Boosting and Maximum Likelihood for Exponential Models

Guy Lebanon John Lafferty

October 6, 2001
CMU-CS-01-144

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Recent research has considered the relationship between boosting and more standard statistical methods, such as logistic regression, concluding that AdaBoost is similar but somehow still very different from statistical methods in that it minimizes a different loss function. In this paper we derive an equivalence between AdaBoost and the dual of a convex optimization problem. In this setting, it is seen that the only difference between minimizing the exponential loss used by AdaBoost and maximum likelihood for exponential models is that the latter requires the model to be normalized to form a conditional probability distribution over labels; the two methods minimize the same Kullback-Leibler divergence objective function subject to identical feature constraints. In addition to establishing a simple and easily understood connection between the two methods, this framework enables us to derive new regularization procedures for boosting that directly correspond to penalized maximum likelihood. Experiments on UCI datasets, comparing exponential loss and maximum likelihood for parallel and sequential update algorithms, confirm our theoretical analysis, indicating that AdaBoost and maximum likelihood typically yield identical results as the number of features increases to allow the models to fit the training data.

This research was partially supported by the Advanced Research and Development Activity in Information Technology (ARDA), contract number MDA904-00-C-2106, and by the National Science Foundation (NSF), grant CCR-9805366.

Keywords: boosting, maximum likelihood, exponential models, convex duality, logistic regression, regularization

I. Introduction

Several recent papers in statistics and machine learning have been devoted to the relationship between boosting and more standard statistical procedures such as logistic regression. In spite of this activity, an easy-to-understand and clean connection between these different techniques has not emerged. Friedman, Hastie and Tibshirani [8] note the similarity between boosting and stepwise logistic regression procedures, and suggest a least-squares alternative, but view the loss functions of the two problems as different, leaving the precise relationship between boosting and maximum likelihood unresolved. Kivinen and Warmuth [9] note that boosting is a form of "entropy projection," and Lafferty [10] suggests the use of Bregman distances to approximate the exponential loss. Mason *et al.* [11] consider boosting algorithms as functional gradient descent and Duffy and Helmbold [6] study various loss functions with respect to the PAC boosting property. More recently, Collins, Schapire and Singer [3] show how different Bregman distances precisely account for boosting and logistic regression, and use this framework to give the first convergence proof of AdaBoost. However, in this work the two methods are viewed as minimizing different loss functions. Moreover, the optimization problems are formulated in terms of a reference distribution consisting of the zero vector, rather than the empirical distribution of the data, making the interpretation of this use of Bregman distances problematic from a statistical point of view.

In this paper we present a very basic connection between boosting and maximum likelihood for exponential models through a simple convex optimization problem. In this setting, it is seen that the only difference between AdaBoost and maximum likelihood for exponential models, in particular logistic regression, is that the latter requires the model to be normalized to form a probability distribution. The two methods minimize the same extended Kullback-Leibler divergence objective function subject to the same feature constraints. Using information geometry, we show that projecting the exponential loss model onto the simplex of conditional probability distributions gives precisely the maximum likelihood exponential model with the specified sufficient statistics. In many cases of practical interest, the resulting models will be identical; in particular, as the number of features increases to fit the training data the two methods will give the same classifiers. We note that throughout the paper we view boosting as a procedure for minimizing the exponential loss, using either parallel or sequential update algorithms as in [3], rather than as a forward stepwise procedure as presented in [8] or [7].

Given the recent interest in these techniques, it is striking that this connection has gone unobserved until now. However in general, there may be many ways of writing the constraints for a convex optimization problem, and many different settings of the Lagrange multipliers (or Kuhn-Tucker vectors) that represent identical solutions. The key to the connection we present here lies in the use of a particular non-standard presentation of the constraints. When viewed in this way, there is no need for special-purpose Bregman distances to give a unified account of boosting and maximum likelihood, as we only make use of the standard Kullback-Leibler divergence. But our analysis gives more than a formal framework for understanding old algorithms; it also leads to new algorithms for regularizing AdaBoost, which is required when the training data is noisy. In particular, we derive a regularization procedure for AdaBoost that directly corresponds to penalized maximum likelihood using a Gaussian prior. Experiments on UCI data support our theoretical analysis, demonstrate the effectiveness of the new regularization method, and give further insight into the relationship between boosting and maximum likelihood exponential models.

II. Notation and Assumptions on Data

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. We denote by $\mathcal{M} = \{m : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+\}$ the set of non-negative measures on $\mathcal{X} \times \mathcal{Y}$, and by $\Delta \subset \mathcal{M}$ the set of conditional probability distributions,

$$\Delta \;=\; \{m \in \mathcal{M} \,|\, \sum_{y\in\mathcal{Y}} m(x,y) = 1, \text{ for each } x \in \mathcal{X}\} \tag{2.1}$$

For $m \in \mathcal{M}$, we will overload the notation $m(x,y)$ and $m(y\,|\,x)$; the latter will be suggestive of a conditional probability distribution, but in general it need not be normalized. Let $f_j : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $j = 1, \ldots, m$, be given functions, which we will refer to as *features*. These will correspond to the *weak learners* in boosting, and to the *sufficient statistics* in an exponential model. Suppose that we have data $\{(x_i, y_i)\}_{i=1}^n$ with empirical distribution $\widetilde{p}(x,y)$ and marginal $\widetilde{p}(x)$; thus, $\widetilde{p}(x,y) = \frac{1}{n}\sum_{i=1}^n \delta(x_i,x)\,\delta(y_i,y)$. We assume, without loss of generality, that $\widetilde{p}(x) > 0$ for all x. Throughout the paper, we assume that the training data has the following property.

Consistent Data Assumption. For each $x \in \mathcal{X}$ with $\widetilde{p}(x) > 0$, there is a unique $y \in \mathcal{Y}$ for which $\widetilde{p}(y\,|\,x) > 0$. This y will be denoted $\widetilde{y}(x)$.

For most data sets of interest, each x appears only once, so that the assumption trivially holds. However, if x appears more than once, we require that it is labeled consistently. We make this assumption mainly to correspond with the conventions used to present boosting algorithms; it is not essential to what follows.

Given f_j, we define the conditional exponential model $q_\lambda(y\,|\,x)$, for $\lambda \in \mathbb{R}^m$, by

$$q_\lambda(y\,|\,x) = \frac{e^{\langle \lambda, f(x,y)\rangle}}{\sum_y e^{\langle \lambda, f(x,y)\rangle}} \tag{2.2}$$

The maximum likelihood estimation problem is to determine parameters λ that maximize the conditional log-likelihood $\ell(\lambda) = \sum_{x,y} \widetilde{p}(x,y) \log q_\lambda(y\,|\,x)$. The objective function to be minimized in the multi-label boosting algorithm AdaBoost.M2 [3] is the *exponential loss* given by

$$\mathcal{E}_{\mathrm{M2}}(\lambda) = \sum_{i=1}^n \sum_{y \neq y_i} e^{\langle \lambda, f(x_i,y) - f(x_i,y_i)\rangle} \tag{2.3}$$

In the binary case, AdaBoost takes $\mathcal{Y} = \{-1, +1\}$, weak learners $f_j : \mathcal{X} \to \mathbb{R}$, and minimizes the loss function

$$\mathcal{E}(\lambda) = \sum_{i=1}^n e^{-y_i \langle \lambda, f(x_i)\rangle} \tag{2.4}$$

This is a special case of the AdaBoost.M2 problem, obtained by taking $f_j(x,y) = \frac{1}{2} y f_j(x)$. For the same $\mathcal{Y}$ and f_j, the logistic model is given by

$$q_\lambda(y\,|\,x) \;=\; \frac{1}{1 + e^{-y\langle \lambda, f(x)\rangle}} \tag{2.5}$$

and the maximum likelihood problem becomes equivalent to minimizing the loss function

$$\bar{\ell}(\lambda) \;=\; \sum_{i=1}^n \log\left(1 - e^{-y_i\langle \lambda, f(x_i)\rangle}\right) \tag{2.6}$$

As has been often noted, the log-loss (2.6) and the exponential loss (2.4) are qualitatively different. The exponential loss (2.4) grows exponentially with increasing negative "margin" $y\,\langle\lambda, f(x)\rangle$, while the log-loss grows linearly.

