Collaborative Regression

Samuel M. Gross[*]   Robert Tibshirani[†]

Departments of Statistics, and Health Research & Policy, Stanford University

Abstract

We consider the scenario where one observes an outcome variable and sets of features from multiple assays, all measured on the same set of samples. One approach that has been proposed for dealing with this type of data is "sparse multiple canonical correlation analysis" (sparse mCCA). All of the current sparse mCCA techniques are biconvex and thus have no guarantees about reaching a global optimum. We propose a method for performing sparse supervised canonical correlation analysis (sparse sCCA), a specific case of sparse mCCA when one of the datasets is a vector. Our proposal for sparse sCCA is convex and thus does not face the same difficulties as the other methods. We derive efficient algorithms for this problem, and illustrate their use on simulated and real data.

1 Introduction

The problem of combining data from multiple assays is an important topic in modern biostatistics. For many studies, the researchers have more data than they know how to handle. For example, a researcher studying cancer outcomes may have both gene expression and copy number data for a set of patients. Should that researcher use both types of predictors in their analysis? Should any care be given to distinguish the fact that these predictors are coming from different assays and may have differing meanings? If the researcher needs to make future predictions based on only gene expression, is there a way that having copy number data in a training set can help those future predictions? All of these are important questions that are still up for debate.

In this paper we propose a method for this problem called "Collaborative Regression", a form of sparse supervised canonical correlation analysis. In Section 2 we define Collaborative Regression (CollRe) and characterize its solution. This involves explicit closed form solutions for the unpenalized algorithm, as well as a discussion of some useful convex penalties that can be applied. Then, in Section 3 we explore the possibility of using CollRe in a prediction framework. While this may seem like an intuitive use case, simulations suggest that CollRe is not able to improve prediction error even over methods that do not take advantage of the secondary dataset.

We look at using CollRe in a sparse sCCA framework in Section 4, including a simulation study where we compare CollRe to one of the leading competitors. We show how the penalized version can be applied to a real biological dataset in Section 5. Finally, in section 6 we explore how to efficiently solve the convex optimization problem given by the penalized form of the algorithm.

2 Collaborative Regression

Collaborative Regression is a tool designed for the scenario where there are groups of covariates that can be naturally partitioned and a response variable. Let us assume that we have observed n

---

[*]email:smgross@stanford.edu

instances of $p_x + p_z$ covariates and a response. We can partition the covariates into two matrices, X and Z, that are $n \times p_x$ and $n \times p_z$ respectively. The response values are stored in a vector, $\boldsymbol{y}$, of length n. Then Collaborative Regression finds the $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{z}}$ that minimize the following objective function:

$$J(\theta_x, \theta_z) = \frac{b_{xy}}{2}\|\boldsymbol{y} - X\boldsymbol{\theta}_{\boldsymbol{x}}\|^2 + \frac{b_{zy}}{2}\|\boldsymbol{y} - Z\boldsymbol{\theta}_{\boldsymbol{z}}\|^2 + \frac{b_{xz}}{2}\|X\boldsymbol{\theta}_{\boldsymbol{x}} - Z\boldsymbol{\theta}_{\boldsymbol{z}}\|^2 \tag{1}$$

This objective function seems natural for the multiple dataset situation. Basically, it says that we want to make predictions of $\boldsymbol{y}$ based on X or Z, but we will penalize ourselves based on how different the predictions are. Essentially, the goal is to uncover a signal that is common to X, Z, and $\boldsymbol{y}$.

Consider trying to maximize the objective function (1). It is easy to show using calculus that the optimal solution, $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{z}}$ will satisfy the following First Order Conditions:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{x}} = \frac{1}{b_{xy} + b_{xz}}(X^T X)^{-1} X^T (b_{xy}\boldsymbol{y} + b_{xz}Z\hat{\boldsymbol{\theta}}_{\boldsymbol{z}}) \tag{2}$$

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{z}} = \frac{1}{b_{zy} + b_{xz}}(Z^T Z)^{-1} Z^T (b_{zy}\boldsymbol{y} + b_{xz}X\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}). \tag{3}$$

By substituting for $\hat{\boldsymbol{\theta}}_{\boldsymbol{z}}$ and solving, we can find a closed form solution for $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{x}} = \left(I - \frac{b_{xz}^2}{(b_{xy} + b_{xz})(b_{zy} + b_{xz})}(X^T X)^{-1} X^T Z(Z^T Z)^{-1} Z^T X\right)^{-1}$$
$$\left(\frac{b_{xy}}{b_{xy} + b_{xz}}(X^T X)^{-1} X^T \boldsymbol{y} + \frac{b_{xy}b_{zy}}{(b_{xy} + b_{xz})(b_{zy} + b_{xz})}(X^T X)^{-1} X^T Z(Z^T Z)^{-1} Z^T \boldsymbol{y}\right) \tag{4}$$

In the above we have assumed that $X^T X$ and $Z^T Z$ are non-singular. Assuming they are, and none of the parameters are zero, then that guarantees the invertibility of $\left(I - \frac{b_{xz}^2}{(b_{xy} + b_{xz})(b_{zy} + b_{xz})}(X^T X)^{-1} X^T Z(Z^T Z)^{-1} Z^T X\right)$. Note that $X^T X$ and $Z^T Z$ will always be nonsingular in the classical case where $\max(p_x, p_z) < n$.

