Predicting Brand Mentions in Using Multiple Output Regression with Output and Task Structures

Jonathan Friedman JONATHANFRIEDMAN@G.HARVARD.EDU

Abstract

Multiple output regression is a general and important problem in industry. Learning the covariance structures of the regression weights and output noise, and imposing sparsity constraints on these structures, can improve predictive performance. A variety of models that learn these structures have been proposed, with various *a priori* assumptions on the structure of the outputs. We consider the problem of predicting brand mentions in online forums, and apply a multiple output regression model that makes few assumptions about output structure. We note the difficulty of this problem and discuss approaches to a better solution.

1. Introduction

Online forums dedicated to a certain type of product, such as clothing, cars, or coffee, often contain discussion of recent and prospective purchases, as well as general discussion of favorite brands and items. Predicting which brands will be mentioned in a discussion could be of interest to advertisers, since it would allow them to serve readers of that discussion with ads relevant to their interests. It is also useful to predict not only which brands will be mentioned, but with what frequency, since a brand mentioned only a few times in a long discussion is unlikely to be relevant. This difficult problem requires predicting a large number of brand frequencies given only the usually short title of the post. One approach to a solution is to exploit the problem's inherent covariance, where similar brands are likely to be mentioned in the same discussion. In this paper, we explore this problem for the Reddit forum /r/malefashionadvice, which is dedicated to discussion of men's clothing. Outside of this particular problem, multiple output regression has been successfully in areas such as manufacturing, stock prediction, and geostatistics (Breiman & Friedman, 1997).

A standard multiple linear regression problem involves predicting a vector of outputs given an input vector. It is assumed that the outputs are a noisy function of the inputs. In this formulation, it is assumed that each output is independent of all others, and the problem reduces to solving a series of single-output linear regression problems. We refer to this approach single target regression. However, in real data, it is often the case that the outputs are related in some way. This structure can be leveraged to improve predictive power by learning the covariance structure of the outputs, called the task structure, which is learned via a covariance structure on the regression weights. Structure unexplained by the regression weights can be leveraged by learning the covariance structure of the output noise, called the output structure. Rai et. al. propose a multiple output regression model that learns both structures while making minimal assumptions about their structure (2012). Since there is no literature on the structure of brand mentions, this model is preferable to one that assumes a specific structure of the outputs.

Code and data are available at `https://github.com/jonathanhfriedman/cs281-brandmentions`. This includes Python code implementing the model, as well as code for data scraping and a driver program to run the model on the datasets.

2. Background

A multiple linear regression problem involves predicting a vector of responses given an input vector. Given an $N \times D$ feature matrix $\mathbf{X}$ and an $N \times K$ response matrix $\mathbf{Y}$, the goal is to learn the $D \times K$ matrix $\mathbf{W}$ and the K-length bias vector $\mathbf{b}$ such that

$$\mathbf{y}_i = \mathbf{W}^T \mathbf{x}_i + \mathbf{b} + \epsilon_i$$

where ϵ_i is noise associated with the i-th output.

The simplest approach to this problem is to solve a series of K independent linear regression problems, one for each output variable. In this paper, we use a model that learns the covariance structure of the regression weights $\mathbf{W}$, as well as the covariance structure of the random noise. These structures are not used directly in prediction, and the prediction step is the same as in the case of standard linear regression with a bias vector:

$$\mathbf{y}_i = \mathbf{W}^T\mathbf{x}_i + \mathbf{b}$$

The graphical model structure of the problem, shown in figure 1 is similar to linear regression, but with a covariance structure on the noise. The covariance structure of the regression weights is not shown in the model because it is assumed to be an inherent, unknown attribute of $\mathbf{W}$, rather than a random variable that influences it.

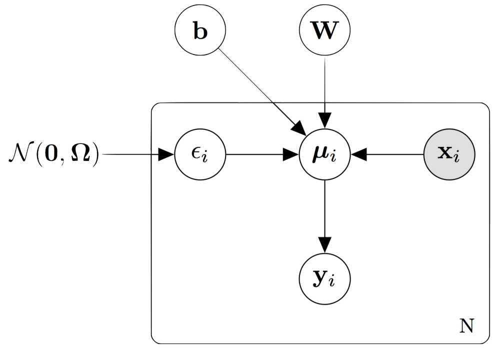


Figure 1. Graphical model structure of multiple output regression with output and task structures.