III. Correspondence Between AdaBoost and Maximum Likelihood

We will work with the (extended) conditional Kullback-Leibler divergence, given by

$$D(p,q) \stackrel{\text{def}}{=} \sum_x \widetilde{p}(x) \sum_y \left(p(y\,|\,x) \log \frac{p(y\,|\,x)}{q(y\,|\,x)} - p(y\,|\,x) + q(y\,|\,x) \right) \tag{3.1}$$

defined on $\mathcal{M} \times \mathcal{M}$ (possibly taking on the value ∞). Note that if $p(\cdot\,|\,x) \in \Delta$ and $q(\cdot\,|\,x) \in \Delta$ then this becomes the more familiar KL divergence for probabilities; see [12] for a nice presentation of the use of the extended KL divergence for alternating minimization problems, including EM and iterative scaling. Let features f_j and a fixed default distribution $q_0 \in \mathcal{M}$ be given. We define the *feasible set* $\mathcal{F}(\widetilde{p}, f) \subset \mathcal{M}$ as

$$\mathcal{F}(\widetilde{p}, f) = \left\{ p \in \mathcal{M} \;\middle|\; \sum_x \widetilde{p}(x) \sum_y p(y\,|\,x) \, \left(f_j(x,y) - E_{\widetilde{p}}[f_j\,|\,x]\right) = 0, \text{ all } j \right\} \tag{3.2}$$

Since $\widetilde{p} \in \mathcal{F}$, this set is non-empty. Note that under the consistent data assumption, we have that $E_{\widetilde{p}}[f\,|\,x] = f(x, \widetilde{y}(x))$. Consider now the following two convex optimization problems, labeled P_1 and P_2.

$$\begin{array}{lll} (P_1) & \textit{minimize} & D(p, q_0) \\ & \textit{subject to} & p \in \mathcal{F}(\widetilde{p}, f) \end{array} \qquad\qquad \begin{array}{lll} (P_2) & \textit{minimize} & D(p, q_0) \\ & \textit{subject to} & p \in \mathcal{F}(\widetilde{p}, f) \\ & & p \in \Delta. \end{array}$$

Thus, problem P_2 differs from P_1 only in that the solution is required to be normalized. As we will show, the dual problem P_1^* corresponds to AdaBoost, and the dual problem P_2^* corresponds to maximum likelihood for exponential models.

This presentation of the constraints is the key to making the correspondence between AdaBoost and maximum likelihood. Note that the constraint $\sum_x \widetilde{p}(x) \sum_y p(y\,|\,x)\, f(x,y) = E_{\widetilde{p}}[f]$, which is the usual presentation of the constraints for maximum likelihood (as dual to maximum entropy), doesn't make sense for unnormalized models, since the two sides of the equation may not be "on the same scale." Note further that attempting to rescale by dividing by the mass of p to get $\sum_x \widetilde{p}(x) \frac{\sum_y p(y\,|\,x)\, f(x,y)}{\sum_y p(y\,|\,x)} = E_{\widetilde{p}}[f]$ would yield *nonlinear* constraints.

We now derive the dual problems formally; the following section gives a precise statement of the duality result. To derive the dual problem P_1^*, we calculate the Lagrangian as

$$\mathcal{L}_1(p, \lambda) = \sum_x \widetilde{p}(x) \sum_y p(y\,|\,x) \left(\log \frac{p(y\,|\,x)}{q_0(y\,|\,x)} - 1 - \left\langle \lambda, f(x,y) - E_{\widetilde{p}}[f\,|\,x] \right\rangle \right) \tag{3.3}$$

For $\lambda \in \mathbb{R}^m$, the connecting equation $q_\lambda \stackrel{\text{def}}{=} \arg\min_{p \in \mathcal{M}} \mathcal{L}_1(p, \lambda)$, is then calculated to be

$$q_\lambda(y\,|\,x) = q_0(y\,|\,x) \, \exp\left(\sum_j \lambda_j \left(f_j(x,y) - E_{\widetilde{p}}[f_j\,|\,x] \right) \right) \tag{3.4}$$

Thus, the dual function $h_1(\lambda) = \mathcal{L}_1(q_\lambda, \lambda)$ is given by

$$h_1(\lambda) \;=\; -\sum_x \widetilde{p}(x) \sum_y q_0(y\,|\,x)\, \exp\left(\sum_j \lambda_j\, \left(f_j(x,y) - E_{\widetilde{p}}[f_j\,|\,x]\right)\right) \tag{3.5}$$

The dual problem is to determine $\lambda^\star = \arg\max_\lambda h_1(\lambda)$. To derive the dual for P_2, we simply add additional Lagrange multipliers μ_x for the constraints $\sum_y p(y\,|\,x) = 1$.

3.1. Special cases

It is now straightforward to derive various boosting and logistic regression problems as special cases of the above optimization problems.

Case 1: AdaBoost.M2. Take $q_0(y\,|\,x) = 1$. Then the dual problem $\max_\lambda h_1(\lambda)$ is equivalent to computing

$$\lambda^\star \;=\; \arg\min_\lambda \sum_i \sum_{y \neq y_i} \exp\left(\sum_j \lambda_j\, (f_j(x_i,y) - f_j(x_i,y_i))\right) \tag{3.6}$$

which is the optimization problem of AdaBoost.M2.

Case 2: Binary AdaBoost. In addition to the assumptions for the previous case, now assume that $y \in \{-1,+1\}$, and take $f_j(x,y) = \frac{1}{2} y\, f_j(x)$. Then the dual problem is given by

$$\lambda^\star = \arg\min_\lambda \sum_i \exp\left(-y_i \sum_j \lambda_j f_j(x_i)\right) \tag{3.7}$$

which is the optimization problem of binary AdaBoost.

Case 3: Maximum Likelihood for Exponential Models. In this case we take the same setup as for AdaBoost.M2 but add the additional normalization constraints: $\sum_y p(y\,|\,x_i) = 1$, $i = 1, \ldots, n$. If these constraints are satisfied, then the other constraints take the form

$$\sum_x \widetilde{p}(x) \sum_y p(y\,|\,x) f_j(x,y) \;=\; \sum_{x,y} \widetilde{p}(x,y) f_j(x,y) \tag{3.8}$$

and the connecting equation becomes

$$q_\lambda(y\,|\,x) = \frac{1}{Z_x} q_0(y\,|\,x)\, \exp(\sum_j \lambda_j\, f_j(x,y)) \tag{3.9}$$

where Z_x is the normalizing constant $Z_x = \sum_y q_0(y|x) e^{\langle \lambda, f(x,y)\rangle}$, which corresponds to setting the Lagrange multiplier μ_x to the appropriate value. In this case, after a simple calculation the dual problem is seen to be

$$h_2(\lambda) \;=\; \sum_{x,y} \widetilde{p}(x,y) \log q_\lambda(y\,|\,x) \tag{3.10}$$

$$\;=\; \sum_x \widetilde{p}(x) \log q_\lambda(\tilde{y}\,|\,x) \tag{3.11}$$

which corresponds to maximum likelihood for a conditional exponential model with sufficient statistics $f_j(x, y)$.