2.1 Infinite Series Solution

Another way to characterize the optimal solution to the objective function (1) is as an infinite series. Instead of solving for $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$ after substituting, consider instead what would happen if we just continued substituting for $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$ or $\hat{\boldsymbol{\theta}}_{\boldsymbol{z}}$ on the RHS. Then, we get an infinite series representation of $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$. Let $P_X = X(X^T X)^{-1} X^T$ be the matrix that performs orthogonal projection onto the column space of X (and let P_Z be defined similarly). Then we can also write $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$ as:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{x}} = \frac{b_{xy}}{b_{xy} + b_{xz}}(X^T X)^{-1} X^T \boldsymbol{y} + \frac{b_{xz}}{b_{xy} + b_{xz}}\frac{b_{zy}}{b_{zy} + b_{xz}}(X^T X)^{-1} X^T P_Z \boldsymbol{y}$$
$$+ \frac{b_{xz}}{b_{xy} + b_{xz}}\frac{b_{xz}}{b_{zy} + b_{xz}}\frac{b_{xy}}{b_{xy} + b_{xz}}(X^T X)^{-1} X^T P_Z P_X \boldsymbol{y}$$
$$+ \frac{b_{xz}}{b_{xy} + b_{xz}}\frac{b_{xz}}{b_{zy} + b_{xz}}\frac{b_{xz}}{b_{xy} + b_{xz}}\frac{b_{zy}}{b_{zy} + b_{xz}}(X^T X)^{-1} X^T P_Z P_X P_Z \boldsymbol{y} \ldots \tag{5}$$

2

If we let

$$
w_i = \begin{cases} \frac{b_{xy}}{b_{xy}+b_{xz}} \left(\frac{b_{xz}}{b_{xy}+b_{xz}} \frac{b_{xz}}{b_{zy}+b_{xz}} \right)^i & \text{if } i \text{ is even} \\ \frac{b_{xz}}{b_{xy}+b_{xz}} \frac{b_{zy}}{b_{zy}+b_{xz}} \left(\frac{b_{xz}}{b_{xy}+b_{xz}} \frac{b_{xz}}{b_{zy}+b_{xz}} \right)^i & \text{if } i \text{ is odd} \end{cases}
$$

$$
\boldsymbol{y}^{(i)} = \begin{cases} (P_Z P_X)^i \boldsymbol{y} & \text{if } i \text{ is even} \\ (P_Z P_X)^i P_Z \boldsymbol{y} & \text{if } i \text{ is odd} \end{cases}
$$

Then

$$
\hat{\boldsymbol{\theta}_x} = (X^T X)^{-1} X^T \sum_{i=0}^{\infty} w_i \boldsymbol{y}^{(i)} \tag{6}
$$

Looking at the infinite expansion can help build some understanding of what CollRe actually does. We note that $\sum w_i = 1$, so essentially CollRe is equivalent to regressing X on the weighted average of the $\boldsymbol{y}^{(i)}$'s. Those $\boldsymbol{y}^{(i)}$'s trace out the path of successive projections onto the column space of X and Z. As the column spaces of X and Z are affine, it is known from Projection onto Convex sets that the sequence will converge to the projection of $\boldsymbol{y}$ onto the intersection of those two spaces. In the case where the columns of X and Z are linearly independent, $\boldsymbol{y}^{(i)}$ will eventually converge to 0. Thus, CollRe is basically shrinking $\boldsymbol{y}$ towards the part that can be explained by both X and Z.

Additionally, we get some picture as to how the parameters $\{b_{xy}, b_{zy}, b_{xz}\}$ affect the solution. $\frac{b_{xy}}{b_{xy}+b_{xz}}$ acts in large part to control the amount of shrinkage imposed on $\hat{\boldsymbol{\theta}_x}$, while $\frac{b_{zy}}{b_{zy}+b_{xz}}$ does the same for $\hat{\boldsymbol{\theta}_z}$.

2.2 Penalized Collaborative Regression

One nice aspect of the objective function (1) is that it is convex. This means that the problem can still be easily solved through convex optimization techniques if we add convex penalty functions to the objective. Thus, we can define Penalized Collaborative Regression (pCollRe) as finding the minimizer of the following objective:

$$
F(\theta_x, \theta_z) = \frac{b_{xy}}{2} \|\boldsymbol{y} - X\boldsymbol{\theta_x}\|^2 + \frac{b_{zy}}{2} \|\boldsymbol{y} - Z\boldsymbol{\theta_z}\|^2 + \frac{b_{xz}}{2} \|X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}\|^2 + P^x(\boldsymbol{\theta_x}) + P^z(\boldsymbol{\theta_z}) \tag{7}
$$

where $P^x(\boldsymbol{\theta_x})$ and $P^z(\boldsymbol{\theta_z})$ are convex penalty functions. Note that some of the convex penalties that may warrant use include:

- The Lasso: $P^x(\boldsymbol{\theta_x})$ is an ℓ_1 penalty on $\boldsymbol{\theta_x}$, namely $P^x(\boldsymbol{\theta_x}) = \lambda_x \|\boldsymbol{\theta_x}\|_1$. The lasso penalty is known to introduce sparsity into $\boldsymbol{\theta_x}$ for sufficiently high values of λ_x.

- Ridge: $P^x(\boldsymbol{\theta_x})$ is a (squared ℓ_2 penalty on $\boldsymbol{\theta_x}$, namely $P^x(\boldsymbol{\theta_x}) = \lambda_x \|\boldsymbol{\theta_x}\|_2^2$. Ridge penalties help to smooth the estimate of $X^T X$ to ensure non-singularity. This can be especially important in the high dimensional case where $X^T X$ is known to be singular.

