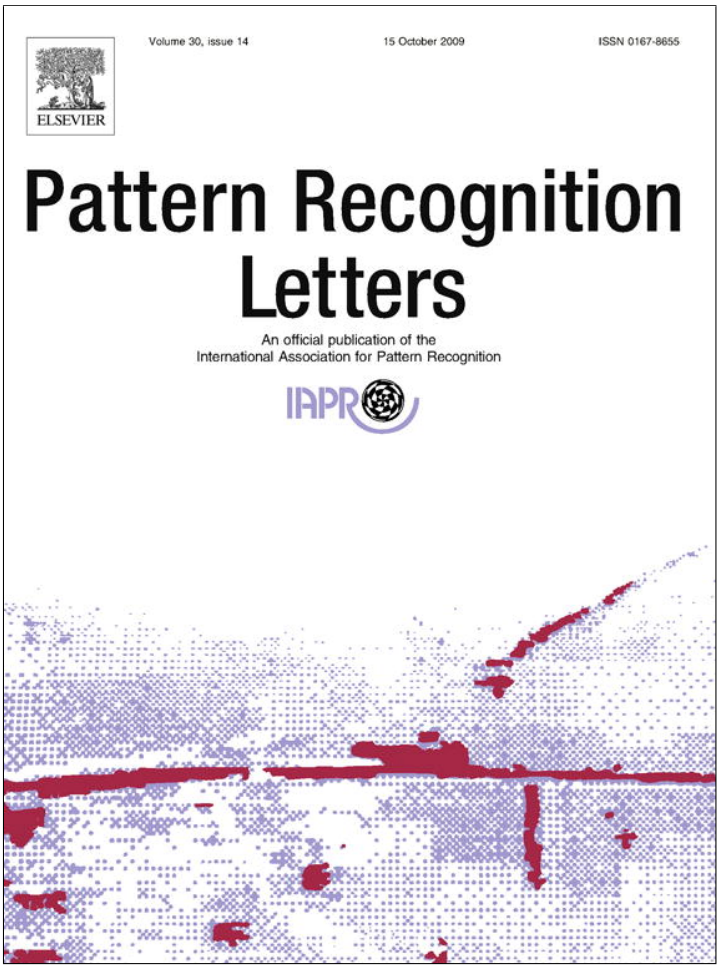

Pattern Recognition Letters 30 (2009) 1295–1300

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Using continuous features in the maximum entropy model ☆

Dong Yu *, Li Deng, Alex Acero

Microsoft Research, One Microsoft Way, Redmond, WA 98052, United States

ARTICLE INFO

Article history:
Received 16 October 2008
Received in revised form 11 May 2009
Available online 24 June 2009

Communicated by R.C. Guido

Keywords:
Maximum entropy principle
Spline interpolation
Continuous feature
Maximum entropy model
Moment constraint
Distribution constraint

ABSTRACT

We investigate the problem of using continuous features in the maximum entropy (MaxEnt) model. We explain why the MaxEnt model with the moment constraint (MaxEnt-MC) works well with binary features but not with the continuous features. We describe how to enhance constraints on the continuous features and show that the weights associated with the continuous features should be continuous functions instead of single values. We propose a spline-based solution to the MaxEnt model with non-linear continuous weighting functions and illustrate that the optimization problem can be converted into a standard log-linear model at a higher-dimensional space. The empirical results on two classification tasks that contain continuous features are reported. The results confirm our insight and show that our proposed solution consistently outperforms the MaxEnt-MC model and the bucketing approach with significant margins.

1. Introduction

The maximum entropy (MaxEnt) model with moment constraints (MaxEnt-MC) on binary features has been shown effective in natural language processing (NLP) (e.g., Berger et al., 1996), speaker identification (e.g., Ma et al., 2007), statistical language modeling (e.g., Rosenfeld, 1996), text filtering and cleaning (e.g., Yu et al., 2005a), machine translation (e.g., Och and Ney, 2002), phonotactic learning (e.g., Hayes, 2008), visual object classification (e.g., Gong et al., 2004), economic modeling (e.g., Arndt et al., 2002), and network anomaly detection (e.g., Gu et al., 2005). However, it is not very successful when non-binary (e.g., continuous) features are used. To improve the performance, quantization techniques such as bucketing (or binning) have been proposed to convert the continuous features into binary features. Unfortunately, quantization techniques provide only limited performance improvement due to its intrinsic limitations. A coarse quantization may introduce large quantization errors and wash out the gain obtained from using the converted binary features, and a fine quantization may increase the number of model parameters dramatically and introduce parameter estimation uncertainties.

