# Classifying Dialogue in High-Dimensional Space

JOSÉ P. GONZÁLEZ-BRENES and JACK MOSTOW
Project LISTEN, School of Computer Science, Carnegie Mellon University

The richness of multimodal dialogue makes the space of possible features required to describe it very large relative to the amount of training data. However, conventional classifier learners require large amounts of data to avoid over-fitting, or do not generalize well to unseen examples. To learn dialogue classifiers using a rich feature set and fewer data points than features, we apply a recent technique, $\ell_1$-regularized logistic regression. We demonstrate this approach empirically on real data from Project LISTEN's Reading Tutor, which displays a story on a computer screen and listens to a child read aloud. We train a classifier to predict task completion (i.e., whether the student will finish reading the story) with 71% accuracy on a balanced, unseen test set. To characterize differences in the behavior of children when they choose the story they read, we likewise train and test a classifier that with 73.6% accuracy infers who chose the story based on the ensuing dialogue. Both classifiers significantly outperform baselines and reveal relevant features of the dialogue.

Categories and Subject Descriptors: I.2.m [**Computing Methodologies**]: Artificial Intelligence—*Spoken Dialogue Systems*; H.4.3 [**Information Systems Applications**]: Communications Applications; D.m [**Software**]: Miscellaneous—*Feature Engineering*

General Terms: Spoken Dialogue Systems, Feature Engineering, Feature Selection

Additional Key Words and Phrases: Task Completion, $\ell_1$-regularized logistic regression, Project LISTEN's Reading Tutor

## 1. INTRODUCTION

Multimodal spoken human-computer dialogue is rich in detail, containing streams of events of different types and grain sizes, whose features may change over different timescales. It is therefore challenging to discover useful characterizations of such dialogue, such as predicting whether, when, or how well it will succeed, or discerning how it is affected by features such as whether the current activity was chosen by the user or by the system.

To be useful, such characterizations should ideally be concise, expressed in terms of a handful of observable features of the dialogue. But in order to discover such descriptions, the discovery process must be able to consider a sufficiently rich set of features to include the desired handful.

The space of possible descriptions explodes combinatorially as the number of features considered increases. This explosion typically requires a commensurate

Authors address: Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA

increase in the size of the training set in order to avoid overfitting the data, so it can generalize to unseen examples.

Fortunately, recent advances in statistical learning have made it possible to search rich feature spaces and find sparse models that generalize well to unseen test data despite using small amounts of training data. These techniques have been applied in computational biology [Shevade and Keerthi 2003], but have not previously been applied to classify spoken dialogue data with many features.

The rest of this paper is organized as follows. Section 2 discusses previous research. Section 3 describes our approach. Section 4 applies it to multimodal spoken tutorial dialogues logged by Project LISTEN's Reading Tutor, which listens to children read aloud and helps them learn to read [Mostow and Aist 2001]. Section 5 tests the approach on two dialogue classification problems: predicting whether the student will complete the task of reading the story, and inferring from the student's reading behavior who chose the story, the tutor or the student. Section 6 concludes.

## 2. RELATION TO PRIOR WORK

Task-oriented spoken dialogue systems are systems where the user and the computer agent try to accomplish a task together by using spoken language. Completion of an entire dialogue is usually required to accomplish the user's goal in a task-oriented system, making task completion an important component of automatic evaluation methodologies [Hajdinjak and Mihelic 2006].

In a task-oriented system, the goal is to complete a task for its own sake with as little work by the user as possible. In contrast, here we analyze data from a tutorial spoken dialogue system. Its goal is for the user to learn, typically by doing as much of the shared task as possible, with as little help from the system as possible. Thus, the shared oral reading goal of the Reading Tutor is for the student to read as much of the text as possible and for the tutor to help only as much as necessary.

Although prior analyses of spoken dialogue systems have focused on task-oriented systems rather than tutoring systems, this paper is nevertheless related to two categories of applications of machine learning from dialogue: (i) dialogue classification and (ii) learning dialogue strategies.

### 2.1 Dialogue Classification

An important example of a dialogue classification problem is predicting task completion: classifying whether the dialogue is going to fail or if it is going to be completed successfully. For example, predicting task completion has been used to direct callers to a human operator, when it is likely that the automatic system will not be able to handle the dialogue correctly [Walker et al. 2000].

Predicting task completion has been studied extensively in the context of automatic evaluation metrics for task-oriented spoken dialogue systems [Walker et al. 2001; Hajdinjak and Mihelic 2006; Möller et al. 2007; Möller et al. 2008; González-Brenes et al. 2009]. Previous work on dialogue classification has used carefully engineered, relatively small feature sets and large training corpora. For example, [Walker et al. 2000] used a data set consisting of around 5,000 dialogues and only 53 features to train a binary classifier to detect, early in the dialogue, cases where the user cannot accomplish a task. Later work formulated multi-class classification problems to predict user satisfaction in the context of the DARPA Communicator

Challenge [Hastie et al. 2002] and the SympaFly corpus of flight booking dialogues [Steidl et al. 2004]. Examples of features relevant to task completion in task-oriented systems include confidence scores output by the speech recognizer, and the number of retrials in a specific dialogue state. These features may or may not be relevant to task completion in a tutoring system.

### 2.2 Learning Dialogue Strategies

A growing body of work has investigated how to use policy learning to improve tutorial effectiveness [Boyer et al. 2010; Boyer et al. 2010; Chi et al. 2010; Chi et al. 2008; Ai et al. 2007; Beck and Woolf 2000; Beck et al. 2000]. Learning a strategy for what to do at each point in a dialogue is a different problem than learning a classifier. However, classifier and policy learning problems encounter some of the same issues, such as combinatorially large feature spaces and sparse training data.

Reinforcement learning models a dialogue system as a Markov Decision Process [Singh et al. 1999], where the system's actions are its utterances and the learning goal is to discover a policy to take actions so as to maximize the expected value of a specified reward function. This approach has been applied to automated tutors to optimize learning gains [Chi et al. 2010]. Depending on the reward function, expected reward can predict task completion. However, it is not clear how to map reinforcement learning to other classification tasks, such as inferring who chose the story.