- The Fused Lasso: $P^x(\boldsymbol{\theta_x}) = \sum_{i=2}^{i=p_x} \lambda_x |(\boldsymbol{\theta_x})_i - (\boldsymbol{\theta_x})_{i-1}|$. The fused lasso will help to ensure that $\boldsymbol{\theta_x}$ is smooth. This can be helpful if there is reason to believe that the predictors can be sorted in a meaningful manner (as with copy number data).

In addition to the convex penalties above, situations may also call for linear combinations of those penalties. For example, the lasso and ridge penalties are often combined to find sparse coefficients for predictors that are highly correlated. The lasso and fused lasso are often combined to find sparse and smooth coefficient vectors. In Section 6 we discuss solving pCollRe efficiently in the case where the penalty terms are asso penalties.

3 Using CollRe for Prediction

One potentially appealing use of CollRe where we want to make predictions of $\boldsymbol{y}$ for future cases where you will only have the variables in X available, and Z is only be available for a training set. Can the information contained in Z be used to help identify the correct direction in X? There are many practical situations in which this framework might be useful. For example, maybe it is much more costly to gather data with a lower amount of noise. Alternatively, it could be that some data is not accessible until after the fact; autopsy results may be very helpful in identifying different types of brain tumors, but it is hard to use that information to help current patients.

CollRe seems like it provides a natural way in which to perform a regression with additional variables present only in the training set. Basically, it is saying that we want our future predictions to agree with what we would have predicted given Z. In this framework, CollRe is similar to "preconditioning" as defined by Paul and others (2008) [6]. Instead of preconditioning on Z and then fitting the regression, we are simultaneously doing the preconditioning and fitting.

Looking at the infinite series solution in Section 2.1 it is clear that performing CollRe is similar to doing ordinary regression after shrinking y. If that shrinkage on y is done in such a way that it reduces noise, we may ultimately expect ourselves to do better in estimating the correct $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$. We investigate this next.

3.1 Simulated Factor Model Example

We decided to generate data from a factor model to test CollRe. A factor model seems natural for this problem, and is a simply way to create correlations between X, Z, and $\boldsymbol{y}$. Another reason the factor model was appealing is because it is relatively easy to analyze, and given $\hat{\boldsymbol{\theta}}_{\boldsymbol{x}}$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{z}}$ it is easy to compute statistics like the expected prediction error or the correlations between linear combinations of the variables. More concretely, given values for parameters $n, p_x, p_z, p_u, s_u, s_x, s_z$, and s_y, we generate data according to the following method:

1. $\boldsymbol{v_y} \in \mathcal{R}^{p_u}$ distributed MVN($0, I_{p_u}$)

2. $\boldsymbol{v_j^x} \in \mathcal{R}^{p_x}$ distributed iid MVN($0, I_{p_x}$) for $j = 1, \ldots, p_u$

3. $V_x = [\boldsymbol{v_1^x}, \ldots, \boldsymbol{v_{p_u}^x}]$

4. $\boldsymbol{v_j^z} \in \mathcal{R}^{p_z}$ distributed iid MVN($0, I_{p_z}$) for $j = 1, \ldots, p_u$

5. $V_z = [\boldsymbol{v_1^z}, \ldots, \boldsymbol{v_{p_u}^z}]$

6. For $i = 1, \ldots, n$:

    (a) $\boldsymbol{u_i} \in \mathcal{R}^{p_u}$ distributed iid MVN($0, s_u^2 I_{p_u}$)

    (b) $y_i = \boldsymbol{v_y^T} \boldsymbol{u_i} + \epsilon_i^y$ with ϵ_i^y distributed N($0, s_y^2$)

    (c) $\boldsymbol{x_i} = V_x \boldsymbol{u_i} + \boldsymbol{\epsilon_x^i}$ with $\boldsymbol{\epsilon_x^i}$ distributed MVN($0, s_x^2 I_{p_x}$)

    (d) $\boldsymbol{z_i} = V_z \boldsymbol{u_i} + \boldsymbol{\epsilon_z^i}$ with $\boldsymbol{\epsilon_z^i}$ distributed MVN($0, s_z^2 I_{p_z}$)

7. $X = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}]^T, Z = [\boldsymbol{z_1}, \ldots, \boldsymbol{z_n}]^T$, and $\boldsymbol{y} = [y_1, \ldots, y_n]^T$

Thus, steps 1-5 generate the factors ($V = [V_X; V_Z; \boldsymbol{v_y}]$) and step 6 generates the loadings (u_i) and noise.

In order to test the performance of CollRe in doing prediction, we generated a set of factors from the above model with $n = 50, p_x = p_z = 10, p_u = 3, s_u = s_x = s_z = s_y = 1$. Then, for each of 80 repetitions, we generated loadings and noise before fitting a range of models. CollRe was fit with $b_{xy} = b_{zy} = 1$ and a variety of values of b_{xz}. Additionally, at each level of b_{xz} we fit
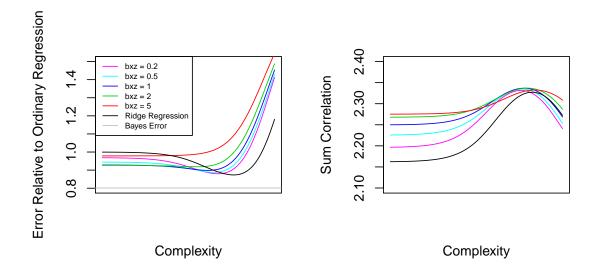
Figure 1: *Results of a simulation study to test the effectiveness of CollRe with ℓ_2 penalty in a prediction framework. Here, the points all the way to the left correspond to no ℓ_2 penalty and the ℓ_2 penalty increases (simpler models) as we move right along the x-axis. The first plot shows prediction error for making future predictions based on X only. The second plot shows the theoretical sum correlation. Values have been averaged over 80 repetitions. As we can see, while CollRe outperforms ordinary regression with no penalty in terms of prediction error (the far left of the first plot), Ridge Regression achieves a lower minimum. The second plot helps illuminate the reason; CollRe does a better job of maximizing the sum correlation, so it is sacrificing some of the correlation between $\boldsymbol{y}$ and $X\hat{\boldsymbol{\theta}_x}$ in order to get a larger correlation between $X\hat{\boldsymbol{\theta}_x}$ and $Z\hat{\boldsymbol{\theta}_z}$*

