Guess-Averse Loss Functions For Cost-Sensitive Multiclass Boosting

Oscar Beijbom* OBEIJBOM@UCSD.EDU
Mohammad Saberian* SABERIAN@UCSD.EDU
David Kriegman KRIEGMAN@UCSD.EDU
Nuno Vasconcelos NVASCONCELOS@UCSD.EDU
University of California, San Diego, 9500 Gilman Drive, 92093 La Jolla, CA

Abstract

Cost-sensitive multiclass classification has recently acquired significance in several applications, through the introduction of multiclass datasets with well-defined misclassification costs. The design of classification algorithms for this setting is considered. It is argued that the unreliable performance of current algorithms is due to the inability of the underlying loss functions to enforce a certain fundamental underlying property. This property, denoted guess-aversion, is that the loss should encourage correct classifications over the arbitrary guessing that ensues when all classes are equally scored by the classifier. While guess-aversion holds trivially for binary classification, this is not true in the multiclass setting. A new family of cost-sensitive guess-averse loss functions is derived, and used to design new cost-sensitive multiclass boosting algorithms, denoted GEL- and GLL-MCBoost. Extensive experiments demonstrate (1) the importance of guess-aversion and (2) that the GLL loss function outperforms other loss functions for multiclass boosting.

1. Introduction

Boosting methods play an important role in classification problems. A prominent example is the Viola-Jones face detection algorithm (Viola & Jones, 2001c) which utilizes a classifier cascade designed with AdaBoost (Freund & Schapire, 1996). Other, more recent, applications include pedestrian and object detection (Dollár et al., 2010; Torralba et al., 2004). However, while boosting classifiers are traditionally trained to minimize the error rate, in many applications some errors are more costly than others. For example, in ImageNet (Deng et al., 2009) the cost of misclassification between, e.g., 'Mountain Gorilla' and 'Western Lowland Gorilla' is lower than, say, 'Giraffe' and 'Chair'. This leads to the more general problem of cost-sensitive multiclass classification, where misclassification costs are specified through a cost matrix. For this problem, current cost-sensitive extensions of multiclass boosting methods, such as AdaBoost.M2 (Freund & Schapire, 1996), frequently result in sub-optimal classifiers (Lozano & Abe, 2008). The problem of binary cost-sensitive learning has received significant attention in the last decade. Early work by (Elkan, 2001) established some fundamental results for learning algorithms that operate in this scenario. This work recommends the explicit computation of class-conditional probabilities from the training set, and the use of these probabilities to obtain optimal decision boundaries, using Bayes rule. A popular example of this strategy is the MetaCost algorithm (Domingos, 1999), which estimates posterior class probabilities through bootstrapping of the training data. The major difficulty of this approach is to obtain accurate conditional probability estimates. In fact, (Masnadi-Shirazi & Vasconcelos, 2011) analyzed this scheme and noted that large-margin methods, such as boosting, only accurately predict class-conditional probabilities close to the cost-*in*sensitive decision boundary. In result, the accuracy of probability estimates tends to be low in the neighborhood of the target cost-sensitive boundary, leading to sub-optimal cost-sensitive decisions. Another approach is to modify the loss function of the boosting algorithm, to take the cost-matrix into account. This approach has received significant attention in the binary classification literature, see e.g. AdaCost (Fan et al., 1999), asymmetric-AdaBoost (Viola & Jones, 2001b), and the more general framework of (Masnadi-Shirazi & Vasconcelos, 2011). These methods usually outperform those based on class-conditional probability estimates. In parallel with algorithmic developments, substantial theoretical work has been devoted to the characterization of different losses, particularly for the binary case. Important

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

*The authors assert joint authorship for this work.

loss properties include classification calibration (Bartlett et al., 2006; Scott, 2012) and margin maximization (Vapnik, 1999). When these properties hold, minimization of the loss function guarantees consistency with Bayes rule and bounds on generalization error, respectively.

For the multiclass scenario, cost-sensitive boosting has received less attention. (Abe et al., 2004) proposed GBSE, based on a combined scheme of instance-weighting, expansion of multiclass labels and gradient descent in functional space (Friedman, 1999; Mason et al., 2000). GBSE was shown to outperform MetaCost (Domingos, 1999), as well as Bagging (Breiman, 1996). Later, (Lozano & Abe, 2008) proposed a cost-sensitive multiclass boosting method, based on a family of p-norm cost functionals, that generalized and improved GBSE. Recently, (Wang, 2013) proposed MultiBoost, which is derived from a simple, yet intuitive, loss function. Alongside these algorithmic contributions, there have been important theoretical contributions on multiclass classification. Notably, (Tewari & Bartlett, 2007; Ramaswamy & Agarwal, 2012) generalized the concept of classification consistency to the multiclass scenario. Other concepts, such as margin maximization are still valid for cost-sensitive multiclass classification. However, losses with these properties do not always produce robust classifiers in practice.

In this work we argue that this is because in the multiclass setting, losses with all the above properties may fail to enforce a simple and intuitive property. This property, which we denote by guess-aversion, is that the loss should encourage correct classifications over the arbitrary guessing that ensues when all classes are equally scored by the classifier. We show that, while this property holds trivially for most binary and multiclass loss functions, this is not true for the cost-sensitive multiclass setting. In fact, loss functions that have lately become popular for cost-sensitive multiclass classification such as (Wang, 2013) do not exhibit it. We then derive a family of cost-sensitive guess-averse loss functions, and use them to derive cost-sensitive extensions of MCBoost (Saberian & Vasconcelos, 2011), denoted GEL- and GLL-MCBoost. Experiments on UCI data and a large-scale biological computer vision dataset show 1) the empirical importance of guess-aversion, and 2) that the GLL loss function outperforms alternative loss functions for cost-sensitive multiclass boosting.

Contributions: This work makes three main contributions: 1) introduces the concept of guess-averse classification losses and empirically demonstrates its importance, 2) proposes a family of guess-averse cost-sensitive multiclass losses, and 3) show that the GLL loss function outperforms alternative loss functions for cost-sensitive multiclass boosting. The MATLAB implementation of the proposed boosting algorithms, along with experimental details is available in supplementary material[1].

2. Problem Definition

A multiclass classifier $h(x)$ is a measurable mapping from an example $x \in \mathcal{X}$ to a class label $z \in \{1, \dots M\}$. This mapping is commonly of the form

$$h(x) = \text{argmax}_k \, S_k(x) \quad k = 1 \dots M, \tag{1}$$

where $S_k : \mathcal{X} \to \mathbb{R}$ is a real-valued function, that is denoted the score of class k for example x. $S_k(x)$ reflects the confidence of the classifier $h(x)$ in the assignment of x to class k. The performance of a classifier, $h(x)$, is evaluated by its classification risk

$$\mathcal{R}[h] \quad = \quad E\left\{C_{Z,h(X)}\right\} \tag{2}$$

where C is a $M \times M$ cost matrix which encodes the cost $C_{j,k} \geq 0$ of classifying an example from class j into class k. We always assume $C_{j,j} = 0 \, \forall j$ and $\sum_{k=1}^{M} C_{j,k} > 0, \forall j$. If $C_{j,k} = 1 \, \forall j \neq k$, the classification problem is said to be cost-insensitive. The *optimal* classifier, i.e. that of minimal risk $\mathcal{R}[h]$, implements Bayes decision rule

$$h^*(x) = \text{argmin}_k \sum_{j=1}^{M} \eta_j(x) C_{j,k}, \tag{3}$$

where

$$\eta_j(x) = P_{Z|X}(j|x) \quad j = 1 \dots M \tag{4}$$

is the posterior probability of class j (Lee et al., 2004). For classifier learning, the risk of (2) is approximated by the empirical risk

$$\widehat{\mathcal{R}}[h] \quad = \quad \frac{1}{n} \sum_{(x_i, z_i) \in \mathcal{D}} C_{z_i, h(x_i)}, \tag{5}$$

where $\mathcal{D} = \{(x_i, z_i)\}_1^n$ is a training set.

