Factor Regression Combining Heterogeneous Sources of Information

Amrish S. Kapoor, Piyush Rai, Hal Daume III
School of Computing
University of Utah
Salt Lake City, UT 84102
amrish.kapoor@utah.edu, {piyush, hal}@cs.utah.edu

Abstract

We present a non-parametric Bayesian factor regression model that combines two heterogeneous sources of information: gene expression arrays and text from their corresponding PubMed abstracts. Our model approximates a pLSI style model and results in improved regression accuracy. We apply this model to gene-expression data analysis, but it is extendable to other problems exhibiting a similar heterogeneous multiplicity in sources of information, like financial analysis, weather prediction and others.

1 Introduction

The abundance of data available today for genomic study has necessitated the use of large-scale computational analysis to arrive at biologically meaningful results. Most approaches are centered around factor analysis and factor regression models [1, 2, 3], where the goal is to uncover latent factors associated with the observed data, and to make predictions using the uncovered factors instead of the actual data (usually to avoid overfitting in "large P, small N" settings [2]). However, these approaches fundamentally work with just one type of information: direct gene expression measurements, primarily from DNA microarrays. We extend these models to allow the inclusion of other sources of information, leading to a more robust and complete model. In particular, we address the problem of incorporating a dissimilar form of information - text - into a factor regression model designed for gene expression analysis.

2 Problem Setting

The standard factor analysis problem formulation assumes that the observed data is generated by a combination of latent factors, and hence can be represented as:

$\mathbf{X} = \mathbf{AF} + \mathbf{E}$, where $\mathbf{X}$ is the observed data matrix of size PxN, containing N samples of P features each. $\mathbf{A}$ is a factor loading matrix of size PxK and $\mathbf{F}$ is a factor matrix of size KxN. $\mathbf{E}$, also of size PxN accounts for the affect of the noise components. The factors, K, represent the underlying processes that combined in some way to produce the observations, $\mathbf{X}$.

In the case of gene analysis, the features are the genes, the samples are the microarray measurements and the ultimate aim is to be able to explain the observed expression profiles by some combinations of factors, which in this case correspond to gene pathways. Various methods have been used to arrive at such a separation, ranging from direct application of techniques such as ICA and PCA to more complex hierarchical factor models. We build on

the formulation developed in [3] by adding our textual analysis model. The model presented in [3] uses a non-parametric Bayesian factor model that accounts for uncertainty in the number of factors, and the relation between them using a sparse variant of the Indian Buffet Process, coupled with a hierarchical model over the factors using Kingman's coalescent.

The dataset we use is a subset of the Oncomine database depicting expression profiles of a range of genes under various clinical conditions reflecting cancer conditions. The dataset also includes DOIs for PubMed publications that we crawled and processed to extract a set of abstracts summarizing the experiments and analysis of each dataset. The entire dataset (115 experiments collated) consisted of P=2834 genes and N=8007 samples (microarray measurements) in a single matrix. However, for our experiments, we only used the subset of genes from this set that exhibited some significant variance, using a cutoff threshold of 0.5. This gave us an expression matrix of size 245 (genes/features) x 8007 (expression values/samples). Such a treatment of NLP models applied to gene expression data to obtain a combined factor model has not been attempted before to the best of the authors' knowledge, though [8] presents just a text-based analysis.

3 Combining Text Analysis with Factor Regression

Our formulation is based on an analysis of the abstract text data to extract the biologically meaningful terms. We use the Unified Medical Language System (UMLS) set of Concept Unique Identifiers (CUI) to represent the biologically relevant terms encountered in the abstract. The MetaMap algorithm was used to map raw text to UMLS concepts [7]. A matrix was built showing the frequency count of each such identifier for each experiment. Thus, we have a matrix, **W**, of size NxC where C is the number of unique CUIs found in all experiments, and N is the number of experiments that were analyzed. The entries in the matrix give the frequency counts for each CUI in each experiment. Thus, we have a word-frequency matrix on which we can perform a Latent Semantic Indexing (LSI) style analysis to obtain an abstract-topic relationship [4]. This is accomplished by subjecting the word-frequency matrix, **W**, to a singular value decomposition to obtain

