Human Guided Linear Regression with Feature-Level Constraints

Aubrey Gress and Ian Davidson
Department of Computer Science, University of California, Davis
1 Shields Avenue, Davis, CA, 95616
{adgress@ucdavis.edu, davidson@cs.ucdavis.edu}

Abstract

Linear regression methods are commonly used by both researchers and data scientists due to their interpretability and their reduced likelihood of overfitting. However, these methods can still perform poorly if little labeled training data is available. Typical methods used to overcome a lack of labeled training data somehow involve exploiting an outside source of labeled data or large amounts of unlabeled data. This includes areas such as active learning, semi-supervised learning and transfer learning, but in many domains these approaches are not always applicable because they require either a mechanism to label data, large amounts of unlabeled data or additional sources of sufficiently related data. In this paper we explore an alternative, **non-data** centric approach. We allow the user to guide the learning system through three forms of feature-level guidance which constrain the parameters of the regression function. Such guidance is unlikely to be perfectly accurate, so we derive methods which are robust to some amounts of noise, a property we formally prove for one of our methods.

1 Introduction

Linear regression methods are often used in practice due to their simplicity, interpretability and resilience to overfitting (Friedman, Hastie, and Tibshirani 2001), but they can perform poorly if too little labeled data is available. This situation can be addressed by several branches of Machine Learning and Statistics, such as semi-supervised learning, transfer learning and active learning, but each of these areas has strong data requirements that may not always hold. Semi-supervised learning (Zhu 2005) methods can make use of unlabeled data, but they require large amounts of it and for the "cluster assumption" to hold (Singh, Nowak, and Zhu 2009). Active Learning (Settles 2010) methods can increase the amount of labeled data, but require both unlabeled data and a trained human annotator. Transfer learning (Pan and Yang 2010) methods can incorporate external data sets, but require the data to be sufficiently similar in order for standard transfer learning techniques to succeed.

We explore an alternative, non-data centric approach - using knowledge the user may have about the *features*. While Machine Learning methods generally treat features as a "black box," users may have a wealth of understanding about the relationships between the features and the output. For example, a scientist performing a longevity study may be confident that life expectancy decreases with respect to smoking habits.

We propose three forms of feature level knowledge which we use to constrain the solution space of linear regression, thereby reducing the necessary amount of training data. These constraints are on: parameter signs, relative parameter effect ordering and pairwise parameter signs. This allows a rich set of feature level guidance to be provided, such as "feature i will have a positive impact on the label" or "feature j will have a more positive impact than feature k." This feature level guidance forms a new set of additional information users can provide even in situations where generating additional labeled data is not possible or cost effective. This type of guidance can be feasible in domains where the features have a semantic meaning, such as text, medicine, education and sales data. Additionally, in order to account for potential errors in the guidance, we develop methods of harnessing this knowledge which are robust to certain amounts of noise in the feature guidance.

Little work has been done on feature-level guidance for regression. The most ubiquitous line of work on constraining the parameters of linear methods is through sparsity promoting regularization such as the Lasso (Tibshirani 1996) and the Elastic Net (Zou and Hastie 2005), but these methods do not take in additional guidance the user may be able to provide. The closest work to our own is on "signed" regression which alleviates the small data problem by taking the standard least squares objective and allowing guidance on the signs of the coefficients of the parameters (Breiman 1995; Chen and Plemmons 2009; Slawski and Hein 2011; Slawski, Hein, and others 2013), but they do not have mechanisms for allowing noisy guidance (i.e. the user is sometimes wrong) and are limited to just constraints on the coefficient signs.

Our contributions are:

- We introduce Parameter Sign Constrained Regression (PSCR), which allows the user to provide guidance on parameter signs and, unlike previous work, is robust to noisy sign guidance (Section 3).
- We introduce Parameter Relative Constrained Regression (PRCR), which incorporates relative ordering of the coef-

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ficients (Section 3). This allows a user to state that feature i should have a greater positive impact on the label than feature j.

- We introduce Pairwise Parameter Sign Constrained Regression (PPSCR), which constrains pairs of coefficients to have the same signs. This allows a user to state that features i and j should have the same signed impact on the label, but is unsure if it is a positive or negative impact.
- We propose a novel transfer mechanism for generating the guidance which makes *weaker* assumptions than prior work, allowing the use of transfer learning in areas where the source data is significantly different from the target (Section 3.5).
- We present a theoretical analysis of our PSCR formulation and an associated bound which shows it can perform well even when the sign guidance is noisy (Section 4).
- Our experimental results show our formulations outperform natural baselines and prior work (Section 5).