In the problem of predicting brand mentions, $\mathbf{x}_i$ is a vector of features derived from the post titles and $\mathbf{y}_i$ is a vector of length equal to the number of brands under consideration. Let $\mathbf{y}_{i,j}$ be the entry in $\mathbf{y}_i$ corresponding to brand j. Then

$$y_{i,j} = \frac{\text{\# mentions of brand } j \text{ in post } i}{\text{\# total brand mentions in post } i}$$

This choice of output has the advantage over a categorical $[0, 1]$ encoding of whether or not a brand was mentioned because it captures the brand's relative popularity in a post, and is superior to an unscaled "# mentions of brand j in post i" output because it is more-or-less invariant to thread size. In other words, a brand mentioned two times in a little-read post with only four total mentions is not treated differently than the same brand mentioned 10 times in a popular post with 20 total mentions. Predicting thread popularity is its own interesting problem, and will not be considered here.

2.1. Related Work

In multiple regression problems, ℓ_1-regularization can improve predictive performance by giving robust estimates of the covariance structure (Tibshirani, 1996). A variety of models that leverage regularization to produce sparse estimates of covariance structure have been proposed, with various assumptions as to the structure of the outputs (Kim & Xing, 2010) (Obozinski et al., 2009), and models that do not assume *a priori* knowledge of the sparsity structure (Sohn & Kim, 2012). In fields such as neuroimaging, facial recognition, and biostatistics, domain knowledge can suggest imposing specific sparsity structures (Jenatton et al., 2011); however, given a lack of knowledge about the structure of brand mentions, we instead choose a more general model.

We are not aware of previous attempts to predict brand mentions in online forums.

3. Model

3.1. Objective Function

We use a model proposed by Rai et. al. that learns the task and output covariance structures along with the regression weights. We sketch the basic structure of the model here. See the original paper for derivations and further details (Rai et al., 2012).

Let $\boldsymbol{\Sigma}$ be the covariance matrix of the regression weights. The prior on $\mathbf{W}$ incorporates $\boldsymbol{\Sigma}$:

$$p(\mathbf{W}) \propto \prod_{k=1}^{K} \mathcal{N}(\mathbf{w}_k|0, \mathbf{I}_D)\mathcal{MN}_{D\times K}(\mathbf{W}|\mathbf{0}_{D\times K}, \mathbf{I}_D \otimes \boldsymbol{\Sigma})$$

The likelihood incorporates $\boldsymbol{\Omega}$:

$$\prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}, \mathbf{b}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n|\mathbf{W}^T\mathbf{x}_n + \mathbf{b}, \boldsymbol{\Omega})$$

Thus, the posterior on $\mathbf{W}$ is

$$\begin{aligned} p(\mathbf{W}|\mathbf{X}, \mathbf{Y}, \mathbf{b}, \boldsymbol{\Omega}, \boldsymbol{\Sigma}) &\propto \prod_{k=1}^{K} \mathcal{N}(\mathbf{w}_k|0, \mathbf{I_D})\mathcal{MN} \\ &= \prod_{k=1}^{K} \mathcal{N}(\mathbf{w}_k|0, \mathbf{I}_D)\mathcal{MN}(\mathbf{W}|\mathbf{0}_{D\times K}, \mathbf{I}_D \otimes \boldsymbol{\Sigma}) \\ &\qquad \prod_{n=1}^{N} \mathcal{N}(\mathbf{y}_n|\mathbf{W}^T\mathbf{x}_n + \mathbf{b}, \boldsymbol{\Omega}) \end{aligned}$$

and the negative log posterior is

$$\begin{aligned} &\text{tr}((\mathbf{Y} - \mathbf{XW} - \mathbf{1b}^T)\boldsymbol{\Omega}^{-1}(\mathbf{Y} - \mathbf{XW} - \mathbf{1b}^T)^T)) \\ &\quad - N\log|\boldsymbol{\Omega}^{-1}| + \text{tr}(\mathbf{WW}^T) + \text{tr}(\mathbf{W}\boldsymbol{\Sigma}^{-1}\mathbf{W}^T) \\ &\quad - D\log|\boldsymbol{\Sigma}^{-1}| \end{aligned}$$