In this paper, we examine the MaxEnt model and the principle behind it. We bring the insight that the key to the success of using the MaxEnt model is providing appropriate constraints. We show that moment constraints on binary features are very strong and fully regularize the distribution of the features. However, moment constraints on continuous features are rather weak and as a result much information contained in the training set is not used by the MaxEnt model. Therefore, using continuous features is less effective than using binary features in the MaxEnt-MC model.

We further discuss how stronger constraints can be included for continuous features by using quantization techniques. We extend the quantization technique to its extreme to introduce the distribution constraint and show that the weights associated with continuous features in the MaxEnt model should not be single values but continuous functions. In other words, the optimization problem is no longer a log-linear problem but a non-linear problem with continuous weighting functions as parameters. We solve this non-linear optimization problem by approximating the continuous weighting function with spline interpolations we recently developed in our variable parameter hidden Markov model (VPHMM) work (Yu et al., 2008, in press). We demonstrate that by using the spline interpolation the optimization problem with non-linear continuous weighting functions can be converted into a standard log-linear problem at a higher-dimensional space where each continuous feature in the original space is mapped into several features. With this transformation, the existing training and testing algorithms (Nocedal, 1980; Riedmiller and Braun, 1993; Malouf, 2002) as well as the recently developed regularization techniques (Chen and Rosenfeld, 1999, 2000; Goodman, 2004; Kazama,

☆ A small portion of this work has been presented at the NIPS 2008 workshop on speech and language: Learning-based methods and systems at Whistler, BC, Canada in December 2008.

* Corresponding author. Fax: +1 425 706 7329.
E-mail addresses: dongyu@microsoft.com (D. Yu), deng@microsoft.com (L. Deng), alexac@microsoft.com (A. Acero).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.06.005

2004; Kazama and Tsujii, 2005) for the MaxEnt-MC model can be directly applied in this higher-dimensional space making our approach very attractive. We validate our insight and the effectiveness of our approach on two classification tasks that contain continuous features and show that our proposed solution consistently outperforms the MaxEnt-MC model and the quantization-based approach with significant margins.

The rest of the paper is organized as follows. In Section 2, we examine the MaxEnt model and discuss why the MaxEnt model with moment constraints performs well for binary features but not for continuous features. In Section 3, we illustrate that continuous weighting functions (instead of single weight values) should be used for continuous features and propose a solution to the optimization problem that contains continuous weighting functions by approximating the weighting functions with spline interpolations. We validate our insight and demonstrate the new approach's superiority over the MaxEnt-MC and quantization-based approaches empirically on two classification tasks in Section 4, and conclude the paper with discussions on many potential applications in Section 5.

2. The MaxEnt model and constraints

In this section, we examine the MaxEnt principle and the MaxEnt model and explain why the MaxEnt model with moment constraints works well for the binary features but not for the continuous features by showing that the moment constraints on binary features are strong while on continuous features weak.

2.1. The MaxEnt principle and MaxEnt model with moment constraints

We consider a random process that produces an output value y from a finite set Y for an input value x. We assume that a training set $(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)$ with N samples is given. The training set can be represented with the empirical probability distribution

$$\tilde{p}(x,y)=\frac{\text{number of times that } (x,y) \text{ occur}}{N}. \tag{1}$$

Our goal is to construct a stochastic model that can accurately represent the random process that generated the training set $\tilde{p}(x,y)$. We denote $p(y\mid x)$ as the probability of outputting by y the model when x is given and assume that a set of constraints C is known either from the training data and/or from *a priori* knowledge.

The MaxEnt principle (Guiasu and Shenitzer, 1985) dictates that from all the probability distributions $p(y\mid x)$ that accord with the constraints C, we should select the distribution that is most uniform. Mathematically, we should select the distribution that maximizes the entropy

$$H(p)=-\sum_{x,y}\tilde{p}(x)p(y\mid x)\log p(y|x), \tag{2}$$

over the conditional probability $p(y\mid x)$.

