FEATURE SELECTION FOR LOG-LINEAR ACOUSTIC MODELS

S. Wiesler, A. Richard, Y. Kubo, R. Schlüter, H. Ney

Human Language Technology and Pattern Recognition
Computer Science Department - RWTH Aachen University
52056 Aachen, Germany
{wiesler, richard, kubo, schlueter, ney}@informatik.rwth-aachen.de

ABSTRACT

Log-linear acoustic models have been shown to be competitive with Gaussian mixture models in speech recognition. Their high training time can be reduced by feature selection. We compare a simple univariate feature selection algorithm with ReliefF - an efficient multivariate algorithm. An alternative to feature selection is ℓ_1-regularized training, which leads to sparse models. We observe that this gives no speedup when sparse features are used, hence feature selection methods are preferable. For dense features, ℓ_1-regularization can reduce training and recognition time. We generalize the well known Rprop algorithm for the optimization of ℓ_1-regularized functions. Experiments on the Wall Street Journal corpus showed that a large number of sparse features could be discarded without loss of performance. A strong regularization led to slight performance degradations, but can be useful on large tasks, where training the full model is not tractable.

Index Terms— feature selection, ℓ_1-regularization, ReliefF, acoustic modeling, log-linear models

1. INTRODUCTION

Almost all statistical speech recognition systems are based on Hidden Markov Models (HMMs) with Gaussian Mixture Models (GMMs) as emission probabilities. They can be trained efficiently according to the maximum likelihood (ML) criterion. In state-of-the-art systems, ML acoustic models are only used as an initialization for discriminative training, e.g. maximum mutual information (MMI) training. In contrast, direct models are inherently discriminative and do not depend on a ML initialization [1]. Recently, log-linear models have gained interest in speech recognition and were successfully applied to phoneme recognition, e.g. [2, 3], and large vocabulary continuous speech recognition (LVCSR) [1, 4]. A major advantage of log-linear models is that their training with respect to the MMI criterion is convex. Log-linear models correspond to linear classifiers which are normalized to provide a probabilistic interpretation. Non-linearity is achieved by an explicit mapping of features into a high dimensional space. In our previous work we employed polynomial features and sparse clustering features, similar to those introduced in [2]. Despite of being theoretically attractive and performing comparably to state-of-the-art systems, a disadvantage of log-linear models is their high training time in comparison to GMMs, which depends on the number of features. Therefore, feature selection is attractive for reducing training time.

Feature selection is an extensive field of research in machine learning [5]. On many problems where overfitting is a severe problem, feature selection even improves the classification accuracy. In speech recognition, feature selection for acoustic features has not gained much attention, because in GMMs only low dimensional feature vectors are employed. Simple feature selection algorithms are univariate, which means features are scored individually. When interaction of features is important, multivariate feature selection algorithms can provide better results. In this work we compare the univariate χ^2-algorithm with the multivariate ReliefF algorithm. Indeed, we show that the multivariate algorithm performs better than the univariate one for typical features used in speech recognition.

Embedded methods jointly learn the structure of the model and the model parameters. Such approaches are conceptually attractive, because they avoid heuristics. A popular embedded method for linear models is ℓ_1-regularized training, which leads to sparse models. For binary classification problems, this corresponds to a feature selection. For multiclass classification, strictly speaking it is not a feature selection, but is often discussed in the context of feature selection. Sparse models are beneficial because of their smaller size. However, as discussed in the next section, when sparse features are used, ℓ_1-regularization does not reduce training or recognition time, therefore feature selection is preferable in this case. For dense features this does not hold. We modify the well known Rprop algorithm [6] in order to directly incorporate the non-differentiable ℓ_1-regularization term in the training procedure of our previous work and apply this method to polynomial features.

The remaining paper is structured as follows. In the next section, we analyze the complexity of log-linear model training and recognition and how it is affected by feature selection and sparse models. In Section 3, we present the feature selection algorithms used in this work and describe our modification of Rprop. Experimental results are given in Section 4. We conclude with a discussion and an outlook.

2. COMPLEXITY OF LOG-LINEAR MODEL TRAINING AND RECOGNITION

Let $X \subset \mathbb{R}^D$ be the feature space and $\mathcal{S}$ a finite set of classes. A *log-linear model* with parameters $\Lambda = (\lambda_{s,i})_{s,i} \in \mathbb{R}^{|\mathcal{S}| \times I}$

is a model for a posterior probability of the form

$$p_\Lambda(s|x) = \exp\Big(\sum_{i=1}^{I} \lambda_{s,i} f_i(x)\Big) \Big/ \sum_{\tilde{s}} \exp\Big(\sum_{i=1}^{I} \lambda_{\tilde{s},i} f_i(x)\Big),$$

where the components of $f : X \to \mathbb{R}^I$ are called *feature functions*. Posterior probability models can be incorporated into HMM recognizers via the hybrid approach, see [7]. Usually, log-linear models are trained according to the MMI-criterion. For a given sequence of training samples $(s_t, x_t)_{t=1,..,T}$ the objective function of *framewise MMI* is defined as

$$F : \mathbb{R}^{|\mathcal{S}|\times I} \to \mathbb{R},\ \Lambda \mapsto \sum_{t=1}^{T} \ln p_\Lambda(s_t|x_t) - Cr(\Lambda).$$

