Computational Statistics and Data Analysis 101 (2016) 226–235

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Data Shared Lasso: A novel tool to discover uplift

Samuel M. Gross [a,b,*], Robert Tibshirani [b,c]

[a] Nuna, 650 Townsend St, San Francisco, CA, United States
[b] Department of Statistics, Stanford University, Stanford, CA, United States
[c] Department of Health, Research, & Policy, Stanford University, Stanford, CA, United States

ARTICLE INFO

Article history:
Received 10 March 2015
Received in revised form 16 February 2016
Accepted 28 February 2016
Available online 12 March 2016

Keywords:
Clinical studies
High dimensional regression
ℓ_1 penalization
Multi-task learning
Sentiment analysis
Uplift

ABSTRACT

A model is presented for the supervised learning problem where the observations come from a fixed number of pre-specified groups, and the regression coefficients may vary sparsely between groups. The model spans the continuum between individual models for each group and one model for all groups. The resulting algorithm is designed with a high dimensional framework in mind. The approach is applied to a sentiment analysis dataset to show its efficacy and interpretability. One particularly useful application is for finding sub-populations in a randomized trial for which an intervention (treatment) is beneficial, often called the *uplift* problem. Some new concepts are introduced that are useful for uplift analysis. The value is demonstrated in an application to a real world credit card promotion dataset. In this example, although sending the promotion has a very small average effect, by targeting a particular subgroup with the promotion one can obtain a 15% increase in the proportion of people who purchase the new credit card.

1. Introduction

We consider the problem of combining data from smaller subproblems that have some shared structure. Namely we have several supervised learning datasets with the same predictors, and we expect that the sparse true coefficients vary in some manner between the different subproblems. This might be a result of the data having been measured differently between the datasets, or from other underlying disparities between the datasets—such as the application of different treatments. We expect that due to the similarities, we can learn more about all of the datasets by sharing the data across the problems in some sensible way. This framework is potentially useful in many applications including biostatistics, sentiment analysis, and clinical trials.

If the true regression coefficients for all of the datasets are the same, then we expect the best solution is obtained by simply pooling all of the data and fitting one model. On the other hand, if the true regression coefficients have little similarity between groups, then it is best to fit separate models for each dataset. Our new approach targets scenarios that fall somewhere in between: some of the coefficients are shared between the true models, but each model also has some coefficients that are different from the shared ones.

In Section 2 we outline notation and give a definition of the Data Shared Lasso (DSL). We leverage the fact that DSL can be characterized as a lasso with augmented data to derive a fast algorithm to solve the problem.

We compare DSL to several similar models and problems in Section 3. There, we see that DSL makes very different assumptions about the coefficients of the model than other typical approaches for the problem. This allows us to fit interpretable models for the cases where we think the DSL framework is more appropriate.

* Corresponding author at: Department of Statistics, Stanford University, Stanford, CA, United States.
E-mail addresses: sam@nuna.com (S.M. Gross), tibs@stanford.edu (R. Tibshirani).

http://dx.doi.org/10.1016/j.csda.2016.02.015
0167-9473/

Section 4 explores the use of DSL on a large sentiment analysis dataset. When splitting movies into groups based on genre, DSL learns more about predicting the rating of a movie from a written review than either a pooled lasso or separate lassos. This example also provides motivation for DSL because the output coefficients nicely elucidate both the similarities and differences between the groups, as seen in Fig. 2.

Finally, in Section 5 we discuss applying DSL to the *uplift* problem where we try to find subpopulations that respond better to a particular treatment, a hot area in marketing and also in personalized medicine. We introduce a new concept: the uplift of a particular treatment assignment plan. By applying DSL to a credit-card-promotion dataset, and using our new concept of uplift, we are able to show that targeting a particular subgroup would result in a 15% increase in the proportion of people who purchase the new credit card.

2. Data Shared Lasso (DSL)

The Data Shared Lasso is a technique that is designed for problems where the observations belong to non-overlapping, pre-specified groups. More formally, we assume we have n observations of the form $(\boldsymbol{x_i}, y_i, g_i)$, where $\boldsymbol{x_i} \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, and $g_i \in \{1, 2, \ldots, G\}$. Here p corresponds to our usual number of predictors and G is the number of groups we consider. We call $\boldsymbol{X}$ the matrix that has the $\boldsymbol{x_i}$'s as rows, $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$, and $\boldsymbol{g} = (g_1, g_2, \ldots, g_n)$.

For simplicity we focus on the regression case now, but DSL is general enough to be applied to any generalized linear model, as will show in the next section. We assume that the y_i are generated as:

$$y_i = \boldsymbol{x_i}^T(\boldsymbol{\beta} + \boldsymbol{\Delta}_{g_i}) + \epsilon_i, \tag{1}$$

where the ϵ_i are independent error terms. An intercept term can be included by making the first column of $\boldsymbol{X}$ a vector of ones.

