Small models for text regression via Lasso, LDA, and NMF

10-701 Final Project Report – Dec 9, 2009

Brendan O'Connor
brenocon@cmu.edu

Abstract

We tested dimension reduction (LDA, NMF, SVD) and regularization (Lasso) for a text regression problem, predicting a company's future stock volatility from the text of its annual financial outlook statement. Lasso regularization is very effective at creating sparse, reasonably interpretable, and highly performant models. Unsupervised dimension reduction only sometimes creates comprehensible models, and they are of no use for the task.

1 Introduction

We approach the problem of text regression on financial legal documents. Every year, publicly held companies in the U.S. are required to file reports to the Security Exchange Commission. One of these filings, the 10-K, must include a section usually titled "managements discussion and analysis of nancial conditions and results of operations," which often contains discussions about potential risks with regards to future risk of the company. [1] was able to predict the future volatility of the company' stock (i.e. its riskiness; specifically, log of the variance of day-to-day returns) from (linear kernel) SVM regression [2] on the unigrams and bigrams from the MD&A text.

We use their released data set,[1] which is fairly large, consisting of 26,806 documents over 10 years, averaging 9,240 words per document. The feature space is also large: there are 220,000 unique unigrams, and 5.6 million bigrams.

It is desirable to derive small and human-interpretable models. First, we want to understand what a predictive model is doing, which may help gain insight into how to pursue future improvements. Second, it may assist financial analysts or researchers understand how a company's communications relates to its financial risk. Finally, aside from interpretability, smaller models may perform better if they are selected in a way such that less effort is spent modeling unimportant phenomena in the data.

Unfortunately, the SVM regression approach does not do this very well. SVMs are supposed to choose a sparse set of support vectors, but at least on this and other text problems, anywhere from 50-80% of the training set's documents end up in the support vector. If the model is viewed in the primal, which is just the weighted sum of features across the support vector, then the final model size is the number of features in the union of the features present in support documents; i.e., hundreds of thousands of features. It is impossible for a human to get a complete view of such a model.

We pursue two approaches to derive smaller models. The first approach is Lasso (L1) regularization [3] on the original input space. It consistently selects only a few hundred features while surpassing performance of the SVM regression. The second approach is to derive a lower dimensional representation of the documents. We compare two approaches, Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF).

[1] http://www.ark.cs.cmu.edu/10K/

2 Lasso regression

We interpret each document as a feature vector $\mathbf{x}_i$ of $\log(1 + count)$ values for each unigram and bigram. The target variable, y_i, is the log-volatility of the company's stock over the 12-month period after the report was filed.[2] Lasso regression optimizes a bias term and feature coefficients for the L1-penalized objective function,

$$\min_{\mathbf{w}} \left\{ \lambda ||\mathbf{w}||_1 + \sum_i \left(y_i - bias - \mathbf{w}^T \mathbf{x}_i \right)^2 \right\} \tag{1}$$

given a predefined regularization constant λ. The L1 penalty has the interesting property that regularization actually shrinks coefficients all the way to 0, so strong regularization does feature selection. In fact, the relationship between λ and the final model size is monotonic.

Furthermore, we use "glmnet" [4], a coordinate descent Lasso implementation that grid searches a path of solutions at a series of λ regularization constants. It starts with the highest value of λ, where all feature coefficients $\mathbf{w}$ are zero. Then for the next-highest value of λ, it performs coordinate descent with a warm start; that is, using the last solution's coefficient vector as the starting point for the coordinate descent towards the next solution. Since as λ decreases, the number of active non-zero features must monotonically increase, this procedure is actually about as fast as performing an optimization for a single low value of λ. Related work includes LARS and forward stagewise regression, both reviewed in [5].

This implementation has an additional feature, to use a weighted combination of L1 and L2 penalties. This is known as elastic net regularization [6]. This is useful because some previous literature has found L2 can sometimes outperform L1 regularization, and also L1 penalization can be less stable in some situations. We experimented minimally with this tradeoff, but found that pure L1 penalty performed best.

We evaluate Lasso in a simulated forecasting setting. For a year t, train on (document, future volatility) pairs from the five-year window $\{t - 4 \ ... \ t\}$, then evaluate on data from the next year $t + 1$. Furthermore, to properly test whether the textual information contributes new information, we combine all text features with a strong non-textual baseline predictor, the historical volatility of the company's stock over the 12 month period before filing the report. We jointly optimize its coefficient along with all others, but do not include it in the regularization penalty. (We also exclude the bias term from the penalty, as is standard practice.)