2 Previous Work

Our work falls into the general area of how to overcome limitations in the amount of training data through feature level guidance.

Regularization. A typical method is to modify regularizers to penalize the use of certain features such as in the Ridge or the Lasso (Hoerl and Kennard 1970; Yuan and Lin 2006). However, this requires the user to specify separate real-valued regularization parameters for each feature, a challenging task because these values do not necessarily correspond to some physical quantity. Methods exist for "learning" these regularization parameters (Boer and Hafner 2005), but this makes the learning problem *harder* by increasing the number of parameters that need to be learned, rather than making it easy.

Feature Level Guidance For Classification. Another line of work considers "labeled features" (Raghavan, Madani, and Jones 2006; Raghavan and Allan 2007; Druck, Mann, and McCallum 2008; Settles 2011). These methods generally have human annotators create associations between features and classes, followed by somehow converting these features into instances and using standard supervised learning algorithms. Alternatively, other methods use the guidance to modify priors over the solution space. These works are different from ours because they only apply to the classification setting, they do not provide any theoretical justification for their methods and they do not explore alternative forms of feature guidance.

Previous Work on Signed Regression. Signed and nonnegative least squares are extensions of least squares which allow the user to specify the signs of the estimated parameters (Breiman 1995; Slawski and Hein 2011; Slawski, Hein, and others 2013). While these methods have a stronger theoretical basis, they are not robust to noisy guidance and can perform arbitrarily poorly if the signed guidance is incorrect. They also do not explore alternative forms of guidance.

Transfer Learning: Transfer learning is a setting where the user wishes to improve performance on a target task with data from one or more source tasks (Pan and Yang 2010). Standard methods for transfer learning generally reduce to somehow transferring the prediction made by the source function, either through regularization or by making the source predictions a new feature (Daume III and Marcu 2006; Tommasi, Orabona, and Caputo 2010; Gong et al. 2012; Fernando et al. 2013; Kuzborskij and Orabona 2013; Patricia and Caputo 2014). However, if the source data is not sufficiently similar then using transfer learning can lead to *worse* performance than simply using the target data (Pan and Yang 2010).

Relative Instance Guidance: A line of work similar in spirit but orthogonal in approach considers alternative forms of supervision for regression (Zhu and Goldberg 2006; 2007; Sculley 2010; Gress and Davidson 2016). In contrast to the standard instance-label guidance used in regression, they consider "relative" supervision of the form "$f(x_i) > f(x_j)$". While both lines of research mitigate the small data problem through alternative forms of guidance, theirs is at the instance level while ours is at the feature level.

3 Our Method

In this section we discuss our methods for adding guidance to regression. First we explore a formulation for signed guidance which is noise tolerant and then we explore relative and pairwise sign constraints. After presenting our framework we describe practical methods for generating this guidance. As in the standard supervised learning setting, for all methods we assume we are given a labeled data set $\{X, Y\}$ where X is a $n \times p$ matrix of n instances and p features and Y is a $n \times 1$ vector of responses.

3.1 Parameter Sign Constrained Regression

We assume that, in addition to the training data, we also have access to e, a $p \times 1$ vector encoding a set of sign guidance where e_i is 1 if the sign of the coefficient should be nonnegative, -1 if nonpositive and 0 if no guidance is provided for the coefficient. Previous work has modeled this guidance using hard constraints, but this can lead to **overfitting the sign guidance**. Thus, our goal is to model this problem in a way that a subset of the guidance can be adaptively ignored if it leads to a worse fit. Letting C and λ be regularization parameters, this can be modeled as the following discrete optimization problem:

$$\begin{aligned} \min_{\beta,\xi} \; & \frac{1}{n}||X\beta - y||^2 + C||\beta||^2 \\ s.t. \; & (1-\xi_i)e_i\beta_i \geq 0, \forall e_i \in e \\ & \xi \in \{0,1\}^p \\ & \sum_i \xi_i \leq \lambda \end{aligned} \tag{1}$$

The objective function is the same as the Ridge (i.e. ℓ_2 regularized linear regression) but with the addition of ξ and the sign constraints on β. The difference between this and previous work is the introduction of the ξ_i, a set of discrete variables which, when set to 1, deactivate the corresponding sign constraint. This problem allows up to λ constraints to be ignored and reduces to previous work when $\lambda = 0$.