Clearly, given enough data from each group we could accurately estimate $\boldsymbol{\beta} + \boldsymbol{\Delta}_g$ for all g using separate regression procedures. Alternatively, if we did not have much data, we could pool all of it together and estimate the coefficients that perform best when averaged across the groups. DSL uses regularization parameters r_g to control the amount of pooling done in an intermediate problem:

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Delta}}_1, \ldots, \hat{\boldsymbol{\Delta}}_G) = \arg\min \frac{1}{2}\sum_i \left(y_i - \boldsymbol{x_i}^T(\boldsymbol{\beta} + \boldsymbol{\Delta}_{g_i})\right)^2 + \lambda\left(\|\boldsymbol{\beta}\|_1 + \sum_{g=1}^{G} r_g \|\boldsymbol{\Delta}_g\|_1\right), \tag{2}$$

where λ is the usual complexity parameter.

Eq. (2) decomposes neatly into two parts. The first is a Residual Sum of Squares term that penalizes a model based on squared distance to the truth. The second term is a lasso penalty on the coefficients. The lasso, or ℓ_1 penalized regression, was popularized by Tibshirani (1996) and is an active area of research in Machine Learning and Statistics (Tibshirani et al., 2012; Donoho, 2006). As λ is increased from 0, a lasso penalty causes increasingly more of the coefficients to be set to exactly zero. Thus, it can be used to fit sparse coefficient vectors. Sparse coefficients are easier to interpret, and especially so in high dimensions ($p \gg n$).

2.1. Fitting the Data Shared Lasso

The Data Shared Lasso can be implemented with any lasso solver using a straightforward augmented data approach:
Let $\boldsymbol{Z}$ be defined as

$$\boldsymbol{Z} = \begin{pmatrix} \boldsymbol{X_1} & r_1\boldsymbol{X_1} & 0 & \cdots 0 \\ \boldsymbol{X_2} & 0 & r_2\boldsymbol{X_2} & \cdots 0 \\ \vdots & & & \\ \boldsymbol{X_G} & 0 & 0 & \cdots r_G\boldsymbol{X_G} \end{pmatrix},$$

where $\boldsymbol{X_j}$ and $\boldsymbol{y_j}$ are the dataset for subproblem j. Namely $\boldsymbol{X_j}$ is the matrix formed by taking as rows all $\boldsymbol{x_i}$ such that $g_i = j$ and $\boldsymbol{y_j}$ is the vector formed by taking as elements all y_i such that $g_i = j$. Finally, let $\tilde{\boldsymbol{y}} = (\boldsymbol{y_1}^T, \boldsymbol{y_2}^T, \ldots, \boldsymbol{y_G}^T)^T$ and $\tilde{\boldsymbol{\beta}} = (\boldsymbol{\beta}^T, \frac{1}{r_1}\boldsymbol{\Delta_1}^T, \ldots, \frac{1}{r_G}\boldsymbol{\Delta_G}^T)^T$.

Then we note that

$$\frac{1}{2}\|\tilde{\boldsymbol{y}} - \boldsymbol{Z}\tilde{\boldsymbol{\beta}}\|^2 + \lambda\|\tilde{\boldsymbol{\beta}}\|_1 = \cdots \frac{1}{2}\sum_i \|y_i - \boldsymbol{x_i}^T(\boldsymbol{\beta} + \boldsymbol{\Delta}_{g_i})\|^2 + \lambda\left(\|\boldsymbol{\beta}\|_1 + \sum_{g=1}^{G} r_g\|\boldsymbol{\Delta}_g\|_1\right), \tag{3}$$

where the left hand side of the equation is simply the lasso objective function for fitting $\tilde{\boldsymbol{y}}$ on $\boldsymbol{Z}$ and the right hand side is our objective function.

Thus, we can solve a DSL problem by passing $\boldsymbol{Z}$ and $\tilde{y}$ into a lasso solver, such as the R package **glmnet** (Friedman et al., 2010). As special case solvers work much faster than general convex solvers, this yields a significant advantage in speed of fitting the model.

Additionally, this augmented data approach makes it clear how we can extend the method of DSL to other problems with ℓ_1 penalties, such as penalized Generalized Linear Models; We build out the same augmented dataset and pass it into the new problem. With the **glmnet** package, this is as easy as changing the `family` argument.

The fact that DSL is a specific case of the lasso does not just provide computational benefits. It also lets us leverage theory that has been designed for the lasso.

2.2. Choosing r_g

The first term of our objective depends only on $\boldsymbol{\beta} + \boldsymbol{\Delta}_{g_i}$ so there is an identifiability concern because it is overparameterized. We resolve this concern with the penalty term $\lambda\left(\|\boldsymbol{\beta}\|_1 + \sum_g r_g \|\boldsymbol{\Delta}_g\|_1\right)$; the r_g parameters control which of the potential solutions DSL selects and also controls the degree of sharing done between the groups in the problem. In general, large values of r_g will correspond to more sharing and small values of r_g will correspond to less sharing.

Assume that we have fixed $\boldsymbol{\beta}^*_{\boldsymbol{g}} \in \mathbb{R}^p$ and r_g for $g = 1, 2, \ldots, G$. What can we say about

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Delta}}_g) = \arg\min \lambda\left(\|\boldsymbol{\beta}\|_1 + \sum_g r_g \|\boldsymbol{\Delta}_g\|_1\right) \quad \text{such that } \boldsymbol{\beta} + \boldsymbol{\Delta}_g = \boldsymbol{\beta}^*_{\boldsymbol{g}}? \tag{4}$$

First, we note the problem is separable in p, so we discuss without loss of generality the case $p = 1$. Further, all values that satisfy the equality constraint fall into a family that can be characterized as $\beta = c$, $\Delta_g = \beta^*_g - c$.

