ISSUES WITH PARTIALLY MATCHING FEATURE FUNCTIONS IN CONDITIONAL EXPONENTIAL MODELS

Carsten Elfers, Hartmut Messerschmidt and Otthein Herzog
Center for Computing and Communication Technologies, University Bremen, Am Fallturm 1, 28357 Bremen, Germany

Keywords: Approximate feature functions, Conditional random fields, Partially matching feature functions, Regularization.

Abstract: Conditional Exponential Models (CEM) are effectively used in several machine learning approaches, e.g., in Conditional Random Fields. Their feature functions are typically either satisfied or not. This paper presents a way to use partially matching feature functions which are satisfied to some degree and corresponding issues while training. Using partially matching feature functions improves the inference accuracy in domains with sparse reference data and avoids overfitting. Unfortunately, the typically used Maximum Likelihood training includes some issues for using partially matching feature functions. In this context three problems (*inequality of influence*, *unlimited weight boundaries* and *local optima in parameter space*) with Improved Iterative Scaling (a popular training algorithm for Conditional Exponential Models) using such feature functions are stated and solved.

1 INTRODUCTION

Conditional Exponential Models (CEM) are effectively used in several machine learning approaches (e.g., in the Maximum Entropy Markov Model (MEMM) (McCallum et al., 2000) and in Conditional Random Fields (CRF) (Lafferty et al., 2001)). CEMs are using features (also called feature functions) to describe the data. Features describe arbitrary or multiple aspects of an observation, like the feature *good weather* which is satisfied if it is warm and sunny. Machine learning methods using CEM are often assuming binary valued features, i.e., they are either satisfied or not. However, in practice there are situations in which no feature is satisfied but a prediction is still desired. For the example the question may occur how to decide if the weather is sunny but not warm without a feature describing this observation? If warm means more than 20 degrees of Celsius, what if it is only 19 degrees? The problem of missing features arises typically in two situations: (1) When not enough features have been specified in advance to represent the data. (2) There is not enough reference data to train the features, i.e., the influence of these features to the inference is unknown (and therefore disregarded). To overcome this problem we introduce the concept of partially matching features, e.g., the feature *good weather* may be satisfied by 50% when the weather is sunny but it is not warm.

The problem of missing reference data has already been investigated for several learning approaches, e.g., in Input-Output Hidden Markov Models (Oblinger et al., 2005) and Markov Models (Anderson et al., 2002). Encouraging experiments regarding the problem of missing features have been made for Conditional Random Fields in (Elfers et al., 2010). In this paper we present the formal basis for CEM with partially matching features (which is a necessary step to overcome the problem of sparse reference data and overfitting) and discuss several problems (and solutions) regarding the training with Improved Iterative Scaling (IIS) (Berger et al., 1996), the most applied training algorithm to CEM.

The paper is organized as follows: In Sec. 2 we introduce Conditional Exponential Models and define partially matching feature functions. In Sec. 3 the influence of partially matching feature functions to the posterior distribution is investigated. In Sec. 4 the problems of Improved Iterative Scaling (IIS) with partially matching feature functions are gathered and solved by extending the algorithm. The paper finishes with the conclusion and outlook in Sec. 5.

Elfers C., Messerschmidt H. and Herzog O..
ISSUES WITH PARTIALLY MATCHING FEATURE FUNCTIONS IN CONDITIONAL EXPONENTIAL MODELS.
DOI: 10.5220/0003855205710578
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (SSML-2012), pages 571-578
ISBN: 978-989-8425-95-9

2 CONDITIONAL EXPONENTIAL MODELS AND PARTIALLY MATCHING FEATURE FUNCTIONS

Conditional Exponential Models are predominantly used in the area of natural language processing (see e.g., (Rosenfeld, 1996)). More recently they are also successfully applied to other domains, e.g., to the domain of intrusion detection (Gupta et al., 2010). Using a CEM allows to relax the strong independence assumptions typically made in the well-known Hidden Markov Model (HMM) (Rabiner, 1989). Contrary to a HMM the corresponding models using a CEM (i.e., MEMM and CRF) allow multiple overlapping and dependent features which are more appropriate to describe a sequential context (i.e., concurrent, previous and possibly next observations).

CEMs describe the data they are generated from by the use of an exponential function. This function is parameterized by a set of weighted feature functions, each representing some aspect of the input data. The weight of each feature function can be seen as a degree of influence of the corresponding feature to the posterior distribution.

Feature functions are typically binary real valued as described in Def. 1 (cf. (Berger et al., 1996)).