### 2.3 Learning with Limited Training Data

Learning with limited training data is a recurrent theme in the dialogue community. For example, Henderson et al. [2008] learned dialogue policies from a large state space and a small corpus of dialogues using a hybrid approach that combined classifier learning with reinforcement learning. Rieser and Lemon [2010] worked around the sparsity of their data by artificially increasing the training data size with simulated dialogue data, generated by bootstrapping from small amounts of real data. Classical feature selection techniques constitute a third approach to taming large feature spaces, and have been explored in the context of learning policies for multimodal dialogue strategies [Rieser and Lemon 2006; 2010]. However, there is both theoretical and experimental evidence that these classical methods are outperformed by newer algorithms based on $\ell_1$-regularization [Yuan and Lin 2006; Park and Hastie 2007].

To our knowledge, we are the first to apply $\ell_1$ regularization to tutorial spoken dialogue systems. We shall describe this technique shortly, but first we summarize why we chose it over some alternatives.

Yang and Pedersen [1997] review different metrics, such as $\chi^2$ and Mutual Information, to rank features that score above a threshold as the means to perform feature selection. Other, more sophisticated greedy algorithms are reviewed in [Lee et al. 2007; Xing et al. 2001]. However, recent results [Zhang and Huang 2008] suggest that greedy methods are not always stable or computationally efficient for high-dimensional data, and that $\ell_1$-regularization should be preferred. Park and Hastie [2007] claim not only that $\ell_1$-regularization offers better performance than forward stepwise feature selection algorithms, but also that it finds models throughout the entire range of complexity. Forward stepwise regression stops before finding the most complex models, whereas $\ell_1$ regularization need not.

## 3. APPROACH

This section describes our approach for modeling dialogues in high-dimensional space. Section 3.1 reviews logistic regression, a classifier learning method used very successfully in language technologies [Rosenfeld 2000] and in the spoken dialogue community [Rieser and Lemon 2006]. Unlike Naïve Bayes classifiers, logistic regression does not assume independence among the features used. Unfortunately, Logistic Regression is prone to overfit the data when used with a high number of features. To address this issue, Section 3.2 introduces $\ell_1$-regularization.

### 3.1 Logistic Regression

Logistic regression, also called maximum entropy classification, is a supervised learning method: given labelled training data, it learns to label unseen test data. More formally, we represent each of the $n$ data points as a $p$-dimensional vector $x$ with a dimension for each feature, and a class label $y$, which for binary logistic regression is either +1 or -1. The probability of classifying $x$ as class +1 is [Ng 2004]:

$$P(y = +1|x; \theta) = \frac{1}{1 + \exp(-\theta \cdot x)} \tag{1}$$

Here we represent the parameters of the model as the vector $\theta \in \mathbb{R}^p$.

We assume that the training data $S = \{(x^{(i)}, y^{(i)})\}_1^n$ are independent and identically distributed (iid), drawn from the true distribution $\mathcal{D}$. Intuitively, training the model fits its parameters by finding the $\theta^*$ that maximizes the probabilities assigned by Equation 1 to the training labels [Ng 2004; Nigam et al. 1999]. More precisely:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_i^n \log P(y^{(i)}|x^{(i)}; \theta) \tag{2}$$

The objective function in Equation 2 is convex, but does not have a closed-form solution. However, it can be solved using numerical methods, such as Improved Iterative Scaling [Berger et al. 1996]. Unfortunately, regardless of the optimization procedure, logistic regression tends to overfit when using a large number of features [Nigam et al. 1999].

### 3.2 High-Dimensional Feature Selection

Using a large number of features allows overfitting the training data so the learned model does not generalize to unseen data. To prevent overfitting, we can transform Equation 2 by adding a complexity penalty proportional to the number of non-zero parameter values. Unfortunately, an exhaustive search for the smallest subset of features to include becomes intractable even with a modest number of features, since optimal feature selection is NP-hard [Guyon and Elisseeff 2003; Amaldi and Kann 1998].

Instead, we use a method based on a complexity-penalized model that is tractable to optimize. $\ell_1$-regularization was first proposed for linear regression in the seminal

LASSO paper [Tibshirani 1996], to optimize the trade-off between fitting the data and model complexity. However, the concept also works for generalized linear models, such as logistic regression. Instead of penalizing the number of (non-zero) parameters, this approach modifies Equation 2 by adding a complexity penalty proportional to the $\ell_1$-norm of the vector of parameters, that is, the sum of their absolute values. This penalty has the effect of being able to find models where many parameters are zero. We refer to features with non-zero parameters as the "features selected." As presented in Ng [2004], training this model means finding $\theta^*$:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i}^{n} \log P(y^{(i)}|x^{(i)};\theta) - \lambda||\theta||_1 \tag{3}$$

Equation 3 combines feature selection and logistic regression to perform them jointly. Here $\lambda$ is a hyper-parameter that controls the trade-off between bias and variance of the model. The higher the value of $\lambda$, the more biased the model and the smaller the number of features selected. Conversely, the smaller the value of $\lambda$, the better the fit to the training data, but the higher the risk of overfitting. $\lambda$ is customarily selected by cross-validation on a held-out subset of the training set.

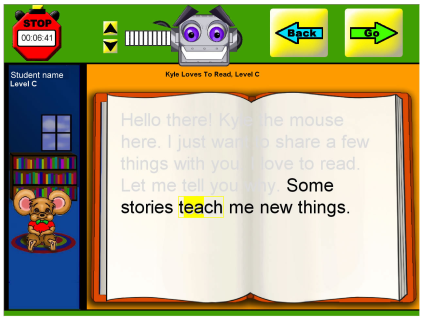
More generally, the $\ell_m$ norm of a vector $\theta$ is defined as:

$$||\theta||_m := (\sum_{i=0}^{p} |\theta_i|^m)^{1/m} \tag{4}$$

Ng [2004] extensively investigates the difference between $\ell_1$ and $\ell_2$ regularization in logistic regression, concluding that although $\ell_1$ regularization is more difficult to optimize, since it is not differentiable at every point, it tends to produce sparser models, where many parameters weights have value 0. Furthermore, $\ell_1$ regularization also gives theoretical guarantees that under some weak assumptions, the number of features selected does not exceed the number of training data points. More importantly, recent theoretical results [Lee et al. 2007] show that the sparsity guarantees of $\ell_1$-regularization allow the number of features to grow exponentially in the number of training samples in density estimation [Dudik et al. 2004] and in logistic regression [Ng 2004], making it possible to learn models where the dimensionality of $x$ may be much larger than the number of training data points (that is, $p \gg n$).

