# End-User Feature Labeling via Locally Weighted Logistic Regression†

**Weng-Keen Wong[1], Ian Oberst[1], Shubhomoy Das[1], Travis Moore[1], Simone Stumpf[2], Kevin McIntosh[1], Margaret Burnett[1]**

[1]Oregon State University,
Corvallis, OR 97331, USA
{wong,obersti,dassh,moortrav,mcintoke,burnett}@eecs.oregonstate.edu

[2]City University London,
London, UK
Simone.Stumpf.1@city.ac.uk

**Abstract**

Applications that adapt to a particular end user often make inaccurate predictions during the early stages when training data is limited. Although an end user can improve the learning algorithm by labeling more training data, this process is time consuming and too ad hoc to target a particular area of inaccuracy. To solve this problem, we propose a new learning algorithm based on Locally Weighted Logistic Regression for *feature labeling* by end users, enabling them to point out which features are important for a class, rather than provide new training instances. In our user study, the first allowing ordinary end users to freely choose features to label directly from text documents, our algorithm was more effective than others at leveraging end users' feature labels to improve the learning algorithm. Our results strongly suggest that allowing users to freely choose features to label is a promising method for allowing end users to improve learning algorithms effectively.

## Introduction

Applications such as email classifiers, recommender systems, and intelligent desktop assistants customize themselves to a particular end user's preferences. This customization cannot happen until *after* the system is deployed and training data from that specific end user is obtained. However, when the application is first deployed, there is often limited training data, resulting in poor predictions by the learning algorithm. To address this problem, the end user could label additional training instances, or an active learning algorithm (Settles 2009) could select informative instances to be labeled by the end user. Labeling instances, however, is a tedious process and a substantial number of instances must often be labeled before a change to the learning algorithm is noticeable to an end user. Secondly, if a rare group of instances is incorrectly classified, the learning algorithm cannot be "corrected" unless the user is fortuitously asked to label an instance with this particular combination of attributes.

To overcome these problems, in this paper we investigate the possibility of end-user *feature labeling* (Sindhwani et al. 2009), namely allowing end users to label features instead of instances. For example, rather than labeling entire documents, an end user could point out which words (features) in the document are most indicative of certain class labels. Raghavan et al. (Raghavan et al. 2006, Raghavan and Allan 2007) found that labeling a feature took roughly a fifth of the time to label than a document and the benefits of feature labeling were greatest when the training set sizes were small. However, their work did not statistically evaluate feature labeling when performed by actual end users.

Allowing end users, who are not likely to be educated in machine learning, to use feature labeling introduces new challenges to learning algorithms. End users' choices of features may be noisy, inconsistent, and might vary greatly in ability to improve the predictive power of the machine learning algorithm. This paper therefore investigates algorithms which are able to stand up to these challenges. We present a new feature labeling algorithm based on Locally Weighted Logistic Regression. We then evaluate our algorithm, first under ideal conditions using feature labels obtained from an oracle, and second under more realistic conditions using feature labels from actual end users. Our results strongly suggest that feature labeling by end users is both viable and an effective solution for allowing end users to improve a learning algorithm.

## Related Work

Algorithms for feature labeling can roughly be divided into supervised and semi-supervised techniques. Raghavan and Allan (Raghavan and Allan 2007) present two



† This paper is a shortened version of (Wong et al. 2011).

supervised feature labeling methods based on Support Vector Machines (SVMs). Method 1 scales features indicated as relevant by the user by a constant *a* and the rest of the features by *d* (where $a \geq d$). Method 2 introduces pseudo-documents which influence the position of the separating hyperplane. In contrast to these two supervised feature labeling techniques, semi-supervised feature labeling methods (Druck et al. 2008, Method 3 of Raghavan and Allan 2007, Stumpf et al. 2009) also leverage information from a large pool of unlabeled data. In our work, we want to measure the gains from feature labeling using only the labeled instances in the training set. As such, we will only compare our work against supervised feature labeling algorithms. We do, however, intend to extend our algorithm to the semi-supervised setting in the future. Other work (Attenberg et al. 2010, Sindhwani et al. 2010) has also investigated *dual supervision,* which is used to describe the process of labeling both instances and features.

Almost all of the prior work in feature labeling evaluates algorithms under ideal conditions, such as feature labels obtained from an oracle (Raghavan et al. 2006, Attenberg et al. 2010, Sindhwani et al. 2010). Our study investigates both the use of ideal oracle feature labels and feature labels provided by real end users.