Definition 1 (Feature Function)**.** *A feature function $f'(\boldsymbol{x},y)$ is a binary valued function dependent to a discrete sequence of observations $\boldsymbol{x} = x_1,\cdots,x_t$ and a label $y \in \boldsymbol{y}$ from the set of all labels $\boldsymbol{y}$:*

$$f'(\boldsymbol{x},y) = \begin{cases} 1 & \text{if the feature matches} \\ & \text{on the given } \mathbf{x} \text{ and } y \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Please note that sets of elements are indicated by bold characters.

In this paper we are extending this to partially matching feature functions as in Def. 2 to solve the problem arising in the absence of matching feature functions:

Definition 2 (Partially Matching Feature Function)**.** *A partially matching feature function $f(\boldsymbol{x},y)$ is a real valued function in the interval $[0,1]$ dependent to a discrete sequence of observations $\boldsymbol{x} = x_1,\cdots,x_t$ and a label $y \in \boldsymbol{y}$ from the set of all labels $\boldsymbol{y}$. The value of such a function is called degree of matching.*

$$f(\boldsymbol{x},y) = \begin{cases} 1 & \text{if the feature matches} \\ ]0,1[\in \mathbb{R} & \text{if the feature matches partially} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

A CEM belongs to the group of discriminative models, which means that they only model the conditional probability of labels **y** (or classes in the case of classification) regarding a sequence of observations **x**. The major difference to generative models (like e.g., Hidden Markov Models) is that they do not learn how to generate samples or observations from the trained model. Any assumptions about the underlying generative process do not need to be modeled.

In the following the notation of Conditional Random Fields for CEM is used:

Definition 3 (Conditional Exponential Model)**.** *CEMs are defined for a label $y \in \boldsymbol{y}$ (the set of labels) conditioned under a vector of observations $\boldsymbol{x}$ regarding a set of real valued weights λ and a corresponding set of real valued feature functions $\boldsymbol{f}$ as:*

$$p(y|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\boldsymbol{x},y)\right) \quad (3)$$

In a CEM a partition function (or normalization function) Z is used to ensure that the result is a probability mass function.

Definition 4 (Partition Function)**.** *The partition function Z of a CEM is defined for an $\boldsymbol{x}$ over the sum of all possible labels $\boldsymbol{y}$ as:*

$$Z(\boldsymbol{x}) = \sum_{y \in \boldsymbol{y}} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\boldsymbol{x},y)\right) \quad (4)$$

The objective during training this model is to find an appropriate combination of weights λ to represent the maximum likelihood solution with respect to the given training data. This solution is typically found by the Improved Iterative Scaling algorithm. However, in the application we found that this algorithm has some problems by using real valued (or partially matching) feature functions which is discussed in this paper.

3 INFLUENCE OF PARTIALLY MATCHING FEATURE FUNCTIONS

In this section the bahavior of CEM with partially matching feature functions is analyzed. In particular how the degree of matching (cf. Def. 2) influences the posterior distribution. Therefore, basic monotonicity requirements are analyzed and proven. These are necessary to preclude unexpected behavior of a CEM using partially matching feature functions. Higher matching feature functions should contribute more to the posterior distribution than lower matching ones.

3.1 Monotonicity

The influence of the degree of matching f_i on the posterior distribution $p(y|\mathbf{x})$ is highly dependent on the assigned weight λ_i to this feature function.

Example 1. *In Fig. 1 this dependency is shown for a given observation x, two labels y and two feature functions. The feature function $f_i(\boldsymbol{x},\bar{y})$ depends on the corresponding weight λ_i in the interval $[-4,4]$ and only matches the plotted label $\bar{y}$, the other feature function is unsatisfied (i.e., zero) and, therefore, is independent of the assigned weight. This setup leads to the plotted equation for the posterior probability $p(\lambda,f)=\frac{\exp(\lambda f)}{\exp(\lambda f)+\exp(0)}$ with $f=f_i(\boldsymbol{x}$ and $\bar{y})$.*

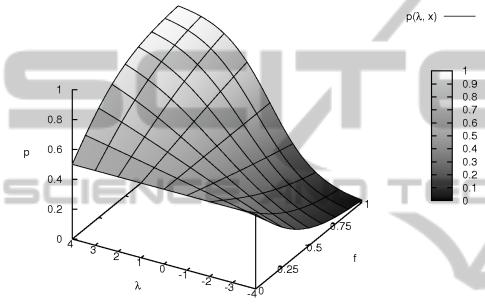

Figure 1: The posterior probability with one partially matching feature function and corresponding weights in the interval $[-4,4]$

The a posteriori probability with weight 4 *increases for the feature function values more rapidly and is more stable for higher degrees of matching. Correspondingly the a posteriori probability decreases for negative weights.*

This gives a first intuition how the exponential model behaves with respect to the degree of matching with arbitrary feature weights and partially matching feature functions. Intuitively, the a posteriori probability increases with respect to the degree of matching for positive weights and decreases for negative weights. This is a fundamental requirement for using partially matching feature functions in CEM. In the opposite case more exact matches of the feature functions would lead to a greater deviation from the Maximum Likelihood solution, which is obviously undesirable. Therefore this monotonicity is one of the most essential properties to show:

Theorem 1 (Value-monotonicity)**.**
The a posteriori probability p behaves strictly monotonic for non-zero weights (and is constant for zero weights) with respect to the feature function value for a given observation $\boldsymbol{x}$ and a fixed label $\bar{y}$. To specify the monotonicity the first derivative of the posterior probability p with respect to $f_n(\boldsymbol{x},\bar{y})$ is used, and denoted as $\frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y})}$:

$$\forall\lambda_n\in\lambda>0.\frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y})}>0. \tag{5}$$

$$\forall\lambda_n\in\lambda<0.\frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y})}<0. \tag{6}$$

$$\lambda_n\in\lambda=0.\frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y})}=0. \tag{7}$$

Proof. From Eqn. 3 and Eqn. 4:

$$p(\bar{y}|\mathbf{x})=\frac{\exp\left(\sum_{i=1}^{n-1}\lambda_i f_i(\mathbf{x},\bar{y})+\lambda_n f_n(\mathbf{x},\bar{y})\right)}{\sum_{j=1}^{m}\exp\left(\sum_{i=1}^{n-1}\lambda_i f_i(\mathbf{x},y_j)+\lambda_n f_n(\mathbf{x},y_j)\right)} \tag{8}$$

We differentiate $p(\bar{y}|\mathbf{x})$ with respect to a given observation $\mathbf{x}$. Then the normalization function Z is constant regarding f_n except for $f_n(\mathbf{x},\bar{y})$. C describes these constant parts:

$$C:=\sum_{y_j\neq\bar{y},j=1}^{m}\exp\left(\sum_{i=1}^{n}\lambda_i f_i(\mathbf{x},y_j)\right)$$

$$p=\frac{\exp\left(\sum_{i=1}^{n-1}\lambda_i f_i(\mathbf{x},\bar{y})+\lambda_n f_n(\mathbf{x},\bar{y})\right)}{C+\exp\left(\sum_{i=1}^{n-1}\lambda_i f_i(\mathbf{x},\bar{y})+\lambda_n f_n(\mathbf{x},\bar{y})\right)}$$

The derivative of the numerator and the denominator are both $\lambda_n\exp\left(\sum_i^{n-1}\lambda_i f_i(\mathbf{x},\bar{y})+\lambda_n f_n(\mathbf{x},\bar{y})\right)$. Therefore the derivative of $p_{f_n}(\bar{y}|x)$ with respect to $f_n(\mathbf{x},\bar{y})$ is as follows:

$$\begin{aligned}\frac{\partial p}{\partial f_n(\mathbf{x},\bar{y})}&=\frac{\lambda_n\exp\left(\sum_{i=1}^{n}\lambda_i f_i(\mathbf{x},\bar{y})\right)C}{\left(C+\exp\left(\sum_{i=1}^{n-1}\lambda_i f_i(\mathbf{x},\bar{y})+\lambda_n f_n(\mathbf{x},\bar{y})\right)\right)^2}\\&=\frac{\lambda_n\exp\left(\sum_{i=1}^{n}\lambda_i f_i(\mathbf{x},\bar{y})\right)\left(\sum_{y_j\neq\bar{y},j=1}^{m}\exp\left(\sum_{i}^{n}\lambda_i f_i(\mathbf{x},y_j)\right)\right)}{\left(\sum_{j=1}^{m}\exp\left(\sum_{i=1}^{n}\lambda_i f_i(\mathbf{x},y_j)\right)\right)^2}\end{aligned}$$

The denominator of the derivative is always positive and based on the fact that the exponential function is always greater zero for real numbers the only way to change the sign (or force the value to be zero) is the parameter λ_n. Therefore, Eqn. 5, Eqn. 6 and Eqn. 7 hold. □

Similarly, a proof can be made for a corresponding feature function weight:

Theorem 2 (Weight-monotonicity)**.** *The a posteriori probability behaves strictly monotonic for a given observation and for non-zero feature function values (and is constant for zero values) regarding the feature function weight. The first derivation of the posterior probability with respect to λ_n is denoted as $\frac{\partial p}{\partial \lambda_n}$:*

$$\forall f_n(\boldsymbol{x},\bar{y}) > 0. \frac{\partial p}{\partial \lambda_n} > 0 \tag{9}$$

$$\forall f_n(\boldsymbol{x},\bar{y}) < 0. \frac{\partial p}{\partial \lambda_n} < 0 \tag{10}$$

$$f_n(\boldsymbol{x},\bar{y}) = 0. \frac{\partial p}{\partial \lambda_n} = 0 \tag{11}$$