Here, $r(\Lambda)$ is a regularization term, e.g. the ℓ_1- or ℓ_2-norm. For both choices, the MMI criterion is a convex optimization problem. For training, gradient based optimization algorithms can be employed. The time consuming part of training is the calculation of the gradient. For MMI training the gradient of the unregularized objective function F at Λ with respect to a parameter $\lambda_{s,i}$ is

$$\frac{\partial F(\Lambda)}{\partial \lambda_{s,i}} = \sum_{t=1}^{T} \big(\delta(s, s_t) - p_\Lambda(s|x_t)\big) f_i(x_t).$$

Hence, for every feature vector the posterior probabilities according to the current model have to be computed and the statistics need to be updated. For the posteriors, the inner product of all parameter vectors and the feature has to be calculated. For dense features, the complexity of this operation is in the order of $O(2T|\mathcal{S}|I)$ floating point operations. The update of the statistics has again a complexity of $O(2T|\mathcal{S}|I)$. Training costs reduce drastically for sparse features. For the inner product of a dense parameter and a sparse feature vector only the non-zero components have to be considered. Thus, the costs of calculating the inner products are in the order of $O(2T|\mathcal{S}|A_f)$, where A_f denotes the average number of active features. Since only the statistics for non-zero features have to be updated, these costs are reduced by the same factor.

Determining the complexity is more subtle when sparse parameter vectors are used as well. For sparse parameter vectors and dense features, the inner products are again cheaper. When in addition the features are sparse, in principle only those components have to be considered, where both parameters and features are different from zero. Unfortunately, this structure can not be exploited efficiently, since then comparisons have to be performed instead of floating point operations. Therefore, the efficiency of a sparse-sparse inner product depends on the computer architecture, but on a modern CPU no speedup can be expected [1].

When using ℓ_1-regularization the structure of the parameters is learned during model training. Therefore the costs for the statistics update do not change when sparse parameters are used. If the structure is known in advance, the costs for the statistics update are proportional to the number of active parameters.

For decoding in the hybrid approach, instead of the acoustic distance calculations for GMMs, the feature vector has to be constructed and the inner products of features and parameters have to be computed. Decoding with hybrid log-linear models with clustering features as in [2, 1] is faster than decoding with GMMs, because the same Gaussians are used for all states. Selecting sparse features and imposing a sparse structure on the parameters corresponding to polynomial features further reduces the complexity as described above.

The same considerations hold for different discriminative training criteria as minimum phone error (MPE) [9] and when generalized iterative scaling (GIS) is used instead of gradient based optimization algorithms.

In summary, feature selection is advantageous in comparison to ℓ_1-regularization for sparse features. For dense features, ℓ_1-regularization can speed up training and may therefore be preferable to feature selection, because of its conceptual attractiveness.

3. FEATURE SELECTION METHODS

As described in the previous section, the costs of using a dense feature are much higher than those for a sparse feature. Therefore sparse and dense features should be treated separately. According to the considerations above, we performed a feature selection for sparse features and regularized the parameters of the model corresponding to dense features with ℓ_1-norm.

3.1. Sparse Features

Feature selection methods can be divided into methods which use the performance of the classifier on a cross-validation set for ranking features or feature subsets (wrapper methods) and methods that are independent of the classifier, (filter methods). Since our goal is to speed up the training of the classifier, we are interested in filter methods.

Simple methods are univariate, which means they rank every feature individually. The χ^2-algorithm is one of the best performing univariate filter methods, especially on very high dimensional problems as text classification [10]. The idea of the algorithm is to apply the well known χ^2-independence test to each feature and the class variable (see [10] for a more detailed description). Higher scores correspond to a higher dependence of the variable to the class label, hence to better features. In order to apply the χ^2-test to continuously valued features, the feature range has to be discretized. Experimental results for the χ^2-algorithm are presented in Section 4.

