Classical FA deals with features, X, that are linear in the latent factors, S. Like any other generative model approach, it can be easily extended to deal with non-linear functions of the latent factors: $X = g(S)$, provided that $g : \mathbb{R}^q \mapsto \mathbb{R}^p$ is one-to-one. This problem is a non-linear embedding into a linear space (the one spanned by S). It can also be seen as the generative counterpart of kPCA (Sec.4.3) or auto-encoders (Sec.4.10).

3.3 Independent Component Analysis (ICA)

Like FA, ICA is a family of latent space models, thus, a *meta-method*. It assumes data is generated as some function of the latent variables S. In many cases this function is assumed to be linear in S so that ICA is compared, if not confused, with PCA and even more so with FA.

The fundamental idea of ICA is that S has a joint distribution of *non-Gaussian independent* variables. This independence assumption, solves the the non-uniquness of S in FA.

Being a generative model, estimation of S can then be done using maximum likelihood, or other estimation principles.

ICA is a popular technique in signal processing, where V is actually the signal, such as sound in Example 3. Recovering V is thus recovering the original signals mixing in the recorded X.

Remark 6 (ICA and FA)**.** The solutions to the (linear) ICA problem can ultimately be seen as a solution to the FA problem with a particular rotation R implied by the probabilistic assumptions on S. Put differently, the formulation of the (linear) ICA problem, implies a unique rotation, which can be thought of as the rotation that returns components that are as far from Gaussian as possible.

Remark 7 (Linear and non-linear embeddings)**.** In classical ICA in Eq.(6) is clearly an embedding to a linear space. The one spanned by S. The probabilistic assumptions on S and ε the embedding itself being non linear, thus solved as an optimization problem, and not via PCA.

4 Purely Algorithmic Approaches

We now discuss dimensionality reduction approaches that are not stated via their generative model, but rather, directly as an algorithm. This does not mean that they cannot be cast via their generative model, but rather they were not motivated as such.

4.1 Multidimensional Scaling (MDS)

Very roughly speaking, MDS can be thought of as a variation on PCA, that (i) begins with a distance graph[6] $\mathfrak{D}$, and (ii) embeds into a two-dimensional space. Regarding (i), we have already seen in Section 1.3 that PCA really only needs a similarity graph, $X'X$, and not the whole X. Regarding (ii), the embedding into two dimensions is motivated by that fact that MDS is typically used for visualization.

MDS aims at embedding a graph of distances, while preserving the original distances. Basic results in graph/network theory [e.g. Graham, 1988] suggest that the geometry of a graph cannot be preserved when embedding it into lower dimensions. The different types of MDSs, such as *Classical MDS*, and *Sammon Mappings*, differ in the *stress function* penalizing for geometric distortion.

[6]The term Graph is typically used in this context instead of Network. But a graph allows only yes/no relations, while a network, which is a weighted graph, allows a continuous measure of similarity (or dissimilarity). It is thus more appropriate.

Sadly, MDS may scale poorly to large dissimilarity matrices, and the optimization may converge to a local minimum. The solution to MDS is an embedding and not an embedding *function*. When new data points are made available, the embedding will thus have to be re-learned.

4.1.1 Mathematics of MDS

We start with either a dissimilarity network $\mathfrak{D} = (d_{i,j})$, or a similarity network $\mathfrak{S} = (s_{i,j})$. Similarities can be thought of as correlations, and dissimilarities as distances (which are indeed the typical measures in use). Define $z_i \in \mathbb{R}^q$ the location of point i in the target linear space of rank q. The z_i's are set to minimize some penalty for geometric deformation called the *stress function*. Typical stress functions include:

Classical MDS A.k.a. *Torgerson scaling* begins with the empirical covariance as the similarity measure,$s_{i,j} := \langle x_i - \bar{x}, x_j - \bar{x} \rangle$, and minimizes the squared average distortion:

$$argmin_{z_1,\ldots,z_n}\{\sum_{i,j=1}^{n}(s_{i,j} - \langle z_i - \bar{z}, z_j - \bar{z} \rangle)^2\}. \tag{7}$$