Thus, solving $(\hat{\beta}, \hat{\Delta}_g)$ is equivalent to solving an unconstrained optimization in one variable: $\hat{c} = \arg\min |c| + \sum_g r_g |\beta^*_g - c|$. If we let $r_0 = 1$ and $\beta^*_0 = 0$, we see that $\hat{c} = \hat{\beta}$ is the weighted median of $\{\beta^*_i\}_{i=0,1,\ldots,G}$ with weights $\{r_i\}_{i=0,1,\ldots,G}$.

Example: $r_g = r \in (\frac{1}{G}, \frac{1}{G-2})\ \forall g \in \{1, 2, \ldots, G\}$.

In this case, we would have nonzero $\hat{\beta}$ if and only if all of the β^*_g are nonzero with the same sign. Then, we would have $\hat{\beta} = \text{sign}(\beta^*_1) \min |\beta^*_i|$. In this setting a coefficient will only be called shared to the extent that all of the groups capture that effect. The rest of the contribution for each group will be captured by a separate effect in $\hat{\Delta}_g$. Note that for the case $G = 2$, the possible values of r range from $1/2$ to ∞.

Example: r_g such that $\sum_g r_g < 1$.

Since $\sum_g r_g < 1$, we are guaranteed that $\hat{\beta} = 0$. This is equivalent to fitting separate lasso models to each group.

Example: r_g such that $\sum_g r_g = 1$.

There will be identifiability issues, so this case should be avoided.

Example: $r_g = r > 1\ \forall g \in \{1, 2, \ldots, G\}$.

Here β^*_0 as defined above will play no role in determining $\hat{\beta}$ except for potentially changing how the weighted median handles ties when G is even. For G odd, $\hat{\beta}$ will simply be the median of $\{\beta^*_g\}_{g=1,\ldots,G}$.

In addition to the above examples, there is another way to interpret the r_g values. Essentially, each r_g represents a discount factor for having those coefficients apply to only a subset of the data instead of the whole dataset. If we consider making a fixed change to either $\hat{\beta}$ or one of the $\hat{\Delta}_g$, the change to $\hat{\beta}$ will have a larger effect on our prediction vector. Thus, we may want to penalize changes to $\hat{\beta}$ at a higher rate. This is a heuristic argument for setting $r_g < 1$.

In the special case where there are the same number of observations in each group, the columns of each $\boldsymbol{X_i}$ are standardized to have mean 0 and unit variance, and we use $r_g = r = \frac{1}{\sqrt{G}}$, $\boldsymbol{Z}$ will just be the standardized version of

$$\begin{pmatrix} \boldsymbol{X_1} & \boldsymbol{X_1} & 0 & \cdots 0 \\ \boldsymbol{X_2} & 0 & \boldsymbol{X_2} & \cdots 0 \\ \vdots & & & \\ \boldsymbol{X_G} & 0 & 0 & \cdots \boldsymbol{X_G} \end{pmatrix}.$$

For this reason, we recommend $\frac{1}{\sqrt{G}}$ as a default value of r_g. Note that for small numbers of groups ($G \leq 3$), this also satisfies the condition that $r \in (\frac{1}{G}, \frac{1}{G-2})$. In the case that prediction error is of primary concern, or a user is interested in looking at different levels of pooling, it may be worth looking at other values of r_g as well. In this case a user could apply a cross validation step to select the value for r_g.

3. Comparison to related models

3.1. Data Enriched Regression

A related approach – Data Enriched Regression – has recently been suggested by Chen et al. (2013) for the special case when $G = 2$ in a situation where one of the groups is the target for analysis, and the other is used only to improve estimates.

If we let $g = 1$ correspond to the target group, they solve:

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Delta}}) = \arg\min \sum_{i:g_i=1} (y_i - \boldsymbol{x_i}^T \boldsymbol{\beta})^2 + \sum_{i:g_i \neq 1} (y_i - \boldsymbol{x_i}^T (\boldsymbol{\beta} + \boldsymbol{\Delta}))^2 + \lambda P(\Delta). \tag{5}$$

Their paper focuses mostly on the case where $P(\Delta) = \sum_{i:g_i=1} (\boldsymbol{x_i}^T \boldsymbol{\Delta})^2$—the quadratic penalty. With this penalty in the location model ($p = 1, x_i = 1$), they are able to show that solving the above problem at an oracle choice of λ achieves the same expected squared estimation error as doing the regression on just group one with an additional $\frac{\sigma_1^2}{\Delta^2}$ observation (where σ_1 is the standard deviation of the Gaussian noise added to observations from group one) as the number of observations in group two goes to infinity. Thus, this partial pooling will do better when the bias between the problems is small, or when the original problem has a large noise level. Since the effect is additive, we expect this procedure to help most on datasets with relatively few observations.

There are three main differences between our problem and Data Enriched regression. First, we will often consider multiple datasets in our setup and are potentially interested in all of them, while they focus on one dataset that they want to learn about and a larger one from which to borrow data. Second, we penalize both the $\boldsymbol{\beta}$ and $\boldsymbol{\Delta}_g$ terms in order to enforce sparsity on both while they only penalize $\boldsymbol{\Delta}$. Finally, they focused mostly on ℓ_2 penalties instead of ℓ_1 due to analytical tractability. Despite these differences, their theoretical results suggest that our algorithm will be able to provide improvement over just fitting each dataset separately; this is demonstrated on real data in Section 4.