A typical type of constraints used in the MaxEnt model is moment constraints. Assume that a set of M features $f_i(x,y)$, $i=1,\ldots,M$ is available, the moment constraint requires that the moment of the features as predicted from the model should be the same as that observed from the training set. In most cases only the constraints on the first-order moment is used, i.e.,

$$E_p[f_i]=E_{\tilde{p}}[f_i],\quad i=1,\ldots,M, \tag{3}$$

where E_p is the expected value over the distribution p defined as

$$E_p[f_i]=\sum_{x,y}\tilde{p}(x)p(y|x)f_i(x,y), \tag{4}$$

and $E_{\tilde{p}}$ is the expected value over the distribution $\tilde{p}$ defined as

$$E_{\tilde{p}}[f_i]=\sum_{x,y}\tilde{p}(x,y)f_i(x,y)=\sum_{x,y}\tilde{p}(x)\tilde{p}(y|x)f_i(x,y). \tag{5}$$

A nice property of the MaxEnt model with moment constraints (Berger et al., 1996) is that its solution is in the log-linear form of

$$p(y|x)=\frac{1}{Z_\lambda(x)}\exp\left(\sum_i\lambda_i f_i(x,y)\right), \tag{6}$$

where

$$Z_\lambda(x)=\sum_y\exp\left(\sum_i\lambda_i f_i(x,y)\right), \tag{7}$$

is a normalization constant to make sure $\sum_y p(y|x)=1$, and λ_i is the weight for the feature $f_i(x,y)$ and is chosen to maximize

$$\Psi(\lambda)=-\sum_x\tilde{p}(x)\log Z_\lambda(x)+\sum_i\lambda_i E_{\tilde{p}}[f_i]. \tag{8}$$

Since this dual problem is an unconstraint convex problem, many algorithms such as generalized iterative scaling (GIS) (Darroch and Ratcliff, 1972), gradient ascent and conjugate gradient (e.g., L-BFGS) (Nocedal, 1980), and RPROP (Riedmiller and Braun, 1993) can be used to find the solution. A comparison on the performance of different learning algorithms can be found in (Malouf, 2002 and Mahajan et al., 2006). Notice that applying the higher-order moment constraints in the MaxEnt model is equivalent to using higher-order statistics as features in the MaxEnt model with mean (i.e., first-order moment) constraint. The MaxEnt-MC model has been improved with regularization techniques (Chen and Rosenfeld, 1999, 2000; Goodman, 2004) and uncertain constraints (Kazama, 2004; Kazama and Tsujii, 2005) in the recent years.

2.2. Moment constraints on binary features and continuous features

The MaxEnt principle basically says one should not assume any additional structure or constraints other than those already imposed on the constraint set C. The appropriate selection of the constraints thus is crucial. In principle, we should include all the constraints that can be validated by (or reliably estimated from) the training set or prior knowledge.

With the binary features where $f_i(x,y)\in\{0,1\}$, the moment constraint described in Eq. (3) is a strong constraint since $E_p[f]=p(f=1)$. In other words, constraining the expected value implicitly constrains the probability distribution. However, the moment constraint is rather weak for continuous features. Constraining the expected value does not mean much to the continuous features because many different distributions can yield the same expected value. That is to say, much information carried in the training set is not used in the parameter estimation if solely moment constraints are used for the continuous features especially when the distribution of the features has multiple modes. This is the most important reason that the MaxEnt-MC model works well for binary features but not so well for non-binary features, especially the continuous features.

Let us illustrate this observation with an example. Consider a random process that generates 0 with probability 1 if $x\in\{1,3\}$, and generates 1 with probability 1 if $x\in\{2\}$, and assume that we have a training set with the empirical joint distributions

$$\begin{aligned}&\tilde{p}(1,0)=0.25,\quad \tilde{p}(1,1)=0,\\ &\tilde{p}(2,0)=0,\quad \tilde{p}(2,1)=0.5,\\ &\tilde{p}(3,0)=0.25,\quad \tilde{p}(3,1)=0,\end{aligned} \tag{9}$$

and features

$$f_1(x,y) = \begin{cases} x & \text{if } y = 0 \\ 0 & \text{otherwise,} \end{cases}$$
$$f_2(x,y) = \begin{cases} x & \text{if } y = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Notice that these features have the same empirical first-order moment and so the same moment constraint since

$$E_{\tilde{p}}[f_1] = 0.25 \times 1 + 0.25 \times 3 = 1, \quad \text{and} \quad (11)$$
$$E_{\tilde{p}}[f_2] = 0.5 \times 2 = 1. \quad (12)$$

However, they have different distributions since

$$\begin{aligned} &\tilde{p}(f_1 = 0) = 0.5, \quad \tilde{p}(f_2 = 0) = 0.5, \\ &\tilde{p}(f_1 = 1) = 0.25, \quad \tilde{p}(f_2 = 1) = 0, \\ &\tilde{p}(f_1 = 2) = 0, \quad \tilde{p}(f_2 = 2) = 0.5, \\ &\tilde{p}(f_1 = 3) = 0.25, \quad \tilde{p}(f_2 = 3) = 0. \end{aligned} \quad (13)$$

This indicates that the moment constraint is not strong enough to distinguish these two different feature distributions and the resulting MaxEnt model performs poorly.