While this models the problem we want to solve, it is a difficult discrete optimization problem which can be hard to solve in practice. Thus, we relax it to the following continuous, convex optimization problem:

Primal Form.

$$\begin{aligned} \min_{\beta,\xi} \ & \frac{1}{n}||X\beta - y||^2 + C||w||^2 \\ s.t. \ & E\beta + \xi \geq 0 \\ & \xi \geq 0 \\ & \sum_i \xi_i \leq \lambda \end{aligned} \tag{2}$$

where E is a diagonal matrix encoding the sign constraints, with $E_{ii} \in \{-1, 0, 1\}$ depending on the sign guidance. The ξ_i are now, in a manner analogous to the Support Vector Machine (SVM), slack variables which control the extent to which the sign constraints can be violated (Friedman, Hastie, and Tibshirani 2001). In the same way that slack variables can prevent SVM from overfitting the data, ξ can prevent our method from overfitting the sign guidance. Additionally, this problem is convex because the objective is a convex function and the constraints are all affine, so it can be solved efficiently using standard convex optimization libraries.

Dual Problem. To better understand equation 2 we now compute its dual problem. The derivation is done using Lagrange Duality and is included in the **supplementary materials**.

Dual Form.

$$\begin{aligned} \min_{\alpha,\gamma} \ & \frac{1}{4C}\alpha'(EE' - B'MB)\alpha + y'MB\alpha + \lambda\gamma \\ s.t. \ & \alpha, \gamma \geq 0 \\ & \alpha \leq \gamma \mathbf{1} \end{aligned} \tag{3}$$

where

1. $B = XE'$
2. $M = (nCI + XX')^{-1}$
3. $\delta = M(-XE'\alpha + 2Cy)$
4. $\beta = \frac{1}{2C}(X'\delta + E'\alpha)$

This dual is similar to the Ridge's dual but with the addition of the dual variables α and γ which capture the constraints from equation 2. Considering this dual formulation is useful because this problem can be less computationally intensive to solve when p (the number of features) is much greater than n (the number of instances) and the number of sign constraints.

Impact of Active Sign Constraints on Dual. We can better understand the impact of sign constraints on PSCR by contrasting the behavior of the dual with another way of enforcing sign constraints - a thresholding procedure wherein the solution to the Ridge β_{Ridge} is first computed, then its entries are thresholded to respect the sign constraints. e.g. if $E_{ii} = 1$, then set β_i to $\max((\beta_{Ridge})_i, 0)$. By considering equation 3 we will show our method has a much different result than thresholding.

From the KKT conditions we know that if $\alpha = 0$, then none of the sign constraints are active (i.e. none of them hold with equality), so the solution reduces to that of the Kernel Ridge ($\beta_{Ridge} = X'My$). This is identical to the solution one would get when using the simple thresholding procedure we just mentioned. If $\alpha \neq 0$, then some subset of the sign constraints are active and α induces a data dependent translation. By plugging in δ to item (4) above we can better understand the impact of an active sign constraint:

$$\begin{aligned} \beta &= \frac{1}{2C}(X'\delta + E'\alpha) && (4) \\ &= \beta_{Ridge} + \frac{1}{2C}(I - X'MX)E'\alpha && (5) \end{aligned}$$

Now consider the case where only a single entry of α is nonzero, indicating that one of the sign constraints is violated by the Ridge solution. Thresholding would solve this by only modifying the corresponding coefficient, but our method translates the Ridge solution by $\frac{1}{2C}(I - X'MX)E'\alpha$, which may potentially modify **all** the other coefficients. Intuitively, our method accommodates the sign constraint by modifying the entries with non-active sign constraints in order to still fit the data well, as opposed to thresholding which considers each coefficient in isolation. In a certain sense, our method finds an **alternative** solution to β that fits the data well while enforcing the relaxed sign constraints.

3.2 Parameter Relative Constrained Regression

The second form of guidance we consider is pairwise guidance. Let P be a set of pairs where each $p_i \in P$ is a tuple (j, k) indicating that coefficient j is greater than coefficient k. This guidance can similarly be used to define a discrete optimization problem, but for the sake of brevity we immediately jump to its continuous relaxation:

$$\begin{aligned} \min_{\beta,\xi} \ & \frac{1}{n}||X\beta - y||^2 + C||\beta||^2 \\ s.t. \ & \beta_j - \beta_k + \xi_i \geq 0, \forall p_i \in P \\ & \xi \geq 0 \\ & \sum_i \xi_i \leq \lambda \end{aligned} \tag{6}$$

Similar to equation 2, the first constraint enforces the guidance the user provides and ξ is a slack parameter to prevent overfitting. Also similarly, this problem is convex. The key difference between this and PSCR is that the user does not need to know the signs of the coefficients, just their relative ordering, which may be easier to generate in some settings. A similar formulation was considered for "nearly" Isotonic Regression (Tibshirani, Hoefling, and Tibshirani 2011).