We impose ℓ_1 sparsity penalties on $\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\Omega}^{-1}$, which, in addition to improving robustness, reflects the intuition that in general, mentions of arbitrary brands are not correlated. The final optimization objective is thus

$$\begin{aligned}\underset{\mathbf{W},\mathbf{b},\mathbf{\Sigma}^{-1},\mathbf{\Omega}^{-1}}{\operatorname{argmin}} \quad & \operatorname{tr}((\mathbf{Y}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^T)\mathbf{\Omega}^{-1}(\mathbf{Y}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^T)^T)\\ & - N\log|\mathbf{\Omega}^{-1}| + \operatorname{tr}(\mathbf{W}\mathbf{W}^T) + \lambda_1\operatorname{tr}(\mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{W}^T)\\ & - D\log|\mathbf{\Sigma}^{-1}| + \lambda_2||\mathbf{\Omega}^{-1}||_1 + \lambda_3||\mathbf{\Sigma}^{-1}||_1\end{aligned}$$

3.2. Minimizing the Objective Function

The objective function is minimized via four-step alternating optimization. The steps involve solving a system of linear equations, evaluating a simple matrix multiplication, and running two iterations of graphical lasso inverse covariance estimation (Friedman et al., 2008). Again, we simply give the relevant equations and refer the reader to (Rai et al., 2012) for details. Note that a few typographical and notational errors in the original paper have been clarified here.

3.2.1. Optimization with Respect to $\mathbf{W}$

$\hat{\mathbf{W}}$ is given by solving

$$\begin{aligned}[(\mathbf{\Omega}^{-1}\otimes\mathbf{X}^T\mathbf{X}) + ((\lambda_1\mathbf{\Sigma}^{-1}+\lambda\mathbf{I}_K)\otimes\mathbf{I}_D)]\operatorname{vec}(\mathbf{W})\\ = \operatorname{vec}(\mathbf{X}^T(\mathbf{Y}-\mathbf{1}\mathbf{b}^T)\mathbf{\Omega}^{-1})\end{aligned}$$

where $\operatorname{vec}(\mathbf{W})$ is the column-major flattening of $\mathbf{W}$. That is, IF $\mathbf{w}_{ij}$ is the entry in the i-th row and j-th column of $\mathbf{W}$, then $\operatorname{vec}(\mathbf{W}) = [\mathbf{w}_{11}, \mathbf{w}_{21}, ..., \mathbf{w}_{D1}, \mathbf{w}_{12}, \mathbf{w}_{22}, ...]$.

3.2.2. Optimization with Respect to $\mathbf{b}$

$\hat{\mathbf{b}}$ is given by

$$\frac{1}{N}(\mathbf{Y}-\mathbf{X}\mathbf{W})^T\mathbf{1}$$

or equivalently

$$\sum_{n=1}^{N}\frac{1}{N}(\mathbf{Y}-\mathbf{X}\mathbf{W})^T$$

where the sum is taken over the rows of $(\mathbf{Y}-\mathbf{X}\mathbf{W})^T$.

3.2.3. Optimization with Respect to $\mathbf{\Sigma}^{-1}$

$\hat{\mathbf{\Sigma}}^{-1}$ is given by

$$\text{gLasso}(\frac{\lambda_1}{D}\mathbf{W}^T\mathbf{W}, \lambda_3)$$

that is, by solving the graphical lasso inverse covariance estimation problem with empirical covariance matrix $\frac{\lambda_1}{D}\mathbf{W}^T\mathbf{W}$ and sparsity penalty λ_3.

3.2.4. Optimization with Respect to $\mathbf{\Omega}^{-1}$

$\hat{\mathbf{\Omega}}^{-1}$ is given by

$$\text{gLasso}(\tfrac{1}{N}(\mathbf{Y}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^T)^T(\mathbf{Y}-\mathbf{X}\mathbf{W}-\mathbf{1}\mathbf{b}^T), \lambda_2)$$

These steps are repeated in sequence until convergence. We defined convergence as $||W_k - W_{k-1}||_2$, that is, the norm of the difference between $\mathbf{W}$ at step k and at step $k-1$, falling below some predefined tolerance. Empirical convergence estimates are given in section 5.3.

4. Data Collection

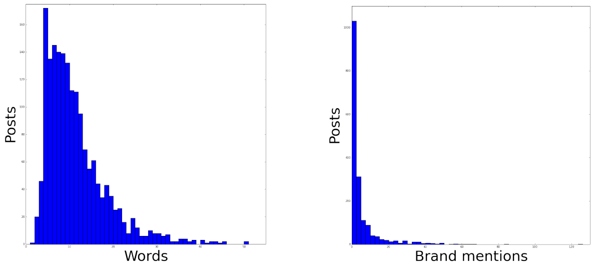
Fashion forum post and comment data were scraped from the popular Reddit forum /r/malefashionadvice on Oct. 28, 2015 using the Python Reddit API Wrapper (PRAW). It is difficult to scrape more than the first 1,000 Reddit posts given a sorting criterion, so we scraped the top 1,000 posts (sorted by total upvotes, Reddit's version of "likes") of the past year and the top 1,000 comments all time and deleted redundant posts. This resulted in a dataset of 1,776 posts and 113,906 comments.