Case 4: Logistic Regression. Returning to the case of binary AdaBoost, we see that when we add normalization constraints as above, the model is equivalent to binary logistic regression, since

$$q_\lambda(1 \mid x) \;=\; \frac{1}{1 + e^{-\langle \lambda, f(x) \rangle}} \tag{3.12}$$

We note that it is not necessary to scale the features by a constant factor here, as in [8]; the correspondence between logistic regression and boosting is direct.

3.2. Duality

Making the Lagrangian duality argument of the previous section rigorous requires care, because of the possibility that the solution may lie on the boundary of $\mathcal{M}$.

Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be defined as the following exponential families:

$$\mathcal{Q}_1(q_0, f) \;=\; \{q \in \mathcal{M} \mid q(y \mid x) \;=\; q_0(y \mid x)\, e^{\langle \lambda, f(x,y) - f(x, \tilde{y}(x)) \rangle},\ \lambda \in \mathbb{R}^m\} \tag{3.13}$$

$$\mathcal{Q}_2(q_0, f) \;=\; \{q \in \Delta \mid q(y \mid x) \;\propto\; q_0(y \mid x)\, e^{\langle \lambda, f(x,y) \rangle},\ \lambda \in \mathbb{R}^m\} \tag{3.14}$$

Thus $\mathcal{Q}_1$ is unnormalized while $\mathcal{Q}_2$ is normalized. We now define the boosting solution $q^\star_{boost}$ and maximum likelihood solution $q^\star_{ml}$ as

$$q^\star_{boost} \;=\; \arg\min_{q \in \overline{\mathcal{Q}_1}} \sum_x \widetilde{p}(x) \sum_y q(y \mid x) \tag{3.15}$$

$$q^\star_{ml} \;=\; \arg\min_{q \in \overline{\mathcal{Q}_2}} \sum_x \widetilde{p}(x) \log \frac{1}{q(\tilde{y} \mid x)} \tag{3.16}$$

where $\overline{Q}$ denotes the closure of the set $Q \subset \mathcal{M}$. It is interesting to cast the boosting objective function in terms of a statistical model. For $q \in \mathcal{Q}_1$, let $\hat{q}(y \mid x)$ be the normalized version of q:

$$\hat{q}(y \mid x) \;=\; \frac{q_0(y \mid x)\, e^{\langle \lambda, f(x,y) - f(x, \tilde{y}(x)) \rangle}}{\sum_y q_0(y \mid x)\, e^{\langle \lambda, f(x,y) - f(x, \tilde{y}(x)) \rangle}} \tag{3.17}$$

$$\;=\; \frac{q_0(y \mid x)\, e^{\langle \lambda, f(x,y) \rangle}}{\sum_y q_0(y \mid x)\, e^{\langle \lambda, f(x,y) \rangle}} \tag{3.18}$$

Then the boosting optimization problem (3.15) can be re-written as

$$q^\star_{boost} \;=\; \arg\min_{q \in \overline{\mathcal{Q}_1}} \sum_x \widetilde{p}(x) \frac{1}{\hat{q}(\tilde{y} \mid x)} \tag{3.19}$$

$$\;=\; \arg\min_{q \in \overline{\mathcal{Q}_2}} \sum_x \widetilde{p}(x) \frac{1}{q(\tilde{y} \mid x)} \tag{3.20}$$

The following result corresponds to Proposition 4 of [4] for the usual KL divergence; the proof for the extended KL divergence carries over with only minor changes. In [5] the duality theorem is proved for a class of Bregman distances, including the extended KL divergence as a special case. Note that we do not require divergences such as $D(\mathbf{0}, q)$ as in [3], but rather $D(\widetilde{p}, q)$, which is more natural and interpretable from a statistical point-of-view.

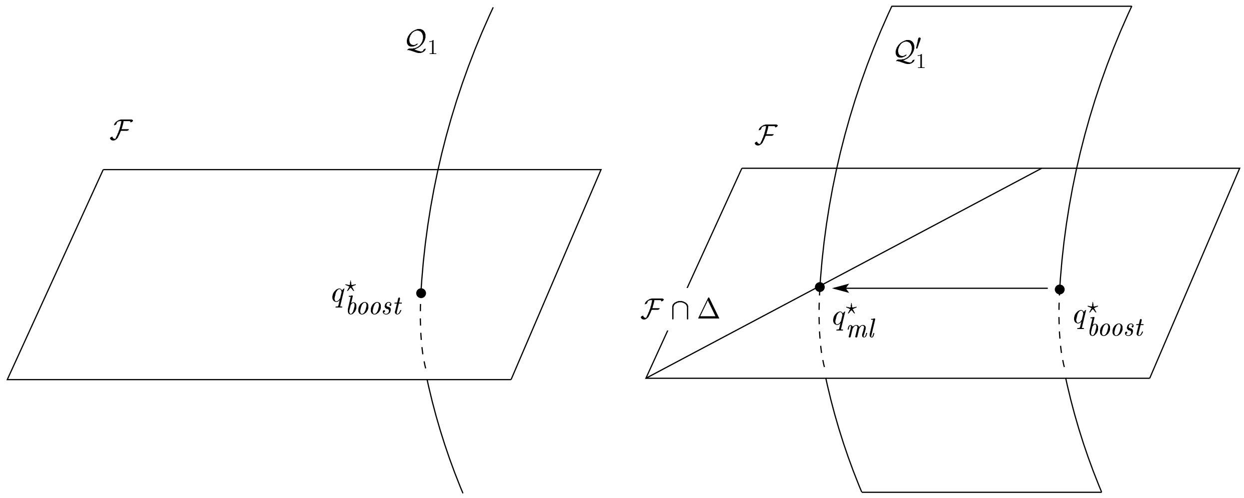

Figure 1: Geometric view of duality. Minimizing the exponential loss finds the member of $\mathcal{Q}_1$ that intersects the feasible set of measures satisfying the moment constraints (left). When we impose the additional constraint that each conditional distribution $q_\lambda(y \mid x)$ must be normalized, we introduce a Lagrange multiplier for each training example x, giving a higher-dimensional family $\mathcal{Q}'_1$. By the duality theorem, projecting the exponential loss solution onto the intersection of the feasible set with the simplex of conditional probabilities, $\mathcal{F} \cap \Delta$, we obtain the maximum likelihood solution. In many practical cases this projection is obtained by simply normalizing by a constant, resulting in an identical model.

Proposition 3.1. *Suppose that $D(\widetilde{p}, q_0) < \infty$. Then $q^\star_{boost}$ and $q^\star_{ml}$ exist, are unique, and satisfy*

$$q^\star_{boost} = \arg\min_{p \in \mathcal{F}} D(p, q_0) = \arg\min_{q \in \overline{\mathcal{Q}_1}} D(\widetilde{p}, q) \tag{3.21}$$

$$q^\star_{ml} = \arg\min_{p \in \mathcal{F} \cap \Delta} D(p, q_0) = \arg\min_{q \in \overline{\mathcal{Q}_2}} D(\widetilde{p}, q) \tag{3.22}$$

Moreover, $q^\star_{ml}$ is computed in terms of $q^\star_{boost}$ as $q^\star_{ml} = \arg\min_{p \in \mathcal{F} \cap \Delta} D(p, q^\star_{boost})$.

This result has a simple geometric interpretation. The unnormalized exponential family $\mathcal{Q}_1$ intersects the feasible set of measures $\mathcal{F}$ satisfying the constraints (3.2) at a single point. The algorithms presented in [3] determine this point, which is the exponential loss solution $q^\star_{boost} = \arg\min_{q \in \overline{\mathcal{Q}_1}} D(\widetilde{p}, q)$ (see Figure 1, left).