This problem can be more concisely written by replacing the first constraint of equation 6 with $E\beta + \xi \geq 0$, where for every pair $p_i = (j, k)$, there is a corresponding row E^i where the entries are $E^i_j = 1$, $E^i_k = -1$ and 0 otherwise. By

using this form the problem becomes identical to equation 2, but with a different matrix encoding the constraints and a different interpretation of the slack variables ξ. Additionally, the **dual will take the same form**.

3.3 Pairwise Parameter Sign Constrained Regression

The final form of guidance we consider is pairwise sign guidance. Let P be a set of pairs where each $p_i \in P$ is a tuple (j, k) indicating that coefficients j and k should have the same (or opposite) sign. Our strategy for modeling this is by constraining the product $E_{jk}\beta_j\beta_k$ to be positive, where $E_{j,k}$ is positive if they should have the same sign, and negative otherwise. This cannot be modeled through the previously proposed formulation because while the other forms of guidance lead to affine constraints, this leads to nonconvex quadratic constraints which most standard continuous optimization libraries cannot model. Thus, we propose the following formulation:

$$\begin{aligned} \min_{\beta,\xi} \ & \frac{1}{n}||X\beta - y||^2 + C||\beta||^2 \\ & + \lambda_2 \sum_{p_i \in P} \max(-(E_{j,k}\beta_j\beta_k + \xi_i), 0) \\ s.t. \ & \xi \geq 0 \\ & \sum_i \xi_i \leq \lambda_1 \end{aligned} \tag{7}$$

where λ_1 and λ_2 are regularization parameters. The key difference between this and previous formulations is that we have moved the guidance constraint to the objective. This term will be 0 when the entire set of pairwise sign guidance is respected, and positive otherwise. This optimization problem is nonconvex, but a local optimum can be found using standard optimization libraries such as SciPy's optimization module (Jones et al. 2001).

3.4 User Generated Guidance

Ideally, parameter constraint guidance could be estimated from the training data, but this guidance will be noisy if little training data is available. The user is a potentially better source, but it is not clear users will be able to accurately provide it because they won't necessarily know such properties of β. This problem is challenging because it requires knowing the relationship between one or more features and the response *within the context of the other features*. This leads to a chicken-and-the-egg type problem: we use this guidance to more accurately estimate β, but to get it we need an accurate estimate of β.

To solve this problem we propose the user answer **simpler** queries that assess more basic properties of the relationships between covariates and the response. Rather than ask for properties of the ground truth coefficients, we instead ask for guidance when considering individual or pairs of features *independently* of the others. The advantage of this method is that while users may not be able to accurately answer questions about the ground truth β when considering *all* the features, they can likely answer these same questions when considering a small *subset* of the features. For example, in a medical domain a user could likely say with confidence that increasing the number of cigarettes smoked per day will reduce average life expectancy.

It is important to note that this guidance is a *proxy* for the sign or pairwise properties of β. For example, just because coefficient i is positive when regressing on only feature i does not mean β_i will be positive. This is why our methods of relaxing the guidance are important - it can prevent overfitting when the guidance is noisy.

3.5 Parameter Constraint Transfer

While our proposed guidance can be provided by domain experts, for domains where this is not possible, it can instead be *learned* through transfer learning. To do this, rather than have a human generate the constraints manually, the constraints are generated by estimating them from a related source domain which shares the same feature set. For example, if the task is to predict the probability of a customer buying a TV given a set of demographics information, then the sign constraints could be generated by estimating customer purchasing habits of other related products, and then use these estimates to generate the sign guidance. Formally, given a source task S, we can compute the Ridge solution β_S on the source data, then generate sign/relative/pairwise sign constraints from the entries of β_S. e.g. if $(\beta_S)_i > 0$, then set $E_{ii} = 1$. This leads to a new form of transfer learning, "Parameter Constraint Transfer," which makes different modeling assumptions from previous work. While many transfer learning methods assume the source and target functions are similar, ours makes the weaker assumption that properties such as sign and relative ordering of the coefficients are shared between the domains. This can allow Parameter Constraint Transfer to succeed in areas where standard transfer learning methods fail.

4 Theory

In this section we present an error bound for Parameter Sign Constrained Regression. This bound provides theoretical justification for our formulation as well as providing insight into the relationship between the performance of our method and the accuracy of the sign guidance. We start with a high-level description of the bound, followed by the bound and an analysis of how the regularization parameter λ relates to the accuracy of the sign guidance.

