DISCRIMINATIVE FEATURE EXTRACTION FOR LANGUAGE IDENTIFICATION

Shuai Huang[1], *Glen A. Coppersmith*[1,2]

[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, MD, USA
{shuaihuang, coppersmith}@jhu.edu

ABSTRACT

In this paper we propose a discriminative feature extraction method, DFE, to address the increasing number of features in language identification (LID) tasks. Similar to linear discriminant analysis (LDA), it extracts the most discriminative features through the maximization of an "approximated" mutual information $I(C;Y)$ between the class labels C and the projected data Y. Compared with other feature extraction methods, experiments done on the CallFriend corpus shows DFE could handle high-dimensional dataset with ease. Furthermore, this feature extraction shows improvements on the LID task over standard feature extraction methods (LDA and principal components analysis).

Index Terms— Feature extraction, mutual information, language identification

1. INTRODUCTION

Language identification (LID) systems try to automatically identify the language spoken in an utterance using features extracted from the acoustic signal [1]. Training a LID system requires the acquisition of features from the speech that are appropriate for discrimination between the languages, those systems that include more features tend to show superior performance. However, including irrelevant or noisy features can degrade performance (to say nothing of increased computational burden). Thus, feature extraction has become a necessary preprocessing step to classification in LID tasks.

Many feature extraction methods exist (both supervised and unsupervised), such as principal component analysis (PCA), linear discriminant analysis (LDA) etc. PCA is an unsupervised method that tries to project the data onto some orthogonal principal components so that the corresponding variances are maximized. Given a term-document matrix X, these principal components can be obtained by selecting the m eigenvectors of $cov(X)$ that correspond to the m largest eigenvalues. The eigenvectors with small eigenvalues correspond to dimensions with small variances, which we expect are either caused by noise or are irrelevant to the task, and thus remove them. However, projecting the data in this manner (with no respect for the downstream inference task, here: LID classification) will not necessarily give the most discriminative features needed for classification.

On the other hand, LDA is a supervised method that extracts the features so that the ratio of between-class variance and within-class variance is maximized, i.e. it tries to separate the class means as much as possible while keeping the within class variance small. The transformed dataset $Y = W^T X$, specifically, the transformation matrix W should maximize the following objective:

$$J(w) = \frac{|W^T S_B W|}{|W^T S_W W|} \tag{1}$$

where S_B is the between-class scatter matrix, and S_W is the within-class scatter matrix. W can be obtained by computing the eigenvectors of $S_W^{-1} S_B$ with largest eigenvalues.

LDA depends on the assumption that the data is distributed according to a unimodal Gaussian for the features extracted to be the most informative for discriminating different classes. When this assumption is violated, performance can be degraded. Although LDA is a simple yet powerful method, it can only extract no more than $|C| - 1$ features, where $|C|$ is the number of classes. Also, LDA wouldn't work when the scatter matrix S_W is singular due to the small sample size problem [2], i.e. when the number of samples $|X|$ is smaller than the dimensionality of the samples.

Given high-dimensional dataset X, we propose a supervised discriminative feature extraction method, DFE, as an approximation to maximize the mutual information between the class labels C and the projected data Y. DFE has no assumption about the dataset distribution and extracts the most discriminative features directly, which makes it a good choice to handle the high number of feature dimensions in LID tasks.

The paper proceeds as follows: a brief introduction of how $I(C;Y)$ is approximated is given in Section 2, followed by the objective function and the proposed method in Section 3. Experiments comparing different techniques are shown in Section 4, concluding remarks and relation to prior work appear in Section 5.