Least Squares A.k.a. *Kruskal-Shepard* starts with a Euclidean distance graph,$d_{i,j} = \|x_i - x_j\|_2$, and minimizes the squared average distortion.

$$argmin_{z_1,\ldots,z_n}\{\sum_{i\neq j}(d_{i,j} - \|z_i - z_j\|)^2\}. \tag{8}$$

Sammon Mapping Also known as *Sammon's stress*, starts with a Euclidean distance graph and aims at minimizing the average *proportion* of distortion:

$$argmin_{z_1,\ldots,z_n}\{\sum_{i\neq j}\frac{(d_{i,j} - \|z_i - z_j\|)^2}{d_{i,j}}\}. \tag{9}$$

Remark 8 (MDS and PCA)**.** MDS with "classical scaling" (a.k.a. Torgerson scaling) returns the exact same embedding as PCA with 2 PCs.

Remark 9 (MDS and Graph Drawing)**.** If the purposes of lowering the dimension is visualization, then the dimension of $\mathcal{M}$ will typically be 2. Embedding in 2 dimensional linear spaces is of great interest for visualization, and indeed there is a whole field called *Graph Drawing*[7], which focuses on these problems. For instance, the very popular *force embedding*[8] in the D3 java-script libraries stems from the Graph Drawing literature.

4.1.2 Non-Metric MDS

The above approaches to MDS start with the actual distances between points. If the actual distances are replaced with some monotonic function of these, then it is only the *ordering* of distances that drives the solution. This is known as *non-metric MDS*.

4.2 Local MDS

Example 5 (Non-Euclidean surface)**.** Consider data of coordinates on the globe. At short distances, constructing a dissimilarity graph with Euclidean distances will capture the true distance between points. At long distances, however, the Euclidean distances as grossly inappropriate. A more extreme example is coordinates on the brain's cerebral cortex. Being a highly folded surface, the Euclidean distance between points is far from the true geodesic distances along the cortex's surface[9].

[7] `https://en.wikipedia.org/wiki/Graph_drawing`

[8] `https://en.wikipedia.org/wiki/Force-directed_graph_drawing`

[9] Then again, it is possible that the true distances are the white matter fibers connecting going within the cortex, in which case, Euclidean distances are more appropriate than geodesic distances. We put that aside for now.

Local MDS is aimed at solving the case where we don't know how to properly measure distances. It is an algorithm that compounds both the construction of the dissimilarity graph, and the embedding. The solution of local MDS, as the name suggests, rests on the computation of *local* distances, where the Euclidean assumption may still be plausible, and then aggregate many such local distances, before calling upon regular MDS for the embedding.

Because local MDS ends with a regular MDS, it can be seen as a non-linear embedding into a linear $\mathcal{M}$.

4.3 Kernel Principal Component Analysis (kPCA)

Back to the height-weight problem in Example1: assume we want to construct a "bigness" score, that best separates between individuals, but we no longer constrain it to be a linear function of the height and weight. Adopting the variance maximization view of PCA, we could try to find the best separating score $g(x)$ by solving

$$argmax_g\{\mathbf{Cov}\left[g(\mathbf{x})\right]\} \tag{10}$$

where $g(x) : \mathbb{R}^p \mapsto \mathbb{R}^q$, maps an individual's p features to a q dimensional score in $\mathcal{M}$.

Without any constraints on $\mathcal{M}$, thus on g, we will overfit and\or not be able to compute g as optimization is done in a infinite dimensional space. We thus have two matters to attend: (i) We need to constrain $g(x)$. (ii) We need the problem to be computable. This is precisely the goal of kPCA.

If we choose the right g's, the solution of Eq.(10) may take a very simple form. The classes of such g's are known as Reproducing Kernel Hilbert Spaces (RKHS). At an intuitive level, RKHS's are subspaces of "the space of all functions", which: (i) Are defined via a *kernel function* we need to specify. Each kernel defines a different subspace. (ii) They have a very particular structure so that the optimal g is not linear in X, but is linear in some simple and known transformation of X– the kernel.