Bound Description. Recall that λ controls the extent to which our method can violate the given guidance. Intuitively, Theorem 1 states that the error of our estimate is bounded by the sum of two key terms: the "excess error" due to making λ too *small*, and λ itself. This leads to a trade off between these two terms - increasing λ will decrease the former term but increase the latter and vice versa. However, when all the sign guidance is completely accurate, the excess error term will be 0 for all λ, so this sum is minimized when $\lambda = 0$. Along these lines, this bound decreases when the guidance is more accurate. Thus, while our method is robust to noise, accurate guidance will lead to better estimates.

Error Bound. For our analysis we assume $y_i = x_i'\beta^* + \epsilon_i$

for all (x_i, y_i), where the ϵ_i are independent and identically distributed Gaussian random variables.

First, some notation:

- β_λ: Closest approximation to β^* that lies within the feasible set of equation 2 for the given value of λ. i.e. it is the solution to the problem $\min_\beta ||\beta^* - \beta||^2$ subject to the constraints in equation 2.
- $\hat{\beta}_\lambda$: Estimate of β from solving equation 2 for hyperparameter λ.
- $\hat{\delta} = \hat{\beta}_\lambda - \beta_\lambda$
- $\hat{\delta}_p, \hat{\delta}_n$: the positive and negative components of $\hat{\delta}$ respectively. Intuitively, the components of $\hat{\beta}_\lambda$ which overestimate and underestimate β^*.

For the sake of clarity we also make the following assumptions:

- $\frac{1}{n}||X'X||_\infty = 1$
- Sign constraints are provided for every feature.
- All the sign constraints are nonnegative. This does not restrict the generality of our results because any nonpositive constraint can be transformed into a nonnegative constraint by scaling the associated feature by -1.

At a high level, the bound is derived by considering how much the coefficients of $\hat{\beta}_\lambda$ over and under-estimates β^*. First we bound the negative components:

Lemma 1 (Bounding $||\hat{\delta}_n||$)**.**

$$||\hat{\delta}_n|| \leq g_n(\hat{\delta}) = ||\beta^*|| + \lambda$$

Bounding the norm of the negative component is straightforward using the constraints from equation 2. Next, we bound the positive components:

Lemma 2 (Bounding $||\hat{\delta}_p||$)**.** *With at least probability* $1 - 2p^{-M^2}$*:*

$$||\hat{\delta}_p|| \leq g_p(\hat{\delta}) = \mathcal{O}(\frac{1}{C}(||\beta^* - \beta_\lambda|| + A + ||\beta_\lambda|| + (C+1)||\hat{\delta}_n||)$$

where $A = \sqrt{\frac{2\log p}{n}}\sigma(1+M)$

The proof of this result is more involved because equation 2 does not explicitly bound the overestimation of $\hat{\beta}_\lambda$. As a result, this bound is a function of several terms including the excess error, a standard concentration of measure term and $||\hat{\delta}_n||$. Finally, we combine these results to get:

Theorem 1 (Error Bound)**.** *With at least probability* $1 - 2p^{-M^2}$*:*

$$||\beta^* - \hat{\beta}_\lambda|| \leq \mathcal{O}(||\beta^* - \beta_\lambda|| + g_p(\hat{\delta}_p) + g_n(\hat{\delta}_n))$$

Given the previous lemmas, this result is straightforward to show using the triangle inequality.

4.1 Interpreting λ

One way of interpreting λ is it controls the extent the algorithm is allowed to deviate from the provided guidance. In the theoretical setting, we can better analyze this by setting $\lambda = ||\beta^*||k$, where k is the relative error between β^* and β_λ (i.e. $k = \frac{||\beta^* - \beta_\lambda||}{||\beta^*||}$). For this value of λ the right hand side of the second item in Theorem 1 becomes a function of $(1+k)||\beta^*||$. This term decreases with the relative error k, which can be done in two ways: by providing more accurate sign guidance or by tuning λ to increase the size of the feasible set of equation 2. Thus, while our method can accommodate noisy guidance, it will perform better with more accurate guidance.

5 Experiments

Our first set of experimental questions explore the effectiveness of our formulations for feature guidance:

- How well do our proposed methods (PSCR, PRCR, PPSCR) perform relative to standard methods which cannot take the guidance we proposed (Table 3)?
- How beneficial is our method of using soft constraints compared to using hard constraints when the same guidance is used (Table 4)?

Our second set of experimental questions explore the effectiveness of our methods for generating the feature guidance. Due to space constraints, for these experiments we focus on sign guidance.

- How well does our proposed method of generating sign constraints via the user compare to estimating the constraints automatically from the training data (Table 5)?
- How well does our proposed method of generating constraints using transfer learning compare to standard methods of performing transfer (Table 6)?