Modern learning algorithms, such as boosting, rely on a *surrogate loss function* $L[C, z, S(x)]$ of the cost matrix C, class-label z, and score function $S : \mathcal{X} \to \mathbb{R}^M$. The optimal score function

$$S^{L^*} = \text{argmin}_S \, \mathcal{R}_L[S], \tag{6}$$

minimizes the risk defined by this loss

$$\mathcal{R}_L[S] \quad = \quad E_{X,Z}\{L[C, z, S(x)]\} \tag{7}$$

which, in practice, is approximated by

$$\widehat{\mathcal{R}}_L[S] \quad = \quad \frac{1}{n} \sum_{i=1}^{n} L[C, z_i, S(x_i)]. \tag{8}$$

[1] A detailed reply to the issues raised by the reviewers of the original submission of the paper is also part of this material.

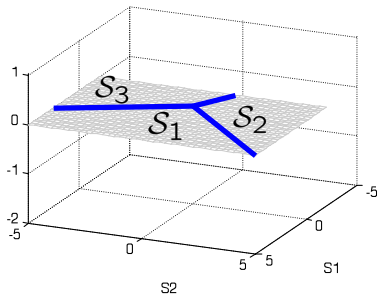


Figure 1. **Sets $\mathcal{S}_k$ for a 3-class classification problem.** The blue lines correspond to decision boundaries.

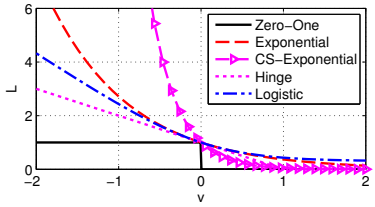


Figure 2. **Common binary loss functions.** The guess-averse property is trivially satisfied by these functions, including the cost-sensitive version of the exponential loss (Masnadi-Shirazi & Vasconcelos, 2011).

The choice of loss function $L(C, z, S(x))$ significantly impacts the classification performance of the optimal score function $S^{L^*}(x)$. Surrogate loss functions have been the subject of extensive research in the last decade (Bartlett et al., 2006; Vapnik, 1999; Tewari & Bartlett, 2007). A number of desirable properties, for both the binary and multiclass classification settings, have been identified including:

Classification Calibration: A loss L is classification calibrated if the use of $S^{L^*}(x)$ in (1) results in the same decision as the Bayes rule of (3) (Bartlett et al., 2006; Tewari & Bartlett, 2007).

Margin Maximizing: Given a training pair (x, z), a loss L is margin maximizing if the minimization of $\widehat{\mathcal{R}}_L$ results in the maximization of the margin

$$\mathcal{M}(z, S(x)) = S_z(x) - \max_{k \neq z} S_k(x). \tag{9}$$

It has been shown that margin maximizing losses have better generalization performance than losses without this property (Vapnik, 1999).

The importance of these conditions is well established for cost-insensitive binary classification. The problems of cost-sensitive and multiclass classification have, however, proven more elusive. This is particularly true for cost-sensitive multiclass classification. In our experiments, losses that satisfy all the conditions above frequently produce poor decision rules. We argue that this is because in the multiclass setting, losses with all the properties above can still lack a simple but important property. We discuss this property next.

3. Guess-averse losses

We start by defining the *support set,* or simply *support,* of class k.

Definition 1. *Support set of class k is set of all score vectors for which S_k is the largest score, i.e.*

$$\mathcal{S}_k = \left\{ S | S \in \mathbb{R}^M, S_k > S_j \ \forall j \neq k \right\}. \tag{10}$$

If an example x belongs to class z, then $\mathcal{S}_z$ is the set of scores $S(x)$ for which the decision rule of (1) assigns x to the correct class. For example Figure 1 shows supports $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$, a the 3-class classification problem[2].

We next define arbitrary guess points.

Definition 2. *A score vector, $S(x) \in \mathbb{R}^M$, is an arbitrary guess point if $S_k(x)$ is independent of k, i.e. $S_k(x) = S_1(x)\ \forall k$. The set of all arbitrary guess points is denoted $\mathcal{A}$.*

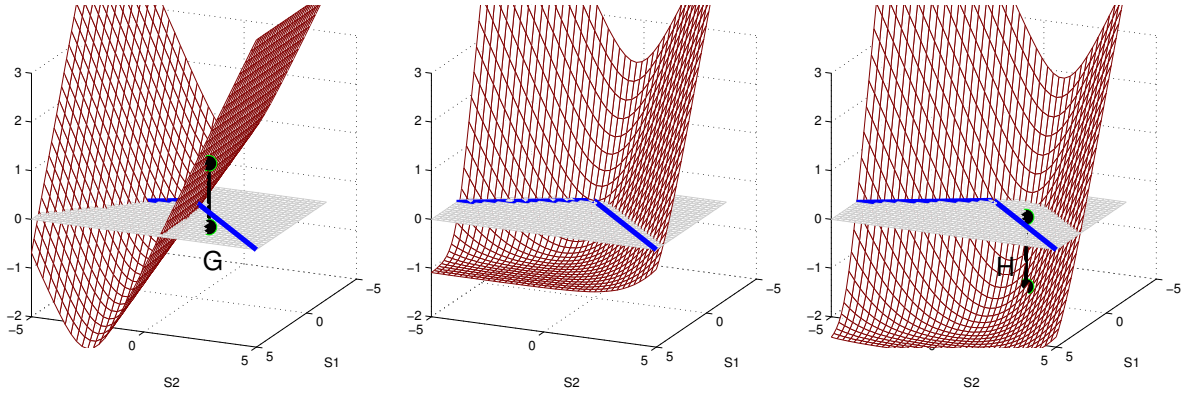
Arbitrary guess points are the points of maximal uncertainty for the classification rule of (1). Since all classes have the same score at these points, (1) produces a tie and the classifier $h(x)$ selects an arbitrary class by guessing. While this is the optimal decision when the probabilities of (4) are indeed identical, it is otherwise sub-optimal. A sensible loss function should steer learning algorithms away from arbitrary guess points for large portions of the example space, $\mathcal{X}$. In particular, the loss should encourage more correct classifications than arbitrary guesses, by encouraging $S(x)$ to be in $\mathcal{S}_z$ for the largest possible subset of $\mathcal{X}$ (where z is the label of x). When this is the case, the loss is said to be *averse to guessing*. This leads to the following definition.

Definition 3. *A loss L is averse to guessing if for any cost matrix C, any class z, any $S \in \mathcal{S}_z$ and any $A \in \mathcal{A}$*

$$L(C, z, S) < L(C, z, A). \tag{11}$$

In binary classification such as boosting or SVM it is common to assume $S_1(x) = -S_2(x) = f(x)$. In these cases, there is only one arbitrary guess point, i.e. $\mathcal{A} = \{(0, 0)\}$ and guess-aversion reduces to $L(C, 1, [0, 0]) > L(C, 1, [v, -v])\ \forall v \in \mathbb{R}^+$. As shown in Figure 2 this property holds for all popular binary losses, e.g. the hinge loss of SVMs (Vapnik, 1999), the logistic loss, the exponential loss of boosting (Freund & Schapire, 1996) and its

[2]For better illustration in the figures of this paper related to 3-class classification, we assume $S_3 = -S_1 - S_2$, and only show a 2D projection of the score space



(a) (b) (c)