Remark 10 (kPCA and Manifold learning)**.** kPCA is seen by some as the "father" of all manifold learning algorithms. Algorithms such as IsoMap, LLE, LaplacianEigenmaps, were motivated as generalizations of kPCA.

Mathematics of kPCA [TODO]

4.4 Isometric Feature Mapping (Isomap)

Isomap, also known as *Principal Coordinate Analysis*, operates very similarly to local MDS, but with a different algorithm to compute the dissimilarity matrix.

4.5 Local Linear Embedding (LLE)

LLE is similar in spirit to Isomap and LocalMDS. It differs, however, in the way similarities are computed, and in the way embeddings are performed. Unlike localMDS, which computes local distances and ends with a *global embedding*, LLE computes local distances and performs *local linear embeddings*. The resulting approximating manifold $\mathcal{M}$, being the "stitching" of many linear spaces, is ultimately non linear.

4.6 Laplacian Eigenmaps

Just like LLE, Laplacian Eigenmaps is an algorithm to measure distances and embed local neighborhoods of the original X space.

4.7 Maximum Variance Unfolding

A different solution to the same problem a LLE.

4.8 Self Organizing Maps (SOM)

SOMs are special in our context in that the target $\mathcal{M}$, is not a bona-fide manifold. SOMs are typically designed for visualization, so they embed into a 2 dimensional space. The target space, $\mathcal{M}$, is a *polygon mesh*. Polygon meshes are easy to compute with, but hard to analyze mathematically. In particular, the "stitching" between polygons is a very irregular structure that does not satisfy the definition of a *manifold*.

4.9 Principal Curves and Surfaces

The algorithm iterates until it returns a curve of a surface. In the curve case, it will return a curve with the *self consistency* property. I.e., a curve with a path that is the average of all it's closest data points. Roughly speaking, one can think of this as a generalization of k-means from points to curves. Each such curve smoothly connects the k-means cluster centres. Using the same, slightly inaccurate depiction, a *principal surface* is manifold with dimension greater than 1, connecting these k-means cluster centres. In both cases, the output is a continuous parametrization of the curve or the surface.

Self Consisntecy

It is highly uncommon to approximate the data with manifolds with a dimension larger than 2, as typically the method is used for projecting the data before visualizing and\or clustering.

4.10 Auto Encoders

To present *auto-encoders*, a.k.a. *auto-associator* or *Diabolo network*, we restate the PCA space embedding formulation of Eq.(4):

$$argmin_{V\in\mathbb{R}^{q\times p},S\in\mathbb{R}^{n\times q}}\{\|X-SV\|_{Frob}\} \tag{11}$$

because of the invariance of the solution to rotations, and at the cost of interpretation, we may freely assume that $VV'=I$. If $q=p$ then at the optimum $X=SV$ so that $S=V'X$. We thus have:

$$argmin_{V\in\mathbb{R}^{q\times p}}\{\|X-XV'V\|_{Frob}\}. \tag{12}$$

This formulation of PCA problem, allows us to see that the single layer *auto-encodres*, assumes that X is some non-linear $g:\mathbb{R}^p\mapsto\mathbb{R}^p$ of scores with loadings W:

$$argmin_{W\in\mathbb{R}^{q\times p}}\{\|X-g(XW')W\|_{Frob}\}. \tag{13}$$

For the right choice of g, autoencoding can be understood as a *neural network* used to predict X latent scores (see Sec 2.4). Alternatively, as a purely-algorithmic counterpart fo non-linear FA and ICA.

4.11 t-SNE

Maaten and Hinton [2008]

4.12 Joint Diagonalization

Miettinen et al. [2017]

4.13 Non-Negative Matrix Factorization

[TODO]

4.14 Information Bottleneck

[TODO]

Remark 11 (Information Bottleneck and ICA)**.** [TODO]

4.15 Bibliographic notes

To read more on almost anything, see Friedman et al. [2001]. For a detailed review of ICA see Hyvärinen and Oja [2000]. For more on MDS Borg and Groenen [2005]. For manifold learning, including kPCA, Isomap, local MDS, Laplacian eigenmaps, etc, see Mohri et al. [2012].