3.2. Data Shared Lasso as an interaction model

It is clear from the augmented data representation in Section 2.1 that DSL is basically a linear model with interaction terms between group and the other predictors. While this is true, it is important to note that we have resolved the identifiability concerns using our penalty instead of a more typical way. There are two common approaches to resolving this identifiability. One is to set one of the groups to act as a baseline and constrain its interaction terms to be zero as in Chen et al. (2013). The other is to impose a constraint that the mean interaction across groups is zero.

An interesting discussion of the nature of interaction effects when you impose a mean zero constraint for a variety of predictor types can be found in Lim and Hastie (2013). They argue for a hierarchical interaction model that only allows interaction terms in the presence of main effects. This is similar to the approach of Bien et al. (2013). Hierarchy makes sense in the main/interaction effects model with a mean zero constraint because a main effect can only be zero if the mean of the interaction effects for that predictor is also zero; this is an event we typically do not expect to occur in practice. In the shared/separate effects approach, the same logic no longer applies because we have resolved the identifiability using our penalization scheme instead of an equality constraint. Allowing nonzero separate effects for predictors with no shared effect is valuable for interpretation, as it allows for a more sparse representation of the same (or similar) effects. As we will see in Section 4, a coefficient can be meaningful for only one of the groups. Our method allows the outputted coefficient to make this claim directly.

Other approaches have also been tried in recent years to address the issue of building interaction models. These approaches include Logic Regression (Ruczinski et al., 2003), Composite Absolute Penalties (Zhao et al., 2009), and tree based methods (Hastie et al., 2001). While all of these methods have uses, we think that the interpretability of the shared/separate effects of DSL makes it attractive for many problems. In addition, the DSL problem has a nice heuristic explanation of allowing effects to enter the model in different ways and letting them compete to give the most succinct explanation of the final effects.

3.3. Multi-task learning

Multi-Task Learning is a framework that is essentially the same as the one we laid out in Section 2. Many of these approaches are similar to our own in setting up a convex optimization problem, but they often make different choices in the sorts of penalties they choose, which can have dramatic effects on the fitted coefficients. For simplicity, we will focus on two examples here and how they differ from DSL.

Argyriou et al. (2008) introduce a formulation that penalizes using the group lasso (Ming and Lin, 2005). Basically, they have a penalty like

$$\sum_{j=1}^{p} \sqrt{\beta_j^2 + (\boldsymbol{\Delta_1})_j^2 + \cdots + (\boldsymbol{\Delta_G})_j^2}.$$

This enforces sparsity on the predictors, but only if the coefficient is 0 for that predictor across all groups. Thus, this model implicitly assumes that each task shares the same predictors, but may have different values for those predictors.

Another approach is given by Gu and Zhou (2009) who suggest a model that finds coefficients vectors so that $\boldsymbol{\beta} + \boldsymbol{\Delta}_g$ is in a low dimensional subspace of $\mathbb{R}^p$. Essentially, this model is equivalent to assuming there are a few linear combinations of your original predictor space that should form the predictor space for your G tasks.

Table 1
Test set mean squared error for IMDB data.

	All	Drama	Comedy	Horror
Pooled	5.63	5.57	**5.78**	5.55
Separate	5.72	5.63	5.97	5.61
Data shared	**5.54**	**5.56**	5.85	**5.07**

These models are interesting and useful, but they imply a very different set of assumptions on the coefficients. DSL only considers coefficients "shared" to the extent that they actually have the same value. These other methods though, consider a variable having an opposite effect to be just as similar as having the same effect (in terms of reduction to the penalty).

That said, the low dimensional subspace model of Gu and Zhou (2009) has some particular advantages as the number of tasks, G, increases. In this case, we may be more interested in summarizing each task more than just to the nonzero coefficients. The implicit low dimensional summary of the subspace approach gives an easy way to represent the similarities/differences between tasks.

Consider applying either of these formulations to the IMDb example in Section 4. Clearly, assuming that the sparsity pattern is similar for the different genres is false. Thus, it makes sense to look to models like DSL that will allow for differing sparsity patterns between groups.

After submitting a first version of this work, we became aware of concurrent works by Ollier and Viallon (2014, 2015) where the authors introduce similar ideas together with a theoretical analysis of their statistical properties.

4. IMDb example

To test our algorithm on some real data, we use a publicly available dataset of movie reviews from IMDb.com. This dataset, aclImdb (Maas et al., 2011), contains 50K written reviews of movies that have been split into a training and test set of equal size. Associated with each written review is an integer rating out of 10 (where 10 is the best). The dataset only contains polarized reviews; half of the reviews are positive (rating ≥ 7) and the other half are negative (rating ≤ 4). We used a binary bag of words representation of the reviews, using only words that were present in at least 5 reviews from our training set; this resulted in $p = 27{,}743$ features in our dataset. Our response value is the integer rating.

To use our method on this dataset, we need some pre-defined groups of observations. In this problem, we focused on the genres of the movies restricting our attention to the three most common in this dataset: drama, comedy, and horror. As a movie can have multiple genres, we only kept reviews for movies that had exactly one of those three genres. This brought us down to $n = 16{,}386$ reviews in our training set (8286 dramas, 5027 comedies, and 3073 horror films).

