A Hybrid Loss for Multiclass and Structured Prediction

Qinfeng Shi, *Member, IEEE,* Mark Reid, Tiberio Caetano, and Anton van den Hengel, *Member, IEEE*

Abstract—We propose a novel hybrid loss for multiclass and structured prediction problems that is a convex combination of log loss for Conditional Random Fields (CRFs) and a multiclass hinge loss for Support Vector Machines (SVMs). We provide a sufficient condition for when the hybrid loss is Fisher consistent for classification. This condition depends on a measure of dominance between labels – specifically, the gap in per observation probabilities between the most likely labels. We also prove Fisher consistency is necessary for parametric consistency when learning models such as CRFs. Moreover, we give PAC-Bayes bounds on the hybrid loss.
We demonstrate empirically that the hybrid loss typically performs as least as well as – and often better than – both of its constituent losses on variety of tasks. Applications on variety of multiclass and structured prediction tasks such as Plant Segmentation, Named Entity Recognition, Joint Image Object Categorization *etc.* demonstrate the excellence of the proposed hybrid loss. In doing so we also provide an empirical comparison of the efficacy of probabilistic and margin based approaches to multiclass and structured prediction and the effects of label dominance on these results.

Index Terms—Conditional Random Fields, Support Vector Machines, Hybrid Loss, Fisher Consistency, PAC-Bayes Bounds, Structured Learning

1 Introduction

Conditional Random Fields (CRFs) and Support Vector Machines (SVMs) can be seen as representative of two different approaches to classification problems. The former is purely *probabilistic* – the conditional probability of classes given each observation is explicitly modelled – while the latter is purely *discriminative* – classification is performed without any attempt to model probabilities. Both approaches have their strengths and weaknesses. CRFs [12, 22] are known to yield the Bayes optimal solution asymptomatically but often require a large number of training examples to do accurate modelling. In contrast, SVMs make more efficient use of training examples but are known to be inconsistent when there are more than two classes [25, 15].

Despite their differences, CRFs and SVMs appear very similar when viewed as optimisation problems. The most salient difference is the loss used by each: CRFs are trained using a log loss while SVMs typically use a hinge loss. In an attempt to capitalise on their relative strengths and avoid their weaknesses, we propose a novel *hybrid loss* which "blends" the two losses. After some background (§2) we provide the following analysis: We argue that Fisher Consistency for Classification (FCC) – a.k.a. classification calibration – is too coarse a notion and introduce a distribution-dependent refinement called Conditional Fisher Consistency for Classification (§3). We prove the hybrid loss is conditionally FCC and give a noise condition that relates the hybrid loss's mixture parameter to a margin-like property of the data distribution (§3.1). We then show that, although FCC is effectively a non-parametric condition, it is also a necessary condition for consistent risk minimisation using parametric models (§3.2). Finally, we empirically test the hybrid loss on various domains including multiclass classification, Chunking, Named Entity Recognition and Joint Image Object Categorisation and show it consistently performs as least as well as – and often better than – both of its constituent losses(§5).

2 Losses for Multiclass Prediction

In classification problems *observations* $x \in \mathcal{X}$ are paired with *labels* $y \in \mathcal{Y}$ via some joint distribution D over $\mathcal{X} \times \mathcal{Y}$. We will write $D(x, y)$ for the joint probability and $D(y|x)$ for the conditional probability of y given x. Since the labels y are finite and discrete we will also use the notation $D_y(x)$ for the conditional probability to emphasise that distributions over $\mathcal{Y}$ can be thought of as vectors in $\mathbb{R}^k$ for $k = |\mathcal{Y}|$. We will use q to denote distributions over $\mathcal{Y}$ when the observations $x \in \mathcal{X}$ are irrelevant. When the number of possible labels $k = |\mathcal{Y}| > 2$ we call the classification problem a *multiclass* classification problem.