Logistic regression finds a linear decision boundary. It is possible to escape this restriction by projecting the features into a high-dimensional space using kernels [Keerthi et al. 2005]. However, efficient solutions of kernel methods assume $\ell_2$-regularization [Hastie et al. 2003]. Moreover, models built using kernel methods can be hard to interpret [Okanohara and Tsujii 2009].

To train $\ell_1$-regularized logistic regression models, we use a MATLAB implementation of the Dual Augmented Lagrangian method (DAL) [Tomioka and Sugiyama 2009], an optimization algorithm that solves regularized minimization problems. In the experiments to be described in Section 5, DAL converged to the global optimum in seconds, much faster than a BFGS-style algorithm [Schmidt et al. 2008].



Hello there! Kyle the mouse here. I just want to share a few things with you. I love to read. Let me tell you why. Some stories teach me new things.

Fig. 1. Project LISTEN's Reading Tutor

## 4. REPRESENTING DIALOGUE IN A HIGH DIMENSIONAL SPACE: EXAMPLE

In this section we apply the $\ell_1$-regularized logistic regression approach to two dialogue classification problems, using data collected by the tutorial spoken dialogue system described in Section 4.1. Section 4.2 describes the classification problems. Section 4.3 explains our feature engineering.

### 4.1 Project LISTEN's Reading Tutor

We use data logged by Project LISTEN's Reading Tutor, which listens to a child read aloud, and takes turns picking stories to read [Mostow and Aist 2001]. The Reading Tutor adapts the Sphinx-II speech recognizer [Huang et al. 1993] to analyze the students' oral reading, and intervenes when it notices the reader make a mistake, get stuck, click for help, or encounter difficulty. Figure 1 shows a screen shot of the 2005 Reading Tutor. The sentence being read is in boldface, and the tutor is giving help on the highlighted word.

### 4.2 Modeling Task Completion and Story Choice Initiative

We used two classification problems to test our approach for modeling dialogues with a rich feature set. For both problems, we will use "multimodal dialogue" to refer to the interaction of a student reading one story with the tutor. Other problems can involve other units of analysis, whether shorter (e.g. a single sentence) or longer (e.g. an entire session from login to logout).

4.2.1 *Task Completion.* We want to predict at runtime, based on the dialogue so far, whether the student will finish reading the story or is about to stop reading. Accordingly, we calculate all features using only information available at prediction time. Thus for positive training examples, we truncate each finished story to a random number of sentence encounters before calculating features. For negative training examples, we use unfinished stories, but we do not truncate them, because our goal is to discover features that signal disengagement, which may not arise until later in a story. Such truncations might well be harder to classify.

4.2.2 *Story Choice Initiative.* The Reading Tutor and the student take turns deciding which story to read. Previous research showed that who has this initiative affects how often students read new material [Aist and Mostow 2000; 2007], the

Table I. Raw dialogue features considered

**Dynamic features**

**Prosodic features:** Various duration, pitch, and intensity features detailed in [Mostow and Duong 2009]

**Sentence features:** Features of a sentence to be read, such as percentage of story read so far, number of word types and tokens, number of clicks for help, and statistics on word length and frequency

**Static features**

**Student features:** Grade(K-6), age, story choice initiative, number of stories read by the student

**Story features:** Story length, popularity of the story, how often do children finish this story, did this student finish the story

difficulty level of the text [Mostow et al. 2003], and how much they learn [Beck 2007].

Here our goal is to discover how story choice initiative affects reading behavior. That is, how does the tutorial dialogue differ as a function of who picked the story? To discover such differences, we train a model to classify who chose the story, based on features of the ensuing dialogue. In contrast to the task completion model, which predicts a future event (finishing a story), the story choice model "predicts" a past event as a way to characterize its effects on the dialogue.

## 4.3 Features

Table I summarizes the sorts of dialogue features we use to model task completion and story-choice initiative. Static features have the same value throughout the entire dialogue, while dynamic features may change over the course of the story because they change over time or differ from sentence to sentence.

To give the model greater expressive power, we systematically derive three additional types of features from each basic feature listed in Table I.

4.3.1 *Feature windows.* Many dialogue features are dynamic rather than static over the course of the dialogue. For instance, the average value of a prosodic feature calculated from the whole dialogue, its beginning, or its end, may yield very different values. We want our algorithm to learn the relative importance of each feature across time.

To include dynamic features in our feature vector, we first decide the size $w$ and position in the dialogue of the window over which to compute them. Figure 2 illustrates three ways to extract dynamic features; each row represents a dialogue, and the circles correspond to different sentence encounters. The shadowed circles represent where the features are extracted, namely the first three sentence encounters (head) of the dialogue ($h_3$), the last three encounters ($t_3$), or the first and last three sentence encounters ($h_3t_3$).

If a dialogue is shorter than $w$, we have missing values for some positions. To assign values to them, we follow standard practice for logistic regression: we use the average over the positions for which we have values. If a dialogue is between $w$ and $2w$ sentence encounters, its head and tail features will overlap, but they are still defined, so we use them anyway.

Once we determine the dialogue window over which to extract dynamic features,

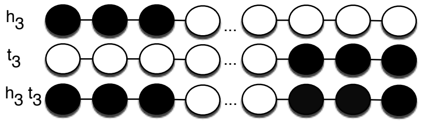


Fig. 2. Comparison of feature extraction strategies

we decide whether to (i) use their individual values as separate features, or (ii) use their average as a single aggregated feature. For the individual features strategy, each colored circle in Figure 2 represents a separate feature. This strategy multiplies the number of dynamic features by the chosen window size $w$ (or $2w$ for a double window $h_w t_w$).