On the other hand, maximum conditional likelihood estimation for an exponential model with the same features is equivalent to the problem $q^\star_{ml} = \arg\min_{q \in \overline{\mathcal{Q}'_1}} D(\widetilde{p}, q)$ where $\mathcal{Q}'_1$ is the exponential family with additional Lagrange multipliers, one for each normalization constraint. The feasible set for this problem is $\mathcal{F} \cap \Delta$. Since $\mathcal{F} \cap \Delta \subset \mathcal{F}$, by the Pythagorean equality we have that $q^\star_{ml} = \arg\min_{p \in \mathcal{F} \cap \Delta} D(p, q^\star_{boost})$ (see Figure 1, right).

IV. Regularization

Minimizing the exponential loss or the log-loss on real data often fails to produce finite parameters. Specifically, this happens when for some feature f_j

$$
\begin{aligned}
& f_j(x,y) - f_j(x,\widetilde{y}(x)) \geq 0 \text{ for all } y \text{ and } x \text{ with } \widetilde{p}(x) > 0 \\
\text{or} \quad & f_j(x,y) - f_j(x,\widetilde{y}(x)) \leq 0 \text{ for all } y \text{ and } x \text{ with } \widetilde{p}(x) > 0
\end{aligned}
\tag{4.1}
$$

This is especially harmful since often the features for which (4.1) holds are the most important for the purpose of discrimination. The parallel update in [3] breaks down in such cases, resulting in parameters going to ∞ or $-\infty$. On the other hand, iterative scaling algorithms work in principle for such features. In practice however, either the parameters λ need to be artificially capped or the features need to be thrown out altogether, resulting in a partial and less discriminating set of features. Of course, even when (4.1) does not hold, models trained by maximizing likelihood or minimizing exponential loss can overfit the training data. The standard regularization technique in the case of maximum likelihood employs parameter priors in a Bayesian framework.

In terms of convex duality, a parameter prior for the dual problem corresponds to a "potential" on the constraint values in the primal problem. The case of a Gaussian prior on λ, for example, corresponds to a quadratic potential on the constraint values in the primal problem. Using this correspondence, the connection between boosting and maximum likelihood presented in the previous section indicates how to regularize AdaBoost using Bayesian MAP estimation for unnormalized models, as explained below.

We now consider primal problems over (p,c) where $p \in \mathcal{M}$ and $c \in \mathbb{R}^m$, where c is a parameter vector that relaxes the original constraints. Define $\mathcal{F}(\widetilde{p},f,c) \subset \mathcal{M}$ as

$$
\mathcal{F}(\widetilde{p},f,c) \quad = \quad \left\{ p \in \mathcal{M} \;\middle|\; \sum_x \widetilde{p}(x) \sum_y p(y\,|\,x)\, \big(f_j(x,y) - E_{\widetilde{p}}[f_j\,|\,x]\big) = c_j \right\}
\tag{4.2}
$$

and consider the primal problem $P_{1,\text{reg}}$ given by

$$
\begin{array}{lll}
(P_{1,\text{reg}}) & \textit{minimize} & D(p,q_0) + U(c) \\
& \textit{subject to} & p \in \mathcal{F}(\widetilde{p},f,c)
\end{array}
$$

where $U : \mathbb{R}^m \to \mathbb{R}$ is a convex function whose minimum is at $\mathbf{0}$.

To derive the dual problem, the Lagrangian is calculated as $\mathcal{L}(p,c,\lambda) = \mathcal{L}(p,\lambda) + U(c)$ and the dual function is then given by $h_{1,\text{reg}}(\lambda) = h_1(\lambda) + U^*(\lambda)$ where $U^*(\lambda)$ is the convex conjugate of U. For a quadratic penalty $U(c) = \sum_j \frac{1}{2}\sigma_j^2\, c_j^2$, we have $U^*(\lambda) = -\sum_j \frac{1}{2}\sigma_j^{-2}\,\lambda_j^2$ and the dual function becomes

$$
h_{1,\text{reg}}(\lambda) \quad = \quad -\sum_x \widetilde{p}(x) \sum_y q_0(y\,|\,x)\, e^{\sum_j \lambda_j\,(f_j(x,y) - f_j(x,\tilde{y}(x)))} - \sum_j \frac{\lambda_j^2}{2\sigma_j^2}
\tag{4.3}
$$

A sequential update rule for (4.3) incurs the small additional cost of solving a nonlinear equation by Newton's method every iteration. See [2] for a discussion of this technique in the context of exponential models in statistical language modeling.

	Unregularized			*Regularized*		
Data	$\ell_{train}(q_1)$	$\ell_{test}(q_1)$	$\epsilon_{test}(q_1)$	$\ell_{train}(q_2)$	$\ell_{test}(q_2)$	$\epsilon_{test}(q_2)$
Promoters	-0.29	-0.60	0.28	-0.32	-0.50	0.26
Iris	-0.29	-1.16	0.21	-0.10	-0.20	0.09
Sonar	-0.22	-0.58	0.25	-0.26	-0.48	0.19
Glass	-0.82	-0.90	0.36	-0.84	-0.90	0.36
Ionosphere	-0.18	-0.36	0.13	-0.21	-0.28	0.10
Hepatitis	-0.28	-0.42	0.19	-0.28	-0.39	0.19
Breast Cancer Wisconsin	-0.12	-0.14	0.04	-0.12	-0.14	0.04
Pima-Indians	-0.48	-0.53	0.26	-0.48	-0.52	0.25

Table 1: Comparison of unregularized to regularized boosting. For both the regularized and unregularized cases, the first column shows training log-likelihood, the second column shows test log-likelihood, and the third column shows test error rate. Regularization reduces error rate in some cases while it consistently improves the test set log-likelihood measure on all datasets. All entries were averaged using 10-fold cross validation.

V. Experiments

We performed experiments on some of the UCI datasets [1] in order to investigate the relationship between boosting and maximum likelihood empirically. The weak learner was `FindAttrTest` as described in [7], and the training set consisted of a randomly chosen 90% of the data. Table 1 shows experiments with regularized boosting. Two boosting models are compared. The first model q_1 was trained for 10 features generated by `FindAttrTest`, excluding features satisfying condition (4.1). Training was carried out using the parallel update method described in [3]. The second model, q_2, was trained using the exponential loss with quadratic regularization. The performance was measured using the conditional log-likelihood of the (normalized) models over the training and test set, denoted ℓ_{train} and ℓ_{test}, as well as using the test error rate ϵ_{test}. The table entries were averaged by 10-fold cross validation.

For the weak learner `FindAttrTest`, only the Iris dataset produced features that satisfy (4.1). On average, 4 out of the 10 features were removed. As the flexibility of the weak learner is increased, (4.1) is expected to hold more often. On this dataset regularization improves both the test set log-likelihood and error rate considerably. In datasets where q_1 shows significant overfitting, regularization improves both the log-likelihood measure and the error rate. In cases of little overfitting (according to the log-likelihood measure), regularization only improves the test set log-likelihood at the expense of the training set log-likelihood, however without affecting test set error.

Next we performed a set of experiments to test how much $q^\star_{boost}$ differs from $q^\star_{ml}$, where the boosting model is normalized to form a conditional probability distribution; see equation (3.17). For different experiments, `FindAttrTest` generated a different number of features (10–100), and the training set was selected randomly. The plots in Figure 2 show for different datasets the relationship between $\ell_{train}(q^\star_{ml})$ and $\ell_{train}(q^\star_{boost})$ as well as between $\ell_{train}(q^\star_{ml})$ and $D_{train}(q^\star_{ml}, q^\star_{boost})$. The trend is the same in each data set: as the number of features increases so that the training data is more closely fit ($\ell_{train}(q_{ml}) \longrightarrow 0$), the boosting and maximum likelihood models become more similar, as measured by the KL divergence.