Figure 3. **The guess-averse property for a three class problem:** S_1 increases along the x-axis, S_2 along the y-axis and $S_3 = -S_2 - S_1$ (not shown). The figures show loss surfaces for an example x from class 1, after a vertical shift such that $L(C, 1, \mathbf{0}) = 0$. As in Figure 1, $\mathcal{S}_1$ is the frontal triangle. 3(a) shows a surface plot of the cost-insensitive version of the loss L_s of (12). Point G correspondent to score vector $[3, 2, -5] \in \mathcal{S}_1$, violates the guess-averse property, since $L_s(C, 1, G) > L_s(C, 1, \mathbf{0})$. The loss thus prefers arbitrary guessing to the correct classification. 3(b) shows a surface plot of the cost-insensitive version of the $L^{\log,\exp}$ loss of (21). In this case, the guess-averse constraint is met for all points in $\mathcal{S}_1$. 3(c) shows a surface plot of $L^{\log,\exp}$ for the cost-sensitive case where $C_{1,2} = 1, C_{1,3} = 10$. This again has the guess-averse property, but one of the loss surface facets (corresponding to the boundary between $\mathcal{S}_1$ and $\mathcal{S}_2$) is shifted away from $\mathcal{S}_1$. Note that while $H = [1.5, 2, -3.5]$ is clearly outside $\mathcal{S}_1$ it has a lower loss than the origin. The loss function thus prefers the score of H to that of arbitrary guessing. This is reasonable: since $C_{1,3}$ is large and mis-classifying x to class 2 has lower risk than using $S = \mathbf{0}$, which assigns it to the third class with probability 0.33.

cost-sensitive extensions (Masnadi-Shirazi & Vasconcelos, 2011). In the multiclass case, however, the guess-averse condition is not as trivial as in the binary case. For example, it does not hold for the loss function

$$L_s(C, z, S(x)) = \sum_{j=1}^{M} C_{z,j} e^{S_j(x)} \tag{12}$$

where $\sum_{j=1}^{M} S_j(x) = 0$. This loss function is classification calibrated and popular for both cost-sensitive (Wang, 2013) and cost-insensitive learning (Lee et al., 2004). To show that (12) is not guess-averse, note that for $[3, 2, -5] \in \mathcal{S}_1$ and $\mathbf{0} = [0, 0, 0] \in \mathcal{A}$,

$$L_s(C, 1, [3, 2, -5]) \approx 7.4 > L_s(C, 1, \mathbf{0}) = 2, \tag{13}$$

where C is the cost-insensitive cost matrix. Therefore this loss *prefers* arbitrary guessing over correct classification, i.e. L_s is not averse to guessing. Figure 3(a) presents a plot of $L_s(C, 1, S(x)) - L_s(C, 1, \mathbf{0})$ for the cost-insensitive case.

Similarly, it is shown in Appendix A[3] that the loss function

$$L_t(C, z, S(x)) = \sum_{k=1}^{M} \sum_{j=1}^{M} C_{z,j} e^{S_j(x) - S_k(x)} \tag{14}$$

is classification calibrated but not guess-averse.

[3] Appendices are available in the supplementary material

Finally, it should be noted that the guess-averse property has some similarity with the "c-calibration" property of (Vernet et al., 2011). C-calibration requires that the loss of correct classification be smaller than the loss of incorrect classification. However, there are fundamental differences. First, the guess-averse property is defined using the set of scores, whereas c-calibration is defined on the probability simplex. Second, and most important, while the guess-averse property relies on the comparison between the score of correct classification and the score of arbitrary guessing, c-calibration compares the score of correct classification to all possible incorrect classifications. It is shown in Appendix B that c-calibration implies guess-aversion, but the converse is not true. For example, the GLL loss defined in (21) is guess-averse but not c-calibrated.

3.1. A family of guess-averse losses

We introduce a family of cost-sensitive multiclass loss functions.

Definition 4. *Let x be an example of class z, C a cost matrix, and $S(x) \in \mathbb{R}^M$ a score vector. Then, for any measurable functions $\gamma(\cdot)$ and $\phi(\cdot)$ the loss function*

$$L^{\gamma,\phi}(C, z, S(x)) = \gamma\left(\sum_{j=1}^{M} C_{z,j} \phi(S_z(x) - S_j(x))\right) \tag{15}$$

is denoted a $\gamma - \phi$ loss.

The following lemma provides a sufficient condition for

$L^{\gamma,\phi}$ to be a guess-averse loss.

Lemma 1. *Let $\gamma : \mathbb{R} \to \mathbb{R}$ be a monotonically increasing function and $\phi : \mathbb{R} \to \mathbb{R}$ a function satisfying*

$$\phi(v) < \phi(0) \quad \forall v > 0. \tag{16}$$

Then, $L^{\gamma,\phi}$ is a guess-averse loss.

Proof. If example x belongs to class z and if $S(x) \in \mathcal{S}_z$ then $S_z(x) - S_j(x) > 0 \; \forall j \neq z$. Therefore, it follows from (16) that

$$\phi(S_z(x) - S_j(x)) < \phi(0) \quad \forall j \neq z,$$

and from the non-negativity of the costs that

$$\textstyle\sum_{j=1}^{M} C_{z,j} \phi(S_z(x) - S_j(x)) < \phi(0) \sum_{j=1}^{M} C_{z,j}.$$

Finally since $\gamma(\cdot)$ is monotonically increasing

$$L^{\gamma,\phi}(C, z, S(x)) < \gamma\left(\phi(0) \textstyle\sum_{j=1}^{M} C_{z,j}\right). \tag{17}$$

On the other hand, for any $A \in \mathcal{A}$, $A_j = A_1 \; \forall j$ and thus

$$\begin{aligned} L^{\gamma,\phi}(C, z, A) &= \gamma\left(\textstyle\sum_{j=1}^{M} C_{z,j} \phi(A_j - A_z)\right) \\ &= \gamma\left(\phi(0) \textstyle\sum_{j=1}^{M} C_{z,j}\right). \end{aligned} \tag{18}$$

It follows from (17) and (18) that $\forall S(x) \in \mathcal{S}_z, \; \forall A \in \mathcal{A}$

$$L^{\gamma,\phi}(C, z, S(x)) < L^{\gamma,\phi}(C, z, A)$$

and thus $L^{\gamma,\phi}$ is guess-averse.

□

It can be shown that various previous losses in the literature, such as the pairwise comparison loss of (Zhang, 2004), are $\gamma - \phi$ losses. For binary cost-insensitive classification, (15) reduces to

$$L^{\gamma,\phi}(C, 1, S(x)) = \gamma\left(\phi(S_1(x) - S_2(x))\right). \tag{19}$$

Defining the scores as $S_1(x) = -S_2(x) = \frac{1}{2} f(x)$ for some function $f(x)$, and using the identity map for γ, i.e. $\gamma(v) = v$, (15) becomes the standard binary margin loss.

We next define two guess-averse, cost-sensitive multiclass $\gamma - \phi$ losses. The first, Generalized Exponential Loss (GEL), is obtained by setting $\gamma(v) = \mathrm{id}(v) = v$ and $\phi(v) = e^{-v}$

$$L^{\mathrm{id,exp}}(C, z, S(x)) = \textstyle\sum_{j=1}^{M} C_{z,j} e^{S_j(x) - S_z(x)}. \tag{20}$$

This is a straightforward cost-sensitive extension of the loss of (Saberian & Vasconcelos, 2011). For $M = 2$, GEL reduces to the cost-sensitive loss of (Viola & Jones, 2001a).

The second, Generalized Logistic Loss (GLL), is obtained by setting $\gamma(v) = \log(1 + v)$ and $\phi(v) = e^{-v}$,

$$L^{\log,\exp}(C, z, S(x)) = \log\left(1 + \textstyle\sum_{j=1}^{M} C_{z,j} e^{S_j(x) - S_z(x)}\right). \tag{21}$$

For cost-insensitive classification, this reduces to the multiclass logistic loss of (Friedman et al., 2000). Figure 3(b) shows a surface plot of the GLL, for the cost-insensitive 3-class problem.