$$\mathbf{W} = \mathbf{U}\,\mathbf{S}\,\mathbf{V},$$

where **U** can be seen as modeling an abstract-topic relationship, **V** represents a topic-word relationship, and **S** contains the actual singular values. Note that the very low singular value terms are ignored, thus leading to a reduction in dimensionality. In our setup, we used a set of N=115 experiments, with a total of C=2834 CUIs, and a complete SVD of the data showed that only 110 of the singular values were numerically significant. As a result, we computed only a 110-point SVD to obtain an abstract-topic matrix, **U** of size 115x110. We then use this matrix to add to the factor analysis representation modeled by [3]. The original data matrix is composed as a simple Genes x Samples matrix, with each row representing a gene and each column representing a time at which the expression was measured. Again, this data is available for multiple experiments (in our case, 115).

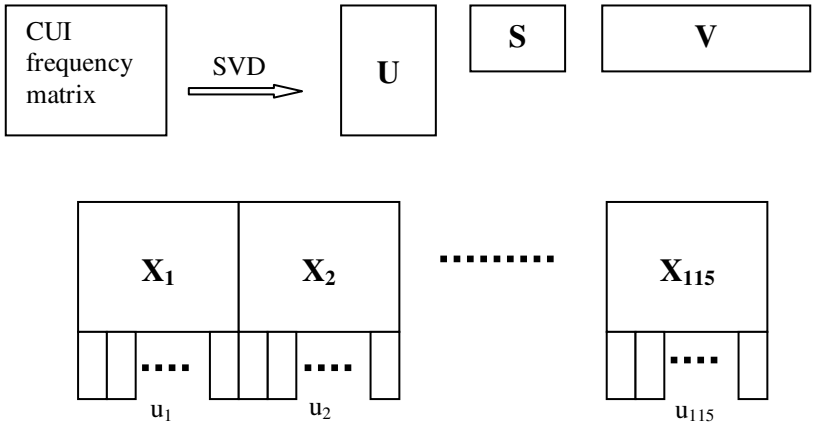

Figure 1: Construction of the data for the combined model

Since our abstract-topic matrix, **U**, is built at the experiment level, and not at a per-sample level, it must be incorporated in such a way as to reflect this fact. We achieved this by using each row of **U** as a column for each experiment, and repeating this column across all measurement time intervals for gene expression for that experiment. This process is repeated by attaching corresponding rows of **U** to the appropriate experiment data. The repeated columns attached to the gene expression profiles can be seen as "constant" across time, representing the textual information associated with this entire dataset. The final result is a large gene+abstract-topic matrix that is used as the input to the factor regression model. This construction is shown in Figure 1. The coalescent version of the infinite hierarchical regression model [3] is then used on this consolidated data matrix to obtain the final factor regression results. This formulation can also be seen as an approximation to the pLSI graphical model [6] - by noting the probabilistic framework that it is subjected to. The graphical representation is shown in Figure 2.

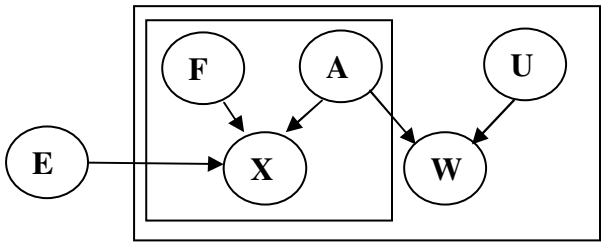

Figure 2: Graphical model of the pLSI style approximation combined with the infinite hierarchical factor regression model of [2]

While this is not a true pLSI model since the probabilities are not used at the time of the abstract-topic matrix computation, it can be thought of as an approximation of one in the context of this complete gene regression formulation. pLSI maximizes a probability modeling term, whereas vanilla LSA minimizes a Frobenius norm on reconstructions.

4 Results

The results obtained using the combined text and expression-profile formulation are shown here. We used the coalescent version of the infinite hierarchical factor regression model [3]. Figure 3 shows the mean reconstruction error curves for the simple expression-profile based regression model in blue dashed lines and those of the combined model in red solid lines for the two best runs of these methods. The reconstruction error was calculated as the difference between the original gene expression profiles and those obtained from the factor model. The text part of the model was used only in the reconstruction, not the error calculation.