The methods and data sets we used are described in Tables 1 and 2. We generated the guidance by simulating a domain expert. Specifically, we randomly selected a subset of features, estimate the signs (or relative orderings) of regression coefficients of each feature separately using ordinary least squares on a validation set. i.e. to generate the sign of coefficient i, we solved the problem $\hat{\beta}_i = \arg\min_{\beta_i} ||X_i\beta_i - Y||^2$ and used the sign to generate the guidance.

For all methods all regularization parameters were tuned on a validation set. Reported results are the mean of 30 train/test splits. Values in parentheses indicate 95% confidence interval. **More extensive experiments and learning curves are presented in the supplementary material. Code and processed data sets are available at https://github.com/adgress/AAAI2018.**

Comparisons with Baselines: For our first experiments we compared the performance of our methods to a set of baseline methods. These results are in Table 3. These results show that our methods combined with our guidance generation methods can dramatically outperform standard methods. In particular, our methods performed much better than

Method	Description
Ridge (Friedman, Hastie, and Tibshirani 2001)	Least squares with ℓ_2 regularization.
Lasso (Tibshirani 1996)	Least squares with ℓ_1 regularization.
Nonnegative (Slawski, Hein, and others 2013)	The Ridge with nonnegative constraints.
Signed Ridge (Slawski, Hein, and others 2013)	The Ridge with sign guidance. For these experiments the sign guidance was generated in the same way as for our method.
Transfer Ridge (Pan and Yang 2010)	The Ridge where source predictions were added as an additional feature.
PSRC: p signs	Our method with sign guidance. The number of sign constraints is equal to the p, the number of features.
PRCR: p pairs	Our method with pairwise guidance. The number of pairwise constraints is equal to p, the number of features.
PPSCR: p pairs	Our method with pairwise sign guidance The number of pairwise sign constraints is equal to p, the number of features.
Transfer PSRC	Our method with sign guidance where the constraints were generated using the method outlined in 3.5 using a related source domain.
PSRC: Training Guidance	Our method with sign guidance where the constraints were estimated from the given training data.

Table 1: Methods we used in our experiments.

the Ridge with nonnegative constraints, showing that providing more accurate sign guidance is better than arbitrarily constraining the signs to be nonnegative.

While relative guidance seemed effective, these experiments suggest sign and pairwise sign guidance performs better. This suggests sign guidance does a better job of constraining the feasible space than relative guidance.

The Benefits of Soft Constraints: For these experiments we tested the impact of the "soft guidance" our method uses. Recall that previous work used hard constraints, which we argue can lead to overfitting the guidance, while our work includes slack variables to prevent this phenomenon. Due to space constraints we focused on sign constraints. Table 4 shows the results of our method, along with the "Signed Ridge," which does not have a mechanism for relaxing the sign guidance, where we added *the same set of sign constraints*. These results show our method performs much better. This shows our relaxation can play an important role in preventing overfitting.

Constraint Generation: For our next experiments we compared the impact of how we generate the guidance. Again, due to space constraints, we focus on sign guidance. Table 5 shows the results of our method when the signs were generated by simulating a human expert and when the sign guidance was generated from the given training set by first computing the Ridge solution β_{Ridge} and using the signs of β_{Ridge} as constraints. These results show our method performs much better when the guidance is generated from an outside source of knowledge. This intuitively makes sense, because estimating the signs from the training data is in some sense "redundant" because the same data is used to estimate the parameters.

Parameter Constraint Transfer: Our final experiments tests generating the constraints using transfer learning. For these experiments we used a related source domain to generate the constraints, which were then used for the target task. The constraints were generated by computing the Ridge solution β_S on the source data, then generating the constraints from the signs of β_S - e.g. if $(\beta_S)_i > 0$ then set E_{ii} to 1, otherwise set it to -1. Table 6 compares the results of our method, with and without transfer, and the "Transfer Ridge" where source predictions were used as an additional feature. While simple, this method experimentally works very well and is representative of many recent transfer learning methods (Tommasi, Orabona, and Caputo 2010; Kuzborskij and Orabona 2013; Patricia and Caputo 2014).

Our results show our method with transfer performs better than the Transfer Ridge, but worse than our method where the sign guidance was generated through a simulated human. This intuitively makes sense because the transfer guidance is in some sense "noisier" due to it coming from a different domain. In spite of this, the fact that our method outperforms the Transfer Ridge is important because it shows Parameter Constraint Transfer can be a more effective means of transferring knowledge.