For the aggregated feature strategy, each group of three colored circles represents an aggregate feature. For example, if the "pitch variation" feature on the first three sentence encounters is 8, 109, and 148, then it can be encoded (i) as three different features, or (ii) as their average (88.3). This strategy leaves the number of dynamic features unchanged for $h_w$ and $t_w$ models, but doubles them for $h_w t_w$ (double window) models. Our experiments included comparison of the individual and aggregated feature extraction strategies.

After adding feature windows, we further tranform the expanded set of features by squaring and thresholding them, as we now describe.

4.3.2 *Quadratic Features.* Squaring the value of a feature is a standard technique to capture quadratic relationships between the feature and the output variable. Accordingly, for every dialogue feature (whether static or dynamic) we define its square as another feature, thereby doubling the number of features.

4.3.3 *Threshold Features.* Although not specific to dialogue systems, learning threshold features automatically is useful in the context of feature engineering. A threshold feature is a binary feature that activates when a dialogue feature is less than a specific value [Dudik et al. 2004]. Introducing different thresholds explodes the size of the feature space, but can express non-linear effects. For our experiments we create threshold features for all non-binary dialogue features. As threshold values we use multiples of 0.1 standard deviations of the value of the dialogue feature in the training set, ranging from $-3.0$ to $+3.0$: $\{-3.0, -2.9, \ldots, +2.9, +3.0\}$. The transformed feature set can express distinctions between feature values as close as 0.1 standard deviations apart, at the cost of multiplying the number of non-binary dialogue features (both static and dynamic) by 61.

Composing the quadratic and thresholding feature transformations with windowing vastly expands the set of features considered.

## 4.4 Data Preparation

We queried the database logged by the Reading Tutor for the raw dialogue features described in Table I, from which we computed the additional features described in Section 4.3. To predict task completion, we use "task completion" as a data label rather than a dialogue feature, because its value is unknown at prediction time. Similarly, to avoid trivial results we refrain from extracting features of the last sentence encounter in the dialogue, such as clicking on the "STOP" button. To model story choice, we similarly use "initiative" as a data label rather than a dialogue feature. We include the rest of the dialogue features except for student features, which we omit because our goal is to characterize how students behaved rather than which students happened to pick more stories.

Using the entire data set, we cap outlier values of continuous features below the $5^{th}$ percentile and above the $95^{th}$ percentile (this preprocessing step is called winsorization). To increase numerical stability and compare the contributions of different features, we perform the standard transformation of centering the feature values as z-scores with mean zero and standard deviation one. To speed up training, it is common practice in machine learning experiments to remove features that have the same value with fewer than 15 exceptions. Instead of the fixed minimum of 15 examples, we use the corresponding percentage of our training data, namely 0.7%, so as to adjust the minimum appropriately in our experiments on varying the amount of training data.

## 5. EXPERIMENTAL EVALUATION

The data set we used was logged by the Reading Tutor while used regularly at elementary schools during the 2005-2006 school year. For both problems, we included only dialogues with at least four sentence encounters, as a minimum amount of data from which to extract dynamic features.

For predicting task completion by 162 children, we used a balanced set of 2,112 story readings, i.e., as many completed as not. The dialogues average 18 sentences long.

Unless noted otherwise, we report all results using 10-fold cross-validation across students. That is, we train on 90% of the students, and test on the other 10%. Following standard practice in machine learning, we optimize the hyper-parameter $\lambda$ (explained in Section 3.2) on a development set held out from the training data. Although the folds vary in size, non-overlap of students between training and test data is necessary to estimate accuracy on unseen students. The conventional alternative, called random $k$-fold cross-validation, uses equal-size folds but allows such overlap. Training and testing on the same students would risk relying on peculiarities of individual students. We compute the classification accuracy on each fold, average across the folds to compute mean accuracy, and report its 95% confidence interval as computed by MATLAB.

The purpose of our evaluation was to answer the following questions. First, how well did our approach work? Section 5.1 analyzes how accurately our approach solves the two classification problems, compared to baseline approaches.

Next, what did we learn about applying our approach to spoken dialogue? Section 5.2 describes how accuracy varies with the number of features. Section 5.3 describes how accuracy varies with the amount of training data.

Finally, what did we learn about the problem domain? Section 5.4 identifies the most predictive features for each problem.

### 5.1 Classification Accuracy

Figures 3 and 4 show classification accuracy for the two problems. Their horizontal axes show window size in sentence encounters, and their vertical axes show classifi-