models with a range of ridge penalties. Ridge Regression models were also fit, which corresponds to $b_{xz} = 0$. To evaluate the success of the fits, we looked at prediction error based on using just X relative to Ordinary Regression as well as the sum correlation. Here, by sum correlation, we mean $\text{cor}(\boldsymbol{x}_*^T\boldsymbol{\theta_x}, y_*) + \text{cor}(\boldsymbol{z}_*^T\boldsymbol{\theta_z}, y_*) + \text{cor}(\boldsymbol{x}_*^T\boldsymbol{\theta_x}, \boldsymbol{z}_*^T\boldsymbol{\theta_z})$, where $(\boldsymbol{x}_*, \boldsymbol{z}_*, y_*)$ is a future observation (corresponding to making another pass through step 6).

The results of the simulation, in Figure 3.1, sheds some light on the effectiveness of using CollRe to improve a regression of $\boldsymbol{y}$ on X. First, we note that ridge regression outperforms CollRe at any choice of b_{xy} and ℓ_2 penalty for this particular problem. At first, this might seem surprising given the fact that CollRe gets the advantage of using Z and ridge regression does not. When looking at the sum correlation though, we see that CollRe outperforms ridge. This suggests that the reason CollRe is doing worse on predicting $\boldsymbol{y}$ is because it is focusing on the distance between $X\hat{\boldsymbol{\theta}_x}$ and $Z\hat{\boldsymbol{\theta}_z}$ instead of just the typical RSS. Essentially, CollRe is giving up a little of the fits involving $\boldsymbol{y}$ in order to get a higher correlation between $X\hat{\boldsymbol{\theta}_x}$ and $Z\hat{\boldsymbol{\theta}_z}$. It seems that CollRe is more naturally suited for supervised canonical correlation analysis, discussed next.

4   Supervised Canonical Correlation Analysis (sCCA)

Canonical Correlation Analysis (CCA) is a data analysis technique that dates back to Hotelling (1996) [4]. Given two sets of centered variables, X and Z, the goal of CCA is to find linear combinations of X and Z that are maximally correlated. Mathematically, CCA performs the following

constrained optimization problem:

$$(\hat{\boldsymbol{\theta_x}}, \hat{\boldsymbol{\theta_z}}) = \arg\max_{\boldsymbol{\theta_x},\boldsymbol{\theta_z}} \boldsymbol{\theta_x}^T X^T Z \boldsymbol{\theta_z} \text{ such that } \boldsymbol{\theta_x}^T X^T X \boldsymbol{\theta_x} \leq 1, \boldsymbol{\theta_z}^T Z^T Z \boldsymbol{\theta_z} \leq 1$$

In this form, it is possible to derive a closed form solution for CCA using matrix decomposition techniques. Namely, $\hat{\boldsymbol{\theta_x}}$ will be the eigenvector corresponding to the largest eigenvalue of $(X^T X)^{-1} X^T Z (Z^T Z)^{-1} Z^T X$. A similar expression can be found for $\hat{\boldsymbol{\theta_z}}$ by switching the roles of X and Z.

CCA might be a useful tool for finding a signal that is common to both X and Z, but there is no guarantee that the discovered signal will also be associated with $\boldsymbol{y}$. To approach this issue, a generalization of CCA called Multiple Canonical Correlation Analysis (mCCA) was developed. mCCA allows for more than 2 datasets and seeks to find a signal that is common to all of the datasets. The case we have, where the third dataset is a vector, can be thought of as a special case of mCCA that we will call Supervised Canonical Correlation Analysis (sCCA).

There are many techniques that approach the mCCA problem. Most of them focus on optimizing a function of the correlations between the various datasets. Gifi (1990) [3] provides an overview of many of the suggestions that have been made for this problem. One example of an optimization problem that people would call mCCA is based on trying to maximize the sum of the correlations:

$$\{\boldsymbol{\theta_i}\}_{i=1,\ldots,k} = \arg\max \sum_{i<j} \boldsymbol{\theta_i}^T X_i^T X_j \boldsymbol{\theta_j} \text{ such that } \boldsymbol{\theta_i}^T X_i^T X_i \boldsymbol{\theta_i} \leq 1 \ \forall i \tag{8}$$

Now the optimization problem above is multiconvex as long as each of the $X_i^T X_i$ are non-singular. This means that a local optimum can be found by iteratively maximizing over each θ_i given the current values of the rest of the coefficients.

4.1 Sparse sCCA

For high dimensional problems (where $p_i >>> n$ for at least one i), several issues emerge when doing sCCA. First, the constraints given in equation (8) are no longer strictly convex constraints because $X_i^T X_i$ is necessarily singular for at least one i. This means that the problem cannot be as easily solved by an iterative algorithm.

