Supervised Principal Components for Classification

An Odyssey

Collin Nolte

September 21, 2018

Table of Contents

Similarity to Sparse PCA

Sparse PCA can be thought of as penalized regression of loadings onto $X_{n \times p} = UDV^T$

- UD represent the (scaled) principal components
- V^T, the loadings of X, is orthonormal, where $VV^T = I$, and each column is associated with a feature in X
- Post multiplying each side by V, we get

$$XV = UD = P$$

- We then choose our loadings $\tilde{v}_i$ for each column of p_i separately such that $\tilde{v}_i$ is subject to the constraints

$$\min_{V} \frac{1}{N} \sum_{i=1}^{N} L(p_i, v_i^T x_i) + \lambda[(1-\alpha)||v_i||_2^2 + \alpha||v_i||_1]$$

Similarity II

- glmnet style regression works best, as traditional Lasso limits total features to n, due to L_1 penalty
- This will create a number of 0 loadings, resulting in sparse principal component matrix
- This construction is "unsupervised", in that the final selection of $\tilde{V}$
- Once $\tilde{V}$ has been determined, we can perform regression in the standard way, using our sparse principal components

$$Y = \beta_0 + \sum_{m=1}^{M} \beta_m \tilde{P}_m + \epsilon$$

Latent Model Assumption

- We imagine a situation in which Y is a linear function of some latent variable U, where

$$Y = \beta_0 + \sum_{m=1}^{M} \beta_m U_m + \epsilon$$

- Further, suppose each feature of X, say, X_j, captures some portion of this latent feature, so that

$$X_j = a_{0j} + \sum_{m=1}^{M} \alpha_{1jm} U_m + \epsilon_j$$

- We reconsider sparse PCA, but seek to select those X_j which best capture the latent variable U

Method

- Represent each X_j as the linear combination of it's principal components (UD), with coefficients $\alpha_{1jm} = V_{[j,m]}$
- We now go about selecting loadings V^*, not so that we retain the structure of X, but so that we capture information related to our latent variable U
- If **s** represents the standardized regression coefficient measuring the univariate effect of each feature of X on Y,

$$s_j = \frac{x_j^T y}{||x_j||}$$

we seek a collection of features C_θ such that $|s_j| > \theta$

Dataset Summary

- Our analysis consists of three independent datasets collected from the National Center for Biotechnology Information (NCBI)
- Common genes were selected amongst the three datasets, and 40 percent were removed at random for memory constraint issues (13325 total)
- The two primary datasets contain gene expressions from heart (313) and liver (77) tissues, without outcomes being heart failure and Type II diabetes
- The third dataset (24) has outcomes related to both heart disease and diabetes, and is used as an additional out of sample measure of prediction performance

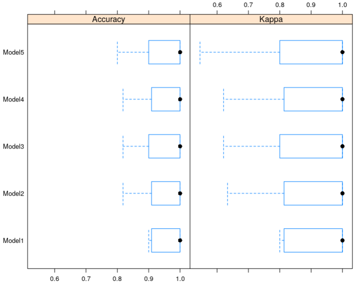
Models Used

For each dataset, the following models were built and considered in caret using 10 fold cross validation

- Supervised PCA with threshold θ as a tuning parameter (fit with glm and lda)
- glmnet with $\alpha = 1/\epsilon$, and tuning parameter λ (sparse PCA)
- AdaBoost.M1 with parameters tree and tree depth
- Partial least squares with number of components as tuning parameters

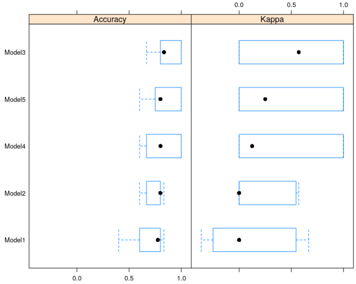
Supervise PCA Performance - Heart

- Model 5 is AdaBoost
- Model 4 is partial least squares
- Model 3 is glmnet
- Models 1 and 2 represent supervised PCA with glm and lda, respectively

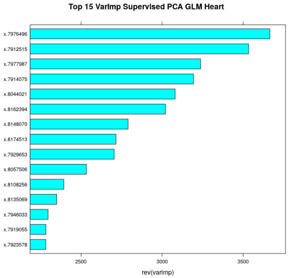


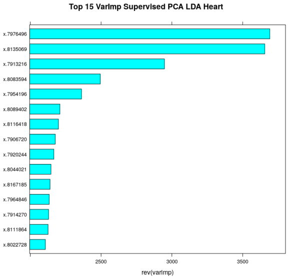
Supervise PCA Performance - Liver

- Model 5 is AdaBoost
- Model 4 is partial least squares
- Model 3 is glmnet
- Models 1 and 2 represent supervised PCA with glm and lda, respectively