We performed an analysis using pooled lasso (equivalent to DSL with $r = \infty$), separate lasso (equivalent to DSL with $r = \frac{1}{4}$), and our method with $r = 1/\sqrt{3}$. In order to summarize the coefficients of each model, we used word clouds where the size of each word represents the absolute value of the coefficient, and the sign of each coefficient is indicated by color (orange positive and blue negative). We also wanted to compare DSL to some of the models mentioned in Section 3.3, but were unable to do so due to the size of the dataset. The X matrix used in this problem is $16{,}386 \times 27{,}743$, which takes about 3.6 GB to store in R. As many types of algorithms require storing further intermediate results in memory, this will quickly overwhelm the total memory on typical computers. We were able to fit such a large problem on a desktop with only 8 GB of memory by utilizing the **glmnet** packages capabilities to use sparse representations of a data matrix (since we are using word count data, almost every value in X is 0). This drastically reduces the memory requirements and allows the model to be fit. To our knowledge, none of the other methods we discussed that perform multi-task learning have been implemented for sparse representations of data. This further highlights the advantages of our model being a special case of the lasso.

Fig. 1 shows the word clouds generated by running separate lassos on each of the datasets. So much of the image is spent conveying the shared coefficients like "worst" and "waste", that it is harder to make out what is different between the groups.

Compare that to the results of running DSL on the dataset as summarized in Fig. 2. By separating the shared and separate coefficients, it becomes much easier to interpret the model. Essentially, the shared words have the same meaning for any of the three genres, and the separate coefficients tell you the words that have special meaning for a specific genre. Compare the ease of reading similarities/differences between the genres from Figs. 2 and 1.

We also calculated the error on the withheld test set. Namely, the average squared difference between the true rating and the predicted rating based on a model learned on our training set. Only one movie is shared between the training and test datasets, so we are not worried about picking up correlations between our training and test sets. Normally we would not allow any reuse between training and test set, but we decided to use the original groupings as published in Maas et al. (2011) to ease comparisons. After screening the test set by genre, we had 18,109 samples with a similar distribution of genres as the training set (9937 dramas, 4774 comedies, and 3398 horror films).

As we see from Table 1, the DSL approach does almost as well as or better than the other approaches for all three genres. It had the biggest gains on Horror, which was the smallest group in the dataset. This is consistent with the theoretical results that Chen et al. (2013) proved about Data Enriched Regression.

We have provided the ***R*** code and additional data needed in order to run the analysis of this section as a supplement to this paper (see Appendix A). This implements a version of DSL that works with sparse matrices.

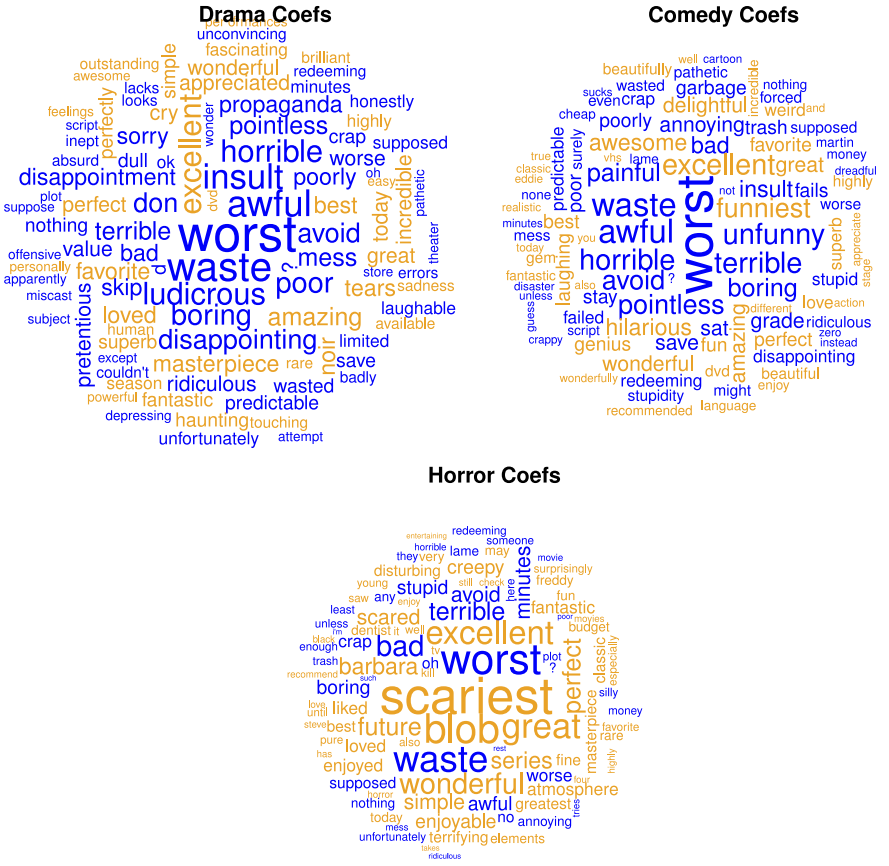

Fig. 1. Here are the coefficients found for each genre from running a separate lasso model on each genre. Orange words are predictive of good ratings and blue words are predictive of bad ratings. Absolute value of coefficients are proportional to size. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. The search for uplift

Personal Treatment Effects (PTEs) are a popular way of studying the effects of an intervention since the introduction of the Rubin Causal Model (Rubin, 1974, 2005). The premise is that in studying the effect of an intervention, we should try to estimate for each individual observation the difference between the response under the treatment and the response under the control. In a typical setting, it is only possible to observe one of these and the other will be a counterfactual. Thus, for a given individual represented by covariates $\boldsymbol{x_i}$, the PTE of that individual is $\mathrm{E}(y_i|\boldsymbol{x_i}, \text{treatment}) - \mathrm{E}(y_i|\boldsymbol{x_i}, \text{control})$. The overall treatment effect is just the average of the PTEs.