## Methodology

Locally Weighted Logistic Regression (LWLR) (Cleveland and Devlin 1988, Deng 1998) is a variant of Logistic Regression in which the logistic function is fit *locally* to a neighborhood around a query point to be classified. Intuitively, LWLR gives more weight to training points that are "closer" to the query point than those farther away. Our approach, called LWLR-FL, modifies the distance function used by LWLR by incorporating information from feature labels. LWLR-FL assigns higher weights to training instances that are similar to the query point according to the feature label information.

We use cosine similarity ie. $cosim(\boldsymbol{x_q},\boldsymbol{x_i}) = 1 - cos(\boldsymbol{x_q},\boldsymbol{x_i})$ as the baseline distance function between a query point $\boldsymbol{x_q}$ and a labeled data instance $\boldsymbol{x_i}$. In order to incorporate feature labels, we modify $cosim(\boldsymbol{x_q},\boldsymbol{x_i})$ by a multiplicative factor, which is determined by the class label of $\boldsymbol{x_i}$ and the end user's feature labels. The overall distance function is shown below:

$$f(\boldsymbol{x_q}, \boldsymbol{x_i}) = \text{cosim}(\boldsymbol{x_q}, \boldsymbol{x_i}) \cdot \left[ 1 - \boldsymbol{R}(y_i)^T \boldsymbol{x_q} + \left( \frac{(\boldsymbol{U} - \boldsymbol{R}(y_i))^T \boldsymbol{x_q}}{M-1} \right) \right]$$

$$w(\boldsymbol{x_q}, \boldsymbol{x_i}) = \exp\left(-\max(0, f(\boldsymbol{x_q}, \boldsymbol{x_i}))^2 / k^2\right)$$

We briefly provide an intuitive explanation of the formula above. For further details, we refer the reader to the extended version of this paper (Wong et al. 2011). In the equation above, the term $\boldsymbol{R}(y_i)^T \boldsymbol{x_q}$ corresponds to the sum of feature values of $\boldsymbol{x_q}$ for those features which are associated with label $y_i$. The higher this value, the more similar $\boldsymbol{x_q}$ will be to $\boldsymbol{x_i}$ according to the labeled features. The term $(\boldsymbol{U} - \boldsymbol{R}(y_i))^T \boldsymbol{x_q}$ corresponds to the sum of the feature values for $\boldsymbol{x_q}$ for the features that are not associated with label $y_i$. The higher this value, the more dissimilar $\boldsymbol{x_q}$ will be to $\boldsymbol{x_i}$ according to the labeled features. We divide this value by *(M-1)* to appropriately balance the difference since there are *(M-1)* class labels excluding $y_i$. Note that the distance function needs to be smaller for more similar instances. We also introduce a *max* term to prevent the distance from becoming negative in certain cases.

## Experiments

### Oracle Study

In our oracle-based experiments, we used three common text classification datasets: 20 Newsgroups (Lang 1995) (2925 articles in *comp.sys.ibm.pc.hardware*, *misc.forsale*, *sci.med*, and *sci.space*), the Modapte split of the Reuters dataset (Lewis 2004) (1092 documents in *earn*, *acq*, *negative_topic*, and *money-fx*), and the Reuters Corpus Volume 1 (RCV1) dataset (Lewis et al. 2004) (6500 documents in *C15*, *CCAT*, *ECAT*, *GCAT*, and *MCAT*). All documents were converted to a normalized TF-IDF representation with a vocabulary of unigrams and with stopwords removed.

We compared LWLR-FL against Method 1 (SVM-M1), Method 2 (SVM-M2) and their combination (SVM-M1M2) from (Raghavan and Allan 2007). For the SVM-based methods we only report results with linear kernels as these performed better than other kernels.

Because prior work (Raghavan and Allan 2007) showed that feature labeling is most effective with smaller training set sizes, we created training sets consisting of only six instances per class. The total training set sizes for our datasets were 24 for 20 Newsgroups, 24 for the Modapte split and 30 for RCV1. Having equal number of data instances from each class avoided biases due to class imbalance. A separate validation set, used for tuning algorithm parameters, was composed of 100 data points equally distributed among all the classes. The testing set consisted of the remaining data instances. For all datasets, the results were averaged over 30 random splits for training, validation and testing.

For simulated users in each dataset, the ten most predictive features were selected for each class ranked by information gain over the respective corpus, giving 40

feature labels for 20 Newsgroups and Modapte, and 50 for RCV1. We experimented with adding these features incrementally in order of information gain (one per class, two per class, so on until ten.) By the nature of their selection and the order of addition, potential gains of these *oracle* features are bound to be optimistic.