Our regularizer tuning procedure was as follows. First, we hold out the last of the training years as a development set, and train a full regularization path on the first four years of the training window. We then evaluate each model's predictive error on the development set to pick the optimal value of λ. (The graph of held-out accuracy against λ is U-shaped.) That value is then used for a final trained model on the entire five-year window, and the MSE on the test year is reported.

Regularization tuning is an important part of creating a performant predictive system, and it's worth noting that the pathwise coordinate descent algorithms makes tuning substantially easier compared to other approaches. This is may be important for problems like this one, where properties of the data change over time; for example, average document size increases four-fold from 1996 through 2006, and the devset-optimal λ values change (slightly) as well.

We also compare to the simple baseline of directly predicting future volatility as historical volatility. In the financial literature, historical trends are known to be strong baseline predictors. Finally, we have a one-variable OLS model from the historical volatility predictor; this approach does better in some years but worse in others. However, this is a better baseline to compare to in order to understand the effect of adding new features, since it's simply the model in 1 with all text coefficients $\mathbf{w}$ set to 0. Indeed, in all years, adding the text yields an improvement over the baseline-only model.

[2] Using the logarithm of the volatility, instead of the volatility, is a standard transformation in finance; it yields a fairly bell curve distribution. Tails are known to be slightly heavier than a Gaussian, but it is much more normal than the raw distribution. Since in the probabilistic interpretation of least-squares regression, the error distribution is Gaussian, it is justified to pursue transformations that make the variables look as normal as possible.

Table 1: Comparison of Lasso, SVR, and baseline, on Mean Sq Error for held-out test years.

Model	2001	2002	2003	2004	2005	2006	wtd. avg.
Lasso Regression	0.1801	0.1512	0.1569	0.1171	0.1209	0.1283	0.1406
SVM Regression [1]	0.1852	0.1792	0.1599	0.1352	0.1307	0.1448	0.1538
Historical volatility (OLS)	0.1922	0.1631	0.2040	0.1272	0.1275	0.1403	0.1578
Historical volatility	0.1747	0.1600	0.1873	0.1442	0.1365	0.1463	0.1576
Total no. training features	2.4mil	2.6mil	3.0mil	3.5mil	3.9mil	4.3mil	
No. Lasso-selected features	282	232	112	310	116	111	

Results are shown in Table 1. The Lasso regression outperforms both the baseline as well as previously reported SVM regression results.

Furthermore, it yields very sparse models. The models for each year varied from 100 to 300 features total, and as can be seen in the table, are impressively smaller than the original number of input features. Unfortunately, the Lasso is not completely responsible for the sparseness of the final models. Limitations in the glmnet implementation forced us to impose very strong feature count thresholds as a preprocessing step, such that each training set had only 30,000 or so input features. This is still a sizable reduction, however; and in any case, it still outperforms the SVR. It is possible that a better implementation, run on all several million features, may yield a larger but more performant model.

One final Lasso model is shown in its entirety in Table 2. Among other things, it seems that the model is picking up on information about mergers as indicators of low risk, and information about costs and lost profits as indicators of high risk. The baseline historical volatility is also shown, and is far stronger than any of the textual features; the only feature within an order of magnitude as powerful is the bigram "merger agreement."

2.1 Dimension reduction as preprocessing

The Lasso creates small models by selecting a small subset of the input feature space. An alternative approach is to use an unsupervised algorithm to reduce the dimensionality of the input features, then use the lower-dimensional representation in a final regression. (Yet another alternative is a joint model that does both simultaneously).

There is a family of models and algorithms that can be viewed as *topic models*, in which a fixed number of K hidden topics are fit to the data. Every topic has weights for words in the vocabulary, and every document has weights for each topic. Thus, instead of representing documents as collections of words, they can be viewed as collections of topics. Latent Semantic Analysis was the first algorithm to take this approach, via the SVD. We test this and two other models, LDA and NMF.

2.1.1 Latent Dirichlet Allocation (LDA)

LDA is a Bayesian approach that models documents as multinomial distributions over topics, and each word as a draw from the document's topic multinomial, then a draw from that topic's distribution over words.

We omit details for lack of space, but training and inference for LDA is intractable, and approximate methods must be used. We use the variational inference method of [7], with the LDA-C implementation.

Under the weights view of topic models, a topic's word weights is the topic's conditional probability distribution across words; and a document's topic weights is the document's conditional distribution across topics.