One approach that some people take to this problem is to add a ridge penalty on the coefficients. As with ridge regression, adding a ridge penalty will effectively replace $X_i^T X_i$ with $X_i^T X_i + \lambda_i I$ (I being the identity matrix), which will then be non-singular. This means that the mCCA problem in equation (8) can be solved by adding a ridge penalty. Examples of works where people have pursued this method include Leurgans and others (1993) [5]. Another approach is pursued by Witten and Tibshirani (2009) [8] where $X_i^T X_i$ is replaced by I in order to ensure strict convexity of the constraints.

Now, even after adjusting to make sure that the constraints (or penalties in the Lagrange form) are convex, there is still another issue that the high dimensional regime adds. For many problems in the high dimensional regime, the goal of the problem is to do some sort of variables selection. After all, it is much more useful for a biologist to uncover 30 genes or pathways that are particularly important in a process than it is to uncover 30,000 coefficient values that are all fairly noisy anyway. Another way to state that is that we want to find coefficients that are sparse (mostly 0). There has been a lot of work in sparse statistical methods following the introduction of the lasso by Tibshirani (1996) [7]. Witten and Tibshirani (2009) [8] offer the following optimization problem to perform sparse mCCA:

$$\{\boldsymbol{\theta_i}\}_{i=1,\ldots,k} = \arg\max \sum_{i<j} \boldsymbol{\theta_i}^T X_i^T X_j \boldsymbol{\theta_j} \text{ such that } \boldsymbol{\theta_i}^T \boldsymbol{\theta_i} \leq 1, \|\boldsymbol{\theta_i}\| < c_i \forall i$$
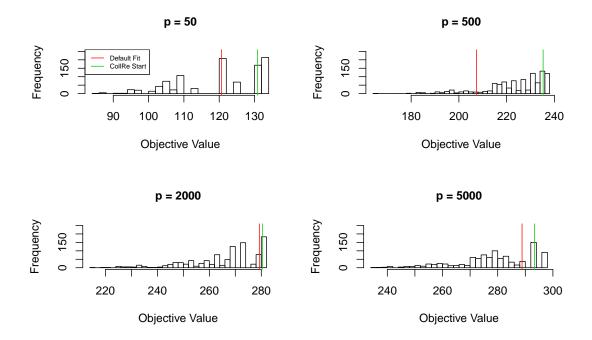
Figure 2: *Result of a simulation to see how close the locally optimal solutions to MultiCCA end up to the global optimum. Histograms of objective values of MultiCCA from 1000 random starts. As we can see, the random starts end up at a variety of local optima, and using the results of CollRe as a starting point often outperforms the default start which is based on an singular value decomposition. In each case, $n = 50$.*

where the c_i can be chosen to impose the desired level of sparsity on each coefficient vector. Note that further convex constraints (or penalties) can be added to the above such as the fused lasso or non-negativity constraints. As with the other methods, this problem is multiconvex and can be solved through an iterative algorithm.

Note however, that a multiconvex problem may be particularly hard to solve in a high dimensional space. While we know the algorithm will converge to a local optimum, we would ideally like to find the global optimum. For a low dimensional space this can be mostly resolved by doing multiple starts from random points in the coefficient space. With enough starts we believe that we can search the space sufficiently well that our best local optimum is at least close to globally optimum. This logic breaks down in high dimensional spaces because it is impossible to sufficiently search the space without exponentially many starting points. This means that while the above methods for Sparse mCCA have outputs, we won't know whether those outputs are even optimizing the criteria in high dimensions.

We generated some data to test the extent to which MultiCCA gets caught in local optima. These datasets have $n = 50, p_u = 30, s_u = \sqrt{1/10}, s_x = s_z = s_y = 1$. For $p = 50, 500, 2000,$ and 5000, we generated a dataset with $p_x = p_z = p$ and then ran MultiCCA from 1000 random (uniform on the unit sphere) start locations. Figure 4.1 shows histograms of the resulting objective values. The vertical lines correspond to the default starting point of MultiCCA, and a starting point that is based on a penalized CollRe solution. As we can see, there are many local optima that emerge especially in higher dimensions. One interesting thing is that the CollRe starts typically end up in a better solution than the default starts provided by the MultiCCA function.

7

Another option to perform sparse high-dimensional sCCA was suggested by Witten and Tibshirani (2009) [8]. She suggests that a method of supervision similar to Bair and others (2006) [1] can be used: before doing a fit, all of the variables are screened against $\boldsymbol{y}$. Only the ones that have correlation above some threshold will be passed along to a CCA model. This method can also be used to add a supervised component to any of the methods that can be used to perform CCA. The main issue with this approach is that it does the supervision in a way that is completely univariate.

4.2 Penalized Collaborative Regression as Sparse sCCA

Consider one of the three terms from our objective function:

$$\min \|X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}\|^2 = \boldsymbol{\theta_x}^T X^T X \boldsymbol{\theta_x} + \boldsymbol{\theta_z}^T Z^T Z \boldsymbol{\theta_z} - 2\boldsymbol{\theta_x}^T X^T Z \boldsymbol{\theta_z} \tag{9}$$

Now let's compare that to the following version of the CCA objective:

$$\min -\boldsymbol{\theta_x}^T X^T Z \boldsymbol{\theta_z} \text{ such that } \boldsymbol{\theta_x}^T X^T X \boldsymbol{\theta_x} \leq 1, \boldsymbol{\theta_z}^T Z^T Z \boldsymbol{\theta_z} \leq 1$$

We can convert the CCA problem from its bounded form into the Lagrange form as follows:

$$\min -\boldsymbol{\theta_x}^T X^T Z \boldsymbol{\theta_z} + \lambda_x \boldsymbol{\theta_x}^T X^T X \boldsymbol{\theta_x} + \lambda_z \boldsymbol{\theta_z}^T Z^T Z \boldsymbol{\theta_z}$$

, where λ_x and λ_z are chosen appropriately to enforce the unit variance constraint. In this way CCA can also be characterized as a penalized optimization problem. The difference between the term from our objective, and the penalized form of CCA is that instead of using λ_x and λ_z in order to enforce unit variance, we choose the values that would result in the objective being convex instead of merely biconvex.