ReliefF [11] is a state-of-the-art feature selection algorithm and can be seen as a compromise between univariate and multivariate methods. Every feature is scored individually, but the score takes the interaction of features into account. The idea of the algorithm is to measure how well a feature separates neighboring training samples in the original space. It can be applied to categorical as well as numerical features. A small subset of m training samples is chosen randomly. For each sample the k nearest instances of the same class (nearest hits)

[1] Note that also in standard sparse vector routine libraries as Sparse BLAS [8] sparse-sparse operations are not supported. In addition, sparse features are typically orders of magnitude sparser than sparse parameters, see [2, 1]

and all other classes (nearest misses) are computed. When a feature of a sample and one of its nearest hits is different, the score of the feature is reduced. Conversely, the feature score is increased, if the feature of the training sample and one of its nearest misses is different. The nearest hits and misses are calculated in the original feature space with the ℓ_2-norm. The expensive part of the ReliefF algorithm is the calculation of the nearest hits and misses, which are chosen from the complete set of training samples. This part can easily be parallelized. Typically, a small m in the range from 50 to 1000 is chosen and regarded as a constant, hence the complexity of the algorithm is linear in the amount of training data. The costs of feature selection with ReliefF are negligible in comparison to the log-linear training. Experimental results and a comparison to the χ^2-algorithm are given in Section 4.

3.2. Polynomial features

For the optimization of ℓ_1-regularized objective functions, special care has to be taken, because the regularization term is non-differentiable. In our previous work we used Rprop for training. Advantages of Rprop are its robustness to tuning parameters and its simplicity. Furthermore, Rprop leads very quickly close to the optimum of the objective function. Our proposed modification to Rprop is analog to that of L-BFGS in [12]. We refer to it as Orthantwise-Rprop in the following (OW-Rprop). First, it is observed that the components of the objective function with ℓ_1-regularization are left and right differentiable, which is used for the definition of the pseudo gradient:

$$\Diamond_{s,i} F(\Lambda) = \begin{cases} \partial^+_{s,i} F(\lambda) & , \text{ if } \partial^+_{s,i} F(\lambda) > 0 \\ \partial^-_{s,i} F(\lambda) & , \text{ if } \partial^-_{s,i} F(\lambda) < 0 \\ 0 & , \text{ otherwise} \end{cases} .$$

A parameter vector Λ maximizes the objective function if and only if the pseudo gradient of F at Λ is zero. The idea of the algorithm is that the objective function is differentiable, when it is restricted to the orthant (a multidimensional quadrant) containing the current iterate and into which the pseudo gradient leads. The concept of orthantwise optimization can directly be applied to Rprop. For OW-Rprop, the pseudo gradient is used instead of the gradient. Furthermore, whenever for Rprop the sign of a parameter changes from one iteration to the next, OW-Rprop sets it to zero. Experimental results for the application of OW-Rprop to the optimization of log-linear models with polynomial features are reported in the next section.

4. EXPERIMENTAL RESULTS

All experiments were performed on the Wall Street Journal corpus with a vocabulary of 5k words (WSJ0), a small LVCSR task. The training corpus consists of 15 hours and the evaluation corpus of 0.4 hours of English read speech. Since the official WSJ0 corpus does not provide a development set, additional 0.5 hours were extracted from the North American Business task. The vocabulary of the task is closed.

The acoustic front end of all experiments uses 16 Mel-frequency cepstral coefficients (MFCC). The MFCC features are normalized by a vocal tract length normalization and augmented with a voicedness feature. Feature vectors from nine consecutive frames are concatenated and a linear discriminative analysis is used to reduce the dimension to 33.

For our log-linear system, a single GMM (independent of the state) with 1024 densities has been trained. The posteriors of each density were used as sparse features. Acoustic context expansion led to 9216 sparse features in total. In addition, second order polynomial features were used. The system uses 1500 generalized triphone states. All log-linear systems were trained with Rprop respectively OW-Rprop until convergence, which takes about 75 iterations. For comparison, a GMM recognizer with the same number of states and 223k Gaussians in total has been trained. All Gaussians share a single diagonal covariance matrix. The recognitions were performed with the baseline bigram language model delivered with the corpus and a trigram language model trained at our group. The GMM system has been trained with the ML criterion and has a word error rate (WER) of 3.6% with the trigram and 5.6% with the bigram language model. To our knowledge, these are the best ML results published on this corpus. The log-linear system achieves a WER of 3.6% with the trigram and 6.6% with the bigram language model.

An alternative to the application of feature selection on the 9216 sparse features is to decrease the acoustic context length. This simple method is considered as the baseline in the following. For the discretization of the feature range needed by χ^2, we simply distinguished between active and non-active features. For the ReliefF algorithm $m = 650$ samples were chosen. The number of nearest hits and misses was set to $k = 50$. We observed that the algorithm seems to be very robust to the choice of the tuning parameters. Initial feature selection experiments were performed on a monophone system with the same features as the final triphone system. Since our interest is the quality of the acoustic model, the recognitions were performed with the bigram language model. The results are shown in Table 1. The performance of the χ^2 feature selection is not satisfactory. The error rate is higher than the baseline in all cases. In contrast, ReliefF allows for discarding almost half of the features by only a slight increase in WER.

sparse feature dim.	**9120**	**7168**	**5120**	**3072**	**0**
context reduction	10.1	10.2	11.3	11.8	14.0
χ^2	10.1	10.4	11.8	12.5	14.0
ReliefF	10.1	10.3	10.4	11.8	14.0

Table 1. Word error rates for feature selection on monophone systems with second order polynomial features and sparse clustering features. Recognitions were performed with a bigram language model.