Thanks to NSF grant No CCF-0728931 for funding.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

2. ESTIMATION OF MUTUAL INFORMATION

When performing feature extraction, we would like to the projected data Y to retain as much information about the class labels C as possible, and the mutual information $I(C;Y)$ [3] provides a quantitative measure of mutual dependence between two random variables C and Y:

$$I(C;Y) = H(Y) - H(Y|C) \tag{2}$$

Computing the entropy $H(Y), H(Y|C)$ usually involves estimation of the probability density distribution of Y, which is infeasible for high-dimensional dataset due to the data sparsity problem cause by "curse of dimensionality". In [4], an estimation of the entropy based on nonparametric statistics is proposed as follows:

$$H(Y) = \frac{m}{N\left(N-1\right)} \sum_{i \neq j} \log \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 + \text{const.} \tag{3}$$

where $N = |Y|$ is the size of the dataset $Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N\}$, m[1] is the cardinality of $\boldsymbol{y}_i \in \mathbb{R}^m$. The above estimation makes it possible to compute the entropy from the dataset directly. Furthermore, suppose there are $K = |C|$ classes, and each class k has $N_k = |Y_k|$ data points, the conditional entropy is thus given by:

$$\begin{aligned} H(Y|C) &= \sum_{k=1}^{K} \frac{N_k}{N} H(Y|C=k) + \text{const.} \\ &= \sum_{k=1}^{K} \frac{m}{N\left(N_k - 1\right)} \sum_{\substack{c_i = c_j = k \\ i \neq j}} \log \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 + \text{const.} \end{aligned} \tag{4}$$

The estimated mutual information can thus be further simplified:

$$I(C;Y) = \sum_{i \neq j} \gamma_{ij} \log \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 + \text{const.} \tag{5}$$

$$\gamma_{ij} = \begin{cases} \frac{m}{N(N-1)} & \text{if } c_i \neq c_j \\ \frac{m}{N(N-1)} - \frac{m}{N(N_k-1)} & \text{if } c_i = c_j = k \end{cases} \tag{6}$$

In other words, the mutual information can be estimated as log sum of projected distance $\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$ between two data points weighted by γ_{ij}. Since $N \geq N_k$, γ_{ij} is positive if they belong to the same class, and negative otherwise.

3. OBJECTIVE FUNCTION

We would like to find the transformation matrix W that maximizes $I(C;Y) = I(C;W^T X)$. Take the one dimensional case for example, without loss of generality, we add the constraint $\boldsymbol{w}^T\boldsymbol{w} = 1$ and let $\boldsymbol{z}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j$. Using the Lagrange multipliers, the Lagrange objective function $J(\boldsymbol{w})$ becomes:

$$J(\boldsymbol{w}) = \sum_{i \neq j} \gamma_{ij} \log \boldsymbol{w}^T \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T \boldsymbol{w} - \lambda(\boldsymbol{w}^T\boldsymbol{w} - 1) + \text{const.} \tag{7}$$

Take the derivative of $J(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$, and set it to 0, we have:

$$\begin{aligned} \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} &= 2 \sum_{i \neq j} \gamma_{ij} \frac{1}{\boldsymbol{w}^T \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T \boldsymbol{w}} \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T \boldsymbol{w} - 2\lambda \boldsymbol{w} \\ &= 2 \sum_{i \neq j} \alpha_{ij} \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T \boldsymbol{w} - 2\lambda \boldsymbol{w} = 0 \end{aligned} \tag{8}$$

where $\alpha_{ij} = \gamma_{ij} \frac{1}{\boldsymbol{w}^T \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T \boldsymbol{w}}$. Furthermore, α_{ij} can be seen as the weight assigned to a pair of points $\boldsymbol{x}_i, \boldsymbol{x}_j$. However, as experiments have shown, if two points are very close in the projected space, $|\alpha_{ij}|$ could become extremely large and $\boldsymbol{z}_{ij}\boldsymbol{z}_{ij}^T$ would dominate $J(\boldsymbol{w})$, which eventually leads to suboptimal solutions. Since the pair of points are expected to be close if they are in the same class, one way to handle this situation is to assign an average weight $\alpha_{ij} = \gamma_{ij}\gamma_k$ to those pairs belonging to the same class and $\alpha_{ij} = \gamma_{ij}\gamma_l$ to the rest. The mutual information can be used as a criterion (score) to tune the parameters γ_k, γ_l on a development set. Specifically, α_{ij} and γ_{ij} are related as follows:

$$\alpha_{ij} = \begin{cases} \gamma_{ij}\gamma_l & \text{if } c_i \neq c_j \\ \gamma_{ij}\gamma_k & \text{if } c_i = c_j = k \end{cases} \tag{9}$$

