L_1 Regularization for Learning Word Alignments in Sparse Feature Matrices

Ergun Biçici EBICICI@KU.EDU.TR
Deniz Yuret DYURET@KU.EDU.TR
Department of Electrical and Computer Engineering
Koç University, Istanbul, Turkey

Abstract

Sparse feature representations can be used in various domains. We compare the effectiveness of L_1 regularization techniques for regression to learn mappings between features given in a sparse feature matrix. We apply these techniques for learning word alignments commonly used for machine translation. The performance of the learned mappings are measured using the phrase table generated on a larger corpus by a state of the art word aligner. The results show the effectiveness of using L_1 regularization versus L_2 used in ridge regression.

1. Introduction

In statistical machine translation, parallel corpora, which contain translations of the same documents in source and target languages, are used to estimate a likely target translation for a given source sentence based on the observed translations. Sparse feature representations can be used in various domains. When the number of instances, m is significantly smaller than the number of features, n, $m \ll n$, then we have an under determined system of equations.

We examine the effectiveness of regression to find the mappings between sparsely observed feature sets. Regularization of the cost function plays an important role to increase the performance; therefore we experiment with L_1 regularization. We analyze and devise instance selection methods for a given source sentence to increase the performance of the word alignment. The performance is estimated by comparing with the phrase table obtained by GIZA++ (Och & Ney, 2003), which is a state of the art word alignment tool commonly used in phrase-based machine translation systems. GIZA++ combines the result of various statistical word alignment models and performs symmetrization of the generated directed alignments.

2. Regression Based Alignment Learning

Let the feature matrices $\mathbf{M}_X \in \mathbb{R}^{N_X \times m}$ and $\mathbf{M}_Y \in \mathbb{R}^{N_Y \times m}$ be obtained from m training instances such that each column of $\mathbf{M}_X$ ($\mathbf{M}_Y$) is obtained by a feature mapper $\Phi_X : X^* \rightarrow \mathbb{R}^{N_X}$ ($\Phi_Y : Y^* \rightarrow \mathbb{R}^{N_Y}$). The ridge regression solution using L_2 regularization is given in Equation 1:

$$\begin{aligned} \mathbf{H}_{L_2} &= \underset{\mathbf{H} \in \mathbb{R}^{N_Y \times N_X}}{\arg\min} \parallel \mathbf{M}_Y - \mathbf{H}\mathbf{M}_X \parallel_F^2 + \lambda \parallel \mathbf{H} \parallel_F^2 &(1)\\ &= \mathbf{M}_Y \mathbf{M}_X^T (\mathbf{M}_X \mathbf{M}_X^T + \lambda \mathbf{I})^{-1} &(2)\\ \mathbf{H}_{L_1} &= \underset{\mathbf{H} \in \mathbb{R}^{N_Y \times N_X}}{\arg\min} \parallel \mathbf{M}_Y - \mathbf{H}\mathbf{M}_X \parallel_F^2 + \lambda \parallel \mathbf{H} \parallel_1 . &(3) \end{aligned}$$

$\mathbf{H}_{L_2}$ does not give us a sparse solution as most of the coefficients remain non-zero. L_1 norm behaves both as a feature selection technique and a method for reducing coefficient values. Equation 3 presents the *lasso* (least absolute shrinkage and selection operator) (Tibshirani, 1996) solution where the regularization term is now the L_1 matrix norm defined as $\parallel \mathbf{H} \parallel_1 = \sum_{i,j} |H_{i,j}|$. $\mathbf{H}_{L_2}$ can be found by taking the derivative but since L_1 regularization cost is not differentiable, $\mathbf{H}_{L_1}$ can be found by optimization or approximation techniques.

We perform experiments with forward stagewise regression (Hastie et al., 2006) (FSR) and quadratic optimization (QP) techniques to find $\mathbf{H}_{L_1}$. The incremental forward stagewise regression algorithm increases the weight of the predictor variable that is most correlated with the residual by a small amount, ϵ, multiplied with the sign of the correlation at each step. As $\epsilon \rightarrow 0$, the profile of the coefficients resemble the *lasso* (Hastie et al., 2001). We can pose *lasso* as a QP problem as follows (Mørup & Clemmensen, 2007). We assume that the rows of $\mathbf{M}_Y$ are independent and solve for each row i, $\mathbf{M}_{y_i} \in \mathbb{R}^{1 \times m}$, using non-negative variables $\mathbf{h}_i^+, \mathbf{h}_i^- \in \mathbb{R}^{N_X \times 1}$ such that $\mathbf{h}_i = \mathbf{h}_i^+ - \mathbf{h}_i^-$:

$$\mathbf{h}_i = \parallel \mathbf{M}_{y_i} - \mathbf{h}_i \mathbf{M}_X \parallel_F^2 + \lambda \sum_{k=1}^{N_X} |h_{i,k}| \qquad (4)$$

$$\mathbf{h}_i = \underset{\tilde{\mathbf{h}_i}}{\arg\min}\ \frac{1}{2} \tilde{\mathbf{h}_i} \widetilde{\mathbf{M}_X} \widetilde{\mathbf{M}_X}^T \tilde{\mathbf{h}_i}^T - \tilde{\mathbf{h}_i} (\widetilde{\mathbf{M}_X} \mathbf{M}_{y_i}^T - \lambda \mathbf{1}) \qquad (5)$$

$$\text{s.t. } \tilde{\mathbf{h}_i} > 0, \ \ \widetilde{\mathbf{M}_X} = \begin{bmatrix} \mathbf{M}_X \\ -\mathbf{M}_X \end{bmatrix}, \ \ \tilde{\mathbf{h}_i} = \begin{bmatrix} \mathbf{h}_i^+ & \mathbf{h}_i^- \end{bmatrix}$$