The same feature configuration was used for the experiments with ℓ_1-regularization. Only the second order features were regularized with ℓ_1-norm. In our experiments OW-Rprop converged as fast as Rprop. Recognition results of the monophone system are shown in Table 2. The results are in between those of the system with all second order features ($C = 0$) and a system which uses only first order and clustering features ($C = \inf$).

Final experiments were performed on the log-linear triphone

C	sparse coefficients (%)	WER (%)
0	0	10.1
100	16.6	10.2
500	67.5	11.2
1000	89.7	11.9
inf	100.0	12.5

Table 2. Results for ℓ_1-regularized monophone systems with bigram language model. In addition to the second order polynomial features, first order features and sparse clustering features have been used.

	sp. coeff.	sparse dim.	3gr.	2gr.
GMM-Baseline	-	-	3.6	5.6
LL-Baseline	0.0	9216	3.6	6.6
ReliefF	0.0	5120	3.6	6.6
ℓ_1	60.0	9216	3.9	7.0
ℓ_1+ReliefF	70.7	5120	4.1	7.0

Table 3. Results of the final triphone system with feature selection and sparse models. WERs are given for recognitions with a bigram and trigram language model.

system (see Table 3). Reducing the number of sparse features with ReliefF to 5120 did not degrade the error rate at all. The application of ℓ_1-regularization with $C = 100$ led to a model where 60.0% of the parameters corresponding to the second order features are zero by only a slight increase in error rate. Unfortunately, with the current training scheme for ℓ_1-regularized training, we could not obtain a speedup. This has two reasons. First, for ℓ_1-regularization the model structure is learned during training. Therefore the costs for the statistics update do not decrease. Furthermore, even though we initialized the parameters with zero, the parameter fill up immediately and then slowly get sparse during training. That means for most iterations the model is not sparse. However, for recognition the final sparse model is used.

5. DISCUSSION AND OUTLOOK

In this paper we investigated the application of feature selection techniques to log-linear acoustic models. The use of the multivariate feature selection algorithm ReliefF allowed us to reduce the sparse feature dimension by almost 50% without increase in WER. For polynomial features, training with ℓ_1-regularization has the potential to speed up training and recognition. In order to incorporate ℓ_1-regularization into our training scheme, we proposed to apply the concept of orthantwise optimization to Rprop. The sparse models performed only slightly worse than the full models.
Feature selection for sparse features is especially useful for huge feature dimensions. For example, feature selection can be employed for the integration of offset features as in MPE into the log-linear model. In the future, we want to investigate techniques for fixing the model structure corresponding to polynomial features, e.g. learning the model structure on a small subset of the training data or preventing parameters to get non-zero after some iterations. With reduced training time, we want to evaluate log-linear acoustic models on larger data sets.

Acknowledgment—This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

6. REFERENCES

[1] S. Wiesler et.al., "Investigations on features for log-linear acoustic models in continuous speech recognition," in *Proc. IEEE Automatic Speech Recognition and Understanding workshop*, Merano, Italy, 2009.

[2] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 354–365, 2009.

[3] A. Gunawardana et.al., "Hidden conditional random fields for phone classification," in *Proc. Interspeech*, 2005.

[4] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *Proc. IEEE Automatic Speech Recognition and Understanding workshop*, Merano, Italy, 2009.

[5] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[6] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm," in *Proc. IEEE International Conference on Neural Networks*, Nagoya, Japan, 1993.

[7] D. Kershaw et.al., "Context-dependent classes in a hybrid recurrent network-HMM speech recognition system," *Advances in Neural Information Processing Systems*, vol. 8, pp. 750–756, 1995.

[8] I.S. Duff, M.A. Heroux, and R. Pozo, "An overview of the sparse basic linear algebra subprograms: The new standard from the BLAS technical forum," *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 239–267, 2002.

[9] D. Povey and P.C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Orlando, USA, 2002.

[10] Y. Yang and J.O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. International Conference on Machine Learning*, Nashville, USA, 1997.

[11] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. European Conference on Machine Learning*, Catania, Italy, 1994.

[12] G. Andrew and J. Gao, "Scalable training of L1-regularized log-linear models," in *Proc. International Conference on Machine Learning*, Corvallis, USA, 2007.