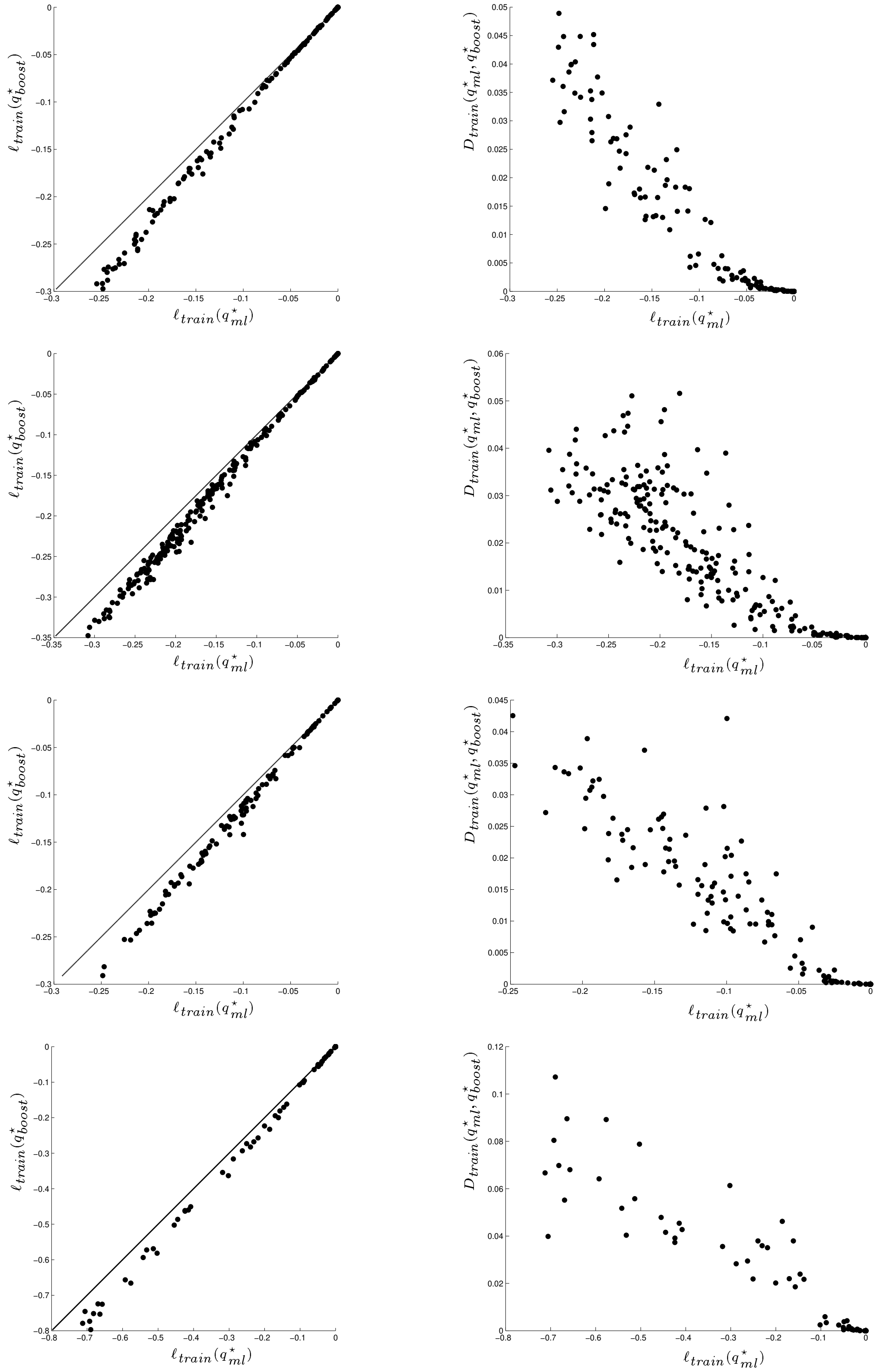

Figure 2: Comparison of AdaBoost and maximum likelihood on four UCI datasets: Hepatitis (top row), Promoters (second row), Sonar (third row) and Glass (bottom row). The left column compares $\ell_{train}(q^\star_{ml})$ to $\ell_{train}(q^\star_{boost})$, and the right column compares $\ell_{train}(q^\star_{ml})$ to $D_{train}(q^\star_{ml}, q^\star_{boost})$.