### User Study

We also performed a user study in which actual end users labeled features on the same 20 Newsgroups dataset mentioned in the previous section. We used a smaller validation set of size 24 instead of 100 to reflect a real-world situation with limited labeled instances. Instead of restricting end users to only select features from a pre-computed list as in (Raghavan et al. 2006), we allowed users to identify *any* feature they considered predictive by freely highlighting text directly in the documents. Consequently, participants could also create and label new features by combining words and punctuation.

Our user study had 43 participants: 24 males and 19 females, all of whom had no background in Computer Science, machine learning, or HCI. The application displayed 24 previously labeled documents in four newsgroups and then gave a participant 12 minutes to identify features that he/she believed would help the computer label future documents.

We used participant-provided feature labels to compare the performance of LWLR-FL, SVM-M1, SVM-M2 and SVM-M1M2. If a participant created a new feature, it was added to that participant's document representation. We analyzed two variants of each algorithm: one variant used participants' labels on existing features only, and the other used all features that participants provided.

## Results

### Oracle Feature Labels

We evaluate the algorithms in terms of the macro-average F1 score (abbreviated as macro-F1), averaged over 30 random training/validation/testing splits. Figure 1 shows that the average macro-F1 scores for the top two algorithms generally increased with addition of more oracle features for all algorithms.

The LWLR-FL algorithm's effectiveness with feature labeling is compared against its baseline LWLR that uses pure cosine similarity as the distance metric. Likewise, a "plain" SVM can be considered as a baseline algorithm for the SVM-based algorithms. The improvement in macro-F1 score, denoted by $\Delta_{baseline}$, over their respective baselines expresses the benefit of incorporating feature labels in all the algorithms. The average $\Delta_{baseline}$ was significant for LWLR-FL in all cases, and significant for the best SVM-based methods in most cases (Wilcoxon signed-rank test, $p < 0.05$). LWLR-FL produced larger average $\Delta_{baseline}$ scores than SVM-M1M2 on the 20 Newsgroups and Modapte datasets, and was tied with SVM-M2 on the RCV1 dataset. Interestingly, on the Modapte dataset, LWLR lagged behind SVM but once more than five oracle feature labels per class were provided, LWLR-FL was able to use those features more effectively than any of the SVM-based methods and outperformed all of them.

LWLR-FL produced or matched the highest mean macro-F1 score on all three datasets; its effectiveness was significantly better than SVM-M1M2 on the 20 Newsgroups dataset at 10 oracle feature labels per class (Wilcoxon signed-rank test, $p < 0.05$).

### End User Feature Labels

We now look at the effects of end user feature labels. In our analysis we show only results for SVM-M1M2 in Figure 2 because it consistently outperformed the other SVM methods. Since the participants provided eight feature labels per class on average, we chose the results of eight oracle feature labels per class as our reference (leftmost group in Figure 2). The middle group presents results when only feature labels on existing features were considered (feature labels on features created by participants were ignored). Finally, the rightmost group of results illustrates the macro-F1 scores when all feature labels are considered, including the new features created by the participants.

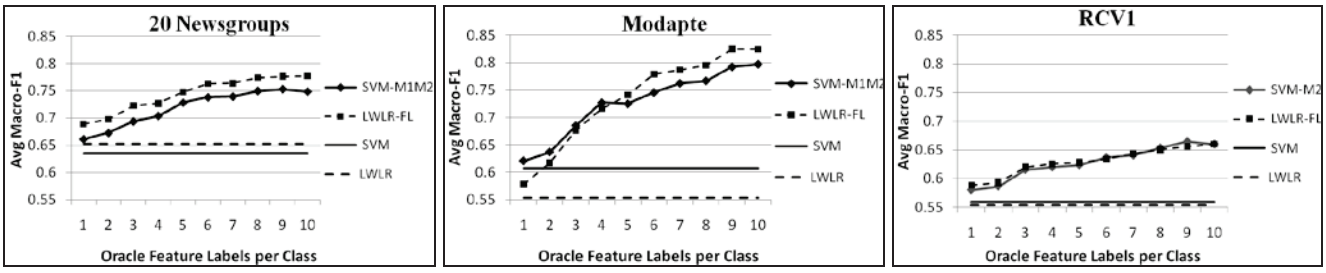


**Figure 1: Average Macro-F1 after adding oracle feature labels on (left) 20 Newsgroups, (middle) Modapte, and (right) RCV1. To avoid clutter on the graph, results for only the two best performing feature labeling algorithms are shown.**

Figure 2 shows that features provided by participants were indeed useful as both algorithms outperformed their baselines by a statistically significant amount for the "all" features case (right). However, only LWLR-FL was significantly better than its baseline with "existing" (middle) features (Wilcoxon signed-rank test, $p < 0.05$). LWLR-FL was significantly better than SVM-M1M2 (Wilcoxon signed-rank test, $p < 0.05$) for both existing and all features. New features created by participants resulted in a slight increase in average macro-F1, indicating that end users can in fact create predictive features.