6 Conclusion

We proposed novel ways of constraining parameter along with formulations that are more robust to overfitting by allowing the guidance to be relaxed. We also presented two practical methods to provide this guidance: through simpler pointwise and pairwise queries and through transfer learning. We also theoretically analyzed our signed guidance method which provides theoretical justification for the method and explains how our method is more robust to noisy guidance. In future work we will consider other forms of coefficient guidance and extend the theoretical framework to the forms of pairwise guidance we proposed.

References

Boer, P., and Hafner, C. M. 2005. Ridge regression revisited. *Statistica Neerlandica* 59(4):498–505.

Breiman, L. 1995. Better subset regression using the nonnegative garrote. *Technometrics* 37(4):373–384.

Chen, D., and Plemmons, R. J. 2009. Nonnegativity constraints in numerical analysis. *The birth of numerical analysis* 10:109–140.

Daume III, H., and Marcu, D. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 101–126.

Druck, G.; Mann, G.; and McCallum, A. 2008. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 595–602. ACM.

Fernando, B.; Habrard, A.; Sebban, M.; and Tuytelaars, T. 2013. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2960–2967.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2001. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2066–2073. IEEE.

Gress, A., and Davidson, I. 2016. Probabilistic formulations of regression with mixed guidance. In *ICDM*, 895–900. IEEE.

Harrison, D., and Rubinfeld, D. L. 1978. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management* 5(1):81–102.

Hoerl, A. E., and Kennard, R. W. 1970. Ridge regression: applications to nonorthogonal problems. *Technometrics* 12(1):69–82.

Jones, E.; Oliphant, T.; Peterson, P.; et al. 2001–. SciPy: Open source scientific tools for Python.

House sales in king county, wa. `https://www.kaggle.com/harlfoxem/housesalesprediction`.

Kuzborskij, I., and Orabona, F. 2013. Stability and hypothesis transfer learning. In *ICML*, 942–950.

Lichman, M. 2013. UCI machine learning repository.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on* 22(10):1345–1359.

Patricia, N., and Caputo, B. 2014. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *CVPR*, 1442–1449.

Raghavan, H., and Allan, J. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *SIGIR*, 79–86. ACM.

Raghavan, H.; Madani, O.; and Jones, R. 2006. Active learning with feedback on features and instances. *JMLR* 7(Aug):1655–1686.

Rousseauw, J.; Du Plessis, J.; Benade, A.; Jordann, P.; Kotze, J.; Jooste, P.; and Ferreira, J. 1983. Coronary risk factor screening in three rural communities. *South African Medical Journal* 64(430-436):216.

Sculley, D. 2010. Combined regression and ranking. In *SIGKDD*, 979–988. ACM.

Settles, B. 2010. Active learning literature survey. *University of Wisconsin, Madison* 52(55-66):11.

Settles, B. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *EMNLP*, 1467–1478. Association for Computational Linguistics.

Singh, A.; Nowak, R.; and Zhu, X. 2009. Unlabeled data: Now it helps, now it doesn't. In *NIPS*, 1513–1520.

Slawski, M., and Hein, M. 2011. Sparse recovery by thresholded non-negative least squares. In *NIPS*, 1926–1934.

Slawski, M.; Hein, M.; et al. 2013. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics* 7:3004–3056.

Tibshirani, R. J.; Hoefling, H.; and Tibshirani, R. 2011. Nearly-isotonic regression. *Technometrics* 53(1):54–61.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Tommasi, T.; Orabona, F.; and Caputo, B. 2010. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, 3081–3088. IEEE.

Vanlehn, K.; Lynch, C.; Schulze, K.; Shapiro, J. A.; Shelby, R.; Taylor, L.; Treacy, D.; Weinstein, A.; and Wintersgill, M. 2005. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education* 15(3):147–204.

Yeh, I.-C. 1998. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research* 28(12):1797–1808.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1):49–67.

Zhu, X., and Goldberg, A. B. 2006. Semi-supervised regression with order preferences. *Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep* 1578.

Zhu, X., and Goldberg, A. B. 2007. Kernel regression with order preferences. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22, 681. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Zhu, X. 2005. Semi-supervised learning literature survey.

Zou, H., and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2):301–320.