To get a better statistical model, quantization techniques such as bucketing (or binning) have been proposed to convert the continuous (or multi-value) features to the binary features and enforce the constraints on the derived binary features. With bucketing, for example, a continuous feature f_i in the range of $[l,h]$ can be converted K into binary features

$$f_{ik}(x,y) = \begin{cases} 1 & \text{if } f_i(x,y) \neq 0 \quad \text{and} \quad x \in [l_k, h_k] \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

where $k \in \{1,2,\ldots,K\}$, and $l_k = h_{k-1} = (k-1)(h-l)/K + l$. Using bucketing we essentially approximate the constraints on the distribution of the continuous features with the moment constraints on each segment. Including constraints in each segment reduces the feasible set of the conditional probabilities $p(y|x)$ and forces the model learned matches the training set more closely. For clarity we use MatEnt-QT to denote the MaxEnt model with quantization techniques for the rest of the paper.

3. MaxEnt model with continuous features

In this section, we introduce the distribution constraint and show that the weights associated with the continuous features should be continuous functions instead of single values in the MaxEnt model. We further propose a solution to this more complex optimization problem by approximating the continuous weighting functions with spline-interpolations.

3.1. Continuous features with continuous weights

The bucketing approach (Eq. (14)) mentioned in Section 2.2 can be modified so that

$$f_{ik}(x,y) = \begin{cases} \frac{h_k + l_k}{2} & \text{if } f_i(x,y) \neq 0 \quad \text{and} \quad x \in [l_k, h_k] \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

Notice that with this reformation, the features are still binary since each feature takes only two values: 0 and $(h_k + l_k)/2$. The only difference this new feature construction approach will cause compared with the original approach in Eq. (14) is that the corresponding weights λ_{ik} learned will be scaled down by $(h_k + l_k)/2$. As we increase the number of buckets, we increase the constraints, better describe the distribution of the continuous features, and reduce the quantization errors. However, increasing the number of buckets also increases the number of weighting parameters λ_{ik} to be estimated and the uncertainty of the constraints since the empirical expected values are now estimated with less training samples. In real applications, a compromise usually needs to be made to balance these two forces if this approach is to be used.

Now assume we have *infinite* number of samples in the training set, we may increase the number of buckets to any large number we want and thus enforce a distribution constraint. Under this condition, we have

$$\lim_{k \to \infty} \sum_k \lambda_{ik} f_{ik}(x,y) = \lambda_i(f_i(x,y)) f_i(x,y), \quad (16)$$

by noticing that only one $f_{ik}(x,y)$ is none-zero for each (x,y) pair, where $\lambda_i(f_i(x,y))$ is a continuous weighting function over the feature values and to be learned. Eq. (16) suggests that for continuous features we should use continuous weighting functions instead of single weight values. In other words, the solution to the MaxEnt model with distribution constraint (MaxEnt-DC) has the form of

$$p(y|x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_{i \in \{\text{continuous}\}} \lambda_i(f_i(x,y)) f_i(x,y) + \sum_{j \in \{\text{binary}\}} \lambda_j f_j(x,y) \right). \quad (17)$$

3.2. Solution with spline interpolation approximation

Two difficulties exist in using continuous weighting functions. First, Eq. (17) cannot be solved with the existing MaxEnt-MC training and testing algorithms. In fact, the model is no longer log-linear. Second, the constraints at each real-valued point are hard to enforce since the number of training samples is usually limited. In this sub-section we propose to convert Eq. (17) into the standard log-linear form by approximating the non-linear continuous weighting functions with spline interpolations.

Spline interpolation is a standard way of approximating continuous functions. Any type of spline may be used. Two most commonly used splines are the linear spline and the cubic spline since the values of these splines can be efficiently calculated. In this study, we use the cubic spline which is smooth up to the second-order derivative. Two typical boundary conditions for the cubic spline are typically used: one for which the first derivative is known and the other where the second derivative is zero. The spline with the latter boundary condition is usually called natural spline and is the one used in this study.

