Progress Report 1: Generative Models Using Probabilistic Principal Component Analysis

Group 9
Aashima Yuthika(1401071), Deep Patel(1401010), Deval Shah(1401060), Kirtan Modi(1401122), Yash Kotadia(1401114)
School of Engineering and Applied Sciences, Ahmedabad University

Subject: Machine Learning & Algorithms and Optimisation for Big Data
March 12, 2017

***Abstract*—The main aim of this project is to be able to build Generative models using the concept of Probabilistic Principal Component Analysis or PPCA. PPCA essentially tries to fill in the data voids using various methods and assumptions that are discussed more in detail later on in the report. These data voids are usually known as the latent (unobservable) variables. Some of the main reasons for using PPCA instead of PCA are: i) The variance-covariance matrix needs to be calculated, this can be very computation-intensive for large datasets with a high dimensions, ii) Outlying data observations can unduly affect the analysis. The Expectation Maximisation Algorithm is often used for estimating these latent variables. Robust PCA and PPCA used with EM are compared in our results for this report.**

I. Introduction

Generative models are being largely studied now by the Machine Learning Community and there are many researchers working on this problem as well. For this project we will be making use of the concept of PPCA to try and achieve this. PPCA is formulated within a maximum-likelihood framework, based on a specific form of Gaussian latent variable model. This leads to multiple mixture models than can combined as a probabilistic mixture, whose parameters can be determined using an EM algorithm. We have discussed both PPCA and EM briefly in the following sections.

II. PPCA - EM

PPCA can be used as a general Gaussian density model in addition to reducing dimensions. Maximum-likelihood estimates can also be computed for elements associated with principal components. Apart from these, PPCA Captures dominant correlations with few parameters. Hence, PPCA can be used to estimate what are known as latent variables.

A latent variable model relates a d-dimensional observation vector t to a corresponding q-dimensional vector of latent(unobserved) variables x. Latent variable model with linear relationship is given by:

$$t = Wx + \mu + \varepsilon \tag{1}$$

Here, w = $d \times q$ matrix that relates 2 sets of variables.
μ = non-zero mean.
ε = error or noise
Latent variables: x $\sim$ N(0, I).
Error (or noise): $\varepsilon \sim$ N(0, ψ).
Location term (mean): μ
The d x q matrix W relates the set of variables, while the parameter vector μ permits the model to have a non-zero mean.

The EM Algorithm basically consists of two steps - the 'E-Step' or the 'Expectation Step' that guesses a probability distribution over completions of missing data in the current model; and the 'M-Step' or the 'Maximisation Step' that re-estimates the model parameter using these completions. It is used in PPCA in the following manner:

- Initialise W, z
- Impute missing X from W and z
- Estimate z on all data based on the current W
- Repeat until convergence
- Estimate W on all data based on the current z

Note, that here we are assuming that we are not given a dataset with completely missing observations.

III. PPCA-EM vs Robust PCA

For Robust PCA, we are given a large data matrix M which can be decomposed as:

$$M = L_0 + S_0$$

Where, L_0 is a low rank matrix and S_0 is a sparse matrix. There is no knowledge of these two matrices. We now want to know if we can recover the low-rank and the sparse components of both of these matrices. Robust PCA is thus am optimisation technique with which L_0 and S_0 are being tried to be recovered from a grossly corrupted data matrix M. Whereas, PPCA is a generative model which probabilistically models latent variables and error from which we can generate our data vector t.

IV. Results

We have tried recovering two types of image matrices - i) with missing values, ii) with corrupted entries. A detailed table of the comparison between the performance of the 2 algorithms is attached with the submission. Note that these observations are for 100 iterations as after that the error remained almost constant but the time taken by the algorithm was more.

For an image with missing entries, the PPCA-EM Algorithm overall gave much less error than the Robust PCA Algorithm, although for a smaller window size (=8) it took more time, but as the windows size increased (=32), the time it took became less than the time that RPCA took.

For an image with corrupted entries, the PPCA-EM Algorithm and the Robust PCA Algorithm both gave almost the same amount of error. However, PPCA-EM had a much better performance in terms of the time complexity.

References

[1] Michael E. Tipping; Christopher M. Bishop, *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 61, No.3 (1999), 611-622.

[2] E.J.Candes, Xiaodong Li, Yi Ma, John Wright, *Robust Principal Component Analysis*, Journal of the ACM, Vol. 58, No. 3, Article 11, May 2011.

[3] https://people.cs.pitt.edu/m̃ilos/courses/cs3750-Fall2007/lectures/class17.pdf

[4] https://www.youtube.com/watch?v=ekEEG7t3sGI