Recently, people have returned to the PTE setup to find regions where the treatment outperforms control (or vice versa). This problem is often referred to as *uplift*, and a recent paper by Guelman et al. (2014) provides a good summary of the setup and recent approaches.

Our model provides a natural approach to uplift. By letting the groups in DSL correspond to the different treatment arms, we see that we can easily estimate the uplift of a given $\boldsymbol{x_i}$ just by looking at the fits under differing treatment assignments; in the case of ordinary regression on a continuous response, as in Eq. (2), the uplift at $\boldsymbol{x_i}$ is just $\boldsymbol{x_i}^T(\boldsymbol{\Delta}_{\mathbf{Treatment}} - \boldsymbol{\Delta}_{\mathbf{Control}})$.

In addition to interaction models, such as DSL, the other main approaches to uplift involve either modified outcomes or modified covariates. Using one of these approaches allows one to model the uplift directly instead of through predictions as we do. Tian et al. (2012) provide a simple approach to finding uplift with modified covariates or modified outcomes. One nice aspect of our model is that it provides an option for more than two treatment arms; as far as we know, this is unique among algorithms that have been proposed for this problem.

In addition to the concept of uplift at a given $\boldsymbol{x_i}$, we introduce the notion of the uplift of a *treatment plan*, $\mathcal{M}$. A treatment plan maps from $\mathbb{R}^p$ into the space of potential treatments $\{1, \ldots, G\}$. For a DSL fit, the *implied treatment plan* is just $\mathcal{M}(\boldsymbol{x_i}) = \arg\max_g \boldsymbol{x_i}^T(\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\Delta}}_g)$; this is the treatment that gives the highest predicted response. Then we define the uplift of $\mathcal{M}$ to be

$$\text{Uplift}(\mathcal{M}) = \mathrm{E}_{x,y}[y|g = \mathcal{M}(x), x] - \frac{1}{G}\sum_{g=1}^{G} \mathrm{E}_{x,y}[y|g, x] \tag{6}$$

Fig. 2. Shared and separate effects found by the Data Shared Lasso model run on an IMDB sentiment analysis dataset. Splitting the effects into shared and separate makes the interpretation much easier.

where the treatment g in the second term has been assigned independently of x, as in a randomized trial. Essentially, we say that the uplift of treatment plan $\mathcal{M}$ is the expected improvement in response that we would obtain by using that plan relative to a random assignment. Then, given an independent dataset $\{(\boldsymbol{x_i}, y_i, g_i)\}_{i=1,\ldots,n}$ where treatment has been randomly assigned, we can form an unbiased estimate of Uplift($\mathcal{M}$) as

$$\widehat{\text{Uplift}}(\mathcal{M}) = \frac{1}{|\{i : \mathcal{M}(\boldsymbol{x_i}) = g_i\}|} \sum_{i:\mathcal{M}(\boldsymbol{x_i})=g_i} y_i - \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{7}$$

Note that because the estimate is just a difference in means, we can use a simple t-test to determine if the uplift of a given treatment plan is significantly different from 0 (being careful to remove the observations that are shared between the two averages). While the random treatment baseline used in Eq. (6) is nice because it guarantees that all of the datapoints will be used in calculating the t-statistic, there are other baselines a user might want as well. These include a baseline of assigning control or using the current standard for treatment assignment.

5.1. Uplift: an example

To demonstrate the applicability of DSL to the uplift problem, as well as the utility of having a concept of uplift of a treatment plan, we ran DSL on a dataset relating to a credit card promotion. A large bank decided to launch a new eco-friendly credit card and they needed to know to whom they should target the card so they conducted a randomized trial. They randomly split 100k eligible customers into either a group that received promotional materials or a group that got no special treatment (control). Associated with each customer is the treatment they received and a binary response value indicating whether they purchased the new card. Additionally, there are 60 predictor variables relating to age and credit that have each been binned into 2–11 distinct values plus a location predictor which has 51 distinct values. The dataset comes divided into a training and test set of 50K customers each.

Of the 25K customers in the training set who received the promotional materials, 20% bought the new card; for the 25K control customers, the number was 19.98%. In other words by sending promotional materials to 25K people, the bank got an additional 10 customers to sign up for the card; these are not very impressive results.