Another advantage of using the average weight α is that the above optimization process can be simplified to the following generalized eigenvalue problem:

$$\textstyle\sum_{i \neq j} \alpha_{ij} \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T \boldsymbol{w} = \lambda \boldsymbol{w} \tag{10}$$

Since α_{ij} is pre-specified and $\boldsymbol{z}_{ij}$ are known, we have a closed form for $\sum_{i \neq j} \alpha_{ij} \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T$, and $\boldsymbol{w}$ is its eigenvector with eigenvalue λ. Naturally if we let $\alpha_{ij} = 0$ for $i = j$ and $\alpha_{ij} = \alpha_{ji}$, the eigenvector can be computed as follows:

$$\begin{aligned} \sum_{i \neq j} \alpha_{ij} \boldsymbol{z}_{ij} \boldsymbol{z}_{ij}^T &= \sum_{i \neq j} \alpha_{ij} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \\ &= \frac{1}{2} \sum_{ij} \alpha_{ij} \left\{ \boldsymbol{x}_i \boldsymbol{x}_i^T + \boldsymbol{x}_j \boldsymbol{x}_j^T - 2 \boldsymbol{x}_i \boldsymbol{x}_j^T \right\} \\ &= \sum_{ij} \boldsymbol{x}_i \alpha_{ij} \boldsymbol{x}_i^T - \sum_{ij} \boldsymbol{x}_i \alpha_{ij} \boldsymbol{x}_j^T \\ &= XDX^T - XAX^T \\ &= X(D - A)X^T \\ &= XLX^T \end{aligned} \tag{11}$$

where D is a diagonal matrix with entries $D_{ii} = \sum_j \alpha_{ij}$, A is a matrix with entries $A_{ij} = \alpha_{ij}$. $L = D - A$ is called

[1] m is the dimensionality of $\boldsymbol{y}$, and M is that of $\boldsymbol{x}$

the Laplacian matrix, and the projection matrix W is the m orthonormal eigenvectors of XLX^T corresponding to the largest eigenvalues.

Under the chosen weights α_{ij}, the approximated objective function $J'(W)$ becomes:

$$J'(W) = \text{tr}\left(W^T XLX^T W\right) = \textstyle\sum_{i=1}^{m} \lambda_i \qquad (12)$$

which is the sum of the m largest eigenvalues λ_i. We can see that $J'(W)$ increases as long as we add $\lambda_i > 0$, and we should keep the eigenvectors that have positive eigenvalues.

While tuning the parameters α, we can simply set $\gamma_l = 1$ and only tune γ_k, the proposed discriminative feature extraction method can be formulated as follows:

Algorithm 1 Discriminative Feature Extraction

```
1. Normalize the dataset and define a set $\boldsymbol{A}_k$ for $\gamma_k$
2. For $\gamma_k^{(i)} \in \boldsymbol{A}_k$
     · Compute the $m$ eigenvectors of $XLX^T$ on training set
         $W = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$
     · Compute the projected data $Y = W^T X$ on development set
     · Compute the mutual information:
         $I^{(i)}(C;Y) = \sum_{i,j} \alpha_{ij} \log \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$
   End
3. Find the set of parameters $\gamma_k$ s.t.
     $\gamma_k = \arg\max_{\gamma_k^{(i)}} I^{(i)}(C;Y)$
```

4. EXPERIMENTAL RESULTS

We then test the proposed feature extraction method on a standard LID task, the dataset used is the CallFriend corpus[5], which contains conversations in 12 languages: Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, Vietnamese. Each document is a 30-second segment of conversational telephone speech converted to a high-dimensional feature vector using articulation "attribute" features, courtesy of [6]. We will increase the number of features gradually and compare different feature extraction methods. For each language we have 800 documents randomly partitioned into training, development and test sets (400, 200 and 200 documents respectively).

