Wrapping up Some of the Techniques we Have Studied

Philippe B. Laval

KSU

Fall 2017
Introduction

We quickly review some of the techniques we have studied so far, working with data and discuss their strengths and weaknesses. We then discuss how some of these weaknesses might be addressed. The techniques discussed include:

- Fourier Transform
- SVD
- PCA
Fourier Transform

1. What are the applications of the Fourier transform?
2. How does it work?
3. What are the pros and cons of the Fourier transform?
4. How can we address these cons?
Fourier Transform: Applications

1. Signal/image processing.
2. Signal/image compression.
3. Remove noise from signal/image.
The Fourier transform maps the time domain into the frequency domain and back.

1. Convert a signal in the frequency domain.
2. Once in the frequency domain, we identify the frequencies corresponding to certain features and we process them.
3. Convert the signal back in the time domain.

For example, to compress a signal, 1) convert it to the frequency domain, 2) remove the frequencies which contribute very little to the signal. This means that now, the signal can be represented with fewer frequencies, hence it requires less space to represent this signal. To recreate the signal, we convert back the new signal with fewer frequencies to the time domain.
Fourier Transform: Pros and Cons

1. The Fourier transform has been around for a long time (early 1800’s).
2. Hence, it is well understood, has been studied extensively and a lot of theory and practice about it is known.
3. There exists very fast algorithms to implement it.

BUT

1. While the Fourier transform captures all the frequencies which occur in a signal, it does not capture when they occur.
Fourier Transform: Solutions for the Cons

- Windowed Fourier transform and the Gabor transform
  - Decompose the signal over the time domain into separate time frames
  - Find the Fourier transform for each frame.

- Wavelets
  - Also decompose the signal over the time domain into separate time frames
  - While for the Fourier transform the signal is decomposed in terms of sine and cosine, for wavelets, the signal is decomposed in terms of what is called a "mother wavelet" which is also scaled and translated. Which mother wavelet we select depends on what features of the signal we want to extract.
1. What are the applications of the SVD?
2. How does it work?
3. What are the pros and cons of the SVD?
4. How can we address these cons?
SVD: Applications

1. One of the most powerful tool in linear algebra.
2. Image compression
3. Computer vision
4. Used to solve least square problems
5. Base tools in many other applications such as PCA
6. Pseudo inverse of a matrix or the Moore-Penrose inverse.
7. Solution of linear systems for which the corresponding coefficient matrix is not square and/or not invertible.
Decomposes an $m \times n$ matrix $A$ into $A = U\Sigma V^T$ where $U$ is $m \times m$, $\Sigma$ is an $m \times n$ diagonal matrix containing the singular values of $A$ (the square root of the eigenvalues of $A^TA$) in decreasing order and $V$ has the corresponding eigenvectors of $A^TA$ as its columns.

The columns of $V$ are the directions which play an important role for the data represented by $A$. How important the role they play is indicated by the size of the corresponding singular values.

The directions for which the corresponding singular values are small can be ignored.
1. Works well if the spectrum of singular values contain few large entries.
2. This is a very useful tool, one used in many applications.

**BUT**

1. Will not work well if $\Sigma$ has full rank and/or if all the singular values of $A$ have a comparable size (why?)
2. The computation of $A^T A$ indicates that we are using the $\ell^2$ norm of a vector or a matrix. See PCA for more details of the implications of this.
See PCA
PCA

1. What are the applications of the PCA?
2. How does it work?
3. What are the pros and cons of the PCA?
4. How can we address these cons?
PCA: Applications

1. Image compression
2. Computer vision
3. Dimension reduction
PCA: How Does it Work?

1. Performs a change of variable to obtain a new set of uncorrelated variables. This is accomplished by computing the covariant matrix then perform a change of variable so that the new covariant matrix is a diagonal matrix.

2. The elements on the diagonal of the diagonal matrix indicate the importance of each of the new variable.

3. We can eliminate the ones which do not play a big role in representing the data.
PCA: Pros and Cons

- Very useful for dimension reduction, which is important for big data.