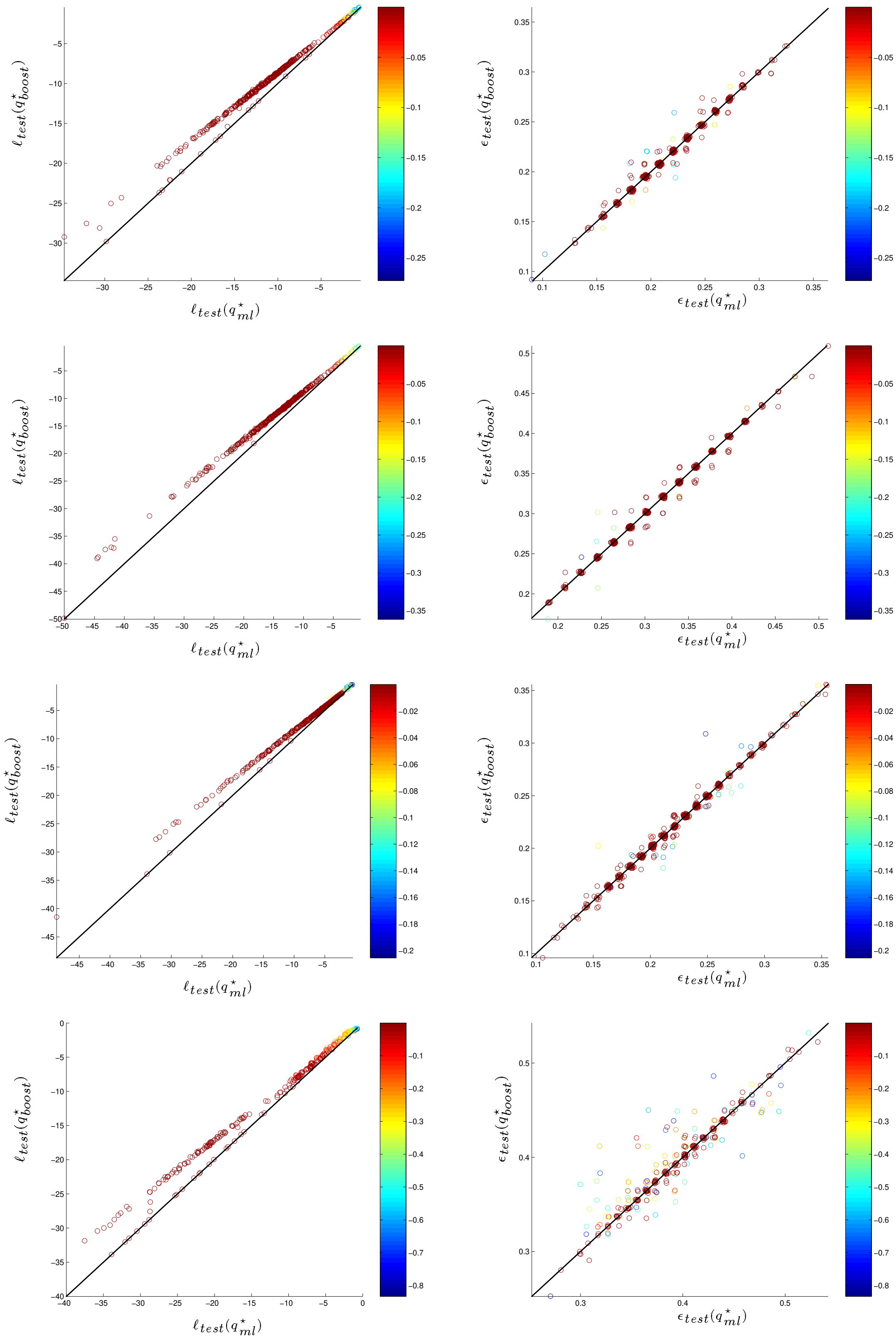

Figure 3: Comparison of AdaBoost and maximum likelihood on the same UCI datasets as in the previous figure. The left column compares the test likelihoods, $\ell_{test}(q^\star_{ml})$ to $\ell_{test}(q^\star_{boost})$, and the right column compares test error rates, $\epsilon_{test}(q^\star_{ml})$ to $\epsilon_{test}(q^\star_{boost})$. In each plot, the color represents the *training* likelihood $\ell_{train}(q^\star_{ml})$; red corresponds to fitting the training data well.

	ρ	α
Promoters	0.99	1.14
Hepatitis	0.99	1.14
Sonar	0.99	1.16
Glass	0.99	1.11

Table 2: Correlation coefficient ρ and linear regression slope α for different datasets, indicating a strong linear relationship with slope close to one.

The plots in Figure 3 show the relationship between the test set log-likelihoods, $\ell_{test}(q^\star_{ml})$ to $\ell_{test}(q^\star_{boost})$, together with the test set error rates $\epsilon_{test}(q^\star_{ml})$ and $\epsilon_{test}(q^\star_{boost})$. In these figures the testing set was chosen to be 50% of the total data. The color represents the *training* data log-likelihood, $\ell_{train}(q^\star_{ml})$, with the color red corresponding to high likelihood. In order to indicate the number of points at each error rate, each circle was shifted by a small random value to avoid points falling on top of each other.

The likelihood plots show a clear linear trend. While the plots in Figure 2 indicate that $\ell_{train}(q^\star_{ml}) > \ell_{train}(q^\star_{boost})$, as expected, on the test data the linear trend is reversed, so that $\ell_{test}(q^\star_{ml}) < \ell_{test}(q^\star_{boost})$. This suggests that the boosting model is smoother and less prone to overfitting. The duality result shows that this is because the model is less constrained due to the lack of normalization constraints, and therefore has higher entropy than the maximum likelihood model. However, as $\ell(q^\star_{ml}) \longrightarrow 0$, the two models come to agree. Table 2 gives the correlation coefficient ρ between $\ell(q^\star_{ml})$ and $\ell(q^\star_{boost})$, as well as linear regression slope coefficient α. It is easy to show (see appendix D) that for any exponential model $q_\lambda \in \mathcal{Q}_2$,

$$D_{train}(q^\star_{ml}, q_\lambda) \quad = \quad \ell(q^\star_{ml}) - \ell(q_\lambda). \tag{5.1}$$

By taking $q_\lambda = q^\star_{boost}$ it is seen that as the difference between $\ell(q^\star_{ml})$ and $\ell(q^\star_{boost})$ gets smaller, the divergence between the two models also gets smaller. Furthermore, since the correlation coefficient ρ is close to 1, we can use the approximation $\ell(q^\star_{boost}) \approx \alpha\, \ell(q^\star_{ml})$ to obtain

$$D_{train}(q^\star_{ml}, q^\star_{boost}) \quad \approx \quad (1-\alpha)\, \ell(q^\star_{ml}) \tag{5.2}$$

The results are consistent with the theoretical analysis. As the number of features is increased so that the training data is fit more closely, the model matches the empirical distribution $\widetilde{p}$ and the normalizing term $Z_\lambda(x)$ becomes a constant. In this case, normalizing the boosting model $q^\star_{boost}$ does not violate the constraints, and results in the maximum likelihood model.

Acknowledgments

We thank Michael Collins, Michael Jordan, Andrew Ng, Fernando Pereira, Rob Schapire, and Yair Weiss for helpful comments on an early version of this paper. Part of this work was carried out while the second author was visiting the Department of Statistics, University of California at Berkeley.

References

[1] C.L. Blake and C.J. Merz. The UCI repository of machine learning databases. University of California, Irvine, Department of Information and Computer Sciences, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[2] S. Chen and R. Rosenfeld. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1), 2000.

[3] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, to appear.

[4] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 1997.

[5] S. Della Pietra, V. Della Pietra, and J. Lafferty. Duality and auxiliary functions for Bregman distances. Technical Report CMU-CS-01-109, Carnegie Mellon University, 2001.

[6] N. Duffy and D. Helmbold. Potential boosters? In *Advances in Neural Information Processing Systems (NIPS)*, volume 12, 2000.

[7] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, 1996.

[8] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 2000.

[9] J. Kivinen and M. K. Warmuth. Boosting as entropy projection. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 1999.

[10] J. Lafferty. Additive models, boosting, and inference for generalized divergences. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 1999.

[11] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, 1999.

[12] J. A. O'Sullivan. Alternating minimization algorithms: From Blahut-Arimoto to Expectation-Maximization. In A. Vardy, editor, *Codes, Curves, and Signals: Common Threads in Communications*, pages 173–192, 1998.

Appendix

In Appendix A and B we rederive the update rules from [3] in our notation. These update rules are derived by minimizing an auxiliary function that bounds from above the reduction in loss. See [3] for the definition of an auxiliary function and proofs that the functions in A and B are indeed auxiliary functions. Appendix A deals with parallel updates and B deals with sequential updates. In Appendix C the regularized formulation is shown in detail and a sequential update rule is derived. Appendix D contains a proof for (5.1).

A. Derivation of the Parallel Updates

Let $x \in \mathcal{X}$ be an example in the training set, which is of size n, and let $\tilde{y}$ be its label. $\mathcal{Y}$ is the set of all possible labels. At a given iteration λ_j denotes the j-th parameter of the model, and $\lambda_j + \Delta\lambda_j$ the parameter at the following iteration.

A.1. Exponential Loss

The objective is to minimize $\mathcal{E}_{exp}(\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}) - \mathcal{E}_{exp}(\boldsymbol{\lambda})$. In the following $h_j(x,y) = f_j(x,y) - f_j(x,\tilde{y})$, $q_{\boldsymbol{\lambda}}(y|x) = e^{\sum_j \lambda_j h_j(x,y)}$, $s_j(x,y) = \text{sign}(h_j(x,y))$, $M = \max_{i,y} \sum_j |h_j(x_i,y)|$, $\omega_{i,y} = 1 - \sum_j \frac{|h_j(x_i,y)|}{M}$. By Jensen's inequality applied to e^x we have

$$\begin{aligned}
\mathcal{E}_{exp}(\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}) - \mathcal{E}_{exp}(\boldsymbol{\lambda}) &= \sum_i \sum_y e^{\sum_j (\lambda_j + \Delta\lambda_j) h_j(x_i,y)} - \sum_i \sum_y e^{\sum_j \lambda_j h_j(x_i,y)} \\
&= \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) e^{\sum_j \Delta\lambda_j \frac{|h_j(x_i,y)|}{M} s_j(x_i,y)\, M} - \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \\
&\leq \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left(\sum_j \frac{|h_j(x_i,y)|}{M} e^{\Delta\lambda_j s_j(x_i,y) M} + \omega_{i,y} - 1 \right) \\
&\stackrel{\text{def}}{=} \mathcal{A}(\Delta\boldsymbol{\lambda}, \boldsymbol{\lambda}). \qquad \text{(A.1)}
\end{aligned}$$