2.1.2 Singular Value Decomosition (SVD)

SVD is the classic matrix factorization approach for dimension reduction. For the the document-word matrix $\mathbf{X}$ (dimensions (n, m), where values are log-counts), SVD finds $\mathbf{W}$ and $\mathbf{H}$ (where rows of $\mathbf{W}$ are projections of documents in the topic space, and columns of H are topic-word weights)

Table 2: The entire 2001-2005 Lasso model (111 selected features). Negative = low financial risk, Positive = high financial risk.

Coef	Feature	Coef	Feature	Coef	Features
-1.22e-01	merger_agreement	-2.86e-03	%	7.97e-04	lender
-2.33e-02	the_proposed	-2.83e-03	subsidiary_of	8.40e-04	financial_covenants
-1.96e-02	exit_or	-2.74e-03	disclosure_provisions	9.72e-04	debenture
-1.92e-02	for_guarantees	-2.69e-03	practices	1.02e-03	$#_for
-1.53e-02	guarantees_issued	-2.63e-03	announcement_of	1.54e-03	fiber
-1.41e-02	poor_s	-2.47e-03	with_gaap	1.78e-03	goodwill_will
-1.39e-02	is_terminated	-2.24e-03	across	2.10e-03	#_page
-1.37e-02	consummation	-2.21e-03	maintenance_revenue	2.31e-03	a_waiver
-1.36e-02	of_merger	-1.86e-03	had_entered	2.57e-03	inventory_costs
-1.28e-02	distribution_system	-1.78e-03	by_operating	2.77e-03	covenant
-1.20e-02	net_income	-1.78e-03	improved	3.25e-03	supersedes_sfas
-1.18e-02	multiple_deliverables	-1.65e-03	as_lower	3.51e-03	business_plan
-1.07e-02	#_amendment	-1.55e-03	insurance	3.53e-03	operating_loss
-9.98e-03	goodwill_amortization	-1.54e-03	changes	4.29e-03	management_will
-9.25e-03	upon_termination	-1.54e-03	such_that	4.57e-03	the_senior
-8.80e-03	rates	-1.51e-03	more_than	4.60e-03	administrative
-8.27e-03	for_stockbased	-1.49e-03	annual	5.21e-03	for_longlived
-8.27e-03	not_completed	-1.49e-03	repurchases	5.30e-03	accounting_be
-8.07e-03	no_impairment	-1.34e-03	on_plan	5.43e-03	assembled
-7.49e-03	gains	-1.18e-03	rate_increases	6.44e-03	its_future
-7.09e-03	in_different	-1.15e-03	lower_interest	7.02e-03	no_assurance
-6.82e-03	billion	-1.13e-03	million_shares	7.09e-03	trade_show
-6.45e-03	merger	-1.01e-03	plan_assets	9.78e-03	warrants_to
-6.38e-03	ratings	-1.00e-03	s_pension	1.04e-02	working
-5.64e-03	termination_fee	-7.52e-04	#_$	1.21e-02	additional_financing
-5.30e-03	final_settlement	-7.46e-04	with_exit	1.23e-02	raise
-4.91e-03	customary	-1.94e-04	consummation_of	1.34e-02	net_loss
-4.66e-03	share_repurchase	-1.51e-05	policies	1.48e-02	be_disposed
-4.32e-03	by_an	2.92e-05	delisted	1.49e-02	waived
-4.24e-03	rate	1.74e-04	#%_per	1.57e-02	business_combinations
-4.00e-03	war	1.95e-04	years_beginning	2.15e-02	profit_decreased
-3.66e-03	tender	3.17e-04	$#_from	2.32e-02	negative_cash
-3.64e-03	changes_in	3.42e-04	is_due	3.68e-02	bid_price
-3.44e-03	earnings	3.71e-04	ending_december	4.06e-02	combinations_initiated
-3.29e-03	tax_rate	6.04e-04	in_default	4.90e-02	a_going
-3.17e-03	the_fair	6.30e-04	financing	6.41e-02	going_concern
-3.13e-03	amortization_as	6.70e-04	establishes_accounting	7.81e-01	NT_before

such that

$$\mathbf{X} \approx \mathbf{WH}$$

under a squared error reconstruction criterion. (The usual definition of SVD has $\mathbf{W} = \mathbf{UD}$ for diagonal $\mathbf{D}$ and $\mathbf{V^T} = \mathbf{H}$ such that the left and right singular matrices $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, but we use this form to facilitate comparisons to other approaches.)

The use of SVD for document-word matrices in this manner is known as Latent Semantic Analysis.