5 Dealing with the high-dimension

As we previously mentioned (Sec.1.2), it is hard to accurately estimates directions of co-variability with little data. For some intuition, think of recovering a "bigness score" (Example 1) from only 3 individuals ($p = 2, n = 3$). Would you trust the recovered index?

In high dimensional problems, i.e., low signal-to-noise regimes, we will introduce some regularization, to reduce variance, as the cost of some bias. For generative models, regularization can always be introduced with Bayesian priors, say, some matrix-valued distribution as a prior on V. For purely algorithmic approaches (such as PCA), regularization has to be be hard-wired into the algorithm.

5.1 Simplified Component Technique LASSO (SCoTLASS)

SCoTLASS can be thought of as PCA, where sparse V are preferred as solutions. By "sparse" we mean that V has many zero entries. We favor sparse solutions both to deal with the very low SNR of high dimensional problems, but also for better interpretation of the PCs.

Think about it. Why is it easier to interpret the PCs when V has many zero entries?

The mathematics of SCoTLASS looks like the variance maximization formulation of PCA with an equality and an inequality constraint.

$$argmax_v\{v'(X'X)v \text{ such that } \|v\|_2 = 1, \|v\|_1 \leq t\} \tag{14}$$

Sadly, Eq.(14) is hard to solve numerically.

5.2 Sparse Principal Component Analysis (sPCA)

Motivated by the numerical difficulties of SCoTLASS, and starting from the linear-space approximation view of PCA (not variance maximization) the sPCA problem is defined as follows:

$$argmin_{V,\theta}\{\|X - \Theta V X\|_{Frob} + \lambda_2\|V\|_2 + \lambda_1\|V\|_1 \text{ such that } \|\Theta\|_2 = 1\}, \tag{15}$$

where V is the matrix of all loadings, $\|.\|_{Frob}$ the Frobenius matrix-norm, $\|.\|_2$ the l_2 matrix-norm, and $\|.\|_1$ the l_1 matrix-norm.

Remark 12 (sPCA and FA rotations)**.** sPCA can be seen as a particular type of rotation, that favors sparse V, since just like the FA problem, when $p > n$ then PCA is also non-identifiable.

5.3 Sparse kernel principal component analysis (skPCA)

If you liked the idea of sparse loadings (for interpretability, or SNR), and you liked the idea of embedding the data into non-linear, infinite dimensional RKHS, why not marry the ideas? See Tipping [2001].

5.4 Random Projections

You feel that the data is SOOOOO large that you don't have time to find embeddings that are optimal in any sense. You thus want to reduce the dimension of the data and don't care how good it is. How about simply multiplying X by a *random* matrix, projecting from $\mathbb{R}^p$ to $\mathbb{R}^q$, for $q < p$. It turns out, that this is not a bad strategy! Look for the Johnson-Lindenstrauss lemma, for example in Mohri et al. [2012].

6 Causal inference

[TODO] Review: Kalisch and Bühlmann [2014]. Relation to ICA: Peters et al. [2014].

References

I. Borg and P. J. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume Not sure about the edition. Probably the free web edition. Springer series in statistics Springer, Berlin, 2001.

R. Graham. Isometric embeddings of graphs. *Selected Topics in Graph Theory*, 3:133–150, 1988.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.

M. Kalisch and P. Bühlmann. Causal structure learning and inference: a selective review. *Quality Technology & Quantitative Management*, 11(1):3–21, 2014.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

J. Miettinen, K. Nordhausen, and S. Taskinen. Blind source separation based on joint diagonalization in r: The packages jade and bssasymp. *Journal of Statistical Software*, 76(1):1–31, 2017. ISSN 1548-7660. doi: 10.18637/jss.v076.i02. URL `https://www.jstatsoft.org/index.php/jss/article/view/v076i02`.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2012.

B. Nadler. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, pages 2791–2817, 2008.

K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

J. Peters, J. M. Mooij, D. Janzing, B. Schölkopf, et al. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(1):2009–2053, 2014.

M. E. Tipping. Sparse kernel principal component analysis. *Advances in neural information processing systems*, pages 633–639, 2001.