BUT

- Assume large variation in data and works well if only a few variables have large variations.
- Assume linearity of the data.
- The data can be represented with its mean and variance. It means the data has a Gaussian distribution
- This process is sensitive to noise and corrupted data because it uses the $\ell^2$ norm of vectors hence the noise is squared.
Regarding linearity of the data, a more general algorithm exists, it is called kernel PCA.

Regarding the problem with noise and the $\ell^2$ norm, another algorithm exists, it is called robust PCA and it uses the $\ell^1$ norm.

MATLAB has code implementing both kernel PCA and robust PCA. However, this code is not part of the main MATLAB distribution. It has to be downloaded.
Vector Norms

Definition

Given a vector space $V$ over a subfield $K$, a norm on $V$ is a function $\rho : V \rightarrow \mathbb{R}$ which satisfies the following properties for every $c \in K$ and $u, v \in V$:

1. $\rho (v) = 0 \iff v = 0$.
2. $\rho (u + v) \leq \rho (u) + \rho (v)$ (triangle inequality)
3. $\rho (cv) = |c| \rho (v)$.

Remark: Properties 1 and 2 imply that $\rho (v) \geq 0$ for any vector $v \in V$. This is sometimes added to the list of properties a norm must satisfy, though it is not necessary since it is implied by the first two properties.
Vector Norms

There exists many possible norms for vectors. We list a few. Let \( \mathbf{v} = (v_1, v_2, \ldots, v_n) \) be a vector.

1. The \( \ell^1 \) norm defined by \( \| \mathbf{v} \|_1 = \sum_{i=1}^{n} |v_i| \), implemented on MATLAB by \texttt{norm(v, 1)}. 

2. The \( \ell^2 \) norm defined by \( \| \mathbf{v} \|_2 = \| \mathbf{v} \| = \sqrt{\sum_{i=1}^{n} (v_i)^2} \), implemented on MATLAB by \texttt{norm(v, 2)} or simply \texttt{norm(v)}.

3. The \( \ell^p \) norm defined by \( \| \mathbf{v} \|_p = \left( \sum_{i=1}^{n} (v_i)^p \right)^{\frac{1}{p}} \), implemented on MATLAB by \texttt{norm(v, p)}.

4. The \( \ell^\infty \) norm defined by \( \| \mathbf{v} \|_\infty = \max_{i} |v_i| \), implemented on MATLAB by \texttt{norm(v, Inf)}. 

Philippe B. Laval (KSU)
Recall that only square matrices with non-zero determinant have an inverse.

The inverse is used for many things. In particular, it can be used to solve equations of the form $Ax = b$ where the solution is $x = A^{-1}b$.

What if $A$ is not a square matrix or not invertible?

This is where the pseudo inverse also known as the Moore-Penrose inverse comes into play.
Definition

Let $A$ be an $m \times n$ matrix. A pseudo inverse of $A$ is an $n \times m$ matrix $A^+$ which satisfies:

1. $AA^+ A = A$
2. $A^+ AA^+ = A^+$
3. $(AA^+)^* = AA^+$ ($AA^+$ is Hermitian)
4. $(A^+A)^* = A^+A$ ($A^+A$ is Hermitian)
1. For any matrix $A$, $A^+$ exists and is unique.
2. If $A$ is also invertible, then $A^+ = A^{-1}$.
3. $(A^+)^+ = A$
4. If $A = U\Sigma V^T$ then $A^+ = V\Sigma^+ U^T$ where $\Sigma^+$ is obtained from $\Sigma$ by taking the inverse of its diagonal entries then its transpose.
If $A$ is $n \times n$ and invertible, then a solution to $Ax = b$ is $x = A^{-1}b$.

What if $A$ is $m \times n$ with $m \neq n$ and/or $A$ does not have an inverse?

1. $Ax = b$ either will have no solution or if it has one, it may not be unique.
2. The solution is found by minimizing $\|Ax - b\|_2$
3. $x = A^+b$ is such a solution.
Summary

- All these techniques use a similar idea.
- We decompose some data into components.
- We identify the components with certain features and process them.
- We recreate new data from the new processed components.