These two losses have the desired behavior for cost-sensitive multiclass classification. Assume that z is the class of example x. If $S_z(x) > S_j(x) \; \forall j \neq z$, then all arguments in the exponent of (20) and (21) are negative and the loss is small. On the other hand, if $\exists j \mid S_j(x) > S_z(x)$, then $S(x) \notin \mathcal{S}_j$ and the loss is larger than $C_{z,j}$. This is sensible, since $C_{z,j}$ is the cost of assigning x to class j and j is a possible outcome of (1). Figure 3(c) shows the loss surface of GLL for an example x from class 1, with costs $C_{1,2} = 1, C_{1,3} = 10$. Note that the loss is still guess-averse but one of its facets (corresponding to the surface between $\mathcal{S}_1$ and $\mathcal{S}_2$) is shifted away from $\mathcal{S}_1$. In fact, while the point $H = [1.5, 2, -3.5]$ is clearly outside $\mathcal{S}_1$, it has lower loss than the origin $\mathbf{0} \in \mathcal{A}$, which means that the loss function prefers the score of H to arbitrary guessing. This is not surprising since using (1) the example will be assigned to the second class, incurring a cost of $C_{1,2} = 1$. On the other hand, the score $\mathbf{0} \in \mathcal{A}$ force the classifier to guess arbitrarily, and results in an expected cost of $\frac{1}{3} C_{1,1} + \frac{1}{3} C_{1,2} + \frac{1}{3} C_{1,3} \approx 3.6$.

4. Algorithm

In this section, we derive Boosting algorithms for the two guess-averse $\gamma - \phi$ losses, $L^{\log,\exp}$ of (20) and $L^{\mathrm{id,exp}}$ of (21), as well as two non guess-averse losses L_s of (14) and L_t of (12). In principle, these losses can be combined with any of the existing multiclass boosting approaches, e.g. AdaBoost-M1 & -M2 (Freund & Schapire, 1996), AdaBoost-MH (Schapire & Singer, 1999), SAMME (Zhu et al., 2009), AdaBoost-MM (Mukherjee & Schapire, 2013), or MultiBoost (Shen & Hao, 2011). In this work, we adopt MCBoost (Saberian & Vasconcelos, 2011) because it simplifies the derivation of the new boosting algorithm and unifies most of the boosting approaches in the literature. We start by briefly reviewing this method, referring the reader to the original paper for further details.

4.1. MCBoost

Given an M-class classification problem, MCBoost learns a multi-dimensional predictor $f(x) = [f_1(x), f_2(x) \dots f_{M-1}(x)] \in \mathbb{R}^{M-1}$ by minimizing the

risk of (8) for cost-insensitive version of (20),

$$L_{MC}(z, S(x)) = \sum_{j=1}^{M} e^{S_j(x) - S_z(x)} \tag{22}$$

where z is the label of example x,

$$S_k(x) = \frac{1}{2}\langle y_k, f(x)\rangle, \tag{23}$$

$\langle \cdot, \cdot \rangle$ the Euclidean dot product and $\mathcal{Y} = \{y_1, y_2, ...y_m\}$ a set of codewords that form a simplex in $\mathbb{R}^{M-1}$ such that $\sum_{j=1}^{M} y_j = 0$ and $\|y_j\|_2 = 1 \; \forall j$.

Given a set, $\mathcal{G}$, of weak learners where $g(x) \in \mathcal{G} : \mathcal{X} \to \mathbb{R}$, and a training set $\mathcal{D} = \{(x_i, z_i)\}_1^n$, MCBoost solves the optimization problem

$$\begin{cases} \min_f & \widehat{\mathcal{R}}_{L_{MC}}[f] = \sum_{i=1}^{n}\sum_{j=1}^{M} e^{-\frac{1}{2}\langle y_{z_i} - y_j, f(x_i)\rangle} \\ s.t & f_r \in \text{span}(\mathcal{G}) \quad \forall r = 1 \dots M-1, \end{cases} \tag{24}$$

using coordinate descent in function space (Mason et al., 2000; Friedman, 1999)[4].

In each boosting iteration, for each coordinate r, MCBoost computes the directional derivative of the risk for an update of $f_r(x)$ along the direction of $g(x)$

$$\delta\widehat{\mathcal{R}}_{L_{MC}}[f; r, g] = \left.\frac{\partial \widehat{\mathcal{R}}_{L_{MC}}[f + \epsilon g \mathbf{1}_r]}{\partial \epsilon}\right|_{\epsilon=0} \tag{25}$$

$$= \sum_{i=1}^{n} \left.\frac{\partial L_{MC}(z_i, (f + \epsilon g)(x_i)\mathbf{1}_r)}{\partial \epsilon}\right|_{\epsilon=0} \tag{26}$$

$$= \sum_{i=1}^{n} g(x_i) w^r(x_i), \tag{27}$$

with

$$w^r(x_i) = \frac{\partial}{\partial f_r(x_i)} \sum_{j=1}^{M} e^{\langle f(x_i), \Delta_{z_i,j}\rangle} \tag{28}$$

$$= \sum_{j=1}^{M} \langle \Delta_{z_i,j}, \mathbf{1}_r\rangle e^{\langle f(x_i), \Delta_{z_i,j}\rangle}, \tag{29}$$

where $\Delta_{z_i,j} = \frac{1}{2}[y_j - y_{z_i}]$ and $\mathbf{1}_r \in \mathbb{R}^{M-1}$ is a vector whose r^{th} element is one and the remaining zero. MCBoost then selects the best weak learner as

$$g_r^* = \text{argmin}_{g \in \mathcal{G}} \, \delta\widehat{\mathcal{R}}_{L_{MC}}[f; r, g] \tag{30}$$

and the optimal step size along g_r^*,

$$\alpha_r^* = \text{argmin}_{\alpha \in \mathbb{R}} \, \widehat{\mathcal{R}}_{L_{MC}}[f + \alpha g_r^* \mathbf{1}_r], \tag{31}$$

[4]While we focus on coordinate descent MCBoost, all results are also valid for gradient descent MCBoost (Saberian & Vasconcelos, 2011).

Algorithm 1 (GLL, GEL, L_s, L_t)-MCBoost

Input: Number of classes M, a set of codewords $\mathcal{Y} = \{y^1, \dots, y^M\} \in \mathbb{R}^{M-1}$, a number of iterations T, a dataset $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^n$, and a cost matrix C.
Initialization: set $f = 0 \in \mathbb{R}^d$
for $t = 1$ to T **do**
  for $r = 1$ to $M - 1$ **do**
    Compute $w^r(x_i)$ using (34 - 37)
    Find $g_r^*(x)$, α_r^* using (30) and (31)
    Update $f_r(x) := f_r(x) + \alpha_r^* g_r^*(x)$
  end for
end for
Output: decision rule: $\arg\max_j \langle y_j, f_j(x)\rangle$

is computed with a line search. Finally, the predictor is updated as

$$f := [f_1, \dots, f_r + \alpha_r^* g_r^*, \dots, f_{M-1}]. \tag{32}$$

Upon convergence, the posterior probabilities of (4) can be estimated as

$$P_{Z|X}(j|x) = \frac{e^{\langle y_j, f(x)\rangle}}{\sum_{k=1}^{M} e^{\langle y_k, f(x)\rangle}}. \tag{33}$$