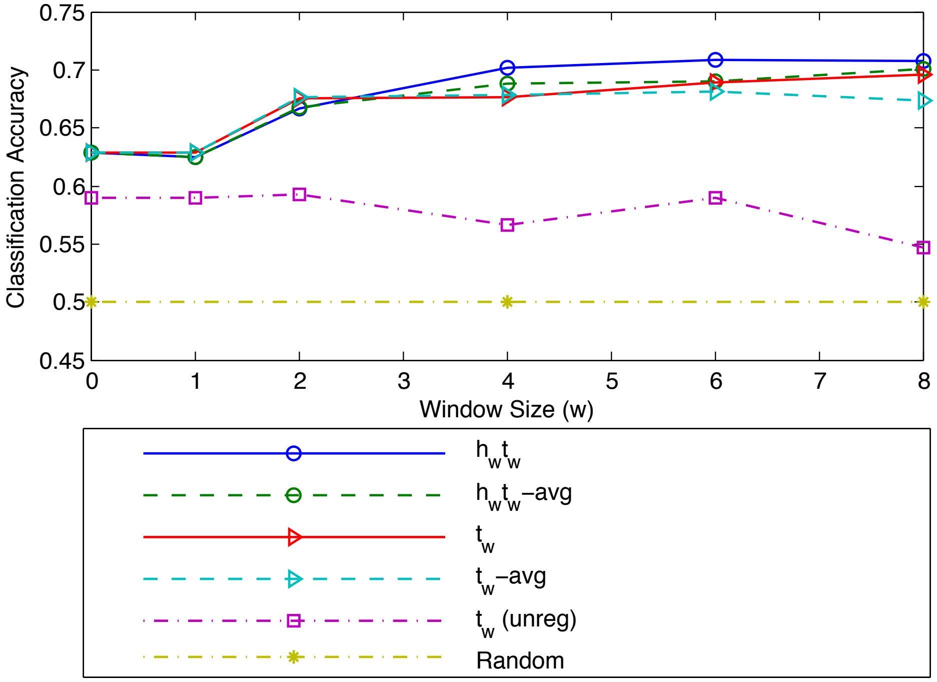


Fig. 3. Classification accuracy for Task Completion

cation accuracy. Accuracy varies with window size, so each figure compares models over a range of window sizes $w$ from 0 (i.e. without dynamic features) to 8. The six models compared in each figure consist of four $\ell_1$-regularized logistic regression models and two baseline models:

(1) $h_w t_w$: uses individual features across a double window
(2) $h_w t_w$-avg: averages features across a double window
(3) $h_w$ or $t_w$: uses individual features across a single window
(4) $h_w$-avg or $t_w$-avg: averages features across a single window
(5) $h_w$ (unreg) or $t_w$ (unreg): an unregularized logistic regression baseline that averages features across a single window. The unregularized solution does not have any hyper-parameters to tune, and its solution is unique. We didn't bother to include $h_w t_w$ (unreg) as a model because adding more features to an unregularized model would have caused even worse overfitting.
(6) Random baseline: guesses randomly, with 50% expected classification accuracy

For predicting task completion, recent sentence encounters turn out to be more informative than initial sentence encounters, so $t_w$ models outpredict $h_w$ models. The reverse is true for story choice initiative. Accordingly, we reduce clutter by omitting $h_w$ models from Figure 3 and $t_w$ models from Figure 4.

As Figures 3 and 4 show, all four $\ell_1$-regularized logistic regression models outperform the random-guess baseline. With one exception, they also outperform the

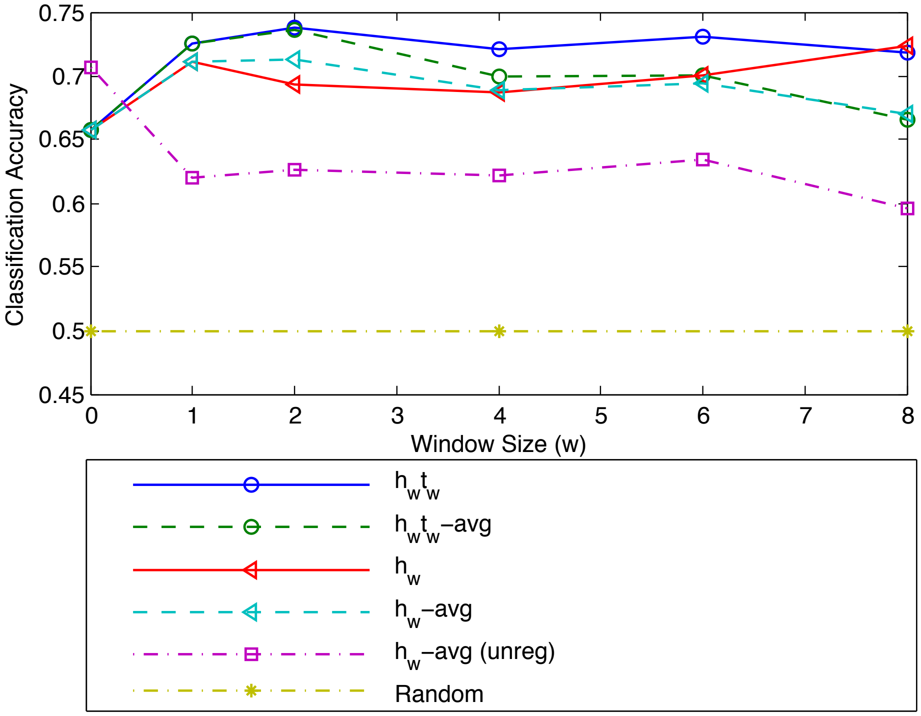


Fig. 4. Classification accuracy for Story Choice Initiative

unregularized logistic regression baseline, whose accuracy ranges from 0.55 to 0.63, and tends to get worse as window size increases. The exception occurs for story choice initiative at $w = 0$, where the 0.71 ±0.07 accuracy of the unregularized baseline exceeds the 0.66 ±0.03 accuracy of the $\ell_1$-regularized methods. However, this difference is not statistically significant (their 95% confidence intervals overlap).

The relation of window size $w$ to accuracy is problem-specific. For predicting task completion, performance improves with window size, and $h_8 t_8$ does best, with accuracy $0.71 \pm 0.04$. For modeling story choice initiative, $h_2 t_2$ performs best, with $0.74 \pm 0.03$ accuracy, but $h_w t_w$ for $w$ as large as 8 do nearly as well, with statistically insignificant differences.

## 5.2 Accuracy Versus Number of Features

Several modeling decisions affect the number of features considered: window size $w$; single window $h_w$ or $t_w$ vs. double window $h_w t_w$; individual features vs. averaging; and the hyper-parameter $\lambda$. How do these decisions affect classification accuracy of the resulting models for the two problems?