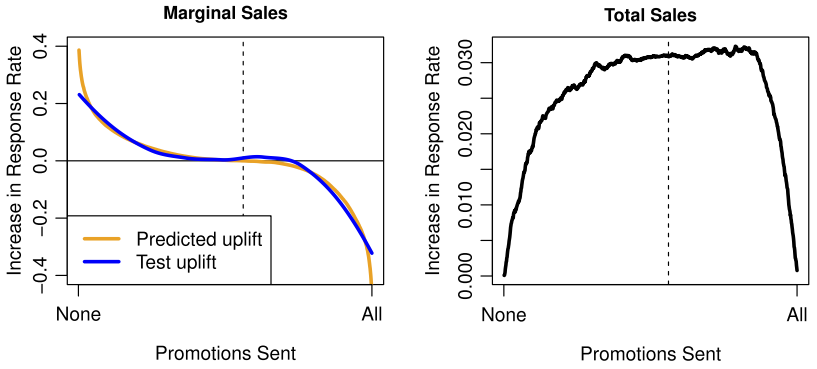

Fig. 3. The left panel is a plot of estimated uplift. Each point represents a bin of 50 observations. The curve in orange shows the predictions from our DSL model and the curve in blue is a `loess` fit to local uplift estimates based on the test set. The *x* axis is an index of the test set observations sorted by predicted uplift. The vertical dashed line in each panel represents the index where the predicted uplift is zero. The right panel shows the estimated uplift of the treatment plan that sends promotions to all customers with predicted uplift at least the predicted uplift of the observation at that index. It is essentially the integral of the test uplift in the left panel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

That said, the goal of the initial campaign was not just to measure the average effect of the treatment, but also to try to identify a sub-population where the treatment might have a larger effect. To investigate this, we applied DSL to the training dataset (using treatment and control as groups). We found that using the implied treatment plan for our DSL model gave an estimated uplift of 0.031 (p-value $\approx 10^{-66}$) relative to random treatment on the test set—a response rate of about 23%. Note how easy it is to interpret the uplift of the treatment plan. It says that if the average gain of selling a card is larger than 0.031 times the cost of promotion, then we would make money by following the DSL treatment plan.

To measure our efficacy in estimating uplift, we sorted our test set by predicted uplift. Then, we split the sorted observations into bins of 50 and estimated the uplift in each bin. The left panel of Fig. 3 shows a `loess` (Cleveland and Grosse, 1991) regression fit to those estimates, as well as the predicted uplift curve from our DSL model. It seems that our model does a good job of estimating the uplift. The right panel of Fig. 3 shows the uplift of the treatment plan for sending the promotions to those customers. Essentially, it is an integrated version of the left panel, and it can aid interpretation.

Our previous analysis assumed that we want to send the promotion to anyone for whom we think it will increase their probability of buying the card. In reality, this is likely not the case. By the time that our model predicts an uplift of zero, we have already sent promotions to many people with an uplift close to zero. This is almost certainly a waste of money. Suppose that sending a promotion costs $10 and buying a new card is worth $100 to the bank. In this case, we would want to send promotions until the marginal expected benefit of sending promotions is 0.1 additional cards sold. By reading the left panel of Fig. 3 we see that this occurs after sending the promotion to about a fifth of the customers. We can then estimate the total effect of this intervention by examining the right panel of the same figure to see that we expect an increase in response of about 2% if we send that number of promotions.

In addition to using our analysis to learn who to target for the next wave of promotions, we can also gain some insight into what kind of people respond either positively or negatively to the promotion. Some care must be taken in analyzing the coefficients from a multivariate model as they represent the effect of the variable in the presence of the other included predictors. In a relatively small model this can be overcome, but our final model ended up using several hundred variables (it can be larger than 60 because our categorical variables have been re-coded as binary variables). Additionally, many of our predictors are correlated. For these reasons, it is revealing to look at the variables one at a time to see how well they correspond to our predicted uplift.

Since we have categorical variables with differing numbers of levels, we used adjusted R-squared to pick the variable that best corresponds to predicted uplift on the test set. The variable that was most predictive is the number of revolving accounts (the most common type of revolving account is a credit card) which had an adjusted R-squared of 0.448 compared to the 2nd best predictor—the number of revolving accounts that satisfy some additional criteria (adjusted R-squared of 0.292). Fig. 4 shows the relationship between number of revolving accounts and predicted uplift. It seems that the promotion is most effective for people with an intermediate number of revolving accounts; people with higher numbers of revolving accounts are likely to be negatively affected by the promotion.

6. Conclusion

In this paper we outline a new technique for partially pooling regression problems, the Data Shared Lasso. Our approach exploits the benefits of an ℓ_1 penalty on the coefficients to add sparsity and to improve interpretability. This was demonstrated by a real world sentiment analysis example.

In addition to presenting DSL and discussing some of the relevant details of fitting the model, we also compared it to some related methods including some from the multi-task learning literature. What we learned is that DSL fits coefficients that

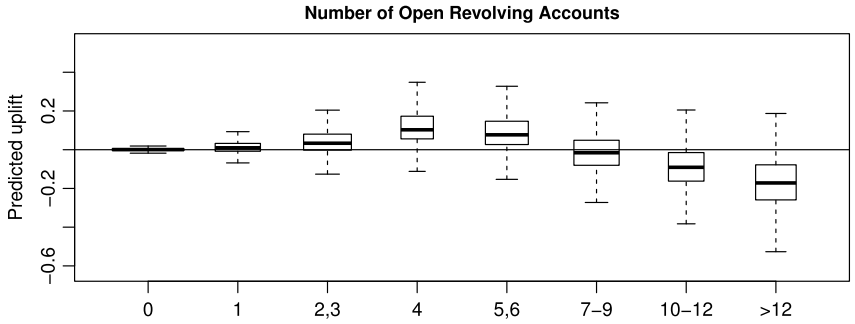

Fig. 4. This boxplot shows the relationship between the number of open revolving accounts and predicted uplift on the test set. Intermediate numbers of revolving accounts correspond to high values of predicted uplift. High numbers of revolving accounts are associated with negative predicted uplift.