Given K knots $\{(f_{ij}, \lambda_{ij}) | j = 1,\ldots,K; f_{ij} < f_{i(j+1)}\}$ in the cubic spline with the natural boundary condition, the value $\lambda_i(f_i)$ of a data point f_i can be estimated as

$$\lambda_i(f_i) = a\lambda_{ij} + b\lambda_{i(j+1)} + c\frac{\partial^2 \lambda_i}{\partial f_i^2}|f_i = f_{ij} + d\frac{\partial^2 \lambda_i}{\partial f_i^2}|f_i = f_{i(j+1)}, \quad (18)$$

where if we define $\Delta f_{ij} = f_{i(j+1)} - f_{ij}$,

$$\begin{aligned} a &= \frac{f_{i(j+1)} - f_i}{\Delta f_{ij}}, \\ b &= 1 - a, \\ c &= \frac{1}{6}(a^3 - a)(\Delta f_{ij})^2, \\ d &= \frac{1}{6}(b^3 - b)(\Delta f_{ij})^2, \end{aligned} \quad (19)$$

are interpolation parameters, and $[f_{ij}, f_{i(j+1)}]$ is the section where the point f_i falls. $\lambda_i(f_i)$ can also be written into the matrix form

$$\lambda_i(f_i) \cong \boldsymbol{a}^T(f_i)\boldsymbol{\lambda}_i, \quad (20)$$

as shown in (Yu et al., 2008, in press), where $\boldsymbol{\lambda}_i = [\lambda_{i1},\ldots,\lambda_{iK}]^T$, $\boldsymbol{a}^T(f_i) = \boldsymbol{e}^T(f_i) + \boldsymbol{f}^T(f_i)\boldsymbol{C}^{-1}\boldsymbol{D}$, is a vector,

$$\boldsymbol{e}^T(f_i) = \left[0 \cdots \underbrace{a}_{j} \underbrace{b}_{j+1} \cdots 0,\right], \qquad \boldsymbol{f}^T(f_i) = \left[0 \cdots \underbrace{c}_{j} \underbrace{d}_{j+1} \cdots 0,\right], \tag{21}$$

$$\boldsymbol{C} = \begin{bmatrix} \frac{\Delta f_{i1}}{6} & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{\Delta f_{i1}}{6} & \frac{\Delta f_{i2}+\Delta F_{i1}}{3} & \frac{\Delta f_{i2}}{6} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & \frac{\Delta f_{i(j-1)}}{6} & \frac{\Delta f_{ij}+\Delta f_{i(j-1)}}{3} & \frac{\Delta f_{ij}}{6} & 0 & \vdots \\ \vdots & \vdots & 0 & \vdots & \vdots & \vdots & 0 \\ 0 & \cdots & \vdots & 0 & \frac{\Delta f_{i(k-2)}}{6} & \frac{\Delta f_{i(k-1)}+\Delta f_{i(k-2)}}{3} & \frac{\Delta f_{i(k-1)}}{6} \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & \frac{\Delta f_{i(k-1)}}{6} \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{1}{\Delta f_{i1}} & -\frac{1}{\Delta f_{i1}}-\frac{1}{\Delta f_{i2}} & \frac{1}{\Delta f_{i2}} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & \frac{1}{\Delta f_{i(j-1)}} & -\frac{1}{\Delta f_{i(j-1)}}-\frac{1}{\Delta f_{ij}} & \frac{1}{\Delta f_{ij}} & 0 & \vdots \\ \vdots & \vdots & 0 & \vdots & \vdots & \vdots & 0 \\ 0 & \cdots & \vdots & 0 & \frac{1}{\Delta f_{i(K-2)}} & -\frac{1}{\Delta f_{i(K-2)}}-\frac{1}{\Delta f_{i(K-1)}} & \frac{1}{\Delta f_{i(K-1)}} \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

If we are given K evenly distributed knots $\{(f_{ik}, \lambda_{ik}) | k = 1, \ldots, K\}$ where $h = \Delta f_{ik} = \Delta f_{ij} > 0, \forall j, k \in \{1, \ldots, K-1\}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ can be simplified as

$$\boldsymbol{C} = \begin{bmatrix} \frac{h}{6} & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{h}{6} & \frac{2h}{3} & \frac{h}{6} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & \frac{h}{6} & \frac{2h}{3} & \frac{h}{6} & 0 & \vdots \\ \vdots & \vdots & 0 & \vdots & \vdots & \vdots & 0 \\ 0 & \cdots & \vdots & 0 & \frac{h}{6} & \frac{2h}{3} & \frac{h}{6} \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & \frac{h}{6} \end{bmatrix},$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{1}{h} & -\frac{2}{h} & \frac{1}{h} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & 0 & \frac{1}{h} & -\frac{2}{h} & \frac{1}{h} & 0 & \vdots \\ \vdots & \vdots & 0 & \vdots & \vdots & \vdots & 0 \\ 0 & \cdots & \vdots & 0 & \frac{1}{h} & -\frac{2}{h} & \frac{1}{h} \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