Proof. This proof can be done analogically to Proof 3.1 by differentiating $p(\bar{y}|x)$ with respect to a given observation. Then the normalization function Z is constant with respect to λ_n except for λ_n occurring together with $f_n(\mathbf{x},\bar{y})$. This leads to the following equation:

$$\frac{\partial p}{\partial \lambda_n} = \frac{f_n(\mathbf{x},\bar{y}) \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\mathbf{x},\bar{y})\right)\left(\sum_{y_j \neq \bar{y}, j=1}^{m} \exp\left(\sum_{i}^{n} \lambda_i f_i(\mathbf{x},y_j)\right)\right)}{\left(\sum_{j=1}^{m} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\mathbf{x},y_j)\right)\right)^2} \tag{12}$$

□

Theorem 1 and Theorem 2 show our presumption that the higher the degree of matching and the absolute value is, the higher is the influence on the a posteriori distribution, respectively. This is essential for working with partially matching feature functions and degrees of matching.

3.2 Shape of Monotonicity

At first the shape of the monotonicity is investigated by the analysis of the previously mentioned gradients. Therefore Eqn. 12 is rearranged to get the dependencies on the regarded variables λ_n and f_n.

$$\frac{\partial p}{\partial \lambda_n} = f_n(\mathbf{x},\bar{y}) \exp\left(\sum_{i=1}^{n-1} \lambda_i f_i(\mathbf{x},\bar{y})\right) \exp\left(\lambda_n f_n(\mathbf{x},\bar{y})\right) \cdot \frac{\left(\sum_{y_j \neq \bar{y}, j=1}^{m} \exp\left(\sum_{i}^{n-1} \lambda_i f_i(\mathbf{x},y_j)\right) \exp\left(\lambda_n f_n(\mathbf{x},y_j)\right)\right)}{\left(\sum_{j=1}^{m} \exp\left(\sum_{i=1}^{n-1} \lambda_i f_i(\mathbf{x},y_j)\right) \exp\left(\lambda_n f_n(\mathbf{x},y_j)\right)\right)^2}$$

Now we assume that λ_n and/or f_n are 0 for a given observation in all cases except $\bar{y}$:$f_n(\mathbf{x}, y \neq \bar{y}) = 0$. In other words the feature function f_n only matches the label $\bar{y}$. With this assumption we can rewrite this equation by introducing two constants C_1 and C_2:

$$\frac{\partial p}{\partial \lambda_n} = \frac{f_n(\mathbf{x},\bar{y}) C_1 \exp\left(\lambda_n f_n(\mathbf{x},\bar{y})\right) C_2}{\left(C_1 \exp\left(\lambda_n f_n(\mathbf{x},\bar{y})\right) + C_2\right)^2} \tag{13}$$

$$C_1 := \exp\left(\sum_{i=1}^{n-1} \lambda_i f_i(\mathbf{x},\bar{y})\right) \tag{14}$$

$$C_2 := \left(\sum_{y_j \neq \bar{y}, j=1}^{m} \exp\left(\sum_{i}^{n-1} \lambda_i f_i(\mathbf{x},y_j)\right) \exp\left(\lambda_n f_n(\mathbf{x},y_j)\right)\right) \tag{15}$$

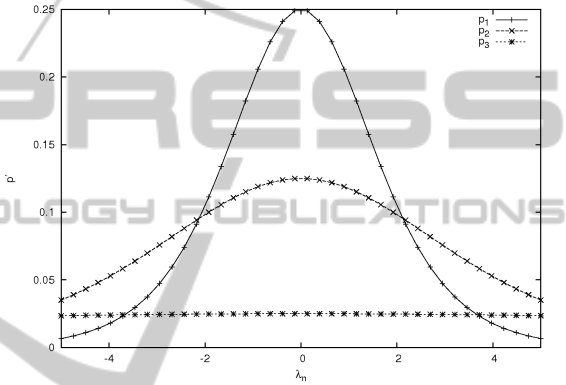

Figure 2: Plotted gradient p' for a feature function that is satisfied by 1.0 (p_1), by 0.5 (p_2) and by 0.1 (p_3) each with $C_1 = 0.1$ and $C_2 = 0.1$.