2.1 From Multiclass to Structured Prediction

A special case of this type of problem is *structured prediction* where the set of labels $\mathcal{Y}$ has some combinatorial structure that typically means k is very large [1]. In structured prediction, each output y involves

- *Q. Shi and A. Hengel are with The Australian Centre for Visual Technologies and The Computer Vision group of The University of Adelaide, SA, Australia.*
  E-mail: qinfeng.shi@ieee.org, anton.vandenhengel@adelaide.edu.au
- *M. Reid and T. Caetano are with The Australian National University and NICTA, ACT, Australia.*
  E-mail:mark.reid@anu.edu.au, Tiberio.Caetano@nicta.com.au

relationships among 'sub-components' of y. For example, the label of a pixel in an image depends on the label of neighbouring pixels. That's where the term 'structured' comes from. However, different y's are typically *not* assumed to possess any joint structure (i.e., it is typically assumed that the data is drawn from $\mathcal{X} \times \mathcal{Y}$). This is why structured prediction is no different in essence than multiclass classification. As seen in the experimental section below a variety of problems, such as text tagging, can be construed as structured prediction problems.

Given m training observations $S = \{(x_i, y_i)\}_{i=1}^m$ drawn i.i.d. from D, the aim of the learner is to produce a predictor $h : \mathcal{X} \to \mathcal{Y}$ that minimises the *misclassification error* $e_D(h) = \mathbb{P}_D[h(x) \neq y]$. Since the true distribution is unknown, an approximate solution to this problem is typically found by minimising a regularised empirical estimate of the risk for a *surrogate loss* ℓ. Examples of surrogate losses will be discussed below.

Once a loss is specified, a solution is found by solving

$$\min_f \frac{1}{m} \sum_{i=1}^{m} \ell(f(x_i), y_i) + \Omega(f) \tag{1}$$

where each *model* $f : \mathcal{X} \to \mathbb{R}^k$ assigns a vector of scores $f(x)$ to each observation and the regulariser $\Omega(f)$ penalises overly complex functions. A model f found in this way can be transformed into a predictor by defining $h_f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} f_y(x)$. We will overload the definition of misclassification error and sometimes write $e_D(f)$ as shorthand for $e_D(h_f)$.

In structured prediction, the models are usually specified in terms of a parameter vector $w \in \mathbb{R}^n$ and a feature map $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ by defining $f_y(x; w) = \langle w, \phi(x, y) \rangle$ and in this case the regulariser is $\Omega(f) = \frac{\lambda}{2}\|w\|^2$ for some choice of $\lambda \in \mathbb{R}$. This is the framework used to implement the SVMs and CRFs used in the experiments described in Section 5. Although much of our analysis does not assume any particular parametric model, we explicitly discuss the implications of doing so in §3.2.

A common surrogate loss for multiclass problems is a generalisation of the binary class hinge loss used for Support Vector Machines [8]:

$$\ell_H(f, y) = [1 - M(f, y)]_+ \tag{2}$$

where $[z]_+ = z$ for $z > 0$ and is 0 otherwise, and $M(f, y) = f_y - \max_{y' \neq y} f'_y$ is the *margin* for the vector $f \in \mathbb{R}^k$. Intuitively, the hinge loss is minimised by models that not only classify observations correctly but also maximise the difference between the highest and second highest scores assigned to the labels.

While there are other, consistent losses for SVMs [25, 15], these cannot scale up to structured estimations due to computational issues. For example, the multiclass hinge loss $\sum_{j \neq y}[1 + f_j(x)]_+$ is shown to be consistent in [15]. However, it requires evaluating f on all possible labels except the true y. This is intractable for structured estimation where the possible labels grow exponentially with the size of the structured output. Since the other known and consistent multiclass hinge losses have similar intractability we will only focus on the margin-based loss ℓ_H which can be evaluated quickly using techniques from dynamic programming, linear programming *etc.* [26, 24, 1].

2.2 Probabilistic Models and Losses

The scores given to labels by a general model $f : \mathcal{X} \to \mathbb{R}^k$ can be transformed into a conditional probability distribution $p(x; f) \in [0, 1]^k$ by letting

$$p_y(x; f) = \frac{\exp(f_y(x))}{\sum_{y \in \mathcal{Y}} \exp(f_y(x))}. \tag{3}$$

It is easy to show that under this interpretation the hinge loss for a probabilistic model $p = p(\cdot; f)$ is given by

$$\ell_H(p, y) = \left[1 - \ln \frac{p_y}{\max_{y' \neq y} p_{y'}}\right]_+$$

Another well known loss for probabilistic models, such as CRFs, is the log loss

$$\ell_L(p, y) = -\ln p_y.$$

This loss penalises models that assign low probability to likely instances labels and, implicitly, that assign high probability to unlikely labels.