2.1.3 Non-negative Matrix Factorization (NMF)

NMF is a matrix factorization approach, that finds a representation of documents in a K-dimensional space of topics. For the the document-word matrix $\mathbf{X}$ (dimensions (n, m), where values are log-counts), NMF finds $\mathbf{W}$ and $\mathbf{H}$ such that

$$\mathbf{X} \approx \mathbf{WH}$$

under a reconstruction error criterion, with the non-negativity constraints $|w_{n,k}| \geq 0$ and $|h_{k,m}| \geq 0$ for all $w \in \mathbf{W}$ and all $h \in \mathbf{H}$. [8] illustrate two different cost functions: squared error, and a KL divergence-like error. The optimization problem is non-convex, but a local minimum can be found through simple multiplicative updates. Solutions found through this approach have been observed to work well in practice.

There should be a theoretical relationship between divergence-reducing NMF and the variational approximation for LDA, which minimize the KL divergence between an approximation and the true probability distribution. We do not further pursue this connection in this work.

2.2 Experiment: discovering topics

We ran LDA and NMF on the set of documents from year 1999; this consists of 2524 documents, and we used the most frequent 30,000 features (unigrams and bigrams). We tried the settings of 20 and 100 topics.

Table 3: Top 5 words for each topic

LDA	NMF	SVD
exchange_gains sfas_no of_declining agreements_with opposed_to	shares stock common_stock shares_of common	consolidation objective come quarter mitigate
company_and available_for increased_personnel from_paying respond_to	currency foreign foreign_currency exchange us	signing privileges hire obligated main
these_costs company_expects growth_for is_expected expanding_its	sales fiscal # fiscal_# the	restructuring yearoveryear telephone taxed agents
liabilities_for respond_to increased_personnel company_while completed_and	# the and of #_#	ultimately weather reconciliation must consists
capacity obsolete differ charge liabilities	# loans #_# the loan	commercially assurance none payment came
available_for respond_to million_$# operating_income the_commercial	the_company company the company_s of	self what implementing evaluation positioning
capacity investments industry liabilities disclosures	services revenues # the revenue	organized intrinsic communicate expand providers
liabilities industry investments charge agencies	restructuring charges charge recorded loss	insured options fluctuate ultimately basin
capacity investments industry liabilities charge	# the properties of and	imports seen earned trucking consists
increased_personnel respond_to of_capital from_$# notes_which	# million $#_million in_# share	remainder preserve acquired receive things
available_for million_$# the_commercial #%_due increased_personnel	the sales # of and	deposits ultimately inception comparable identifying
investments capacity industry liabilities charge	the # portfolio investment of	regulation june stockholder auditors accelerated
foreign_tax services_during see_note repaid_in y#k_issue	gas oil prices production oil_and	as believe meeting returned banks
increased_personnel the_# company_believed #_see program_in	matters environmental regulations state court	than numerous revenue as electronics
operating_income #%_above million_through of_one have_historically	competitors operating_results adversely adverse_effect could_have	factory specified carryforwards options revenue
available_for increased_personnel the_commercial million_$# a_preferred	sfas no_# accounting standards accounting_standards	bear currently insured industry military
respond_to operating_income increased_personnel the_commercial #%_due	research the development and research_and	regulation segment signing previously carryforward
investments industry capacity liabilities disclosures	the # of in to	ultimately cannot bear actuarial at
loss_and credit_which issues_associated these_investments #_net	our we of_our we_have #_we	bear eliminating comparable substantially portfolios
capacity had charge transaction working	facility $#_million million credit_facility credit	limits less gain delayed remainder

We show the most likely, or highest scoring, words for each topic in the $K = 20$ models in Table 3. We feel LDA and NMF look reasonable, but SVD less so. We presented the above word clusters for LDA and NMF to several people during our poster session, and asked them to evaluate which was more seemed a more coherent clustering. There was no consensus either way.

An interesting difference between the two is that LDA selects more bigrams. It should be noted that using both unigrams and bigrams is an odd practice for LDA, which makes conditional independence assumptions for the words in a document. It is well-known that word counts in a document do not obey conditional independence; but including the bigrams, each of which deterministically implies the presence of its two component unigrams, seems like a more egregious violation than usual.

Also interesting is that NMF picks up on many function words (we performed no stoplisting), but LDA and SVD mostly avoid them.