Orthogonality of the coefficient matrix can be desirable since the L_2 regularization parameter penalizes in proportion to $\mathbf{H}^T\mathbf{H}$ and setting $\mathbf{H}^T\mathbf{H} = \mathbf{H}\mathbf{H}^T = \mathbf{I}$ corresponds to assuming that features are selected independently (i.e. correlation of source and target features is identity). Therefore, we also experiment with symmetric coefficient matrix $\mathbf{H}_S = \sqrt{\mathbf{H} \times \overleftarrow{\mathbf{H}}^T}$, where $\times$ stands for the element-wise multiplication operator and $\overleftarrow{\mathbf{H}}$ is the coefficient matrix obtained when solving the inverse problem (i.e. estimating $\mathbf{M}_X$ by using $\overleftarrow{\mathbf{H}}\mathbf{M}_Y$).

3. Experiments

Training set contains about 80K English-German parallel news articles available from WMT2009 (Koehn & Haddow, 2009). We conducted experiments on 10 sentences with 10 tokens (*short*) and another 10 sentences with 20 tokens (*long*). The feature mappers are 3-spectrum counting word kernels, which consider all N-grams up to order 3 weighted by the number of tokens in the feature. Proper selection of training instances plays an important role to learn feature mappings within limited time and at expected accuracy levels. Instance selection is performed with the *tf-idf* (term frequency, inverse document frequency) weighting using the cosine similarity. We experiment with different instance selection methods: (i) per source sentence, (ii) per source sentence feature, (iii) instances' longest common matches per source sentence feature. Selection (ii) selects instances per feature (*ipf*) either proportional to the *length* of the feature, f, ($ipf = n \times \text{length}(f)$) or *dynamically* proportional to $n/\log(1 + \text{idfScore}(f)/9.0)$. Dynamic instance selection select more instances from rare features whose *idf* scores are higher. Selection (iii) uses only the longest matching parts to try to remove features coming from irrelevant tokens. We discard features that are observed less than three times from the training set.

Evaluation: We evaluate the performance of the coefficient matrix, $\mathbf{H}$, by measuring the precision, recall, and fmeasure when compared with the entries in the phrase table, *PT*, obtained by GIZA++ using the full training set. Let T contain the training indices of the target features in the *PT* that match the source sentence features, S, found in $\mathbf{H}$ whose values are greater than zero, then we define:

$$\textit{precision} = \frac{\sum_{i \in S} \sum_{j \in T} \mathbf{H}_{j,i}\ PT_{i,j}}{\sum_{j \in T} \sum_{* > 0} \mathbf{H}_{j,*}} \tag{6}$$

$$\textit{recall} = \frac{\sum_{i \in S} \sum_{j \in T} \mathbf{H}_{j,i}\ PT_{i,j}}{\sum_{i \in S} \sum_{j \in T} PT_{i,j}} \tag{7}$$

$$\textit{fmeasure} = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}} \tag{8}$$

where $\mathbf{H}_{j,i}$ stands for the coefficient for target feature j, t_j, and source feature i, s_i, $\sum_{*>0} \mathbf{H}_{j,*}$ sums over all the entries in row j that are greater than 0, and $PT_{i,j}$ is the multiplication of the lexical translation probabilities $p(s_i|t_j)$ and $p(t_j|s_i)$ found in *PT*. We also use *top3%*, which measures the percentage of observing the top 3 scored target features in the phrase table translations, *sqLoss*, which measures the squared loss of the estimation with respect to the target sentence, and *cov.*, which measures the average coverage of the training set in representing the target sentence. Table 1 presents our evaluation of the performances of different techniques when training instances are selected dynamically with $n = 4$. The effectiveness of selection (iii) can be seen in the increase in the precision, recall, and fmeasure metrics and decrease in computation time in Table 2.

Conclusion: Our findings are listed below:

- L_1 regularization helps improve the performance. L_2 solution performs worse. QP in general perform better than FSR but takes very long time.
- Symmetrization helps in improving precision, recall, and fmeasure score. It reduces *sqLoss* in FSR and sometimes in QP solutions.
- *Coverage* and *top3%* increase as we select more instances, but this decreases precision and *sqLoss* due to adding more noise.
- QP quickly becomes infeasible due to increased computation time when N_X and N_Y increase. Selection (iii) helps us increase precision, recall, and fmeasure without increasing the *sqLoss* too much.