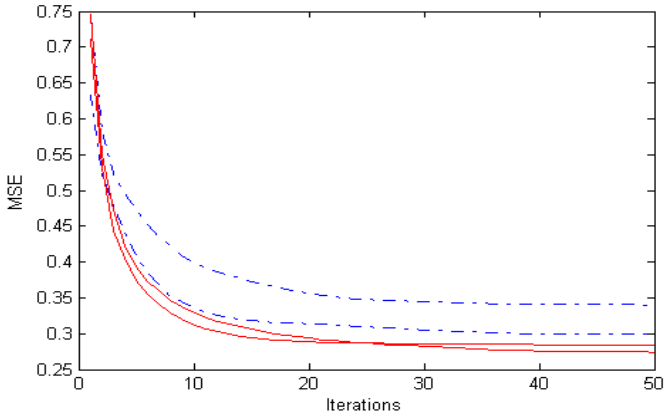

Figure 3: Plot of the MSE's of the two best runs using the original factor regression model (blue dashed) and the model augmented with text data (red solid)

Since inference is via MCMC, repeated runs expectedly show some deviations, but the combined model consistently seems to result in final values that are close to or better than the corresponding values for the original system. On an average, repeated runs show that there is a range of 5-9% improvement obtained over the simple model. This is a non-trivial gain considering that only the text data from the abstract was used. Since there were only 110 numerically significant SVDs, this was used as the size of the **U** matrix. It can also be seen that increasing the number of singular values up to 110 shows a gradual improvement in regression performance. It can be expected that increasing the number past 110 should not improve performance further (maybe even deteriorate it a little) since the remaining dimensions represent no useful information, and are more likely to be noise. This was also verified during our experiments[1].

5 Further Work

Our results show reasonable performance gains in combining various sources of information in the factor regression model. The accuracy could possibly be increased further if more textual data were available at a finer granularity (per gene level). CUIs in UMLS also provide an oncology string that could be exploited. An LDA-style topic model, where each pathway represents one topic could also be attempted. This methodology, or its variants, can also find application in other areas where heterogeneity of information sources is even more common, such as financial forecasting and weather modeling. Possible future work includes incorporation of other sources of information, such as other experiments referring to data from this set. Heterogeneity in the sources of information also occurs at the level of granularity - as in our case where the abstract text corresponds to a *set* of gene expression profiles rather than each one individually. Accounting for this in a clean way also remains an important open question.

6 Summary

We have shown a simple and effective way of using additional textual sources of information within a factor regression model, and demonstrated its applicability in the gene analysis problem. We also showed that this is an approximation of a pLSI-style text model attached to factor regression. We further postulated that this method should perform even better in the presence of increased textual information, and should find broader application in other fields such as finance learning

References

[1] Pournara, I. & Wernisch, L. (2007) Factor Analysis for Gene Regulatory Networks and Transcription Factor Activity Profiles, *BMC Bioinformatics*, vol. 8 article 61.

[2] West, M. (2003) Beyesian Factor Regression Models in the "Large p, Small n" paradigm, *Bayesian Statistics* **7**: 723-732.

[3] Rai, P. & Daume H. (2008) The Infinite Hierarchical Factor Regression Model, *NIPS 2008*.

[4] Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W. & Harshman, R.A. (1990) Indexing by latent semantic analysis, *Journal of the Society for Information Science*, 41(6), 391-407.

[5] Hyvarinen, A. (1999) Fast and Robust Fixed-Point Algorithms for Independent Component Analysis, *IEEE Transactions on Neural Networks*, 10(3):626-634.

[6] Hofmann, T. (1999) Probabilistic latent semantic indexing, *Proceedings of SIGIR-99*, 35-44.

[7] Aronson, A. R. (2001) Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program, *Proceedings of the Annual AMIA Symposium*, 17-21.

[8] Butte, A.J. & Kohane, I.S. (2006) Creation and Implications of a Phenome-Genome Network, *Nature Biotechnology* **24**, 55-62.

[1] In addition to the SVD model of text data, we also attempted to use an ICA [5] based model to estimate the independent components, instead of abstract-topic modeling. This proved to be at best as good as the SVD method, with no further improvement.