Individual and aggregate features models are equivalent for window size $w$ of 0 or 1, but diverge starting at $w = 2$. The top models for both classification problems use double windows, for which individual features work as well as or better than averaging them. This trend extends to single windows except for a

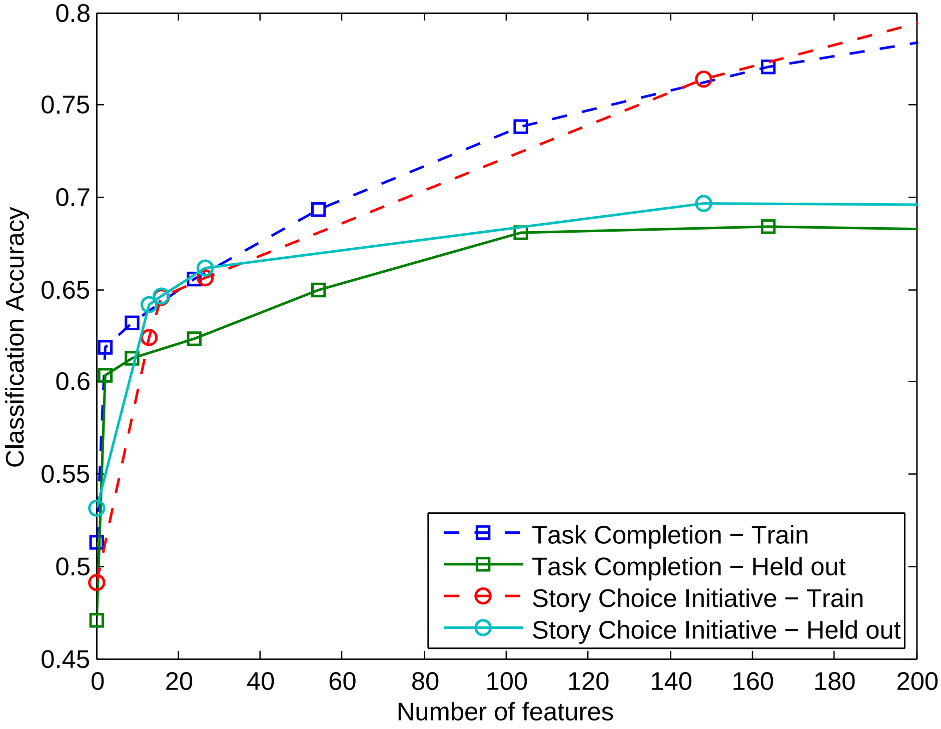


Fig. 5. Classification accuracy for different numbers of features

statistically insignificant difference in the opposite direction between $h_2$ and $h_2$-avg. Double windows with individual features have the most features for a given window size, suggesting that accuracy may tend to increase with the number of features considered. This hypothesis is consistent with the $h_8t_8$ model doing best in Figure 3, and although the $h_2t_2$ model beats $h_8t_8$ in Figure 4, the difference is not statistically significant.

By varying the hyper-parameter $\lambda$ of Equation 3, $\ell_1$-regularized logistic regression finds models along a spectrum of complexity. Since all of the dialogues have at least four sentence encounters, we analyze how accuracy changes in $h_4t_4$ models that select different numbers of features. As Figure 5 shows, test set classification accuracy on both problems increases the most between 0 and 20 features, and more gradually thereafter.

The most complex $h_8t_8$ models select on average only 523 of 68,721 features considered for predicting task completion and 187 of 72,402 features considered for modeling story choice initiative. By limiting the number of features selected, the regularization penalty prevents overfitting.

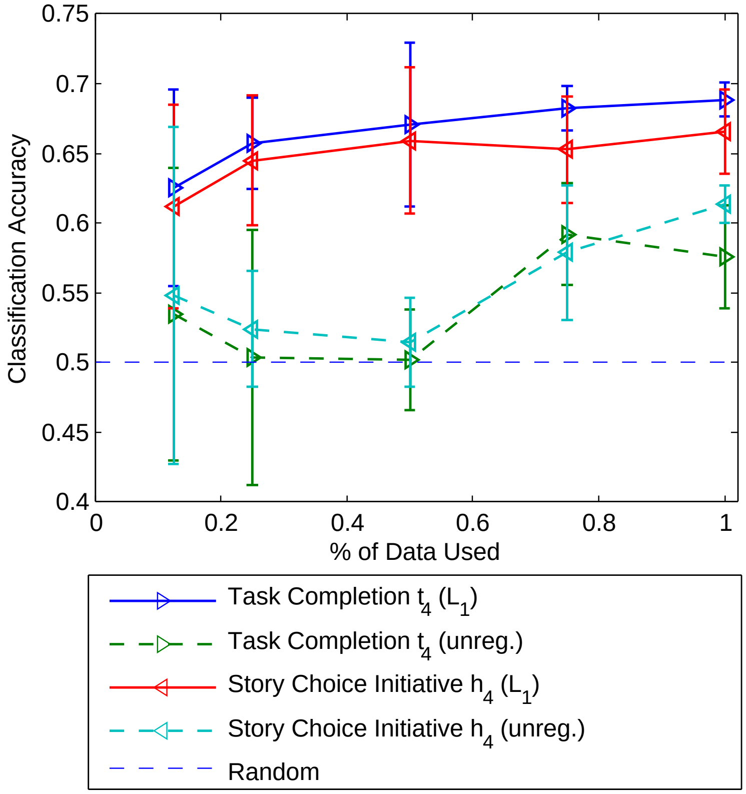


Fig. 6. Classification accuracy for different corpora size (for averaged models)

### 5.3 Accuracy Versus Amount of Training Data

To evaluate robustness to data scarcity, we vary the amount of training data in order to determine how much is required for our approach versus for the unregularized logistic regression baseline. Figure 6 compares their random 10-fold cross-validated accuracy on the two classification problems. The horizontal axis shows the amount of training data as a percentage of the original training corpus: 10%, 25%, 50% 75%, or 100%. To make the comparison as favorable as possible to unregularized logistic regression, we compare models with single rather than double windows and averaging rather than independent features.

As the error bars show, $\ell_1$-regularized logistic regression significantly outperforms unregularized logistic regression (and the random baseline) across the board, except perhaps when they use 10% of the training data (only 211 dialogues for predicting story completion, and 166 dialogues for modeling story initiative). Morever, unregularized logistic regression does not significantly outperform the random baseline except by using over 50% of the training data.