Notice that with Eq. (20) we have

$$\lambda_i(f_i) f_i \cong \boldsymbol{a}^T(f_i) \boldsymbol{\lambda}_1 f_i = [\boldsymbol{a}^T(f_i) f_i] \boldsymbol{\lambda}_i = \sum_k \lambda_{ik} [a_k(f_i) f_i], \tag{22}$$

where $a_k(f_i)$ is the k-th element of $\boldsymbol{a}^T(f_i)$. Eq. (22) indicates that the product of a continuous feature with its continuous weight can be approximated as a sum of the products of K transformed features in the form of $a_k(f_i)f_i$ with the corresponding K single-valued weights. Eq. (17) can thus be converted into

$$p(y|x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_{i \in \{\text{continuous}\}, k} \lambda_{ik} f_{ik}(x, y) + \sum_{j \in \{\text{binary}\}} \lambda_j f_j(x, y) \right), \tag{23}$$

where

$$f_{ik}(x, y) = a_k(f_i(x, y)) f_i(x, y), \tag{24}$$

only depends on the original feature $f_i(x, y)$ and the locations of the knots and depends on the weights to be estimated.

Although the spline-approximation may carry errors, this approach has several advantages over using the continuous weighting functions directly. First, we can better trade-off between the uncertainty of the constraints and the accuracy of the constraints since the weight value at each knot is estimated using not only the information at the knot but also information from many other samples in the training set. For example, when cubic-spline is used, each original continuous feature will affect four features in the higher-dimensional space. Second, Eq. (23) is in the standard log-linear form and can be efficiently solved with existing algorithms (Nocedal, 1980; Riedmiller and Braun, 1993; Malouf, 2002) for the MaxEnt-MC model except the algorithms that cannot handle negative values, e.g., GIS (Darroch and Ratcliff, 1972), since the derived features may be negative. In addition, all the recent advances in the MaxEnt-MC model such as the regularization techniques (Chen and Rosenfeld, 1999, 2000; Goodman, 2004) and uncertain constraints (Kazama, 2004; Kazama and Tsujii, 2005) can be directly applied to the converted optimization problem. Compared to the quantization approaches, our approach has better theoretical justification, has less approximation errors, and generally obtains better performance as shown in Section 4.

There are several practical considerations in using either bucketing or the novel approach proposed in this paper for continuous features. First, $f_i(x, y) = 0$ essentially turns off the feature and so the original continuous feature should not have values across 0 if a bias term is not used. Second, both the bucketing approach and our approach require the lower and higher bounds of the features and therefore, we should normalize the features into a fixed range. We suggest that we map the features f into the range of [1,2] so that it also satisfies the first consideration. This can be done by first limiting the range of the features into $[l, h]$ with sigmoid function and then convert the features with $f' = (f + h - 2l)/(h - l)$. Third, knots with both equal-distance and non-equal-distance can be used. Equal-distance knots are simpler and more efficient but less flexible. This problem can be alleviated by either increasing the number of knots or normalizing the features so that the distribution of the samples is close to uniform. Notice that using a small number of knots may model the constraints less accurately and effectively reduce the classification accuracy. Increasing the number of knots forces the model obtained to follow more closely to the distribution observed in the training data and may decrease the generalization ability of the model. A balance can be achieved by choosing the number of knots based on a development set when this approach is applied to a new task.

4. Experiments

To validate our insight and theory, we have compared the MaxEnt-DC model, where distribution constraint is used, with the MaxEnt-MC model, where each continuous feature is constrained on the lowest K-order moments, and the MaxEnt-QT model, where each continuous feature is quantized into K segments, on two classification tasks from the UCI data repository (Asuncion and Newman, 2007). The first data set used is the handwritten letter recognition data set which has 16 continuous features and contains 20,000 samples, out of which 16,000 samples are used for training and 4000 samples are used for testing. The second data set is the MAGIC gamma telescope data set. It has 10 continuous features and contains 19,020 samples, out of which 15,020 samples are used for training and 4000 samples were used for testing. We want

to point out that in all the experiments we conduct, we do not use the structures specific to the data set and/or the task since our goal is not to find the best model for the specific task but to compare different approaches that use the MaxEnt model. For this reason, we treat all the features available in the data sets as blind final features without interpreting the meanings of each feature and/or transforming the features. The RPROP (Riedmiller and Braun, 1993) training algorithm is used to train all these models since experiments have shown that it performs best among all the most popular training algorithms (Malouf, 2002; Mahajan et al., 2006). In addition, the Gaussian prior (Chen and Rosenfeld, 1999, 2000) is used to regularize the weights in all the experiments. We have also normalized the higher-order statistics in the MaxEnt-MC model because we and others (Mahajan et al., 2006) have noticed that the error rate is very high without the proper normalization since the existing algorithms are typically optimized for features taking values in the similar range.