4.2. Cost Sensitive MCBoost

Under the MCBoost framework, deriving a boosting algorithm for the minimization of $\widehat{\mathcal{R}}_{L^{\text{id,exp}}}$, $\widehat{\mathcal{R}}_{L^{\text{log,exp}}}$, $\widehat{\mathcal{R}}_{L_s}$ and $\widehat{\mathcal{R}}_{L_t}$ reduces to recomputing directional gradients. Using (23), (25) and (27), this results in the following definitions of the boosting weights

$$w^r_{L^{\text{id,exp}}}(x_i) = \sum_{j=1}^{M} C_{z_i,j}\langle \Delta_{z_i,j}, \mathbf{1}_r\rangle e^{\langle f(x_i), \Delta_{z_i,j}\rangle} \tag{34}$$

$$w^r_{L^{\text{log,exp}}}(x_i) = \frac{\sum_{j=1}^{M} C_{z_i,j}\langle \Delta_{z_i,j}, \mathbf{1}_r\rangle e^{\langle f(x_i), \Delta_{z_i,j}\rangle}}{1 + \sum_{j=1}^{M} C_{z_i,j} e^{\langle f(x_i), \Delta_{z_i,j}\rangle}} \tag{35}$$

$$w^r_{L_s}(x_i) = \sum_{j=1}^{M} C_{z_i,j}\langle y_j, \mathbf{1}_r\rangle e^{\langle f(x_i), y_j\rangle} \tag{36}$$

$$w^r_{L_t}(x_i) = \sum_{k=1}^{M}\sum_{j=1}^{M} C_{z_i,j}\langle \Delta_{j,k}, \mathbf{1}_r\rangle e^{\langle f(x_i), \Delta_{j,k}\rangle} \tag{37}$$

These cost-sensitive extensions of MC-Boost are presented in Algorithm 1.

5. Experiments

Various experiments were designed to evaluate the importance of the guess-averse property. These experiments used 10 UCI datasets and a large scale computer vision dataset for coral classification (Beijbom et al., 2012).

Table 1. Characteristics of the used UCI datasets

Dataset	#Training	#Testing	#Attributes	# Classes
Breast Tissue	81	25	9	6
Ecoli	258	78	7	8
Image Segmentation	210	2,100	19	7
Libras	292	68	90	15
Vertebral	239	71	6	3
Vehicle	692	154	18	4
Shuttle	43,500	14,500	9	7
Pen Digit	7,494	3,498	16	10
Optical Digits	3,823	1,797	64	10
Satellite Image	4,435	2,000	36	6

We used the boosting algorithm derived in section 4.2 to compare the performance of the two guess-averse losses $L^{\log,\exp}$ of (20) and $L^{\mathrm{id},\exp}$ of (21), with the two non guess-averse losses L_s of (14) and L_t of (12). For the sake of completeness, we also compared these algorithms with two baseline algorithms 1) the cost-insensitive MCBoost algorithm of (Saberian & Vasconcelos, 2011) which uses the loss function of (22) and 2) a cost-sensitive version of MCBoost, denoted P-MCBoost, which computes posterior probabilities with (33), and uses them in the cost-sensitive Bayes decision rule of (3). In all experiments, regression classifiers were used as weak learners, i.e. $g(x) = ax + b$ where a, b are found by solving a weighted least square problem (Friedman et al., 2000). Performance was evaluated by the classification risk of (5) on the test set.

In our experiments, we focus on symmetric cost-matrices ($C_{j,k} = C_{k,j}\ \forall j, k$), which are natural for many classification problems. For example, when the class structure is given by a taxonomy one often imposes symmetric misclassification costs for class pairs, which vary depending on class distances in the taxonomical tree. This is the case for various popular computer vision datasets such as ImageNet (Deng et al., 2009), CalTech birds (Welinder et al., 2010), or MLC (Beijbom et al., 2012). Other examples are losses based on ordinal regression or the hamming distance between class labels (Ramaswamy & Agarwal, 2012).

In summary, since the goal of the experiments was to investigate the importance of guess-aversion, we fixed the boosting framework (MCBoost), weak learners (regression $[ax + b]$) and the number of boosting iterations. This setup enables a fair comparison between the loss functions.

5.1. UCI Datasets

We start with an evaluation on the ten UCI datasets of Table 1. For these, training/testing partition are either predefined or the data is randomly split into 80% training and 20% testing. For each dataset, a random symmetric cost matrix was generated, $C_{j,k}\ j \neq k$ drawn uniformly from $[1, 10] \in \mathbb{R}$, and all boosted classifiers were trained with 100 iterations. The procedure was repeated 50 times per dataset, and the average classification risk of (5) is reported for each classifier in Table 2. We also ranked the algorithms according to their performance on each dataset. The average rank is shown in the last row of Table 2.

Several observations can be made. First, according to the average ranks, all algorithms based on guess-averse losses had significantly better performance than the non guess-averse algorithms. Note that, as shown in (Wang, 2013) and Appendix A, L_s-MCBoost and L_t-MCBoost are cost-sensitive and classification calibrated, i.e. with infinite training examples and suitable weak learners they implement Bayes rule. However, they are outperformed by all guess-averse losses. In particular, they are outperformed by GLL-MCBoost, which is not classification calibrated. This experiment therefore suggests that guess-aversion may be more important than classification calibration in practice.

Second, GLL-MCBoost consistently outperformed the alternative guess-averse boosting algorithms by a large margin. Not surprisingly, it has better performance than MCBoost, which does not account for the cost matrix. The gains over GEL-MCBoost have a more subtle justification. While this is a cost-sensitive method, it can be shown that the GEL loss is insensitive to symmetric cost-matrices (Appendix C). This explains why GEL-MCBoost performs on par with the cost-insensitive MCBoost. On the other hand, the cost matrix is taken into account by P-MCBoost, which is based on Bayes rule and posterior probability estimates. However, as argued by (Masnadi-Shirazi & Vasconcelos, 2011), these estimates are accurate only in the neighborhood of the cost-insensitive decision boundary. Since the cost-sensitive boundary can be far from its cost-insensitive counterpart, the posterior estimates can be inaccurate in the regions of interest for cost-sensitive classification. Still, P-MCBoost shows an improvement over MCBoost.

5.2. Moorea Labeled Corals

We next evaluated the efficacy of the proposed boosting algorithm on a large scale computer vision dataset. In addition to the boosting methods of the previous section, we compare against three method based on Support Vector Machines.

Moorea Labeled Corals (MLC) comprises more than $400K$ expert annotations on more than $2K$ coral reef survey images, assembled over the years $2008-2010$ (Beijbom et al., 2012). MLC has a natural hierarchy. For example, misclassification between coral genera is less severe than misclassification between coral and algae, or algae and sand. Using this natural class hierarchy, we derived an appropriate cost-matrix, provided in Appendix D, and used it in all MLC experiments. We adopted the feature representation of (Beijbom et al., 2012), and considered all images and

Table 2. **Cost-sensitive classification risk** (5) **on the UCI datasets**. Results indicated as (mean $\pm$ standard error) for 50 random cost-matrix trials. The strongest results for each dataset are marked in bold with significance determined by a paired t-test at the 5% level.