We now propose a novel *hybrid loss* for probabilistic models that is a convex combination of the hinge and log losses

$$\ell_\alpha(p, y) = \alpha \ell_L(p, y) + (1 - \alpha)\ell_H(p, y) \tag{4}$$

where mixture of the two losses is controlled by a parameter $\alpha \in [0, 1]$. Setting $\alpha = 1$ or $\alpha = 0$ recovers the log loss or hinge loss, respectively. The intention is that choosing α close to 0 will emphasise having the maximum gap between the largest and second largest label probabilities while an α close to 1 will force models to prefer accurate probability assessments over strong classification.

3 Fisher Consistency For Classification

A desirable property for a loss is that, given enough data, the models obtained by minimising the loss at each observation will make predictions that are consistent with the true label probabilities at each observation.

Formally, we say vector $f \in \mathbb{R}^{|\mathcal{Y}|}$ is *aligned* with a distribution q over $\mathcal{Y}$ whenever maximisers of f are also maximisers for q. That is, when $\operatorname{argmax}_{y \in \mathcal{Y}} f_y \subseteq \operatorname{argmax}_{y \in \mathcal{Y}} q_y$. If, for all label distributions q, minimising the conditional risk $L(f) = \mathbb{E}_{y \sim q}[\ell(f, y)]$ for a loss ℓ yields a vector f^* aligned with q we will say ℓ is *Fisher consistent for classification* (FCC) [1] – or *classification*

[1] Note that the Fisher consistency for classification is weaker than Fisher consistency for density estimation. The former requires the same prediction only, while the latter requires the estimated density is the same as the true data distribution. In this paper, we focus on the former only.

calibrated [25]. This is an important property for losses since it is equivalent to the asymptotic consistency of the empirical risk minimiser for that loss [25, Theorem 2].

The standard multiclass hinge loss ℓ_H is known to be inconsistent for classification when there are more than two classes [15, 25]. The analysis in [15] shows that the hinge loss is inconsistent whenever there is an instance x with a *non-dominant* distribution – that is, $D_y(x) < \frac{1}{2}$ for all $y \in \mathcal{Y}$. Conversely, A distribution is *dominant* for an instance x if there is some y with $D_y(x) > \frac{1}{2}$. In contrast, the log loss used to train non-parametric CRFs is Fisher consistent for probability estimation – that is, the associated risk is minimised by the true conditional distribution – and thus ℓ_C is FCC since the minimising distribution is equal to $D(x)$ and thus aligned with $D(x)$.

3.1 Conditional Consistency of the Hybrid Loss

In order to analyse the consistency of the hybrid loss we introduce a more refined notion of Fisher consistency that takes into account the true distribution of class labels. If $q = (q_1, \dots, q_k)$ is a distribution over the labels $\mathcal{Y}$ then we say the loss ℓ is *conditionally FCC with respect to* q whenever minimising the conditional risk w.r.t. q, $L_q(f) = \mathbb{E}_{y\sim q}[\ell(f,y)]$ yields a predictor f^* that is consistent with q. Of course, if a loss ℓ is conditionally FCC w.r.t. q for *all* q it is, by definition, (unconditionally) FCC.

Theorem 1: Let $q = (q_1, \dots, q_k)$ be a distribution over labels and let $y_1 = \max_y q_y$ and $y_2 = \max_{y\neq y_1} q_y$ be the two most likely labels. Then the hybrid loss ℓ_α is conditionally FCC for q whenever $q_{y_1} > \frac{1}{2}$ or

$$\alpha > 1 - \frac{q_{y_1} - q_{y_2}}{1 - 2q_{y_1}}. \tag{5}$$

Proof: We use $L_\alpha(p, D) = \mathbb{E}_{y\sim D}[\ell\alpha(p,y)]$ and $\Delta(\mathcal{Y})$ to denote distributions over $\mathcal{Y}$. Since we a free to permute labels within $\mathcal{Y}$ we will assume without loss of generality that $D_1 = \max_{y\in\mathcal{Y}} D_y$ and $D_2 = \max_{y\neq 1} D_y$. The proof now proceeds by contradiction and assumes there is some minimiser $p = \operatorname{argmin}_{q\in\Delta(\mathcal{Y})} L_\alpha(q, D)$ that is not aligned with D. That is, there is some $y^* \neq 1$ such that $p_{y^*} \geq p_1$. For simplicity, and again without loss of generality, we will assume $y^* = 2$.