Table II. Top 4 Most Predictive Features For Task Completion

| Feature Name | $\theta$ |
|---|---|
| Percentage of words accepted | 0.1484 |
| Words per minute$_{<0.1}$ | -0.0152 |
| Correlation with adult word durations | 0.0134 |
| Percentage of story completed so far | 0.0105 |

### 5.4 Features Selected

The value of a model lies not only in its predictive accuracy but in what it reveals about the domain. Tables II and III show the most predictive features for the two problems using $h_8t_8$-avg models. Each cross-validation fold can return different features, so we report only the results of the first fold. Averaging features over an 8-sentence window should smoothe out noise and make results more interpretable. In Table II, the subscript "$< 0.1$" means that the feature is a binary threshold to represent when the z-score for the words per minute is less than 0.1. In Table III, a feature with a superscript "2" means that the value of the feature was squared. For consistency, all features were squared, although squaring binary features does not change their values.

As Table III illustrates, a model may contain two or more equivalent features, which may seem counter-intuitive for a method to attain a given level of accuracy with as few features as possible. Logistic regression is guaranteed to find an optimal solution to Equation 3, but multiple equally good solutions may exist, because the problem is convex but not strictly convex. For example, if two features are equal, reweighting them in the model without changing their weighted sum or the sum of the absolute values of their coefficients yields the same optimal value for the objective function in Equation 3.

For predicting task completion, positive feature weights contribute to the probability of finishing the story. The most informative features in this case reflect three aspects of oral reading fluency: oral reading accuracy, speed, and prosody. The more fluently the student is reading, the likelier he will finish the story. The fourth feature means that the further the student has gotten in the story, the likelier he will finish.

For story choice initiative, positive feature weights contribute to the probability that the student chose the story. The most informative features in this case involve the level of difficulty and how often students (not just this student) finish the story. Students are likelier than the tutor to pick easy stories that students often finish.

Both models make sense after the fact. However, neither model was known beforehand, nor did we know which of many features considered would turn out to be most informative. Omitting some informative features could reveal more interesting ones. For example, removing story difficulty from the feature set might reveal more about children's behavior once a story is chosen.

## 6. CONCLUSIONS

We have described a recent technique, $\ell_1$-regularized logistic regression, that makes it feasible to classify dialogues in a high-dimensional space, even when the number of features exceeds the number of training examples. We used our approach to solve two problems in classifying children's oral reading dialogue with Project LISTEN's

Table III. Most Predictive Features for Story Choice Initiative

| Feature Name | $\theta$ |
|---|---|
| Story level = $2^{nd}$grade | 0.1446 |
| Story level = $2^{nd}$grade$^2$ | 0.1446 |
| Story level = $1^{st}$grade | 0.0808 |
| Story level = $1^{st}$grade$^2$ | 0.0808 |
| Number of times this story has been completed$^2$ | 0.0367 |
| Number of times this story has been completed | 0.0330 |

Reading Tutor: predicting which stories they would finish and characterizing student behavior according to who chose stories, respectively achieving 71.1% and 73.6% cross-validated classification accuracy on balanced sets of data from unseen students.

The significance of this work lies in the freedom to consider a large set of candidate features without worrying that they will overfit the training data and therefore fail to generalize to unseen data. Modeling dialogue in high-dimensional space opens the door from small, manually generated sets of features to richer, automatically generated sets of features. In this paper, we formulated raw features manually and expanded them automatically into a much larger set by windowing, squaring, and thresholding them, relying on $\ell_1$-regularized logistic regression to select a few good features. Future work should explore large, systematically generated sets of features.

## Acknowledgments

This work was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080628 to Carnegie Mellon University. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or U.S. Department of Education. We thank the educators, students, and LISTENers who helped generate, collect, and analyze our data, and the reviewers for their helpful comments. The first author was partially supported by the Costa Rican Ministry of Science and Technology (MICIT).

## REFERENCES

Ai, H., Tetreault, J. R., and Litman, D. J. 2007. Comparing user simulation models for dialog strategy learning. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. NAACL-Short '07. Association for Computational Linguistics, Stroudsburg, PA, USA, 1–4.

Aist, G. and Mostow, J. 2000. Improving story choice in a reading tutor that listens. In *Proceedings of the Fifth International Conference on Intelligent Tutoring Systems*. Springer-Verlag, Montreal, 645–646.

Aist, G. and Mostow, J. 2007. Balancing Learner and Tutor Control by Taking Turns: Faster and Better Mixed-Initiative task choice in a reading tutor that listens. In *The path of speech technologies in computer assisted language learning: from research toward practice*, M. Holland, V. Holland, and F. Fisher, Eds. Routledge, New York, 220–240.

Amaldi, E. and Kann, V. 1998. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science 209,* 1-2, 237–260.

Beck, J. and Woolf, B. P. 2000. High-level student modeling with machine learning. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*. Springer-Verlag, London, UK, 584–593.

Beck, J. E. 2007. Does learner control affect learning? In *Proceeding of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 135–142.

Beck, J. E., Woolf, B. P., and Beal, C. R. 2000. Advisor: A machine learning architecture for intelligent tutor construction. In *Proceedings of the Seventeen National Conference on Artificial Intelligence*. Austin, Texas, 552–557.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist. 22*, 39–71.

Boyer, K., Phillips, R., Ingram, A., Ha, E., Wallis, M., Vouk, M., and Lester, J. 2010. Characterizing the effectiveness of tutorial dialogue with hidden markov models. In *Intelligent Tutoring Systems*, V. Aleven, J. Kay, and J. Mostow, Eds. Lecture Notes in Computer Science, vol. 6094. Springer Berlin / Heidelberg, 55–64.

Boyer, K. E., Phillips, R., Ha, E. Y., Wallis, M. D., Vouk, M. A., and Lester, J. C. 2010. Leveraging hidden dialogue state to select tutorial moves. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*. IUNLPBEA '10. Association for Computational Linguistics, Stroudsburg, PA, USA, 66–73.

Chi, M., Jordan, P., VanLehn, K., and Hall, M. 2008. Reinforcement learning-based feature selection for developing pedagogically effective tutorial dialogue tactics. In *Proceedings of the 1st International Conference on Educational Data Mining*. Montreal, 30–36.