We used our experience with fashion forums to construct a list of 74 popular brands by hand, along with common misspellings, abbreviations, and representative items (for example, "APC", "A.P.C.", "Petit New Standard" (A.P.C.'s signature jeans), and "PNS" (an abbreviation of Petit New Standard) all contribute toward mentions of the brand A.P.C.). The choice of brands, as well as the list of misspellings, abbreviations, and representative items is by nature arbitrary, but we believe that it represents a fair approximation of the most popular brands on /r/malefashionadvice. We also consider an alternate list of the 25 most popular brands in the dataset. Assuming that more popular brands see more to gain from targeted advertising on a forum, the smaller sample of brands focuses on the brands most relevant to our original targeted advertising application. Furthermore, it reduces the training time of the model, which proved useful, given the time constraints of the project.

The data is highly unbalanced, and there is little information on which to base predictions. As seen in figure 2, post titles generally contain few words, and the comments of the average post contain few brand mentions. This proves to be a serious impediment, as described in section 5.

As a sanity check, we also apply our model to real-world multi-output datasets that have been previously benchmarked (Spyromitros-Xioufis et al., 2014). We test it on the Airline Ticket Price datasets ATP1d and ATP7d[1]. These small datasets are ideal candidates for testing our model

[1] These datasets are available at `http://mulan.sourceforge.net/datasets-mtr.html`



(a) Words per post (b) Brand mentions per post

Figure 2. Words and brand mentions per post.

and ensure that it is performing properly. This is especially important given the sparse and unbalanced brand data, since it is difficult to tell whether a model performs poorly because of the challenges of the data or because the model is inherently bad or poorly implemented.

5. Experiments & Results

As suggested in the original paper (Rai et al., 2012), we set $\lambda = 0.01$. The other hyperparameters $\lambda_1, \lambda_2, \lambda_3$ are chosen via four-fold grid search cross-validation. Hyperparameter selection was complicated by the long training time of the model combined with the time constraints of the project and the necessity of performing a grid search over three interacting hyperparameters. Surprisingly, solving the system of DK linear equations was by far the most costly step in the optimization procedure, taking longer than even the graphical lasso steps.

In order to compare with the benchmarks given in (Spyromitros-Xioufis et al., 2014), model performance on the OES data is evaluated using the average relative root mean squared error (aRRMSE). RRMSE is given by

$$RRMSE(D_{test}) = \sqrt{\frac{\sum_{y_j \in D_{test}} (\hat{y}_j - y_j)^2}{\sum_{y_j \in D_{test}} (\bar{Y}_j - y_j)^2}}$$

where D_{test} is the test set, $\hat{y}_j$ is the estimate of target variable y_j, and $\bar{Y}_j$ is the mean of y_j over the training set. aRRMSE is given by the average RRMSE over each output variable. Model performance on the brand data is evaluated using standard RMSE.

Due to the small size of the airline data, aRRMSE is estimated using 10-fold cross validation. The brand data is evaluated using an 80/20 train-test split.

5.1. Real-World Data

We test the model on real-world datasets from the airline industry. The goal is to predict next day (ATP1d) or seven day (ATP7d) ticket prices based on a variety of boolean, categorical, and real valued features over non-stop, one-stop, and two-stop flights, over a range of times (Spyromitros-Xioufis et al., 2014). It is reasonable to expect that a feature like minimum ticket price on a given day is related to the mean ticket price on the day before, so this dataset is an excellent candidate on which to test a model like MROTS that learns the structure of the regression weights. We also display results from (Spyromitros-Xioufis et al., 2014) for single target regression and for their high-performance ensemble regressor chain corrected (ERCC) multi-output model.

ATP1d	
Method	**RMSE**
MROTS	35.33
Single Target	37.35
ERCC	37.24

ATP7d	
Method	**RMSE**
MROTS	53.83
Single Target	52.48
Zero	51.24

Table 1. aRRMSE on real-world datasets for various regression models.

Results are shown in table 1. We stress that we make no claims as to the statistical significance of these results, and that the goal of this paper is not to compare multiple output regression models. However, it appears that MROTS performs competitively with other regression models.