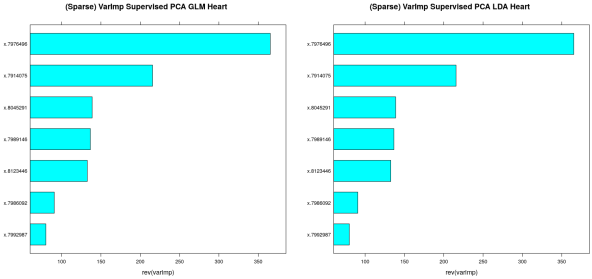
Variable Importance Heart Data





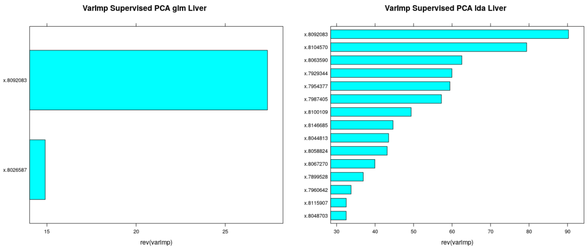
- Define $varImp(x_j) = \sum_{i=1}^{m} < x_j, u_{\theta,i} >$
- GLM retained 3155 out of 13325 genes, and LDA retained 1988 genes
- GLM retained each of the 1988 retained in best LDA model

Focus on Sparsity - Heart



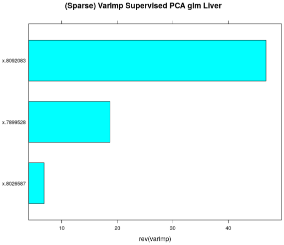
- In most sparse matrices, 7 genes are retained in each

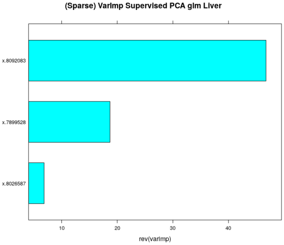
Variable Importance Liver Data



- Define $varImp(x_j) = \sum_{i=1}^{m} < x_j, u_{\theta,i} >$
- GLM retained 2 out of 13325 genes, and LDA retained 15 genes
- LDA retained both genes from GLM

Focus on Sparsity - Liver





- In most sparse matrices, 7 genes are retained in each

Final Model Summaries - Heart

	GLM in-sample	GLM out-sample	LDA in-sample	LDA out-sample
Accuracy	0.932	0.942	0.942	0.916
Kappa	0.862	0.779	0.881	0.779

Table: Best fitting models from Sparse PCA (3155 and 1988 genes)

	GLM in-sample	GLM out-sample	LDA in-sample	LDA out-sample
Accuracy	0.952	0.916	0.942	1.00
Kappa	0.902	0.780	0.881	1.00

Table: Sparsest Models using Sparse PCA (7 genes)

Final Model Summaries - Liver

	GLM in-sample	GLM out-sample	LDA in-sample	LDA out-sample
Accuracy	0.760	0.708	0.800	0.760
Kappa	0.667	0.030	0.000	0.030

Table: Best fitting models from Sparse PCA (2 and 15 genes)

	GLM in-sample	GLM out-sample	LDA in-sample	LDA out-sample
Accuracy	0.760	0.666	0.760	0.666
Kappa	0.342	0.000	0.342	0.030

Table: Sparsest Models using Sparse PCA (3 genes)

Final Summaries

- By comparison, glmnet retained 17 and 26 genes in the heart and liver.

	glmnet Heart in-sample	glmnet Heart out-sample	glmnet Liver in-sample	glmnet Liver out-sample
Accuracy	0.966	1.00	0.920	0.292
Kappa	0.931	1.00	0.781	0.000

Table: glmnet in Heart and Liver (17 and 16 samples, respectively)

Conclusions

- Supervised PCA achieves high sparsity in the loadings of X while retaining structure of latent variable U
- Performs better, relative to others, with larger sample sizes
- Out of sample prediction is also superior in some cases
- Remarkably, retains a set of genes entirely independent from those selected in glmnet.

Sources

- R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.
- Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan and Tyler Hunt. (2018). caret: Classification and Regression Training. R package version 6.0-79. https://CRAN.R-project.org/package=caret
- Max Kuhn and Hadley Wickham (2018). recipes: Preprocessing Tools to Create Design Matrices. R package version 0.1.2. https://CRAN.R-project.org/package=recipes
- BIOS 6720 course notes
- Matt Dowle and Arun Srinivasan (2017). data.table: Extension of ‘data.frame‘. R package version 1.10.4-3. https://CRAN.R-project.org/package=data.table

Sources cont.

- Hadley Wickham (2018). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.3.0. https://CRAN.R-project.org/package=stringr
- Orchestrating high-throughput genomic analysis with Bioconductor. W. Huber, V.J. Carey, R. Gentleman, ..., M. Morgan Nature Methods, 2015:12, 115.
- Davis, S. and Meltzer, P. S. GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. Bioinformatics, 2007, 14, 1846-1847
- "Elements of Statistical Learning", Trevor Hastie, Robert Tibshirani, Jerome Friedman.
- Bair, Eric, et al. "Prediction by Supervised Principal Components." Journal of the American Statistical Association, vol. 101, no. 473, 2006, pp. 119–137., doi:10.1198/016214505000000628.