Now it is worth noting that an unenviable fact about the penalty used in equation (9) is that it results in the minimum being achieved by setting all of the coefficients equal to zero. Fortunately, CollRe avoids this issue because the two terms involving $\boldsymbol{y}$.

Thus, CollRe with $b_{xy} = b_{zy} = b_{xz} = 1$ is very similar to doing a sum of correlations mCCA as in the equation (8), with the exception that we have picked the penalties that allow for convexity instead of the penalties that correspond to unit variance.

As discussed in Section 2.2 one of the advantages of CollRe is the simplicity with which convex penalties can be added to the objective function. Thus, it is easy to convert CollRe into a form that is appropriate for sparse sCCA by adding penalties just as in Witten and Tibshirani (2009) [8].

To compare CollRe against a competing algorithm for sparse sCCA, we generated data from the above model with $n = 50, p_x = p_z = 20, p_u = 3, s_u = 1, s_x = s_z = s_y = 1/2$. Then, we added 40 variables to both X and Z that were generated from 3 new factors that have no effect of $\boldsymbol{y}$. These 40 variables act as confounding variables that reflect an effect we do not want to uncover. This could correspond to a batch effect in the measurements, or maybe some other underlying difference among the sampled patients. Finally, we added another 440 columns to X and Z that were just independent gaussians to act as null predictors. We ran both CollRe with a lasso penalty and Wittens MultiCCA from the PMA package with an ℓ_1 constraint each over a range of parameter values. This process was repeated 80 times with new loadings and noise each time but the same factors. Figure 4.2 shows the average (over repetitions) theoretical sum correlation for future observations, as well as the recovery of true predictors, against a range of nonzero coefficients that corresponds to a range of penalty parameters.

From the results, we can see that CollRe does a much better job of finding coefficients that have high sum correlation. MultiCCA seems to get caught in the trap set by the confounding variables, which makes it harder to raise the sum correlation much above 1 (a perfect correlation between $\boldsymbol{x}_*^T \boldsymbol{\theta_x}$ and $\boldsymbol{z}_*^T \boldsymbol{\theta_z}$ with no relation to y_*). Interestingly, while MultiCCA does worse than CollRe on recovery of true variables for the first 70 or so variables added, it seems to do a better job of
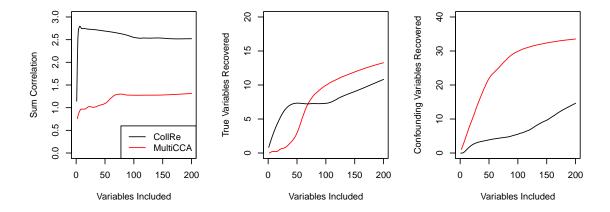
Figure 3: *Results of a simulation to compare CollRe and MultiCCA in performing sparse supervised mCCA. For each repetition, a dataset with $n = 50, p_x = p_z = 500$ is created. For both X and Z, 40 of the predictors are confounding variables and 20 of the predictors are true variables (the rest are null). Confounding variables are the ones that share a signal between X and Z, but not $\mathbf{y}$. The true variables share a signal between all three datasets. Values have been averaged over 80 repetitions. MultiCCA is much more susceptible to picking up the confounding variables, and thus has a much harder time achieving high correlations. Interestingly, while CollRe finds many more true variables at first, after 70 or so included variables MultiCCA starts finding more.*

recovering true variables after that point. It is unclear what exactly is causing that transition in this problem.

5 Real Data Example

To demonstrate the applicability of penalized CollRe, we also ran it on a high dimensional biological dataset. We used a neoadjuvant breast cancer dataset that was provided by our collaborators in the Division of Oncology at the Stanford University School of Medicine. Details about the origins of the data can be found at ClinicalTrials.gov using the identifier NCT00813956. This dataset consists of $n = 74$ patients who underwent a particular breast cancer treatment. Before treatment, the patients had measurements taken on their gene expression as well as copy number variation. In all, after some pre-processing, there were $p_x = 54,675$ gene expression measurements per patient and $p_z = 20349$ copy number variation measurements. Additionally, each patient was given a RCB score six months after treatment that corresponds to how effective the treatment was. The RCB score is essentially a composite of various metrics on the tumor: primary tumor bed area, overall % cellularity, diameter of largest axillary metastasis, etc.

The goal of the analysis is to select a set of gene expression measurements that are highly correlated with a particular pattern of copy number variation gains or losses. That said, we are only interested in sets that also correlate with the RCB value. As such, it is the perfect opportunity to employ CollRe.