The shape of the monotonicity is independent of the amount of labels and feature functions (under the mentioned assumptions) but depends on $f_n(\mathbf{x},\bar{y})$ and λ_n as obvious from Eqn. 13. After this first conclusion an example will demonstrate a problem regarding the shape of monotonicity:

Example 2. *In Fig. 2, three gradients p_1, p_2, p_3 of λ_n occurring together with a certain $f_n(\boldsymbol{x},\bar{y})$, i.e. $p' = \frac{\partial p}{\partial \lambda_n}$ are printed with $C_1 = C_2 = 0.1$ (i.e., the influence of a certain feature function value in dependency of λ_n). p_1 has the feature value $f_n(\boldsymbol{x},\bar{y}) = 1$, p_2 has $f_n(\boldsymbol{x},\bar{y}) = 0.5$ and p_3 has $f_n(\boldsymbol{x},\bar{y}) = 0.1$. The influence of fully satisfied feature functions is converging faster with respect to λ_n than less satisfied feature functions. The higher the weight the faster the experienced convergence of the influence (the sharpener the graph). This leads to the fact that if the assigned weight increases over the intersection point with respect to another feature function's gradient, the feature function with the most increasing influence changes. On the one hand p_1 increases faster than p_2 and p_3 for $\lambda = 0$ and, on the other hand, p_1 increases slower than p_2*

and p_3 for $\lambda_n = 4$ which might be undesirable depending on the application domain or at least lead to counterintuitive inference results. This also touches the problem of overfitting since the influence decreases very rapidly for increasing/decreasing λ_n, e. g. if a feature is satisfied by .1 the posterior probability might increase from .0 to .8, by increasing the same feature to .2 the posterior probability might increase only by .1 to .9.

This leads to the first problem in using partially matching feature functions with CEMs:

Problem 1 (Inequality of Influence)**.** *The increase of influence of less satisfied feature functions may be greater than more satisfied feature functions for some assigned weights:*

$$\frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y}) = v_1} \geq \frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y}) = v_2}, \text{ with } v_1 \leq v_2 \text{ for some unknown } \lambda_n. \tag{16}$$

At next we want to investigate "the unknown λ" stated in the problem. Therefore, the mentioned intersection point in Example 2 can be determined by the following equation for two feature function values $f_n = v_1$ and $f_n = v_2$:

$$\frac{v_1 C_1 C_2 \exp(\lambda v_1)}{(C_1 \exp(\lambda v_1) + C_2)^2} = \frac{v_2 C_1 C_2 \exp(\lambda v_2)}{(C_1 \exp(\lambda v_2) + C_2)^2} \tag{17}$$

Theorem 3 (Monotonicity of the Increase of Influence)**.**
The increase of influence of a feature function is always greater or equal than the influence of another feature function with the same weight and a lower degree of matching if $-1 \leq \lambda \leq 1$:

$$\frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y}) = v_2} \geq \frac{\partial p}{\partial f_n(\boldsymbol{x},\bar{y}) = v_1}, \text{ with } v_2 > v_1 \text{ if } -1 \leq \lambda_n \leq 1. \tag{18}$$

Proof. It is easy to see from Eqn. 13 that $\frac{\partial p}{\partial f_n(\mathbf{x},\bar{y})=v_2} \geq \frac{\partial p}{\partial f_n(\mathbf{x},\bar{y})=v_1}$, with $v_2 > v_1$ holds for $\lambda = 0$. At next we show with respect to λ the condition at which the feature function with the most influence may change. Therefore, Eqn. 17 is rearranged to:

$$\ln\left(\frac{v_1 \exp(\lambda v_1)}{(C_1 \exp(\lambda v_1) + C_2)^2}\right) = \ln\left(\frac{v_2 \exp(\lambda v_2)}{(C_1 \exp(\lambda v_2) + C_2)^2}\right) \tag{19}$$

$$\ln(v_1) + \lambda v_1 - 2\ln(C_1 \exp(\lambda v_1) + C_2) = \ln(v_2) + \lambda v_2 - 2\ln(C_1 \exp(\lambda v_2) + C_2)$$

The equation $\ln(x+y) = \ln(x) + ln\left(1 + \frac{y}{x}\right)$ is used to rearrange to:

$$\ln(v_1) - \lambda v_1 - 2\ln\left(1 + \frac{C_2}{C_1 \exp(\lambda v_1)}\right) = \ln(v_2) - \lambda v_2 - 2\ln\left(1 + \frac{C_2}{C_1 \exp(\lambda v_2)}\right)$$

$$\lambda(v_2 - v_1) = \ln(v_2) - \ln(v_1) + 2\ln\left(1 + \frac{C_2}{C_1 \exp(\lambda v_1)}\right) - 2\ln\left(1 + \frac{C_2}{C_1 \exp(\lambda v_2)}\right)$$

$$\lambda = \frac{\ln v_2 - \ln v_1}{v_2 - v_1} + \frac{2}{v_2 - v_1} \ln\left(\frac{1 + \frac{C_2}{C_1 exp(\lambda v_1)}}{1 + \frac{C_2}{C_1 \exp(\lambda v_2)}}\right) \tag{20}$$