Tables 1 and 2 compare the classification error rate on the letter recognition data set and the MAGIC gamma telescope data set with different number of moments/buckets/knots for each continuous feature respectively. From these tables, we have the following four clear observations:

- First, our proposed approach consistently outperforms the MaxEnt-MC model and the MaxEnt-QT model with significant margins. All the improvements shown in the tables are statistically significant at the significance level of 1%.
- Second, the bucketing approach may underperform the MaxEnt-MC model with mean (i.e., the first-order moment) constraint when the number of buckets is small. This is due to the fact that the quantization error is large under these conditions and so the gain from additional constraints is wiped out by the quantization errors introduced. As the number of buckets increases, the error rate decreases and eventually the bucketing approach outperforms the MaxEnt-MC model with mean constraint. However, the bucketing approach with K-buckets typically underperforms the MaxEnt-MC model with constraints on the K lowest-order moments. The MaxEnt-DC model, however, performs significant better than the MaxEnt-MC model with higher-order moment constraints even when the number of knots is smaller than the order of moments used.
- Third, with 4 knots, our approach can outperform the MaxEnt-MC model with 8 lowest-order moments and the bucketing approach with 8 buckets. In other words, our approach performs better than the MaxEnt-MC model and the MaxEnt-QT model with even half of the parameters.
- Fourth, for the MAGIC gamma telescope data set, over-fitting behavior starts to show with both the MaxEnt-MC model with 8 lowest-order moment constraints and the MaxEnt-QT model when the number of buckets is 8. However, the MaxEnt-DC model still get some gain on the test set when the number of knots is 8. This indicates that our approach typically has better ability to avoid over-fitting than the MaxEnt-MC and MaxEnt-QT models.

All these observations confirm the superiority of our approach against the MaxEnt-MC model and the bucketing approach.

Notice that with our approach, the classification error drops significantly when the number of knots changes from 4 to 5 and continuously decreases as the number of knots increases. However, the reduction of the classification error decreases as the number of knots further increases, which indicates that a trade-off between accuracy and generalization needs to be determined and the critical point can be estimated with a development set.

5. Summary and discussion

In this paper, we have examined the MaxEnt principle and the MaxEnt model. We showed that for continuous features, the weights should be continuous functions instead of single values. We provided a solution to the optimization problem that contains continuous weighting functions. The beauty of our solution is that we can spread and expand each original feature into several features at a higher-dimensional space through a non-linear mapping. With this feature transformation, the optimization problem with continuous weighting functions is converted into a standard log-linear feature combination problem and the existing MaxEnt-MC algorithms and improvements can thus be directly used. We have empirically validated our insight and the effectiveness of our solution compared to the MaxEnt-MC model and the MaxEnt-QT model using two classification tasks.

We see great impact of this work on using MaxEnt models. In the past, although great efforts have been put on using the models in the MaxEnt family to improve the systems' performances, the improvements are either small or negative when continuous features are involved. The work presented in this paper sheds lights to these tasks. For example, in the natural language and speech processing fields, our approach can be applied but not limited to the following areas:

- System combination: where classification scores such as posterior probabilities from different systems can be combined to achieve better accuracy.
- Confidence calculation: where acoustic model (AM) and language model (LM) scores can be combined with other features to estimate the confidence.

Table 1
Classification error rate (%) on the letter recognition data set with different approaches and different number of moments/buckets/knots.

# Of buckets/knots	1	2	3	4	5	6	7	8
MaxEnt-MC	22.82	20.75	19.18	18.80	18.38	18.25	17.98	17.85
MaxEnt-QT				35.52	29.32	24.18	19.93	19.23
MaxEnt-DC				15.60	14.55	14.18	13.88	12.93

Table 2
Classification error rate (%) on the MAGIC gamma telescope data set with different approaches and different number of moments/buckets/knots.

# Of buckets/knots	1	2	3	4	5	6	7	8
MaxEnt-MC	20.20	18.13	17.60	17.20	16.60	16.50	16.45	16.55
MaxEnt-QT				20.68	19.43	17.77	17.43	17.70
MaxEnt-DC				15.63	15.00	14.43	14.18	14.13

- Call routing: where counts or frequency of the unigram/bigram can be used to determine where to rout the call.
- Document classification: where counts or frequency of the unigram/bigram can be used to determine the document type.
- Conditional random field (CRF) and hidden CRF (HCRF) (Mahajan et al., 2006; Yu et al., 2009) based AM: where cepstrum, LM score and long-range dependency features (Deng et al., 2005; Yu et al., 2005b, 2006) can be used to build a conditional speech recognition model.