5.2. Brand Data

5.2.1. Feature Selection

We extracted a mix of features from the post titles:

- Total number of words in title
- Average word length in title
- Average tf-idf score over all words in title
- Indicator variables encoding which of the 150 most popular words appear in a post title (bag of words)

All but the bag of words are self-explanatory. The 150 word cutoff was chosen as a tradeoff between including too many features and not enough. As seen in figure 3, the marginal utility of adding an additional word to the bag begins to tail off at around 150 words, so a bag size of 150 is a reasonable choice.

For the brand data, hyperparameter selection was complicated by the fact that the MROTS model did not perform well on the data (as the good results section 5.1 show, this is a problem with the data, rather than the model). During cross-validation for hyperparameter selection, MROTS produced similar results for each choice of hyperparameters, so it is not clear if the optimal choice of $\lambda_1 = 10^{-3}$, $\lambda_2 = 10^{-5}$, $\lambda_3 = 10^{-3}$ was truly optimal. Fur-

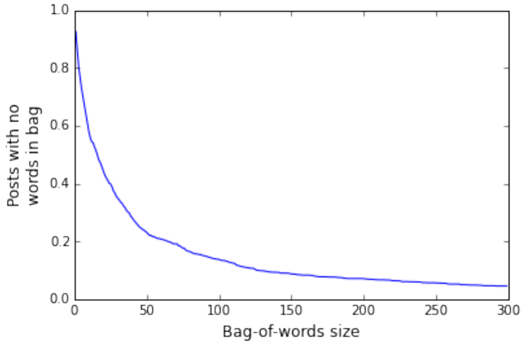


Figure 3. The change in the proportion of posts with no words in the bag of words as bag size increases.

25 Brands	
Method	**RMSE**
MROTS	0.12267
Linear Regression	0.12281
Training Mean	0.12348
Zero	0.12642

74 Brands	
Method	**RMSE**
MROTS	0.07100
Linear Regression	0.07116
Training Mean	0.07173
Zero	0.07233

Table 2. RMSE on the brand dataset.

thermore, the long training time of the model prevented a search over a large grid of hyperparameters.

MROTS is tested against three baselines:

- **Linear regression:** This is the single target regression described earlier, which solves K independent regression problems.
- **Predict mean:** This predicts that every entry in the test matrix of output vectors $\mathbf{Y}_{test}$ is equal to

  $$\frac{\sum \mathbf{Y}_{train}}{NK}$$

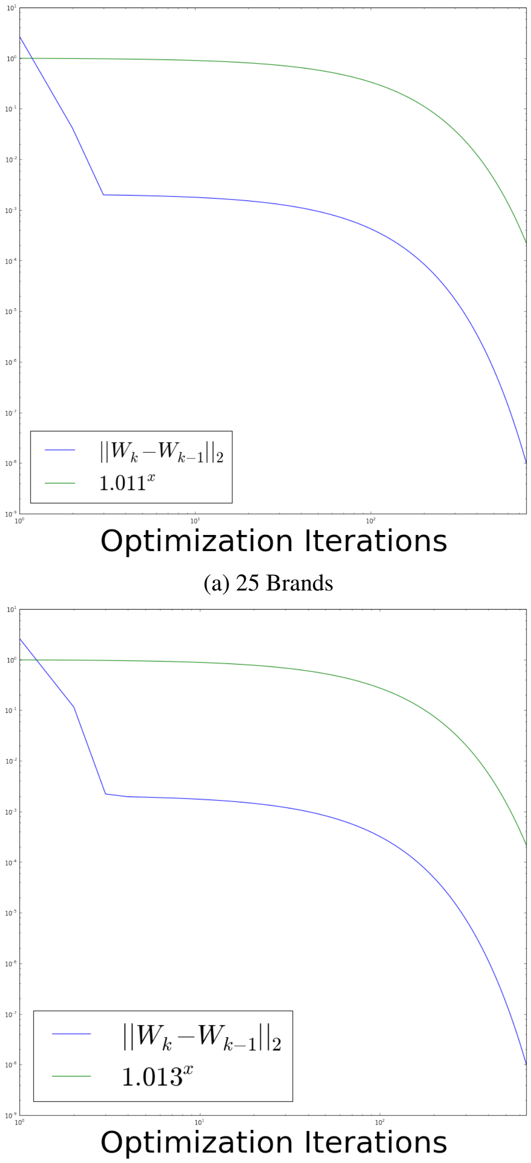
  the average value of the training output matrix.
- **Predict zero:** This predicts that every entry in the test matrix will be zero.