The term $\frac{2}{v_2 - v_1} \ln\left(\frac{1 + \frac{C_2}{C_1 exp(\lambda v_1)}}{1 + \frac{C_2}{C_1 \exp(\lambda v_2)}}\right)$ from Eqn. 20 is always positive under the assumption $v_2 > v_1$ (from the theorem) and $\lambda \geq 0$.[1] Therefore this term is omitted to find a lower bound λ for the intersection point:

$$\lambda \leq \ln\left(\frac{v_2}{v_1}\right) \frac{1}{v_2 - v_1} \tag{21}$$

$$\lambda(v_2 - v_1) \leq \ln\left(\frac{v_2}{v_1}\right) \tag{22}$$

At next the power series for the natural logarithm is taken: $\ln \frac{v_2}{v_1} = 2\frac{\frac{v_2}{v_1} - 1}{\frac{v_2}{v_1} + 1} + R_1(\frac{v_2}{v_1})$, where $R_1(\frac{v_2}{v_1})$ is positive if $v_2 > v_1$ which is assumed in the theorem. This leads to the inequality:

$$2\frac{v_2 - v_1}{v_2 + v_1} \leq \ln\left(\frac{v_2}{v_1}\right) \tag{23}$$

Due to the definition of the feature function weights we can assume $0 \leq v_2 + v_1 \leq 2$ which leads to:

$$c(v_2 - v_1) \leq \ln\left(\frac{v_2}{v_1}\right) \tag{24}$$

with $c = \frac{2}{v_1 + v_2} \geq 1$. Compared to Eqn. 22 we have proven that the inequality holds for $\lambda \geq 1$. This proof can be done correspondingly to a negative λ to find the upper bound of λ which is $\lambda \leq -1$. This inequality shows that the only possible intersection points of λ (of the gradients) are outside (or exactly on the border) of the interval $[-1, 1]$.

□

[1] This equation is always negative under the assumption $\lambda < 0$. This is helpful for the proof of the upper bound.

Solution 1 (Inequality of Influence)**.** *Proof 3.2 shows that Problem 1 can be easily solved by ensuring that all weights are in the interval* $[-1,1]$.

This solution is also applicable to a second problem regarding the training of a CEM with Improved Iterative Scaling (IIS). The general absence of a limitation of the weights' interval during training with IIS leads to a nearly unpredictable influence of partially matching feature functions to the a posteriori probability (this has already been discussed in the sense of regularization, e.g., in (Jin et al., 2003)).

Problem 2 (Unlimited Weight Boundaries)**.** *The influence of partially matching feature functions directly depends on the assigned weights. There exists no boundary (neither an upper nor a lower boundary) of the weights which makes the influence of partially matching feature functions nearly unpredictable (e.g., due to possibly infinite weights).*

Solution 2 (Unlimited Weight Boundaries)**.** *This problem is already solved by Solution 1. However, to solve this problem a less restrictive solution is possible: It is sufficient to ensure that all weights are in a limited interval.*[2] *Additionally, the change of influence with respect to* λ *may also be regarded as a feature to tune the model in the way how partially matching feature functions should be integrated in the inference process. As the weight interval increases, the possible influence of partially matching feature functions increases as well. This might be a reason to choose a less restrictive interval than in Solution 1, however one must be aware of loosing the corresponding properties from Theorem 3.*

3.3 Exemplification

In the previous section we have proven the monotonic properties of exponential models and their probability space. In this section we investigate the behavior of this model for multiple partially matching and complementary feature functions by examples. Specifically the behavior of the model with weights over 1 are demonstrated.

Example 3. *Fig. 3 shows a setup with two feature functions (*$f_1 = v_1$ *and* $f_2 = v_2$*) matching on the first of two possible labels (*$y = 1$*), i.e. they have a positive value and weight for this case. The second feature function's weight has double the weight of the first one, i.e.* $\lambda_2 = 2\lambda_1$ *and* $\lambda_1 = 0.5$*. The feature functions matching the second label (*$y = 2$*) are the complement of the feature functions matching the first label, i.e.* $f_1(y = 1) = v_1$, $f_1(y = 2) = 1 - v_1$, $f_2(y = 2) = v_2$, $f_2(y = 2) = 1 - v_2$ *which can be regarded as a typical example with respect to partially matching feature functions.*

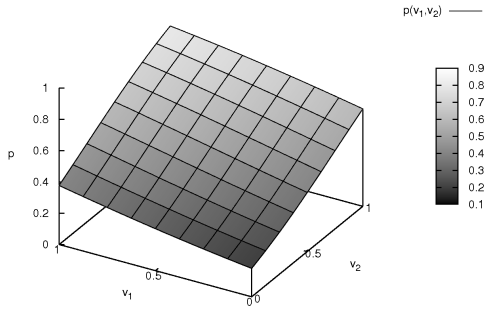

Figure 3: CEM with two feature functions ($f_1 = v_1$, $f_2 = v_2$) and two labels, $\lambda_1 = 0.5$, $\lambda_2 = 1$.