References

Arndt, C., Robinson, S., Tarpc, F., 2002. Parameter estimation for a computable general equilibrium model: A maximum entropy approach. Econ. Model. 19 (3), 375–398.

Asuncion, A., Newman, D.J., 2007. UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

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

Chen, S.F., Rosenfeld, R., 1999. A gaussian prior for smoothing maximum entropy models. In: Technical Report CMU-CS-99-108, Carnegie Mellon University.

Chen, S.F., Rosenfeld, R., 2000. A survey of smoothing techniques for ME models. IEEE Trans. Speech Audio Process. 8 (1), 37–50.

Darroch, J., Ratcliff, D., 1972. Generalized iterative scaling for log-linear models. Ann. Math. Statist. 43, 1470–1480.

Deng, L., Li, X., Yu, D., Acero, A., 2005. A hidden trajectory model with bi-directional target-filtering: Cascaded vs. integrated implementation for phonetic recognition, In: Proc. of ICASSP 2005, vol. 1, pp. 337–340.

Gong, Yi., Han, M., Hua, W., Xu, W., 2004. Maximum entropy model-based baseball highlight detection and classification. Computer Vision and Image Understanding 96 (2), 181–199. Special Issue on Event Detection in Video.

Goodman, J., 2004. Exponential priors for maximum entropy models. In: Proc. of the HLT-NAACL, pp. 305–311.

Gu, Y., McCallum, A., Towsley, D., 2005. Detecting anomalies in network traffic using maximum entropy estimation. In: Proc. of Internet Measurement Conf., pp. 345–350.

Guiasu, S., Shenitzer, A., 1985. The principle of maximum entropy. Math. Intell. 7 (1).

Hayes, B., 2008. A maximum entropy model of phonotactics and phonotactic learning. Linguist. Inq. 39 (3), 379–440.

Kazama, J., 2004. Improving maximum entropy natural language processing by uncertainty-aware extensions and unsupervised learning. Ph.D. Thesis, University of Tokyo.

Kazama, J., Tsujii, J., 2005. Maximum entropy models with inequality constraints: A case study on text categorization. Mach. Learn. 60 (1-3), 159–194.

Ma, C., Nguyen, P., Mahajan, M., 2007. Finding speaker identities with a conditional maximum entropy model. In: Proc. of ICASSP 2007, vol. IV, pp. 261–264.

Mahajan, M., Gunawardana, A., Acero, A., 2006. Training algorithms for hidden conditional random fields. In: Proc. of ICASSP 2006, vol. I, pp. 273–276.

Malouf, R., 2002. A comparison of algorithms for maximum entropy parameter estimation. In: Proc. of CoNLL, vol. 20, pp. 1–7.

Nocedal, J., 1980. Updating quasi-newton matrices with limited storage. Math. Comput. 35, 773–782.

Och, F.J., Ney, H., 2002. Discriminative training and maximum entropy models for statistical machine translation. In: Proc. of the 40th Annual Meeting of the ACL, pp. 295–302.

Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster back-propagation learning: The RPROP algorithm. In: Proc. of IEEE ICNN, vol. 1, pp. 586–591.

Rosenfeld, R., 1996. A maximum entropy approach to adaptive statistical language modeling. Comput. Speech Lang. 10, 187–228.

Yu, D., Mahajan, M., Mau, P., Acero, A., 2005a. Maximum entropy based generic filter for language model adaptation. In: Proc. of ICASSP 2005, vol. I, pp. 597–600.

Yu, D., Deng, L. Acero, A., 2005b. Evaluation of a long-contextual-span hidden trajectory model and phonetic recognizer using A* lattice search, In: Proc. of Interspeech 2005, pp. 553–556.

Yu, D., Deng, L., Acero, A., 2006. Structured speech modeling. IEEE Trans. Audio, Speech, Lang. Process. 14 (5), 1492–1504.

Yu, D., Deng, L., Gong, Y., Acero, A., 2008. Discriminative training of variable-parameter hmms for noise robust speech recognition. In: Proc. of Interspeech 2008, vol. I, pp. 285–288.

Yu, D., Deng, L., Acero, A., 2009. Hidden conditional random field with distribution constraints for phone classification, In: Proc. of Interspeech 2009.

Yu, D., Deng, L., Gong, Y., Acero, A., in press. A novel framework and training algorithm for variable-parameter hidden markov models. IEEE Trans. Audio, Speech, Lang. Process.