References

Hastie, T., Taylor, J., Tibshirani, R., & Walther, G. (2006). Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, *1*.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*. Springer-Verlag.

Koehn, P., & Haddow, B. (2009). Edinburgh's submission to all tracks of the WMT 2009 shared task with reordering and speed improvements to Moses. *Proceedings of the Fourth Workshop on Statistical Machine Translation* (pp. 160–164). Athens, Greece: Association for Computational Linguistics.

Mørup, M., & Clemmensen, L. H. (2007). Multiplicative updates for the LASSO. *MLSP 2007*.

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*, 19–51.

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

Table 1. Numbers represent averages. Time is in seconds. S suffix is for symmetrized techniques.
Top: Performances of different techniques when training instances are selected dynamically with $n = 4$.
Bottom: Selection (i) results using *long* set of sentences. 50 and 100 instances per sentence are selected.

n=4, dynamic		prec.	recall	fmeas.	top3%	sqLoss	time
short	**L2**	0.007	0.038	0.011	0.206	39.521	0.204
	L2S	0.006	0.039	0.010	0.223	69.514	0.674
	QP	0.061	0.062	0.061	0.377	26.208	335.121
	QPS	0.162	0.072	0.098	0.377	30.281	352.124
	FSR	0.038	0.070	0.049	0.335	62.973	24.490
	FSRS	0.193	0.076	0.106	0.318	32.456	23.674
long	**L2**	0.0	0.034	0.009	0.297	69.240	0.960
	L2S	0.0	0.037	0.008	0.276	129.042	1.932
	QP	0.066	0.081	0.072	0.419	51.146	1105.915
	QPS	0.189	0.095	0.125	0.419	51.172	1058.366
	FSR	0.056	0.094	0.069	0.362	99.644	73.107
	FSRS	0.239	0.102	0.141	0.353	53.125	79.008

n, selection (i)		prec.	recall	fmeas.	top3%	sqLoss	time
50	**L2**	0.010	0.033	0.015	0.172	43.242	0.067
	L2S	0.009	0.036	0.014	0.186	50.194	0.137
	QP	0.091	0.056	0.068	0.350	37.885	31.119
	QPS	0.255	0.054	0.087	0.335	31.685	30.879
	FSR	0.051	0.085	0.063	0.285	79.713	3.906
	FSRS	0.321	0.085	0.131	0.275	33.421	3.190
100	**L2**	0.007	0.035	0.011	0.251	55.511	0.363
	L2S	0.006	0.038	0.010	0.254	74.590	0.815
	QP	0.089	0.071	0.079	0.426	43.309	416.296
	QPS	0.257	0.082	0.123	0.417	39.404	423.335
	FSR	0.053	0.085	0.065	0.377	84.525	31.043
	FSRS	0.294	0.091	0.137	0.358	43.266	27.953

Table 2. Numbers represent averages taken over the *long* set of sentences. Time is in seconds.
Top: QP performance when training instances are selected *dynamically* and with proportion to *length*.
Bottom: QP performance when training instances are selected *dynamically* with n and only matching parts are used as training sentences.

QP	n	ipf	cov.	prec.	recall	fmeas.	top3%	sqLoss	time
dynamic	**1**	1.616	0.324	0.083	0.074	0.077	0.330	42.534	113.205
	2	1.663	0.328	0.081	0.073	0.076	0.342	44.195	143.112
	3	2.111	0.360	0.076	0.080	0.077	0.359	49.779	508.252
	4	2.704	0.378	0.066	0.081	0.072	0.419	51.146	1105.915
length	**1**	1.616	0.324	0.083	0.074	0.077	0.330	42.534	114.167
	2	1.954	0.347	0.074	0.075	0.073	0.359	48.205	411.712
	3	2.439	0.365	0.066	0.079	0.071	0.394	49.872	1132.119
	4	3.113	0.385	0.057	0.079	0.066	0.435	51.777	2508.383

n	m	N_X	N_Y	ipf	cov.	prec.	recall	fmeas.	top3%	sqLoss	time
2	81.000	385.700	427.500	1.737	0.243	0.095	0.072	0.081	0.222	41.411	29.661
3	103.500	428.900	474.000	2.214	0.250	0.106	0.084	0.093	0.250	43.289	25.440
4	133.600	433.700	479.900	2.849	0.254	0.113	0.091	0.100	0.263	43.243	52.357
5	162.000	441.600	490.200	3.450	0.256	0.120	0.091	0.102	0.279	43.399	52.019
6	190.300	441.600	494.100	4.048	0.262	0.122	0.096	0.105	0.283	44.083	92.475
7	216.500	442.000	495.800	4.605	0.264	0.129	0.101	0.112	0.286	44.110	89.570
8	242.400	442.300	497.600	5.148	0.270	0.131	0.101	0.112	0.287	44.310	90.859
9	266.800	442.700	498.700	5.662	0.270	0.134	0.100	0.113	0.296	44.249	155.055
10	290.800	443.000	500.100	6.165	0.273	0.136	0.099	0.113	0.298	44.343	175.650