We observe that this small model (with few feature functions) and limited weights (positive and less than one) cannot represent all results in the probability space, e.g. if both feature functions are fulfilled the posterior probability is not one as expected. However, allowing negative weights or having more matching feature functions overcomes this problem. Please note the smooth distribution of the probability space for such a small weight interval and that the feature function value v_2 has a higher influence to the posterior distribution than v_1 as expected.

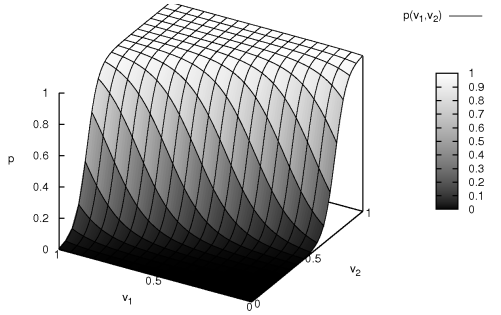

Figure 4: CEM with two feature functions ($f_1 = v_1$, $f_2 = v_2$) and two labels, $\lambda_1 = 5$, $\lambda_2 = 10$.

Example 4. *Fig. 4 shows the same setup as in Example 3 but with ten times the weights, i.e.* $\lambda_1 = 5$ *and* $\lambda_2 = 10$.

It is obvious that this distribution is not as smooth as the previous one. Specifically less matching feature functions have a higher influence to the posterior distribution, e.g. $v_1 = v_2 = 0.75$ is nearly one. The next thing to mention is the rapid decrease of the

[2]This is easily possible while using Improved Iterative Scaling due to the dependency on the update value to all the model parameters. A combination with other regularization methods such as fuzzy maximum entropy (cf. (Chen and Rosenfeld, 2000)) may be desirable but is out of the scope of this paper.

influence for barely satisfied feature functions, e.g. $v_1 = v_2 = 0.25$ is nearly zero. This gives an idea why it might be preferable to lower the restriction regarding the weights as stated in Solution 1 in special cases, e.g. in a domain in which Problem 1 is of minor importance and perhaps a stronger influence of partially satisfied feature functions is desired.

4 ISSUES WITH IMPROVED ITERATIVE SCALING

The Improved Iterative Scaling (IIS) algorithm (Berger et al., 1996) uses a lower bound on the gradient to optimize the weights of the Conditional Exponential Model regarding maximum likelihood (and maximum entropy). The idea of this algorithm is that each weight can be optimized (by gradient descent) independently to the other weights in an iterative way. Algorithm 1 shows the draft of this approach.

Algorithm 1 (Improved Iterative Scaling Algorithm)**.**

- *Start with* $\lambda_i = 0$
- *Do for all* λ_i *until convergence:*
  - *Determine a weight update value* δ_i
  - *Update* $\lambda_i \leftarrow \lambda_i + \delta_i$

The problem of this algorithm with partially matching feature functions is that if the model is trained for either satisfied or unsatisfied feature functions, these values are optimized regarding maximum likelihood (and maximum entropy) but the behavior of partially matching feature functions during inference is not fully constrained by IIS. Bancarz et al. (Bancarz and Osborne, 2002) found that there exists a single global optimum in the likelihood space but multiple local optima in the space of model parameters. This leads to Problem 3:

Problem 3 (Local Optima in Parameter Space)**.** *Improved Iterative Scaling converges the model parameters to a single global optimum in the likelihood space but to unspecified local optima in the space of model parameters. This (also) leads to an unpredictable influence of the partially matching feature functions on the a posteriori probability.*

Bancarz et al. showed that the global maximum can lead to different performances already for binary valued feature functions. However, the problem has a greater impact for partially matching feature functions due to the unpredictable influence on the a posteriori probability.

Example 5. *Consider two feature functions with different weights, but both leading to a posterior probability of* 100% *for some label if they are fully satisfied. This is generally possible as stated in (Bancarz and Osborne, 2002). If these feature functions are both satisfied by only* 50% *this leads to a preference to one label without any rational reason (because of the multiple solutions for the model parameters).*

Bancarz et al. suggested a simple solution to this problem by initializing all weights with zero.[3] However, this is not enough due to the update at each iteration through IIS which results in a faster update of some weights (and therefore to an unjustified divergence of the model paramerters). This problem can be seen in the gradient used in the IIS algorithm from (Berger, 1997):

$$\frac{\partial B(\Lambda)}{\partial \delta_i} = \sum_{x,y} \tilde{p}(x,y) f_i(x,y) - \sum_x \tilde{p}(x) \sum_y p_\Lambda(y|x) \sum_i f_i(x,y) \exp\left(\delta_i f^{\#}(x,y)\right)$$