We proceed by finding the stationary point of the auxiliary function with respect to $\Delta\lambda_j$:

$$\begin{aligned}
0 = \frac{\partial \mathcal{A}}{\partial \Delta\lambda_j} &= -\sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) h_j(x_i,y) e^{\Delta\lambda_j s_j(x_i,y)\, M} \\
&= -\sum_y \sum_{i:s_j(x_i,y)=+1} q_{\boldsymbol{\lambda}}(y|x_i) h_j(x_i,y) e^{\Delta\lambda_j M} - \sum_y \sum_{i:s_j(x_i,y)=-1} q_{\boldsymbol{\lambda}}(y|x_i) h_j(x_i,y) e^{-\Delta\lambda_j M} \\
\Rightarrow\ e^{2M\Delta\lambda_j} \sum_y \sum_{i:s_j(x_i,y)=+1} h_j(x_i,y) q_{\boldsymbol{\lambda}}(y|x_i) &= \sum_y \sum_{i:s_j(x_i,y)=-1} |h_j(x_i,y)| q_{\boldsymbol{\lambda}}(y|x_i) \\
\Rightarrow\ \Delta\lambda_j &= \frac{1}{2M} \log \left(\frac{\sum_y \sum_{i:s_j(x_i,y)=-1} |h_j(x_i,y)| q_{\boldsymbol{\lambda}}(y|x_i)}{\sum_y \sum_{i:s_j(x_i,y)=+1} |h_j(x_i,y)| q_{\boldsymbol{\lambda}}(y|x_i)} \right)
\end{aligned}$$

A.2. ML exponential model

For the normalized case, the objective is to maximize the likelihood or minimize the log-loss. In this section, the previous notation remains except for $q_{\boldsymbol{\lambda}}(y|x) = \frac{e^{\sum_j \lambda_j h_j(x,y)}}{\sum_y e^{\sum_j \lambda_j h_j(x,y)}}$. The log-likelihood is

$$\ell(\boldsymbol{\lambda}) = \sum_i \log \frac{e^{\sum_j \lambda_j f_j(x_i,y_i)}}{\sum_y e^{\sum_j \lambda_j f_j(x_i,y_i)}} = -\sum_i \log \sum_y e^{\sum_j \lambda_j (f_j(x_i,y) - f_j(x_i,y_i))}$$

The loss that we want to minimize is then

$$
\begin{aligned}
\ell(\boldsymbol{\lambda}) - \ell(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}) &= \sum_i \log \frac{\sum_y e^{\sum_j (\lambda_j + \Delta\lambda_j)(f_j(x_i,y) - f_j(x_i,y_i))}}{\sum_y e^{\sum_j \lambda_j (f_j(x_i,y) - f_j(x_i,y_i))}} \\
&= \sum_i \log \frac{\sum_y e^{\sum_j (\lambda_j + \Delta\lambda_j) h_j(x_i,y)}}{\sum_y e^{\sum_j \lambda_j h_j(x_i,y)}} \\
&= \sum_i \log \sum_y q_{\boldsymbol{\lambda}}(y|x_i) e^{\sum_j \Delta\lambda_j h_j(x_i,y)} \\
&\leq \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) e^{\sum_j \Delta\lambda_j h_j(x_i,y)} - n && \text{(A.2)} \\
&= \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) e^{\sum_j \Delta\lambda_j \frac{|h_j(x_i,y)|}{M} s_j(x_i,y)\, M} - n \\
&\leq \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left(\sum_j \frac{|h_j(x_i,y)|}{M} e^{\Delta\lambda_j s_j(x_i,y)\, M} + \omega_{i,y} \right) - n && \text{(A.3)} \\
&\stackrel{\text{def}}{=} \mathcal{A}(\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda}) && \text{(A.4)}
\end{aligned}
$$

where in (A.2) we used the inequality $\log x \leq x - 1$ and in (A.3) we used Jensen's inequality. The derivative of (A.3) with respect to $\Delta\boldsymbol{\lambda}$ will be identical to the derivative of (A.1) and so the log-loss update rule will be identical to the exponential loss update rule, but with $q_{\boldsymbol{\lambda}}(y|x)$ representing a normalized exponential model.

B. Derivation of the Sequential Updates

The setup for sequential updates is similar to that for parallel updates, but now only one parameter gets updated in each step, while the rest are held fixed.

B.1. Exponential Loss

We now assume that that only λ_k gets updated. We also assume (with no loss of generality) that each feature takes values in $[0,1]$, making $h_k(x_i,y) \in [-1,1]$.

$$
\begin{aligned}
\mathcal{E}_{exp}(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}) - \mathcal{E}_{exp}(\boldsymbol{\lambda}) &= \sum_i \sum_y e^{\sum_j \lambda_j h_j(x_i,y) + \Delta\lambda_k h_k(x_i,y)} - \sum_i \sum_y e^{\sum_j \lambda_j h_j(x_i,y)} \\
&= \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left(e^{\left(\frac{1+h_k(x_i,y)}{2}\right)\Delta\lambda_k + \left(\frac{1-h_k(x_i,y)}{2}\right)(-\Delta\lambda_k)} - 1 \right) && \text{(B.1)} \\
&\leq \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left(\frac{1+h_k(x_i,y)}{2} e^{\Delta\lambda_k} + \frac{1-h_k(x_i,y)}{2} e^{-\Delta\lambda_k} - 1 \right) \\
&\stackrel{\text{def}}{=} \mathcal{A}(\boldsymbol{\lambda}, \Delta\lambda_k) && \text{(B.2)}
\end{aligned}
$$

The stationary point of $\mathcal{A}$ (with respect to $\Delta\lambda_k$) is

$$
\begin{aligned}
0 &= \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left(\frac{1 + h_k(x_i, y)}{2} e^{\Delta\lambda_k} + \frac{h_k(x_i, y) - 1}{2} e^{-\Delta\lambda_k} \right) \\
\Rightarrow \quad & e^{2\Delta\lambda_k} \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i)(1 + h_k(x_i, y)) = \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i)(1 - h_k(x_i, y)) \\
\Rightarrow \quad \Delta\lambda_k &= \frac{1}{2} \log \left(\frac{\sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i)(1 - h_k(x_i, y))}{\sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i)(1 + h_k(x_i, y))} \right)
\end{aligned}
$$

B.2. Log-Loss

$$
\begin{aligned}
\ell(\boldsymbol{\lambda}) - \ell(\boldsymbol{\lambda} + \Delta\lambda_k) &= \sum_i \log \frac{\sum_y e^{\sum_j \lambda_j h_j(x_i, y) + \Delta\lambda_k h_k(x_i, y)}}{\sum_y e^{\sum_j \lambda_j h_j(x_i, y)}} = \sum_i \log \sum_y q_{\boldsymbol{\lambda}}(y|x_i) e^{\Delta\lambda_k h_k(x_i, y)} \\
&\leq \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) e^{\Delta\lambda_k h_k(x_i, y)} - n \qquad \text{(B.3)}
\end{aligned}
$$

Equation (B.3) is the same as (B.1), except that $q_{\boldsymbol{\lambda}}$ is now the normalized model. This leads to exactly the same form of update rule as in the previous subsection.