Due to computational limitations and issues with noise in the underlying measurements, some further pre-processing was done to the data. First, the gene expression measurements were screened by their variance across the subjects. Only the top 28835 gene expression genes were kept. For the

copy number variation measurements we needed to account for the fact that for each patient the copy number variation measurements are VERY highly autocorrelated because they had already been run through a circular binary segmentation algorithm (a change point algorithm used to smooth copy number variation data). We use a fused lasso penalty to help correct for the fact that we don't really have gene level measurements. However, doing fused lasso solves can be very slow for large p, so we took consecutive triples of the copy number variation measurements and averaged them. This reduced the number of copy number variation measurements to 6783. Our new X and Z matrices were scaled and centered, and then CollRe on the dataset with $b_{xy} = b_{xz} = b_{zy} = 1$ and the following parameters and penalty terms:

$$P^x(\boldsymbol{\theta_x}) = \lambda_x(.9\|\boldsymbol{\theta_x}\|_1 + .1\frac{1}{2}\|\boldsymbol{\theta_x}\|_2^2)$$

$$P^z(\boldsymbol{\theta_z}) = 4\|\boldsymbol{\theta_z}\|_1 + 200\sum_{i=2}^{i=p_z}|(\boldsymbol{\theta_z})_i - (\boldsymbol{\theta_z})_{i-1}|$$

We searched a grid of λ_x in order to find a solution with about 50 nonzero coefficients in each set of variables. This corresponds roughly with the number of genes a collaborator thought she would be able to reasonably examine for plausible connections. The penalty terms on $\boldsymbol{\theta_z}$ were chosen in a way that the selected coefficients looked reasonably smooth. The resulting $\hat{\boldsymbol{\theta_z}}$ vector can be seen in Figure 5

6   Solving CollRe with Penalties

In Section 2.2 we mentioned that CollRe is solvable with a variety of penalty terms added. In fact, due to the nature of the CollRe objective, it can often be solved for common penalty terms using out of the box penalized regression solvers. To make this concrete, let us focus on CollRe with the addition of ℓ_1 penalties.

Consider then, the objective function with penalty terms:

$$J(\boldsymbol{\theta_x}, \boldsymbol{\theta_z}; X, Z, \boldsymbol{y}, b_{xy}, b_{zy}, b_{xz}, \lambda_x, \lambda_z) = \frac{b_{xy}}{2}\|\boldsymbol{y} - X\boldsymbol{\theta_x}\|^2 + \frac{b_{zy}}{2}\|\boldsymbol{y} - Z\boldsymbol{\theta_z}\|^2 + \frac{b_{xz}}{2}\|X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}\|^2 +$$
$$\lambda_x\|\boldsymbol{\theta_x}\|_1 + \lambda_z\|\boldsymbol{\theta_z}\|_1 \quad (10)$$

We note that (10) is a convex function, so we can optimize it by iteratively optimizing over $\boldsymbol{\theta_x}$ and $\boldsymbol{\theta_z}$. For a given value of $\boldsymbol{\theta_z}$, the optimal $\boldsymbol{\theta_x}$ is given by:

$$\hat{\boldsymbol{\theta_x}} = \text{LASSO}(X, \boldsymbol{y^*}, \frac{\lambda_x}{b_{xy} + b_{xz}}) \text{ , where } \boldsymbol{y^*} = \frac{b_{xy}}{b_{xy} + b_{xz}}\boldsymbol{y} + \frac{b_{xz}}{b_{xy} + b_{xz}}Z\boldsymbol{\theta_z} \quad (11)$$

Here, $\text{LASSO}(\tilde{X}, \tilde{\boldsymbol{y}}, \tilde{\lambda})$ is the solution to the ℓ_1 penalized regression problem:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\tilde{\boldsymbol{y}} - \tilde{X}\boldsymbol{\beta}\|^2 + \tilde{\lambda}\|\boldsymbol{\beta}\|_1 \quad (12)$$

An equivalent solution of $\hat{\boldsymbol{\theta_z}}$ given $\boldsymbol{\theta_x}$ can be found by symmetry. Thus, by iterating back and forth between these two ℓ_1 penalized regression problems, we are guaranteed to the optimum of equation (10). This means it is trivial to write a solver for CollRe using ℓ_1 penalties as long as you have access to a solver for regression with ℓ_1 penalties. Many such functions can be found in R packages, including the popular **glmnet** function in the self titled package.
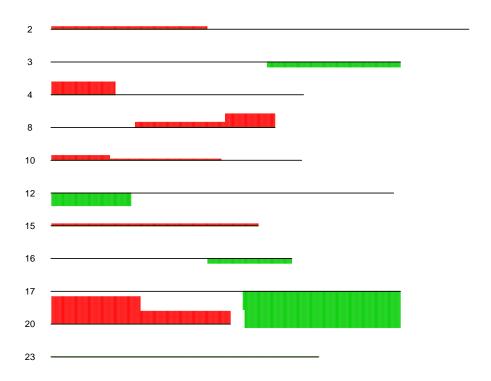
Figure 4: *The resulting vector of coefficients for the copy number variation data from running CollRe on the RCB dataset. Regions with positive coefficients (amplification associated with higher RCB) are darker and appear above the line. Regions with negative coefficients are lighter and appear below the line. The size of the bars are proportional to the coefficient values. Missing chromosomes had no nonzero coefficients. The piece-wise constant nature of the coefficient vector is due to the use of a fused lasso.*

6.1 Proof of Correctness of Algorithm (CollRe with ℓ_1 Penalty)

Let $\tilde{J}$ be the LASSO criterion:

$$\tilde{J}(\tilde{\boldsymbol{\beta}}; \tilde{X}, \tilde{\boldsymbol{y}}, \tilde{\lambda}) = \|\tilde{\boldsymbol{y}} - \tilde{X}\tilde{\boldsymbol{\beta}}\|^2 + \tilde{\lambda}\|\tilde{\boldsymbol{\beta}}\|_1 \tag{13}$$