Data Set	Description
Synthetic Linear	A synthetic linear regression data set with 10 covariates.
Boston Housing (Harrison and Rubinfeld 1978; Lichman 2013)	Predicting housing values in Boston as a function of various socioeconomic and geographic features. For the transfer experiments we created domains based on the LSTAT (percentage of lower status of the population).
Wine (Lichman 2013)	The UCI wine data set. Predicting the quality of wine given a set of chemical and visual characteristics. For the transfer experiments we used the red wine as the target and the white wine as the source.
Concrete (Yeh 1998; Lichman 2013)	Predicting the compressive strength of concrete as a function of its age and ingredients. For the transfer experiments we split the data based on the age.
King County Housing (kcH)	Predicting housing prices in King County, Washington, as a function of a number of features such as location and number of bedrooms.
ITS (Vanlehn et al. 2005)	The USNA Physics (Fall 2008) data set, which contains the performance of 69 university students using the Andes physics intelligent tutoring system (ITS). Task is to predict student performance on the "Angular Momentum" subset of the system as a function of the students' performance on the other sections of the system.
Heart (Rousseauw et al. 1983)	Predicting heart disease in males from a high-risk heart disease area of Western Cape, South Africa, as a function of various lifestyle and biometric features.

Table 2: Data sets we used in our experiments.

	Nonnegative	Ridge	Lasso	PSRC: p Signs	PRCR: p Pairs	PPSCR: p pairs
Synthetic	$0.183(0.030)$	$0.179(0.029)$	$0.180(0.029)$	**0.141(0.026)**	$0.155(0.031)$	$0.161(0.030)$
BH	$0.212(0.022)$	$0.202(0.036)$	$0.183(0.023)$	**0.149(0.018)**	$0.176(0.029)$	$0.163(0.020)$
Wine	$0.101(0.013)$	$0.100(0.013)$	$0.105(0.013)$	**0.088(0.010)**	$0.093(0.011)$	$0.091(0.010)$
Concrete	$0.255(0.022)$	$0.275(0.020)$	$0.293(0.018)$	**0.220(0.017)**	$0.232(0.019)$	$0.229(0.018)$
Housing	$0.432(0.041)$	$0.478(0.043)$	$0.482(0.049)$	$0.409(0.038)$	$0.451(0.042)$	**0.399(0.037)**
ITS	$0.568(0.092)$	$0.625(0.095)$	$0.700(0.118)$	**0.525(0.089)**	$0.540(0.091)$	$0.570(0.093)$
Heart	$2.129(0.220)$	$2.159(0.220)$	$2.190(0.187)$	**2.007(0.191)**	$2.044(0.209)$	$2.124(0.229)$

Table 3: Error (with 95% confidence intervals in parentheses) of our methods using feature level guidance, and competing methods which cannot take feature level guidance. These results show our method is able to successfully exploit the feature level guidance.

	PSRC	Signed Ridge
Synthetic	**0.141(0.026)**	$0.150(0.026)$
BH	**0.149(0.018)**	$0.163(0.022)$
Wine	**0.088(0.010)**	$0.094(0.012)$
Concrete	**0.220(0.017)**	$0.232(0.018)$
Housing	**0.409(0.038)**	$0.433(0.040)$
ITS	**0.525(0.089)**	$0.586(0.095)$
Heart	**2.007(0.191)**	$2.128(0.220)$

Table 4: Errors (with 95% confidence intervals in parentheses) of our method and the Signed Ridge, which takes the same feature level guidance but lacks the robustness to noisy guidance of our method . These results show our method performs better, indicating our mechanism for handling noisy guidance can work well.

	PSRC	PSRC: Training Guidance
Synthetic	**0.141(0.026)**	$0.175(0.029)$
BH	**0.149(0.018)**	$0.176(0.030)$
Wine	**0.088(0.010)**	$0.098(0.013)$
Concrete	**0.220(0.017)**	$0.266(0.020)$
Housing	**0.409(0.038)**	$0.437(0.042)$
ITS	**0.525(0.089)**	$0.608(0.102)$
Heart	**2.007(0.191)**	$2.156(0.222)$

Table 5: Errors (with 95% confidence intervals in parentheses) of our method when the feature guidance is provided from an outside source versus when the guidance is estimated from the training data. These experiments show generating the guidance from the training data does not work well, indicating the need for the guidance to come from an outside source.

	Ridge	Lasso	Transfer Ridge	PSRC: p Signs	Transfer PSRC: p Signs
Synthetic	$0.597(0.101)$	$0.515(0.126)$	$0.539(0.102)$	**0.430(0.092)**	$0.496(0.109)$
BH	$0.337(0.062)$	$0.390(0.080)$	$0.323(0.062)$	$0.247(0.042)$	**0.242(0.045)**
Wine	$0.217(0.026)$	$0.224(0.023)$	$0.268(0.076)$	**0.179(0.020)**	$0.194(0.022)$

Table 6: Errors of our method (with 95% confidence intervals in parentheses) when the guidance is transferred from a related source data set versus the Transfer Ridge, in which the source data is used by treating the source predictions as an extra feature. These results show using the source data to generate feature level guidance performs better than the standard method of transfer.