have a different sparsity pattern than other related methods. In particular, DSL is the only model examined that decomposes the effects into shared and separate effects. This can be contrasted to the mean and interaction effects found by typical interaction models. While the average effect of a word on the rating of a movie may have some value in interpretation, it is clear that the shared effect from DSL has value; this is because it represents the extent to which a meaning is common across genres. As we saw in Section 3.3, some of the multi-task learning approaches consider a coefficient similar across groups even if it has differing signs. While this may be appropriate for some problems, it clearly is not for examples such as the IMDb problem.

We also discussed *uplift*, an increasingly important problem in modern statistics. We used a consumer data example in this case because it has a large sample size and thus made for easy evaluation of success. There are also many uplift problems relating to modern medicine. When clinical drug trials are conducted, it is possible that some drugs will be discounted because they do not have a net positive effect. However, as shown in the example in Section 5.1, it is possible for a subgroup to have a significant treatment effect even if the overall treatment effect is negligible. Application of DSL to clinical trial data could help investigators find specific patient strata where a given drug is more effective than its competitors. This could spawn future studies to examine the efficacy on that subgroup.

DSL is a new technique, so there is much more research that can be done on its use. One obvious addition that could be made is the appropriation of some sort of significance level for each of the coefficients. As DSL is an augmented lasso, it should be possible to extend the results of Lockhart et al. (2014) or Lee et al. (2013) to derive valid p-values for selected predictors.

Acknowledgments

The second author was supported by NSF Grant DMS-99-71405 and National Institutes of Health Contract N01-HV-28183.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.csda.2016.02.015.

References

Argyriou, A., Evgeniou, T., Pontil, M., 2008. Convex multi-task feature learning. Mach. Learn. 73, 243–272.
Bien, J., Taylor, J., Tibshirani, R., 2013. A lasso for hierarchical interactions. Ann. Statist. 42, 1111–1141.
Chen, A., Owen, A.B., Shi, M., 2013. Data enriched linear regression. ArXiv e-prints. arXiv:1304.1837.
Cleveland, W.S., Grosse, E., 1991. Computational methods for local regression. Stat. Comput. 1, 47–62.
Donoho, D.L., 2006. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. Comm. Pure Appl. Math. 59, 797–829.
Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. 33, 1.
Gu, Q., Zhou, J., 2009. Learning the shared subspace for multi-task clustering and transductive transfer classification. In: Ninth IEEE International Conference on Data Mining, 2009. ICDM'09. IEEE, pp. 159–168.
Guelman, L., Guillen, M., Pérez-Marín, A.M., et al. 2014. Optimal personalized treatment rules for marketing interventions: A review of methods, a new proposal, and an insurance case study. UB Riskcenter Working Papers Series.
Hastie, T., Tibshirani, R., Botstein, D., Brown, P., 2001. Supervised harvesting of expression trees. Genome Biol. 2, 1–0003.
Lee, J.D., Sun, D.L., Sun, Y., Taylor, J.E., 2013. Exact post-selection inference with the lasso. arXiv Preprint arXiv:1311.6238.
Lim, M., Hastie, T., 2013. Learning interactions through hierarchical group-lasso regularization. arXiv Preprint arXiv:1308.2719.
Lockhart, R., Taylor, J., Tibshirani, R.J., Tibshirani, R., 2014. A significance test for the lasso. Ann. Statist. 42, 413–468. http://dx.doi.org/10.1214/13-AOS1175.
Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C., 2011. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Portland, Oregon, USA, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.
Ming, Y., Lin, Y., 2005. Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. Ser. B 68, 49–67.
Ollier, E., Viallon, V., 2014. Joint estimation of K related regression models with simple L_1-norm penalties. arXiv Preprint, arXiv:1411.1594.

Ollier, E., Viallon, V., 2015. Regression modeling on stratified data: automatic and covariate-specific selection of the reference stratum with simple L_1-norm penalties. arXiv Preprint, arXiv:1508.05476.
Rubin, D.B., 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. J. Educ. Psychol. 66, 688.
Rubin, D.B., 2005. Causal inference using potential outcomes. J. Amer. Statist. Assoc. 100.
Ruczinski, I., Kooperberg, C., LeBlanc, M., 2003. Logic regression. J. Comput. Graph. Statist. 12, 475–511.
Tian, L., Alizadeh, A., Gentles, A., Tibshirani, R., 2012. A simple method for detecting interactions between a treatment and a large number of covariates. arXiv Preprint arXiv:1212.2995.
Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B 58, 267–288.
Tibshirani, R., Bien, J., Friedman, T., Hastie, J., Simon, J., Taylor, N., Tibshirani, R., 2012. Strong rules for discarding predictors in lasso-type problems. J. R. Stat. Soc. B 245–266.
Zhao, P., Rocha, G., Yu, B., 2009. The composite absolute penalties family for grouped and hierarchical variable selection. Ann. Statist. 3468–3497.