Chi, M., VanLehn, K., and Litman, D. 2010. Do Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning to Induce Pedagogical Tutorial Tactics. In *Intelligent Tutoring Systems*, V. Aleven, J. Kay, and J. Mostow, Eds. Lecture Notes in Computer Science, vol. 6094. Springer Berlin / Heidelberg, Berlin, 224–234.

Dudik, M., Phillips, S. J., and Schapire, R. E. 2004. Performance guarantees for regularized maximum entropy density estimation. In *Proceedings of the 17th Annual Conference on Computational Learning Theory*. Springer Verlag, Banff, Canada, 472–486.

González-Brenes, J. P., Black, A. W., and Eskenazi, M. 2009. Describing Spoken Dialogue Systems Differences. In *International Workshop on Spoken Dialogue Systems*. Springer–Verlat, Irsee, Germany.

Guyon, I. and Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research 3*, 1157–1182.

Hajdinjak, M. and Mihelic, F. 2006. The PARADISE evaluation framework: Issues and findings. *Computational Linguistics 32,* 2, 263–272.

Hastie, H. W., Prasad, R., and Walker, M. 2002. What's the trouble: automatically identifying problematic dialogues in darpa communicator dialogue systems. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Association for Computational Linguistics, Morristown, NJ, USA, 384–391.

Hastie, T., Tibshirani, R., and Friedman, J. 2003. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. ed. Springer, New York.

Henderson, J., Lemon, O., and Georgila, K. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics 34*, 487–511.

Huang, X., Alleva, F., Hon, H., Hwang, M., Lee, K., and Rosenfeld, R. 1993. The SPHINX-II speech recognition system: an overview. *Computer Speech & Language 7,* 2, 137–148.

Keerthi, S. S., Duan, K. B., Shevade, S. K., and Poo, A. N. 2005. A fast dual algorithm for kernel logistic regression. *Maching Learning 61*, 151–165.

Lee, S.-I., Ganapathi, V., and Koller, D. 2007. Efficient structure learning of markov networks using $l_1$-regularization. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, Cambridge, MA, 817–824.

Möller, S., Engelbrecht, K., and Schleicher, R. 2008. Predicting the quality and usability of spoken dialogue services. *Speech Communication 50,* 8-9, 730–744.

Möller, S., Smeele, P., Boland, H., and Krebber, J. 2007. Evaluating spoken dialogue systems according to de-facto standards: A case study. *Computer Speech and Language 21,* 1, 26 – 53.

Mostow, J. and Aist, G. 2001. Evaluating tutors that listen: an overview of project LISTEN. In *Smart machines in education*, K. Forbus and P. Feltovich, Eds. MIT Press, Boston, 169–234.

Mostow, J., Aist, G., Burkhead, P., Corbett, A., Cuneo, A., Eitelman, S., Huang, C., Junker, B., Sklar, M. B., and Tobin, B. 2003. Evaluation of an automated reading tutor that listens: Comparison to human tutoring and classroom instruction. *Journal of Educational Computing Research 29,* 1, 61–117.

Mostow, J. and Duong, M. 2009. Automated Assessment of Oral Reading Prosody. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED2009)*. IOS Press, Brighton, UK, 189–196.

Ng, A. Y. 2004. Feature selection, $\ell_1$ vs. $\ell_2$ regularization, and rotational invariance. In *ICML '04: Proceedings of the Twenty-first International Conference on Machine learning*. ACM, New York, NY, USA, 78–86.

Nigam, K., Lafferty, J., and McCallum, A. 1999. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, T. Dean, Ed. Morgan Kaufmann, Stockholm, 61–67.

Okanohara, D. and Tsujii, J. 2009. Learning combination features with l1 regularization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. NAACL-Short '09. Association for Computational Linguistics, Stroudsburg, PA, USA, 97–100.

Park, M. Y. and Hastie, T. September 2007. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69*, 659–677(19).

Rieser, V. and Lemon, O. 2006. Using logistic regression to initialise reinforcement-learning-based dialogue systems. In *IEEE/ACL 2006 Workshop on Spoken Language Technology Workshop*. IEEE, Aruba, 190 –193.

Rieser, V. and Lemon, O. 2010. Learning human multimodal dialogue strategies. *Natural Language Engineering 16,* 01, 3–23.

Rosenfeld, R. 2000. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE 88,* 8, 1270–1278.

Schmidt, M., Murphy, K., Fung, G., and Rosales, R. 2008. Structure learning in random fields for heart motion abnormality detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Anchorage, Alaska, 1 –8.

Shevade, S. and Keerthi, S. 2003. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics 19,* 17, 2246.

Singh, S. P., Kearns, M. J., Litman, D. J., and Walker, M. A. 1999. Reinforcement learning for spoken dialogue systems. In *Advances in Neural Information Processing Systems*. Vol. 12. MIT Press, Cambridge, MA, 956–962.

Steidl, S., Hacker, C., Ruff, C., Batliner, A., Nöth, E., and Haas, J. 2004. Looking at the Last Two Turns, Id Say This Dialogue Is Doomed–Measuring Dialogue Success. In *Proceedings of the 7th International Conference on Text, Speech and Dialogue*. Springer, Brno, Czech Republic, 629–636.

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

Tomioka, R. and Sugiyama, M. 2009. Dual Augmented Lagrangian Method for Efficient Sparse Reconstruction. *IEEE Signal Processing Letters 16,* 12, 1067.

Walker, M., Kamm, C., and Litman, D. 2001. Towards developing general models of usability with PARADISE. *Natural Language Engineering 6,* 3, 363–377.

Walker, M., Langkilde, I., Wright, J., Gorin, A., and Litman, D. 2000. Learning to predict problematic situations in a spoken dialogue system: experiments with how may i help you? In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics Conference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 210–217.

Xing, E. P., Jordan, M. I., and Karp, R. M. 2001. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 601–608.

YANG, Y. AND PEDERSEN, J. O. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*. ICML '97. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 412–420.

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.

ZHANG, C. AND HUANG, J. 2008. The sparsity and bias of the lasso selection in high-dimensional linear regression. *Annals of Statistics 36,* 4, 1567–1594.