The value of the trained model, denoted as $p_\Lambda(y|x)$, is used in the gradient to determine the weight updates and the updated weights itself are used to determine the value of the model $p_\Lambda(y|x)$ as shown in Algorithm 1.[4] Therefore, we need an additional constraint to ensure that the model parameters are also equal if the expected value of the feature functions $\tilde{p}(f) = \sum_{x,y} \tilde{p}(x,y) f(x,y)$ are equal. This constraint must be independent of the iteratively chosen model parameters:

$$\forall i . \forall j . \; (\tilde{p}(f_i) = \tilde{p}(f_j)) \Rightarrow (\lambda_i = \lambda_j) \qquad (25)$$

Solution 3 (Local Optima in Parameter Space)**.** *Problem 3 can be easily avoided by splitting the loop into an update determination step for all weights and a separate update step (as seen in Algorithm 2). This leads to an equal treatment of the partially matching feature functions and satisfies the additional constraint Eqn. 25.*

The application of all solutions results in the following algorithm:

Algorithm 2 (Additionally Constrained Improved Iterative Scaling Algorithm (AC-IIS))**.**

(1) Start with $\lambda_i = 0$
Do until convergence:
Do for all λ_i*:*
(2) Determine a weight update value δ_i

[3] This has already been suggested in (Berger et al., 1996), however in (Pietra et al., 1997; Berger, 1997) any initial value for the weights have been allowed.

[4] In this equation the notion of (Berger et al., 1996) has been kept. $f^{\#}(x,y) = \sum_i f_i(x,y)$.

Do for all λ_i:

(3) Update $\lambda_i \leftarrow \lambda_i + \delta_i$

(4) Ensure that λ_i is in a given weight interval

Step (1) and the splitting of the convergence loop into (2) and (3) solves Problem 3, step (4) solves Problem 1 and Problem 2.

5 CONCLUSIONS AND OUTLOOK

In this paper the Conditional Exponential Model (which is used in Maximum Entropy Markov Models and Conditional Random Fields) has been extended to be used with partially matching feature functions. This work enables the use of partially matching feature functions with Conditional Exponential Models and Improved Iterative Scaling in a well-defined way to overcome the problem of missing features. It has been shown that the influence of partially matching feature functions on the posterior probability changes in the correct direction (i.e., monotonicity). Further the impact of the weights has been analyzed. Problems regarding IIS have been identified and a solution in a modified algorithm has been developed. Additionally the problem of overfitting is addressed by allowing potentially all feature functions to be satisfied to some degree of matching (and therefore smooth the posterior distribution). In future work we are going to show how partially matching feature functions may be defined in a semantically intuitive way and present empirical results of such a combined method. First steps have already been done in the domain of intrusion detection.

ACKNOWLEDGEMENTS

This work was supported by the German Federal Ministry of Education and Research (BMBF) under the grant 01IS08022A.

REFERENCES

Anderson, C. R., Domingos, P., and Weld, D. S. (2002). Relational Markov models and their application to adaptive web navigation. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 143–152, New York, NY, USA. ACM.

Bancarz, I. and Osborne, M. (2002). Improved iterative scaling can yield multiple globally optimal models with radically differing performance levels. In *Proceedings of the 19th International Conference on Computational Linguistics*, volume 1, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Berger, A. (1997). The improved iterative scaling algorithm: A gentle introduction.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. In *Computational Linguistics*, volume 22, pages 39–71, Cambridge, MA, USA. MIT Press.

Chen, S. and Rosenfeld, R. (2000). A survey of smoothing techniques for ME models. In *Speech and Audio Processing, IEEE Transactions on*, volume 8, pages 37 –50.

Elfers, C., Horstmann, M., Sohr, K., and Herzog, O. (2010). Typed linear chain conditional random fields and their application to intrusion detection. In *Proceedings of the 11th International Conference on Intelligent Data Engineering and Automated Learning*, Lecture Notes in Computer Science. Springer Verlag Berlin.

Gupta, K. K., Nath, B., and Ramamohanarao, K. (2010). Layered approach using conditional random fields for intrusion detection. In *IEEE Transactions on Dependable and Secure Computing*.

Jin, R., Yan, R., Zhang, J., and Hauptmann, A. G. (2003). A faster iterative scaling algorithm for conditional exponential model. In *Proceedings of the 20th International Conference on Machine Learning*, pages 282–289.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

McCallum, A., Freitag, D., and Pereira, F. C. N. (2000). Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 591–598, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Oblinger, D., Castelli, V., Lau, T., and Bergman, L. D. (2005). Similarity-based alignment and generalization. In *Proceedings of ECML 2005*.

Pietra, S. D., Pietra, V. D., and Lafferty, J. (1997). Inducing features of random fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 380 –393.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286.

Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modeling. In *Computer, Speech and Language*, volume 10, pages 187–228.