	Guess-averse				Non guess-averse	
Dataset	MCBoost	P-MCBoost	GEL-MCBoost	GLL-MCBoost	L_s-MCBoost	L_t-MCBoost
Breast Tissue	1.83 ± 0.06	1.63 ± 0.06	1.72 ± 0.06	$\mathbf{1.48 \pm 0.05}$	1.88 ± 0.07	1.83 ± 0.08
Ecoli	0.96 ± 0.03	0.84 ± 0.03	0.94 ± 0.03	$\mathbf{0.82 \pm 0.03}$	1.12 ± 0.04	1.21 ± 0.03
Image Segmentation	0.47 ± 0.02	0.46 ± 0.02	0.47 ± 0.01	$\mathbf{0.45 \pm 0.02}$	0.97 ± 0.03	1.06 ± 0.04
Libras	1.24 ± 0.03	1.16 ± 0.03	1.31 ± 0.03	$\mathbf{1.01 \pm 0.03}$	2.32 ± 0.06	2.43 ± 0.06
Vertebral	0.81 ± 0.05	$\mathbf{0.77 \pm 0.05}$	$\mathbf{0.77 \pm 0.04}$	0.78 ± 0.04	0.87 ± 0.05	0.92 ± 0.06
Vehicle	1.42 ± 0.06	1.41 ± 0.06	1.41 ± 0.06	$\mathbf{1.40 \pm 0.06}$	1.59 ± 0.07	1.55 ± 0.07
Shuttle	0.36 ± 0.02	0.33 ± 0.02	0.36 ± 0.02	$\mathbf{0.17 \pm 0.01}$	0.83 ± 0.04	0.86 ± 0.05
Pen Digits	0.51 ± 0.01	0.49 ± 0.01	0.50 ± 0.01	$\mathbf{0.49 \pm 0.01}$	1.28 ± 0.03	1.54 ± 0.04
Satellite Image	0.88 ± 0.03	$\mathbf{0.85 \pm 0.03}$	0.89 ± 0.03	0.88 ± 0.03	1.38 ± 0.04	1.36 ± 0.04
Optical Digits	0.26 ± 0.00	$\mathbf{0.24 \pm 0.00}$	0.27 ± 0.01	0.26 ± 0.00	0.64 ± 0.01	0.81 ± 0.02
Avg. Rank	3.50	1.70	3.20	**1.50**	5.30	5.70

annotations from 2008, split randomly 10 times so that $2/3$ of the images were used for training and $1/3$ for testing.

The boosting baselines of the previous section were complemented by three linear SVM algorithms. The first, SVM-OVA, implements the popular one-versus-all approach; the second, SVM-MC, is a multiclass SVM based on the formulation of (Crammer & Singer, 2002); and the third, SVM-CSMC, uses the cost-sensitive multiclass formulation of (Branson et al., 2013). As shown in Appendix E, loss functions of SVM-MC and SVM-CSMC are both guess-averse. Among the three SVM algorithms, SVM-CSMC is the only one that is cost-sensitive. For the first two methods we use the LIBLINEAR implementation (Fan et al., 2008), and for the third the publicly available code of (Branson et al., 2013). Cross-validation was performed on the training set to tune the SVM regularization parameter, $\lambda = \{10^{-17} \dots 10^{7}\}$, so as to minimize the classification risk of (5). The boosting methods were trained with 500 iterations.

The classification risk of (5) on the test sets are shown in Figure 4. Again, the two methods based on non guess-averse losses (L_s-MCBoost and L_t-MCBoost) were significantly outperformed by the guess-averse methods. Among the guess-averse methods, cost-insensitive MCBoost was again the weakest, with a classification risk of ≈ 0.66. The probability based P-MCBoost and GEL-MCBoost achieved roughly the same performance. SVM-MC and SVM-OVA were stronger, achieving a classification risk ≈ 0.64. Nevertheless, their performance was weaker than those of GLL-MCBoost and SVM-CSMC, which optimize cost-sensitive and guess-averse losses. These methods had similar performance, achieving the lowest classification risks of ≈ 0.62. This similar performance is not surprising since the GLL loss is a differential approximation of the multiclass hinge loss function used in SVM-CSMC.

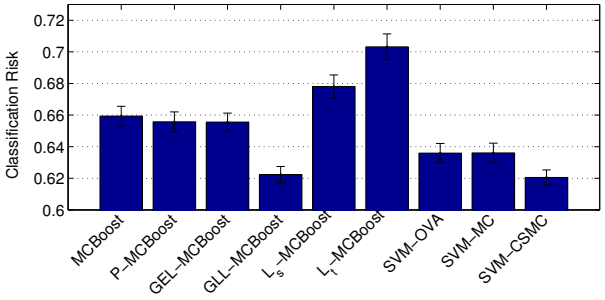


Figure 4. **Cost-sensitive classification risk** (5) **on Moorea Labeled Corals**. Bars indicate average classification risk and standard error for 10 random splits of the data.

6. Conclusion

In this work, we have proposed that guess-aversion is an important property for multiclass loss functions. Guided by this property, we derived a family of guess-averse loss functions, and developed boosting algorithms based on two members of this family, GLL-MCBoost, and GEL-MCBoost. Extensive experiments have demonstrated the importance of guess-aversion and that the GLL loss function outperforms alternative loss functions for cost-sensitive multiclass boosting.

Acknowledgements: This work was partially funded by NSF awards IIS-1208522, CCF-0830535, ATM-0941760 and the Korean Ministry of Trade, Industry & Energy, grant no. 10041126.

References

Abe, Naoki, Zadrozny, Bianca, and Langford, John. An iterative method for multi-class cost-sensitive learning. In *SIGKDD*, 2004.

Bartlett, Peter L. Jordan, Michael I. and McAuliffe,

Jon D.˙ Convexity, classification, and risk bounds. *JASA*, 2006.

Beijbom, Oscar, Edmunds, Peter J, Kline, David I, Mitchell, B Greg, and Kriegman, David. Automated annotation of coral reef survey images. In *CVPR*, 2012.

Branson, Steve, Beijbom, Oscar, and Belongie, Serge. Efficient large-scale structured learning. In *CVPR*, 2013.

Breiman, Leo. Bagging predictors. *ML*, 1996.

Crammer, Koby and Singer, Yoram. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2002.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Dollár, Piotr, Belongie, Serge, and Perona, Pietro. The fastest pedestrian detector in the west. In *BMVC*, 2010.

Domingos, Pedro. Metacost: a general method for making classifiers cost-sensitive. In *SIGKDD*, 1999.

Elkan, Charles. The foundations of cost-sensitive learning. In *IJCAI*, 2001.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. Liblinear: A library for large linear classification. *JMLR*, 2008.

Fan, Wei, Stolfo, Salvatore J, Zhang, Junxin, and Chan, Philip K. Adacost: misclassification cost-sensitive boosting. In *ICML*, 1999.

Freund, Yoav and Schapire, Robert E. Experiments with a new boosting algorithm. In *ICML*, 1996.

Friedman, J. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1999.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 2000.

Lee, Yoonkyung, Lin, Yi, and Wahba, Grace. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *JASA*, 2004.

Lozano, Aurélie C and Abe, Naoki. Multi-class cost-sensitive boosting with p-norm loss functions. In *SIGKDD*, 2008.

Masnadi-Shirazi, Hamed and Vasconcelos, Nuno. Cost-sensitive boosting. *PAMI*, 2011.

Mason, L., Baxter, J., Bartlett, P., and Frean, M. Functional gradient techniques for combining hypotheses. *Advances in Large Margin Classifiers*, 2000.

Mukherjee, Indraneel and Schapire, Robert E. A theory of multiclass boosting. *JMLR*, 2013.

Ramaswamy, Harish G and Agarwal, Shivani. Classification calibration dimension for general multiclass losses. In *NIPS*, 2012.

Saberian, M. and Vasconcelos, N. Multiclass boosting: Theory and algorithms. In *NIPS*, 2011.

Schapire, Robert E. and Singer, Yoram. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999.

Scott, Clayton. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 2012.

Shen, Chunhua and Hao, Zhihui. A direct formulation for totally-corrective multi-class boosting. In *CVPR*, 2011.

Tewari, Ambuj and Bartlett, Peter L. On the consistency of multiclass classification methods. *JMLR*, 2007.

Torralba, Antonio, Murphy, Kevin P, and Freeman, William T. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004.

Vapnik, Vladimir. *The nature of statistical learning theory*. springer, 1999.

Vernet, E., Williamson, R., and Reid, M. Composite multiclass losses. *NIPS*, 2011.

Viola, Paul and Jones, Michael. Fast and robust classification using asymmetric adaboost and a detector cascade. In *NIPS*, pp. 1311–1318, 2001a.

Viola, Paul and Jones, Michael. Fast and robust classification using asymmetric adaboost and a detector cascade. *NIPS*, 2001b.

Viola, Paul and Jones, Michael. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001c.