4.1. Attribute-based Features

In [6], they proposed "manner" and "place" of articulation "attributes" as a universal acoustic characterization of all spoken languages. Manner contains 6 items: vowel, fricative, nasal, approximant, stop and silence; while place contains ten items: coronal, dental, glottal, high, labial, low, mid, palatal, silence and velar. Attribute transcriptions are obtained from the phoneme transcripts via a phoneme-to-attribute mapping table. The transcriptions are obtained for each document (X_i for $i \in (1, \cdots, N)$), and each document is then converted into a M-dimensional feature vector $\mathbf{x}_i$ by collecting and concatenating n-gram counts of manner and place attributes items respectively.

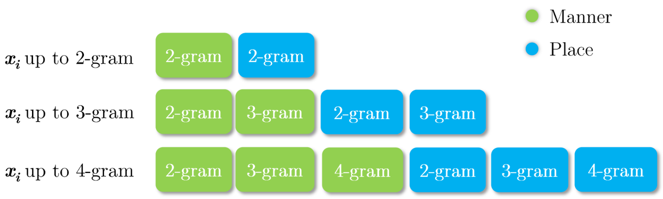

● Manner
● Place
$\boldsymbol{x}_i$ up to 2-gram
2-gram
2-gram
$\boldsymbol{x}_i$ up to 3-gram
2-gram
3-gram
2-gram
3-gram
$\boldsymbol{x}_i$ up to 4-gram
2-gram
3-gram
4-gram
2-gram
3-gram
4-gram

Fig. 1: Construction of feature vector by concatenating the manner and place n-gram features, the first feature vector (up to 2 gram) is of length 152, the rest two re of length 1368 and 12664 respectively

The n-gram attribute features are ideal for our tasks since we can collect an arbitrary number of features by increasing n. The following experiments use $n = 2$ to $n = 4$. Figure 1 shows the construction of n-gram features. In each condition, we compute the average error rate μ and standard deviation σ. Support vector machines (SVMs) with RBF kernel [7] are used for classification.

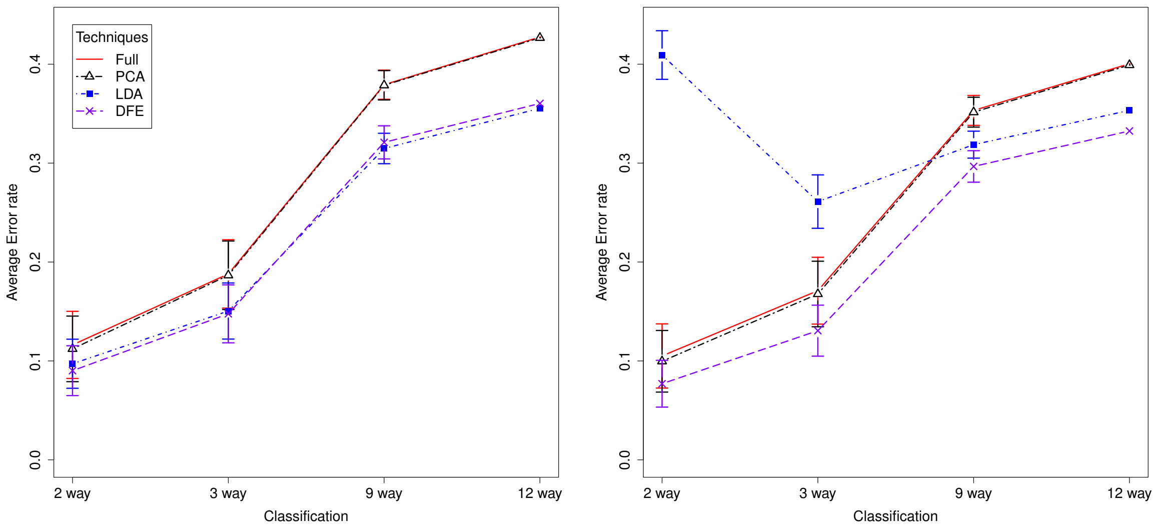
4.2. Comparison of Feature Extraction Methods

We compare the proposed discriminative feature extraction (DFE) method with standard approaches such as PCA, LDA. Hence we only compare those methods using features up to 3-gram statistics. Specifically we did 2-way[2], 3-way, 9-way and 12-way experiments, the results $\left(\mu \pm \frac{\sigma}{2}\right)$ using different number of features are shown in Figure 2.