Then we see that $\tilde{J}$ has subgradient:

$$\frac{\partial \tilde{J}}{\partial \tilde{\boldsymbol{\beta}}} = \tilde{X}^T \tilde{X}\tilde{\boldsymbol{\beta}} - \tilde{X}^T \tilde{\boldsymbol{y}} + \tilde{\lambda}s(\tilde{\boldsymbol{\beta}}) \tag{14}$$

Compare this to the subgradient of J with respect to $\boldsymbol{\theta_x}$:

$$\frac{\partial J}{\partial \boldsymbol{\theta_x}} = (b_{xy} + b_{xz})X^T X\boldsymbol{\theta_x} - X^T(b_{xy}\boldsymbol{y} + b_{xz}Z\boldsymbol{\theta_z}) + \lambda_x s(\boldsymbol{\theta_x}) \tag{15}$$

Dividing (15) by $b_{xy} + b_{xz}$ and substituting $\boldsymbol{y^*} = \frac{b_{xy}}{b_{xy}+b_{xz}}\boldsymbol{y} + \frac{b_{xz}}{b_{xy}+b_{xz}}Z\boldsymbol{\theta_z}$ completes the proof.

6.2 Augmented Data Version

For some selections of penalties, parameters, and solvers, CollRe can be fit using an augmented data approach. This means that the solution can be found in just one call to a solver instead of having to iterate. In practice, this can increase the rate of convergence and reduce total computation time.

Let us return to the example of trying to fit CollRe with the addition of an ℓ_1 penalty. Consider the following LASSO problem:

$$\tilde{X} = \begin{bmatrix} \sqrt{b_{xy}}X & 0 \\ 0 & \sqrt{b_{zy}}\frac{\lambda_x}{\lambda_z}Z \\ \sqrt{b_{xz}}X & -\sqrt{b_{xz}}\frac{\lambda_x}{\lambda_z}Z \end{bmatrix}, \tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y} \\ \sqrt{b_{zy}}\boldsymbol{y} \\ 0 \end{bmatrix}, \tilde{\boldsymbol{\beta}} = \begin{bmatrix} \boldsymbol{\theta_x} \\ \frac{\lambda_z}{\lambda_z}\boldsymbol{\theta_z} \end{bmatrix} \tag{16}$$

$$\hat{\tilde{\boldsymbol{\beta}}} = \arg\min_{\tilde{\boldsymbol{\beta}}} \|\tilde{X}\tilde{\boldsymbol{\beta}} - \tilde{\boldsymbol{y}}\|^2 + \lambda_x\|\tilde{\boldsymbol{\beta}}\|_1 \tag{17}$$

It can be easily verified that equation (17) is exactly the CollRe with ℓ_1 penalty fit for the parameters given. Essentially, this means that instead of iterating between LASSO solves with ($\tilde{n} = n, \tilde{p} = p_x$) and ($\tilde{n} = n, \tilde{p} = p_z$) until convergence, we only do one solve with ($\tilde{n} = 3n, \tilde{p} = p_x + p_z$). Because we expect $n <<< \max(p_x, p_z)$, we don't expect tripling $\tilde{n}$ to have much effect on run time. Further, due to active set rules that are built into packages like the R package **glmnet**, even if we double $\tilde{p}$ it should not have too large an effect on run time (Friedman and others (2010) [2]).

We ran some simulations that involve generating X, Z, and $\boldsymbol{y}$ from independent standard normal draws. we then fit CollRe with Elastic Net to the data setting all of the parameters equal to one (except $\lambda_2^x = \lambda_2^z = 0$). For $n = 100$, $p_x = p_z = 2000$ the normal version of CollRe with ℓ_1 penalty ($\lambda_x = \lambda_z = 1$) takes about 2.2 seconds to run on a 2010 Macbook Pro. The augmented version only takes 0.6 seconds to run. The augmented version also achieves a lower value for the objective function (8.127974 compared to 8.128410), so the speedup is not just coming from a premature convergence.

7 Discussion

In this paper, we introduced a new model called Collaborative Regression, which can be used in settings where one has two sets of predictors and a response variable for a set of observations. We explored the possibility of using CollRe in a prediction framework, but ultimately decided that it was not particularly well suited for that task.

We then discussed the problem of sparse supervised Canonical Correlation Analysis, which seems to be an increasingly interesting problem for biostatistics. While current approaches to sCCA are biconvex and don't necessarily lend themselves to a sparse generalization, CollRe does not suffer from those same issues. We used several simulations and real data to explore both the issues of biconvexity in high dimensions, as well as the performance of CollRe.

Acknowledgments

References

[1] Eric Bair, Trevor Hastie, Debashis Paul, and Robert Tibshirani. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101:119–137, 2006.

[2] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

[3] Albert Gifi. *Nonlinear multivariate analysis*. Wiley Chichester, 1990.

[4] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[5] S. E. Leurgans, R. A. Moyeed, and B. W. Silverman. Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(3):pp. 725–740, 1993.

[6] Debashis Paul, Eric Bair, Trevor Hastie, and Robert Tibshirani. "pre-conditioning" for feature selection and regression in high-dimensional problems. *Annals of Statistics*, 36(4):1595–1618, 2008.

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[8] D.M. Witten and R. Tibshirani. Extensions of sparse canonical correlation analysis, with application to genomic data. *Statistical Applications in Genetics and Molecular Biology*, 8(1):Article 28, http://www.bepress.com/sagmb/vol8/iss1/art28, 2009.