Wang, Junhui. Boosting the generalized margin in cost-sensitive multiclass classification. *JCGS*, 2013.

Welinder, Peter, Branson, Steve, Mita, Takeshi, Wah, Catherine, Schroff, Florian, Belongie, Serge, and Perona, Pietro. Caltech-ucsd birds 200. *Caltech*, 2010.

Zhang, Tong. Statistical analysis of some multi-category large margin classification methods. *JMLR*, 2004.

Zhu, Ji, Zou, Hui, Rosset, Saharon, and Hastie, Trevor. Multi-class adaboost. *Statistics and Its Interface*, 2009.

Supplementary Material: Guess-Averse Loss Functions For Cost-Sensitive Multiclass Boosting

Oscar Beijbom OBEIJBOM@UCSD.EDU
Mohammad Saberian SABERIAN@UCSD.EDU
David Kriegman KRIEGMAN@UCSD.EDU
Nuno Vasconcelos NVASCONCELOS@UCSD.EDU
University of California, San Diego, 9500 Gilman Drive, 92093 La Jolla, CA

A. Properties of $L_t(C, z, S(x))$ loss function

Lemma 1. *The loss function*

$$L_t(C, z, S(x)) = \sum_{j,k=1}^{M} C_{z,j} e^{S_j(x) - S_k(x)} \tag{1}$$

is Classification Calibrated.

Proof. Using (1), risk of classification is

$$\begin{aligned} \mathcal{R}_{L_t}[S] &= E_{X,Z}\{L_t[C, z, S(x)] && (2)\\ &= \sum_{z=1}^{M} \eta_z(x) L_t(C, z, S(x)) && (3)\\ &= \sum_{z=1}^{M} \eta_z(x) \sum_{j,k=1}^{M} C_{z,j} e^{S_j(x) - S_k(x)} && (4)\\ &= \sum_{z=1}^{M} \sum_{j,k=1}^{M} \eta_z(x) C_{z,j} e^{S_j(x) - S_k(x)}. && (5) \end{aligned}$$

To find the optimal scores, $S^*(x)$, we start by setting derivatives to zero, where

$$\begin{aligned} 0 = \frac{\partial \mathcal{R}_{L_t}[S]}{\partial S_i(x)} &= \sum_{z=1}^{M} \eta_z(x) C_{z,i} e^{S_i(x)} \sum_{k=1}^{M} e^{-S_k(x)} && (6)\\ &\quad - e^{-S_i(x)} \sum_{z=1}^{M} \sum_{j=1}^{M} \eta_z(x) C_{z,j} e^{S_j(x)}, && (7) \end{aligned}$$

results in

$$e^{-2S_i(x)} = \sum_{z=1}^{M} \eta_z(x) C_{z,i} \frac{\sum_{k=1}^{M} e^{-S_k(x)}}{\sum_{z,j=1}^{M} \eta_z(x) C_{z,j} e^{S_j(x)}}. \tag{8}$$

Assuming $\sum_{i=1}^{M} S_i(x) = 0$, and defining

$$\psi(i) = \sum_{z=1}^{M} \eta_z(x) C_{z,i} \tag{9}$$

results in

$$S^*{}_i(x) = -\frac{1}{2} \log\left(\psi(i)\right) + \frac{1}{2M} \sum_{j=1}^{M} \log(\psi(j)). \tag{10}$$

Therefore $S^*{}_i(x)$ will be inversely proportional to Bayes cost of i^{th} class and thus (1) will be classification calibrated. □

Lemma 2. *$L_t(C, z, S(x))$ is not guess-averse.*

Proof. The proof is based on a counter example. Assume a cost insensitive problem i.e $C_{i,j} = 1 \quad i \neq j$, with $M = 3$ and $S(x) = [3, 0, -3]$ where example x belongs to the first class. In this case $S(x)$ results in the correct prediction but its loss is greater than random guessing since $L(C, 1, 0) = 6 < L(C, 1, S(x)) \approx 22.19$. Therefore (1) is not guess-averse. □

B. Comparing Guess-aversion and c-calibration

We start with following lemma that shows that c-calibration (Vernet et al., 2011) implies guess-aversion.

Lemma 3. *If a loss function $L(C, z, S(x))$ is c-calibrated, then it will be guess-averse.*

Proof. If $L(C, z, S(x))$ is c-calibrated, then according to c-calibration definition

$$\forall s_1 \in \mathcal{S}_z \quad \forall s_2 \notin \mathcal{S}_z, L(C, z, s1) < L(C, z, s2). \tag{11}$$

In addition note that, $\mathbf{0} \notin \mathcal{S}_z$ and using (11)

$$\forall s_1 \in \mathcal{S}_z \quad L(C, z, s1) < L(C, z, \mathbf{0}). \tag{12}$$

which is definition of guess-aversion. □

We next show that guess-aversion does not guarantee c-calibration.

Lemma 4. *If a loss function $L(C, z, S(x))$ is guess-averse, then it may not be c-calibrated.*

Proof. The proof is based on counter example. Consider the cost-insensitive GLL-loss of Figure 3-b. Since it satisfies Lemma 1 in the paper this loss is guess-averse. However, for sufficiently small $\epsilon > 0$, the set $\mathcal{A}_\epsilon = \{S | S \in \mathcal{S}_2, L(C, 1, S) < L(C, 1, 0) - \epsilon\}$ is non-empty. Similarly, since the loss surface is continuous and smooth in Figure 3-b, there exists a point $p_0 \in \mathcal{S}_1$ such that $L(C, 1, p_0) > L(C, 1, 0) - \epsilon$. Therefore for any $q_0 \in A_\epsilon$, $L(C, 1, p_0) > L(C, 1, q_0)$ which is contradictory to c-calibration, since p_0 results in correct classification and q_0 does not. □

C. Properties of the Generalized Exponential Loss

Lemma 5. *If $C_{i,j} \geq 0 \, \forall i, j = 1 \dots M$ and $\exists i, j : C_{i,j} > 0$ then*

$$\begin{aligned} \mathcal{R}_{L^{id,\exp}} &= E_{X,Z}\{L^{id,\exp}(C, z, S(x))\} \\ &= \sum_{z,j=1}^{M} \eta_z(x) C_{z,j} e^{S_j(x) - S_z(x)}, \end{aligned} \quad (13)$$

is strictly convex with respect to $S(x) \in \mathbb{R}^M$.

Proof. Denoting $\beta_{i,j} = \mathbf{1}_i - \mathbf{1}_j$, we start by computing first and second order derivatives,

$$\begin{aligned} \frac{\partial \mathcal{R}_{L^{\text{id},\exp}}}{\partial S} &= \frac{\partial}{\partial S} \sum_{z,j=1}^{M} \eta_z C_{z,j} e^{\langle S, \beta_{j,z} \rangle} \\ &= \sum_{z,j=1}^{M} \eta_z C_{z,j} \beta_{j,z} e^{\langle S, \beta_{j,z} \rangle} \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial^2 \mathcal{R}_{L^{\text{id},\exp}}}{\partial S^2} &= \frac{\partial}{\partial S} \sum_{z,j=1}^{M} \eta_z C_{z,j} \beta_{j,z} e^{\langle S, \beta_{j,z} \rangle} \\ &= \sum_{z,j=1}^{M} \eta_z C_{z,j} [\beta_{j,z} \beta_{j,z}^T] e^{\langle S, \beta_{j,z} \rangle}. \end{aligned} \quad (15)$$

where we omitted x for simplicity. Note that $[\beta_{j,z} \beta_{j,z}^T]$ is positive definite for all z, j, moreover $C_{i,j} \geq 0 \quad \forall i, j = 1 \dots M$ and $\exists i, j : C_{i,j} > 0$. Therefore the hessian is a sum of positive definite matrices, and is a positive definite matrix. Therefore $\mathcal{R}_{L^{\text{id},\exp}}$ is strictly convex. □

Lemma 6. *If the cost matrix, C, is symmetric then the minimizer of $\mathcal{R}_{L^{id,\exp}(C,z,S(x))}$, (13), is independent of C.*

Proof. We start by setting (14) to zero, therefore

$$\sum_{z,j=1}^{M} \eta_z C_{z,j} \mathbf{1}_j e^{\langle S, \beta_{j,z} \rangle} = \sum_{z,j=1}^{M} \eta_z C_{z,j} \mathbf{1}_z e^{\langle S, \beta_{j,z} \rangle} \quad (16)$$

Table 1. Cost Matrix for MLC (Beijbom et al., 2012).