When comparing different techniques, we project the high-dimensional data onto $m = C - 1$ dimensional subspace $\mathbb{R}^m$ for LDA, DFE. The optimal value for m should be the number of positive eigenvalues, generally this number is $C - 1$. Since the dimensionality of subspace $\mathbb{R}^m$ has a huge influence over the performance when using PCA, we evaluate performance for a range of m and report the best performing one. In the 2-gram case we can see that the classification performances using supervised feature extraction methods (LDA, DFE) are superior (with p-value$\leq 2.2 \times 10^{-16}$) to the unsupervised method (PCA) or full dimensional features. LDA and DFE give almost the same result, and the results are not statistically significant. Also, using unsupervised method (PCA) doesn't give us any gain over the full-dimensional case.

In the 3-gram case, there are 1386 features, which is much larger than the number of samples $N = 800$ in 2-way classification. LDA's inability to handle more features becomes evident. Specifically, in 2-way and 3-way classifications ($M >$

[2] 2-way means there are 2 classes in the dataset



(a)

(b)

Fig. 2: Comparison of different feature extraction methods (a) using up to 2-gram features; (b) using up to 3-gram

N), LDA fails to extract useful features and gives high error rates. While the proposed DFE consistently gives the lowest error rate (with p-value$\leq 2.2 \times 10^{-16}$).

Including 4-gram statistics in the feature vectors, the classification error rate can be further reduced. Since $M >> N$, we will not include LDA in the comparison, the results $(\mu \pm \frac{\sigma}{2})$ are shown in Table 1. All the experiments show that the performance of the LID system can be improved by using the correct feature extraction method.

Methods	2-way	3-way	9-way	12-way
Full	0.102±0.032	0.167±0.034	0.348±0.015	0.396
PCA	0.096±0.031	0.162±0.033	0.343±0.015	0.390
DFE	0.076±0.023	0.124±0.026	0.276±0.013	0.321

Table 1: Comparison of different feature extraction methods

5. CONCLUSION

In this paper we propose a discriminative features extraction (DFE) method based on maximization of the non-parametric estimation of the mutual information $I(C;Y)$ [4], which can be solved by a generalized eigenvalue problem. The inclusion of labeling information enables DFE to extract features that are relevant for classification tasks. Compared with other methods, DFE can handle the increasing features with ease and improve classification performance. The work presented in this paper is related to that of [4] in that they both try to maximize the mutual information between the class label C and the projected features $Y = W^T X$: $I(C;Y)$. In [4] a gradient method is used to perform the optimization with the constrain $-\lambda \|W\|^2$, which might lead to suboptimal or extreme solutions. We used simplified approximation of $I(C;Y)$ to avoid suboptimal solutions and treat the optimization as a generalized eigenvalue problem. Experimental results show the proposed method is viable and effective.

6. REFERENCES

[1] Marc A. Zissman and Kay M. Berkling, "Automatic language identification," *Speech Comm.*, vol. 35, pp. 115–124, 2001.

[2] W.J. Krzanowski, P. Jonathan, W.V McCarthy, and M.R. Thomas, "Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data," *Applied Statistics*, vol. 44, pp. 101–115, 1995.

[3] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley and Sons, Inc., Hoboken, NJ. 2006.

[4] Lev Faivishevsky and Jacob Goldberger, "Dimensionality reduction based on non-parametric mutual information," *Neurocomputing*, pp. 31–37, 2012.

[5] CallFriend Corpus, *CallFriend Corpus, Linguistic Data Consortium*, 1996, http://www.ldc.upenn.edu/Catalog/.

[6] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Exploring universal attribute characterization of spoken languages for spoken language recognition," in *Proc. of Interspeech*, Brighton, UK, 2009.

[7] K. Hornik A. Karatzoglou, A. Smola and A. Zeileis, "kernlab – an S4 package for kernel methods in R," in *Journal of Statistical Software*, 2004, vol. 11.