Computational and Theoretical Analysis of Novel Dimensionality Reduction Algorithms in Data Mining

Brandon Guo*, Dr. Guangliang Chen†

*Monta Vista High School, †San Jose State University

INTRODUCTION

- Modern data mining is interested in understanding data in bulk.
- Past Developments: Easy linear regressions, neural networks
- Given a $m \times n$ dataframe, optimize a function f that best describes the data

THE PROBLEM: **What if n is large?**

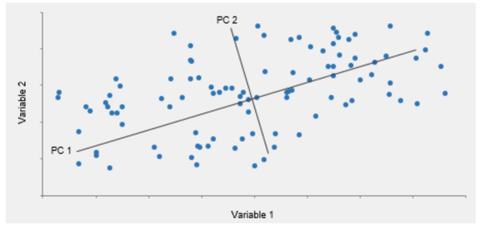
- The run time τ is given by $\tau \propto n^{-\frac{x}{2x+d}}$ [1].
- Many factors often leads to overfitting [2].

Goal: Given $m \times n$ dataframe, derive an optimal $m \times d$ dataframe with $d < n$ that captures important information but also alleviates size-related problems.

Techniques:

1. Principal Component Analysis (PCA)
2. Kernel Component Analysis (KCA)
3. Nonnegative Matrix Factorization (NMF)
4. Independent Component Analysis (ICA)

PRINCIPAL COMPONENT ANALYSIS (PCA)



Given data matrix X $\in \mathbb{R}^{m \times n}$, one can compute its covariance matrix

$$C = \frac{1}{n} X^T X$$

which has eigenvalues λ_i that satisfy the eigenproblem

$$Cv_i = \lambda_i v_i$$

corresponding with eigenvectors v_i represent the **principal components** of the data.

The eigenvectors $\{v_1, \dots, v_n\}$ are ordered by the amount of variance captured by each component. We claim that the largest eigenvalues λ_k corresponding with the v_k that captures the most variance in the data.

PCA chooses the d largest eigenvectors $\{v_1, \dots, v_d\}$ to create a basis $\mathcal{B}$ spanned by the d eigenvectors.

The reduced dataset is simply the projection of the data onto the new d-space. Specifically, the resultant dataframe X_f is simply:

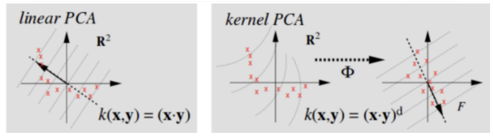
$$X_f = proj_{\mathcal{B}} X \in \mathbb{R}^{m \times d}$$

Value of d is selected by user based on how much variance to capture.

$d = 0$ captures no variance and $d \sim n$ captures almost all variance.

The optimal d varies between each dataset [3].

KERNEL COMPONENT ANALYSIS (KCA)



Instead of mapping onto linear bases, transformation into nonlinear bases, or **contours**.

We define kernel $K = k(\boldsymbol{x}, \boldsymbol{y})$ as a mapping of the standard coordinates to a more complicated function. Some examples of such functions:

Polynomial: $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{z})^2$

Gaussian: $K(\boldsymbol{x}, \boldsymbol{y}) = e^{-||x-y||/2\sigma^2}$

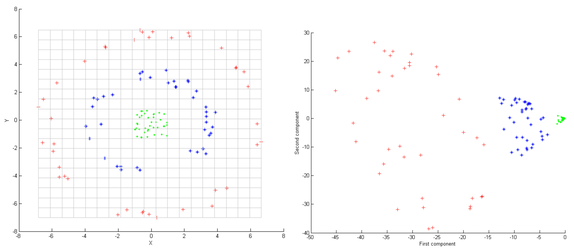
Sigmoid: $K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\boldsymbol{x}^T \boldsymbol{z})$

Then, this eigenproblem can be solved to find eigenvectors v_i:

$$Kv_i = \lambda_i v_i$$

and the kernel components $y_i(x)$ can be calculated.

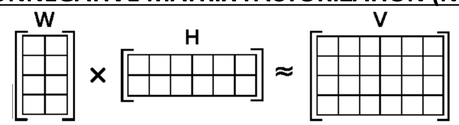
$$y_i(x) = \sum_{i=1}^{n} v_i K(\boldsymbol{x}, \boldsymbol{y})$$



The data on the left cannot be linearly decomposed effectively. A more effective method of reduction is possible via the kernel analysis with $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y} + 1)^2$.

The resulting data is univariate, as the response variable is explained wholly by one variable [4].

NONNEGATIVE MATRIX FACTORIZATION (NMF)



Precondition: dataframe has only nonnegative values.

Approximate two matrices $W \in \mathbb{R}^{m \times d}$ and $H \in \mathbb{R}^{d \times n}$ that multiply to the dataframe $V \in \mathbb{R}^{m \times n}$. Dimensional analysis suggests that the value of d can be anything.

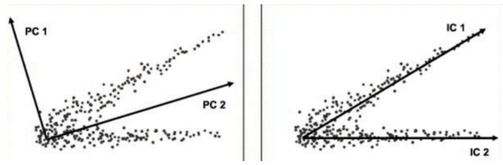
Not only makes dimensions more manageable, extracts easily interpretable factors for analysis.

This is NP-hard, but approximations make the work more manageable.

Columns of V are simply linear combinations of the estimated vectors (w, h).