	CCA	*Turf*	*Macro*	*Sand*	*Acro.*	*Pav.*	*Mon.*	*Pocil.*	*Porit*
CCA	0	1	1	2	4	4	4	4	4
Turf	1	0	1	2	4	4	4	4	4
Macro	1	1	0	2	4	4	4	4	4
Sand	2	2	2	0	4	4	4	4	4
Acropora	4	4	4	4	0	1	1	1	1
Pavona	4	4	4	4	1	0	1	1	1
Monti	4	4	4	4	1	1	0	1	1
Pocill	4	4	4	4	1	1	1	0	1
Porit	4	4	4	4	1	1	1	1	0

and thus

$$\sum_{j=1}^{M} \eta_k C_{k,j} e^{S_j - S_k} = \sum_{z=1}^{M} \eta_z C_{z,k} e^{S_k - S_z}. \quad (17)$$

However note that when C is symmetric,

$$S_k(x) = \frac{1}{2} \log(\eta_k(x)) - \frac{1}{2M} \sum_{j}^{M} \log(\eta_j(x)) \quad (18)$$

satisfies (17). This is because $e^{S_j - S_k} = \frac{\sqrt{\eta_j}}{\sqrt{\eta_k}}$ and thus left and right sides of (17)

$$\sum_{j=1}^{M} \eta_k C_{k,j} e^{S_j - S_k} = \sum_{j=1}^{M} C_{k,j} \sqrt{\eta_j \eta_k} \quad (19)$$

$$\sum_{z=1}^{M} \eta_z C_{z,k} e^{S_k - S_z} = \sum_{z=1}^{M} C_{z,k} \sqrt{\eta_z \eta_k}. \quad (20)$$

become equal. Therefore (18) is a minimizer of $\mathcal{R}_{L^{\text{id},\exp}(C,z,S(x))}$. In addition according to lemma (5), $\mathcal{R}_{L^{\text{id},\exp}}$ is strictly convex and thus (18) will be the unique minimizer. □

D. MLC - Cost Matrix

The cost matrix for the Moorea Labelled Corals dataset is shown in Table 1. The costs are set with an coral ecology application in mind. There, the most important goal is a binary estimate of the amount of corals versus everything else. Thus, the cost of confusion between the coral genera (classes 5-9) and the non-corals (classes 1-4) is set to a high value, 4. Cost of confusion among corals is low, 1, and similarly for cost of confusion among algae (classes 1-3). Finally, confusion between any algae and the sand class is worse than confusion within algae, but not as bad as confusion to (or from) corals. These values are set to 2.

E. Structured SVMs are guess-averse

Let $\mathcal{Y} = \{Y_1, \dots Y_M\}$ be a set of structured outputs. For a training set $\mathcal{D} = \{(x_i, Y_{z_i})\}_1^n$, where $z_i \in \{1 \dots M\}$, a

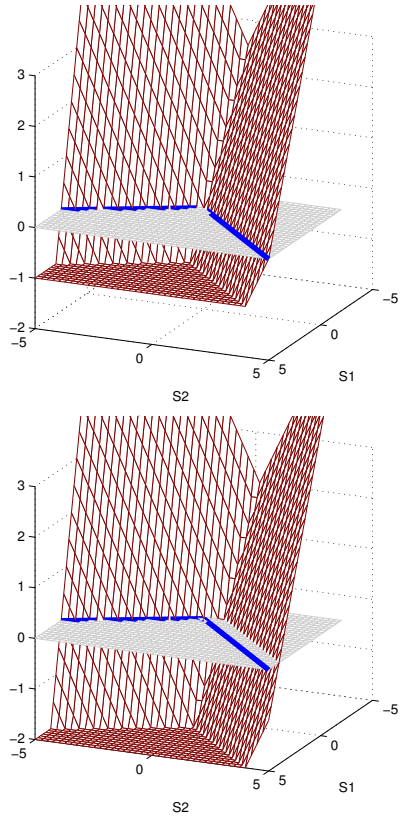


Figure 1. **Structured SVM loss functions:** Cost-insensitive (top), and cost-sensitive, with $C_{1,2} = 1, C_{1,3} = 2$, (bottom).

structured SVM (Tsochantaridis et al., 2004) solves

$$\begin{cases} \min_{\mathbf{w},\epsilon} & \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \epsilon_i \\ \text{s.t.,} & \forall_{i,z\in\mathcal{Z}},\ \ S_z(x_i) + C_{z_i,z} \leq S_{z_i}(x_i) + \epsilon_i \\ & \epsilon_i \geq 0\ \forall i, \end{cases} \tag{21}$$

where

$$S_z(x) = \langle \mathbf{w}, \Psi(x, z) \rangle \tag{22}$$

is the score of structure Y_z for the example x, $\Psi(x, z)$ is a feature vector extracted with respect to structure Y_z, and $C_{z_i,z} \geq 0$ is the cost of assigning structure Y_z instead of the true structure Y_{z_i}.

An equivalent way of writing (21) is

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i L_H(C, z_i, S(x_i)) \tag{23}$$

where

$$L_H(C, z_i, S(x_i)) = \max_k (S_k(x_i) + C_{z_i,k} - S_{z_i}(x_i)), \tag{24}$$

is the loss function for structured SVM. Similar to Figure 3 of the paper, loss surfaces for L_H, (24), are shown in Figure 1. Note how, in the bottom figure, the surface shifts away the boundary between $\mathcal{S}_1$ and $\mathcal{S}_2$, in a similar manner as the cost-sensitive $L^{\log,\exp}$ did. This is not surprising as the logistic function approximates the hinge loss. Finally, the following lemma shows that L_H is guess-averse.

Lemma 7. *The loss function for structured SVM,(24), is guess-averse.*

Proof. Let x be a sample corresponding to a structure Y_z, $S \in \mathbb{R}^M$ the classifier score vector and C a non-negative cost function. First note that using (24) if $S = \mathbf{0} \in \mathbb{R}^M$,

$$L_H(C, z, \mathbf{0}) = \max_k C(z, k). \tag{25}$$

Second, if x is correctly classified, i.e. $S(x) \in \mathcal{S}_z$, then $S_z(x) > S_k(x)\ \forall k \neq z$ and thus using (24), (25)

$$\begin{aligned} L_H(C, z, S(x)) &= \max_k \left[C_{z,k} + (S_k(x) - S_z(x))\right] \\ &< \max_k (C_{z,k}) \\ &= L_H(C, z, \mathbf{0}). \end{aligned}$$

Therefore if $S(x) \in \mathcal{S}_z$, then $L_H(C, z, S(x)) < L_H(C, z, \mathbf{0})$ and thus L_H is guess-averse. □

References

Beijbom, Oscar, Edmunds, Peter J, Kline, David I, Mitchell, B Greg, and Kriegman, David. Automated annotation of coral reef survey images. In *CVPR*, 2012.

Tsochantaridis, Ioannis, Hofmann, Thomas, Joachims, Thorsten, and Altun, Yasemin. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

Vernet, E., Williamson, R., and Reid, M. Composite multiclass losses. *NIPS*, 2011.