Results for both 25 and 74 brands are shown in table 2

MROTS performs marginally better than other methods, and linear regression performs marginally better than either predicting the mean or predicting zero. However, the difference is slight compared to the magnitude of the RMSE. Because of the model's performance on real-world data, we are confident that its poor performance here is a result of the data, rather than the model. Still, the fact remains that this fairly sophisticated model does little better than the most naïve baselines.

5.3. Convergence

Rai et. al. do not prove convergence of the optimization procedure described in section 3.2.1, nor do they give estimates of convergence speed. In the tests on brand data, we observed exponential convergence in $||W_k - W_{k-1}||_2$ as seen in figure 4, though with a small base around 1.01. This suggests that, assuming solving DK linear equations does not pose too much of a bottleneck, training the model would be feasible even for large input and output spaces.



(a) 25 Brands

(b) 74 brands

Figure 4. Convergence of the optimization procedure.

6. Conclusions & Future Work

In this paper, we have studied the problem of predicting brand mentions in online fashion forums using multiple output regression with output and task structures. We have observed that the four step alternating optimization procedure of section appears to give exponential convergence, and that the most expensive step in the procedure is solving the system of linear equations for $\mathbf{W}$.

It seems that the problem is intractable without more data or more sophisticated methods. One could, with some ingenuity, scrape more than 1,000 posts based on a given search criterion, but just having more comments would probably not result in any sort of impressive performance increase. Instead, new types of data would need to be introduced to overcome the difficulties inherent in this problem A few posts are representative of these difficulties:

- A post title describes building a wooden box to store his ties. The comments largely discuss woodworking and do-it-yourself fashion projects rather than tie brands, as one might expect.
- A post title is “Backpacks?” and nothing else. This is very little information on which to make a prediction.

If only post titles are considered, it is difficult to imagine a model that could learn the subtle differences between a post about tie brands and a post about tie boxes, or one that could do well on a post titled “Backpacks?” while also performing well on a post titled “Everlane info beyond the backpacks?” that asks for information on the brand Everlane while explicitly requesting *no* information about backpacks. Successfully overcoming these obstacles would probably require either some combination of more, and different, data; a model that could learn complex nonlinear relationships (perhaps a neural network); or more advanced natural language processing to perform better feature selection.

We could have trained the model on the text of posts, rather than just the titles. However, since post text can be edited and titles cannot, this would require dealing with posts whose edits were influenced by the posts’ comments. In other words, we would risk including information that was not available on a new post, which could result in poor predictive performance on new, unedited posts.

We could also incorporate a user’s post history into the model. One might expect that a user who has, in the past, posted frequently about jeans is likely to post about jeans in the future, and that their posts will lead to more discussions about jeans. One could also model the historical distribution of brand posts on the forum as a whole using a hidden Markov model, with the distribution of brand mentions in a given time interval is a function of some hidden “true” inclination of commenters, as a whole, to mention a given brand. Again, this raises issues of including information that was not available at the time of the post.

Incorporating post text and user history would probably improve predictive power. However, if such a problem were actually considered in targeted advertising, an even more powerful tool would be available: user browsing data. Advertisers often track websites visited using cookies, thus giving them information on a poster’s preferences. This direct, specific brand preference information, combined with post text and user history, would undoubtedly lead to a much more powerful model. This is purely hypothetical, since user browsing data is, of course, proprietary.

Finally, given the numerous difficulties of predicting the specific proportion of brand mentions, one could consider the easier problem of predicting whether or not a brand will be mentioned in a post’s comments. Such information would be less useful, since one would not know how significant those mentions are (in a post with 100 total mentions, a brand mentioned once would be treated the same as a brand mentioned 99 times). If the original problem were to prove intractable even with additional data, a good solution to this categorical problem would be better than nothing.

References

Breiman, L. and Friedman, J. H. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 59 (1):3–54, 1997.

Friedman, J. H., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Jenatton, R., Audibert, J., and Bach, F. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

Kim, S. and Xing, E. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27 th International Conference on Machine Learning*, 2010.

Obozinski, G., Wainwright, M., and Jordan, M. Union support recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1):1–47, 2009.

Rai, P., Kumar, A., and Daumè III, H. Simultaneously leveraging output and task structures for multiple-output regression. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

Sohn, K. and Kim, S. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 2012.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., and Vlahavas, I. Multi-label classification methods for multi-target regression. e-Print (arXiv), 2014. `1211.6581v4[cs.LG]`.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.