## Conclusion

This paper has shown the viability of feature labeling in real circumstances, with end users freely choosing features to label directly from text documents. Our results show that LWLR-FL outperformed or matched SVM-based methods under ideal conditions in an oracle study. More significantly, we evaluated feature labeling algorithms under more realistic conditions with actual end users. Although the gains were not as large as those under oracle conditions, they were still significant improvements over the baseline algorithms without feature labels. In addition, the LWLR-FL algorithm outperformed the SVM-based methods in our user study. These results are promising, as they showed that end users who knew nothing about machine learning could use flexible feature labeling to significantly improve machine learning algorithms trained on small data sets. Feature labeling can be especially useful for learning algorithms that customize themselves to the preferences of a specific individual.

Our results point to promising future research directions. We plan to develop a semi-supervised version of the LWLR-FL algorithm and to investigate end-user feature engineering, in which end users are able to interactively create features and update the learning algorithm's representation of the data. In addition, we will examine the design of suitable user interfaces to help end users choose and create features to label. Finally, we will apply feature labeling to domains other than text mining, such as image classification, which represent data instances with features that are not intuitively understood by end users.

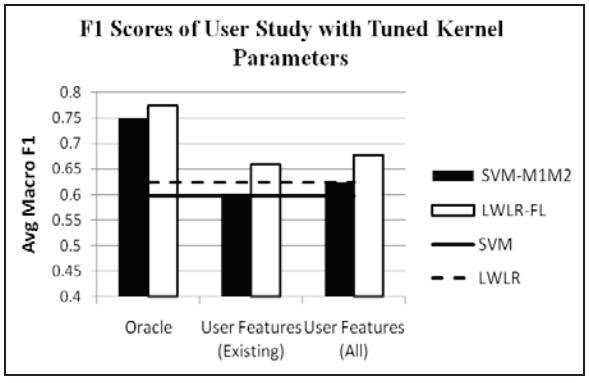


**Figure 2: Average macro-F1 scores for incorporating end user feature labels to the 20 Newsgroups dataset: (Left) incorporating 8 oracle feature labels per class, incorporating end-user feature labels only for existing features (Center) and for all end-user feature labels (Right).**

## Acknowledgements

This work was supported in part by NSF grant 0803487.

## References

Attenberg, J., Melville, P., and Provost, F. 2010. A unified approach to active dual supervision for labeling features and examples. In *Proc. ECML*, 40-55. Berlin, Heidelberg: Springer-Verlag.

Cleveland, W., and Devlin, S. 1988. Locally-weighted regression: An approach to regression analysis by local fitting. *J. American Statistical Assn* 83(403): 596–610.

Deng, K. 1998. Omega: On-line Memory-Based General Purpose System Classifier. PhD Dissertation. Carnegie Mellon University, Pittsburgh, PA.

Druck, G., Mann, G., and McCallum, A. 2008. Learning from labeled features using generalized expectation criteria. In *Proc. SIGIR*, 595-602. New York, NY: ACM Press.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *Proc. ICML*, 331-339. San Mateo, CA: Morgan Kaufmann.

Lewis, D. 2004. Reuters-21578. Available at http://www.daviddlewis.com/resoursce/testcollections/reuters21578.

Lewis, D. D., Yang, Y., Rose, T., Li, F. 2004. RCV1: A new benchmark collection for text categorization research. *JMLR* 5: 361-397. http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf.

Raghavan, H., Madani, O., and Jones, R. 2006. Active Learning with Feedback on Both Features and Instances. *JMLR* 7: 1655-1686.

Raghavan, H. and Allan, J. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines, In *Proc. SIGIR*, 79-86. New York, NY: ACM.

Settles, B. 2009. Active learning literature survey. Technical Report 1648, Department of Computer Science, University of Wisconsin–Madison, Madison, WI.

Sindhwani, V., Melville, P., and Lawrence, R. 2009. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proceedings of ICML*, 953-960. New York, NY: ACM.

Stumpf, S. Rajaram, V., Li, L., Wong, W.-K., Burnett, M., Dietterich, T., Sullivan, E., and Herlocker J. 2009. Interacting meaningfully with machine learning systems: Three experiments. *Int. J. Human-Computer Studies 67(8):* 639-662.

Wong, W-K., Oberst, I., Das, S., Moore, T., Stumpf, S., McIntosh, K., and Burnett, M. 2011. End-user feature labeling: A locally-weighted regression approach. In Proc. *Int. Conf. Intelligent User Interfaces,* 115-124. New York, NY: ACM.