To approximate W and H, one applies the *multiplicative update rule* [5], an a form of descent method that updates W and H until stability is achieved, using a cost function defined simply by $C = ||V - WH||$.

INDEPENDENT COMPONENT ANALYSIS (ICA)

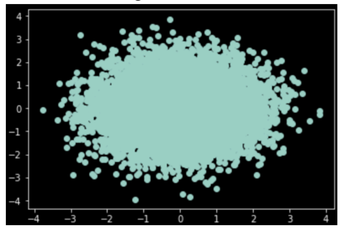


Inspired by the "cocktail party problem" and the existence of confounding variables in data

Goal: minimize mutual information between components

Opposite of PCA (maximize variation)

This is equivalent to minimizing correlation between the vectors.



To decrease mutual information, components must make data less Gaussian [6].

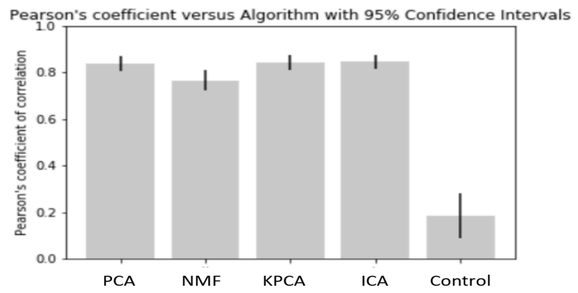
Seek non-zero kurtosis, a measure of Gaussianity given by $E(x^4) - 3$, where x is a vector — kurtosis is zero for normal data [7].

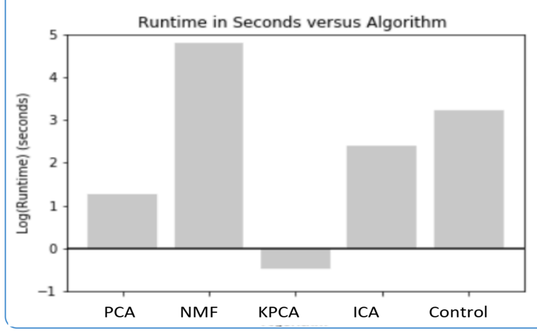
Converge at components that yield least Gaussian data [8].

EXPERIMENTATION AND RESULTS

Perform DR algorithms on large datasets and analyze the results for speed and power $(m, n > 5000)$.

Use a linear regression on the reduced datasets, tabulate average runtime as well R-value of the line of best fit model.





ON EFFICIENCY

Given runtime τ and r-value r we postulate that the efficiency η behaves:

$$\eta \propto r^j \tau^{-k}$$

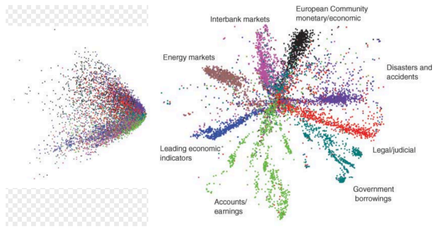
where j, k are positive reals.

Based on the data, why aren't PCA and KCA used all the time?

Latent Relationships – relationships between the data that cannot be directly observed through classical methods (regression testing).

However, through DR techniques, such patterns may emerge.

- ICA can expose trends in data points, particularly values that protrude out of the general cluster.
- NMF mandates that data points are estimated as purely linear combinations, which is important for understandability.



Buduma, 2017 [9]

CONCLUSION

Based on the figures, a few observations can be confidently made:

- Using reduction techniques for large datasets is beneficial
- PCA and KPCA are the quickest ways to consolidate data
- NMF and ICA suffer from lower accuracy, worse runtime, or both when compared to PCA/KPCA.

When considering the nature of the NMF and ICA algorithm, however, it is important to see that they each hold value in **exploring** the data, not merely modeling it.

Whether through component analysis or matrix factorization, DR algorithms each have their unique strengths to making a large dataset more manageable.

REFERENCES

[1] Blumer, Anselm & Ehrenfeucht, Andrzej & Haussler, David & K. Warmuth, Manfred. (1989). Learnability and the Vapnik-Chervonenkis Dimension. J. ACM. 36. 929-965. 10.1145/76359.76371.

[2] Subramanian, J., & Simon, R. (2013). Overfitting in prediction models – Is it a problem only in high dimensions? Contemporary Clinical Trials, 36(2), 636–641. https://doi.org/10.1016/j.cct.2013.06.011

[3] Shlens, J. (2005). A Tutorial on Principal Component Analysis. CoRR, abs/1404.1100.

[4] Python Software Foundation. Python Language Reference, version 3.0. Available at http://www.python.org.

[5] Keogh, E., & Mueen, A. (2017). Curse of Dimensionality. In Encyclopedia of Machine Learning and Data Mining (pp. 314–315). Springer US.

[6] Burred. (2014). Detailed derivation of multiplicative update rules for NMF.

[7] Burges, C. J. C. (2009). Dimension Reduction: A Guided Tour. Foundations and Trends® in Machine Learning, 2(4), 275–364. https://doi.org/10.1561/2200000002

[8] Foldiak, P & Young, M (1995). Sparse coding in the primate cortex. The Handbook of Brain Theory and Neural Networks, 895- 898. (MIT Press, Cambridge, MA).

ACKNOWLEDGEMENTS

Professor Guangliang Chen

Professor Yingze Zhang

JSM 2019