C. Regularized Loss Functions

C.1. Problem setting

$$
\begin{aligned}
&\textit{Minimize} \qquad D(p, q_0) + U(c) = \sum_x \tilde{p}(x) \sum_y p(y|x) \left(\log \frac{p(y|x)}{q_0(y|x)} - 1 \right) + U(c) \\
&\textit{subject to} \qquad f_j(p) = \sum_{x,y} \tilde{p}(x) p(y|x) h_j(x, y) = c_j, \quad j = 1, \ldots, m
\end{aligned}
$$

where $c \in \mathbb{R}^m$ and $U : \mathbb{R}^m \to \mathbb{R}$ is a convex function whose minimum is at $\mathbf{0}$. The Lagrangian turns out to be

$$
\mathcal{L}(p, c, \lambda) = \sum_x \tilde{p}(x) \sum_y p(y|x) \left(\log \frac{p(y|x)}{q_0(y|x)} - 1 - \langle \lambda, h(x, y) \rangle \right) + U(c).
$$

We will derive the dual problem for $U(c) = \sum_i \frac{1}{2} \sigma_i^2 c_i^2$. The convex conjugate U^* is

$$
\begin{aligned}
U^*(\lambda) &\stackrel{\text{def}}{=} \inf_c \sum_i \lambda_i c_i + U(c) = \inf_c \sum_i \lambda_i c_i + \sum_i \frac{1}{2} \sigma_i^2 c_i^2 \qquad \text{(C.1)} \\
0 &= \lambda_i + \sigma_i^2 c_i \quad \Rightarrow \quad c_i = -\frac{\lambda_i}{\sigma_i^2} \\
U^*(\lambda) &= -\sum_i \frac{\lambda_i^2}{\sigma_i^2} + \sum_i \frac{1}{2} \sigma_i^2 \frac{\lambda_i^2}{\sigma_i^4} = -\sum_i \frac{\lambda_i^2}{2\sigma_i^2}
\end{aligned}
$$

The dual problem for the exponential loss is then

$$
\begin{aligned}
\lambda^{\star} &= \arg\max_{\lambda} h_{1,reg}(\lambda) \\
&= \arg\max_{\lambda} -\sum_x \tilde{p}(x) \sum_y q_0(y|x) e^{\sum_j \lambda_j h_j(x,y)} + U^*(\lambda) \qquad \text{(C.2)} \\
&= \arg\min_{\lambda} \sum_x \tilde{p}(x) \sum_y q_0(y|x) e^{\sum_j \lambda_j h_j(x,y)} + \sum_j \frac{\lambda_j^2}{2\sigma_j^2}
\end{aligned}
$$

Next, a sequential update rule for the exponential loss is derived.

C.2. Exponential Loss–Sequential update rule

As before, $q_0 = 1$ and we replace $\frac{1}{2\sigma_k^2}$ by β.

$$
\begin{aligned}
&\mathcal{E}_{exp}(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}) - \mathcal{E}_{exp}(\boldsymbol{\lambda}) \\
&= \sum_i \sum_y \left(e^{\sum_j \lambda_j h_j(x_i,y) + \Delta\lambda_k h_k(x_i,y)} - e^{\sum_j \lambda_j h_j(x_i,y)} \right) + \beta(\lambda_k + \Delta\lambda_k)^2 - \beta\lambda_k^2 \qquad \text{(C.3)} \\
&= \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left(e^{\left(\frac{1+h_k(x_i,y)}{2}\right)\Delta\lambda_k + \left(\frac{1-h_k(x_i,y)}{2}\right)(-\Delta\lambda_k)} - 1 \right) \\
&\quad + 2\beta\, \lambda_k\, \Delta\lambda_k + \beta\, \Delta\lambda_k^2 \\
&\leq \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left(\frac{1+h_k(x_i,y)}{2} e^{\Delta\lambda_k} + \frac{1-h_k(x_i,y)}{2} e^{-\Delta\lambda_k} - 1 \right) \\
&\quad + 2\beta\, \lambda_k\, \Delta\lambda_k + \beta\, \Delta\lambda_k^2 \\
&\stackrel{\text{def}}{=} \mathcal{A}(\boldsymbol{\lambda}, \Delta\lambda_k)
\end{aligned}
$$

The stationary point will be at the solution of the following equation

$$
0 = \frac{\partial \mathcal{A}}{\partial \Delta\lambda_k} = \frac{1}{2} \sum_i \sum_y q_{\boldsymbol{\lambda}}(y|x_i) \left((1 + h_k(x_i,y)) e^{\Delta\lambda_k} + (h_k(x_i,y) - 1) e^{-\Delta\lambda_k} \right) + 2\beta\,(\lambda_k + \Delta\lambda_k).
$$

A solution to this equation can be found with Newton's method. Since the second derivative $\frac{\partial^2 \mathcal{A}}{\partial \Delta\lambda_k^2}$ is positive, the auxiliary function is strictly convex and Newton's method will converge.

D. Divergence Between Exponential Models

The log-likelihood of $q_{\boldsymbol{\lambda}}$ is:

$$
\begin{aligned}
\ell(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_i \log \frac{e^{\sum_j \lambda_j f_j(x_i,y_i)}}{Z_{\boldsymbol{\lambda},i}} = \frac{1}{n} \sum_j \lambda_j \sum_i f_j(x_i,y_i) - \frac{1}{n} \sum_i \log Z_{\boldsymbol{\lambda},i} \\
&= \sum_j \lambda_j E_{\tilde{p}}[f_j] - \frac{1}{n} \sum_i \log Z_{\boldsymbol{\lambda},i}
\end{aligned}
$$

$$
\begin{aligned}
D(q_{\boldsymbol{\lambda}_1}, q_{\boldsymbol{\lambda}_2}) &= \frac{1}{n}\sum_i \sum_y q_{\boldsymbol{\lambda}_1}(y|x_i) \log \frac{q_{\boldsymbol{\lambda}_2}(y|x_i)}{q_{\boldsymbol{\lambda}_1}(y|x_i)} \\
&= \frac{1}{n}\sum_i \sum_y q_{\boldsymbol{\lambda}_1}(y|x_i) \left(\log \frac{Z_{\boldsymbol{\lambda}_2,i}}{Z_{\boldsymbol{\lambda}_1,i}} + \log \frac{e^{\sum_j \lambda_{1,j} f_j(x_i,y)}}{e^{\sum_j \lambda_{2,j} f_j(x_i,y)}} \right) \\
&= \frac{1}{n}\sum_i \log \frac{Z_{\boldsymbol{\lambda}_2,i}}{Z_{\boldsymbol{\lambda}_1,i}} + \frac{1}{n}\sum_i \sum_y q_{\boldsymbol{\lambda}_1}(y|x_i) \sum_j f_j(x_i,y)(\lambda_{1,j} - \lambda_{2,j}) \\
&= \frac{1}{n}\sum_i \log \frac{Z_{\boldsymbol{\lambda}_2,i}}{Z_{\boldsymbol{\lambda}_1,i}} + \sum_j (\lambda_{1,j} - \lambda_{2,j}) E_{q_{\boldsymbol{\lambda}_1}}[f_j]
\end{aligned}
$$

This corresponds to the fact that the KL divergence between exponential models is the Bregman distance, with respect to the cumulant function, between the natural parameters.

If $q_{\boldsymbol{\lambda}_1}$ is $q_{\boldsymbol{\lambda}}^{ml}$, since the moment constraints are satisfied, it follows that

$$
D(q_{\boldsymbol{\lambda}}^{ml}, q_{\boldsymbol{\lambda}}) = \frac{1}{n}\sum_i \log \frac{Z_{\boldsymbol{\lambda},i}}{Z_{\boldsymbol{\lambda},i}^{ml}} + \sum_j (\lambda_j^{ml} - \lambda_j) E_{\tilde{p}}[f_j] = \ell(\boldsymbol{\lambda}^{ml}) - \ell(\boldsymbol{\lambda})
$$