2.3 Experiment: predicting volatility

Table 4: Prediction results for held-out year 2001, trained on 1999-2000

Model	Test MSE	Model size	Notes
Hist. vol. (OLS)	0.1929	1	
NMF	0.1950	21	
SVD	0.1924	21	
LDA	0.1929	21	
Lasso at $\lambda = 0.36$	0.1877	21	see Table 5
Lasso at $\lambda = 0.18$	0.1874	123	optimal for dev set

We tested whether these topic representations could help the volatility prediction problem. For convenience, we used a single test year (2001) with a two-year training window (1999-2000). Experiments which used regularization used 1999 vs. 2000 as the dev split as outlined in the first section 2. We continue the practice of always using the baseline as an unregularized parameter in the regression.

Interestingly, this experiment highlighted the very large computational differences between LDA and matrix factorization. Training LDA took longer, but furthermore, to infer topics for a new document, LDA requires an optimization step, whereas NMF and SVD do a simple dot product against the topic-word loadings. To execute inference on 7544 documents, LDA-C took 7 hours, but our trivial NMF and SVD dot product script spent less than 10 minutes.

Can topics stand-in for words? In a first experiment, we took the topic representations of documents as the sole textual features, discarding all original unigrams and bigrams, and tried this for all six unsupervised models. As expected given the small dimensionality of the problem, no regularization was found to be helpful, as was expected given the small dimensionality of the problem. All models failed to add information past the baseline, though NMF was slightly worse. Results for $K = 20$ are reported in Table 4 for the 20 topic models.

Can topics compete against words? In a second experiment, we supplemented the original unigram and bigram text features with additional features for the documents' topic representations. No improvements were made; results are not reported. In most cases, the Lasso threw out almost all of the topic variables, finding the original data was more useful. The best case for topics was LDA at $K = 100$, where four LDA topics made it in to the final model, displacing a few of the original words.

Both questions had an answer of "no," indicating that these topics are not useful at all for the prediction task.

Table 5: The unigram/bigram Lasso model at $df = 21$. All linear combinations of the topics in Table 3 perform worse than this.

Coef	Feature	Coef	Feature
		0.0002	development
-0.06	merger_agreement	0.0006	general_and
-0.02	estate	0.002	and_marketing
-0.02	and_plan	0.002	product
-0.01	properties	0.004	financial_covenants
-0.01	net_income	0.005	personnel
-0.005	mortgage_notes	0.007	net_loss
-0.004	mortgage	0.01	be_successful
-0.001	operating_partnership	0.01	additional_financing
-3e-05	average	0.02	administrative
		0.05	a_going
		0.8	NT_before

There is one last possible justification for topics: if they are easy for humans to understand, perhaps we can regress on a very small number of topics to create a model that's even smaller and easier to read than the unigram/bigram Lasso model, which uses 122 text features.

This can be tested. The Lasso regularizer path can also be viewed as a path across final model sizes, since λ is monotonic with the final model size (number of selected features) df. We refined the path and found the range of regularizers where the final model used 20 features, the same size as the ineffective topic regressions. Here, there was barely any decrease in performance compared to the larger λ that was optimal for the development set.

3 Conclusion

We tested two approaches towards making smaller models for text regression. The first approach was to use L1 regularization, which turned out to select very small subsets of features while achieving good performance. The second approach was to use a two-step process of first using unsupervised learning to derive a low-dimensional topic representation, then using those representations for the regression. This was not successful.

It is an open question whether joint semi-supervised topic modeling approaches, that learn topics while optimizing the volatility target, could help for this task. Given that we found that simple topic information was not helpful at all, it may be difficult to see topic modeling approaches yielding any gains.

In a flat linear model, individual words are individual indicators of a target outcome. Perhaps this is the best approach because words are reasonable units of meaning. Words do, of course, combine to form more complex meanings, but they do this in complicated ways, often governed by local contexts and structure, and whatever these methods of interaction are, they may not correspond to observable behavior at the level of document co-occurrence statistics, which are the only information that topic models can capture.

This also could be an explanation for the widespread failure of generic non-linear modeling in NLP — after a period of experimentation with kernelized SVMs, neural networks, boosted decision trees, etc., most NLP research today is performed with (specialized) linear models. If words are the only easy-to-extract unit of meaning, then combining non-linear models with simple word-level information will show no improvement over linear models.

References

[1] Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[2] Vladimir N. Vapnik and Sayan Mukherjee. Support vector method for multivariate density estimation. In *Advances in Neural Information Processing Systems*, pages 659–665. MIT Press, 2000.

[3] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[4] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularized paths for generalized linear models via coordinate descent. *Technical Report, Stanford University*, 2009.

[5] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, second edition, July 2009.

[6] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[7] D. M Blei, A. Y Ng, and M. I Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[8] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.