The first case to consider is when p_2 is a maximum and $p_1 < p_2$. Here we construct a q that "flips" the values of p_1 and p_2 and leaves all the values unchanged. That is, $q_1 = p_2$, $q_2 = p_1$ and $q_y = p_y$ for all $y = 3, \dots, k$. Intuitively, this new point is closer to D and therefore the CRF component of the loss will be reduced while the SVM loss won't increase. The difference in conditional risks satisfies

$$\begin{aligned} L_\alpha(p,D) - L_\alpha(q,D) &= \sum_{y=1}^{k} D_y.(\ell_\alpha(p,y) - \ell_\alpha(q,y)) \\ &= D_1.(\ell_\alpha(p,1) - \ell_\alpha(q,1)) \\ &\quad + D_2.(\ell_\alpha(p,2) - \ell_\alpha(q,2)) \\ &= (D_1 - D_2)(\ell_\alpha(q,2) - \ell_\alpha(q,1)) \end{aligned}$$

since $\ell_\alpha(p,1) = \ell_\alpha(q,2)$ and $\ell_\alpha(p,2) = \ell_\alpha(q,1)$ and the other terms cancel by construction. As $D_1 - D_2 > 0$ by assumption, all that is required now is to show that $\ell_\alpha(q,2) - \ell_\alpha(q,1) = \alpha \ln\frac{q_1}{q_2} + (1-\alpha)(\ell_H(q,2) - \ell_H(q,1))$ is strictly positive.

Since $q_1 > q_y$ for $y \neq 1$ we have $\ln\frac{q_1}{q_2} > 0$, $\ell_H(q,2) = \left[1 - \ln\frac{q_2}{q_1}\right]_+ > 1$, and $\ell_H(q,1) = \left[1 - \ln\frac{q_1}{q_y}\right]_+ < 1$, and so $\ell_H(q,2) - \ell_H(q,1) > 1 - 1 = 0$. Thus, $\ell_\alpha(q,2) - \ell_\alpha(q,1) > 0$ as required.

Now suppose that $p_2 = p_1$ is a maximum. In this case we show a slight perturbation $q = (p_1 + \epsilon, p_2 - \epsilon, p_3, \dots, p_k)$ yields a lower for $\epsilon > 0$. For $y \neq 1, 2$ we have $\ell_L(p,y) - \ell(q,y) = 0$ and since $p_2 > p_y$ and $q_1 > q_y$ thus $\ell_H(p,y) - \ell_H(q,y) = 1 - \ln\frac{p_y}{p_2} + 1 - \ln\frac{q_y}{q_1} = \ln\frac{p_2}{q_1} > 1 - \frac{q_1}{p_2} = -\frac{\epsilon}{p_2}$ since $-\ln x > 1 - x$ for $x \in (0,1)$ and $q_1 = p_1 + \epsilon = p_2 + \epsilon$. Therefore

$$\ell_\alpha(p,y) - \ell_\alpha(q,y) > -\epsilon\frac{(1-\alpha)}{p_1} \tag{6}$$

When $y = 1$, $\ell_L(p,1) - \ell_L(q,1) = -\ln\frac{p_1}{q_1} > \frac{q_1 - p_1}{p_1} = \frac{\epsilon}{p_1}$ and $\ell_H(p,1) - \ell_H(q,1) = (1 - \ln\frac{p_1}{p_2}) - (1 - \ln\frac{q_1}{q_2}) = \ln\frac{q_1}{q_2} = \ln\frac{p_1+\epsilon}{p_1-\epsilon}$ since $p_1 = p_2$. Thus $\ell_H(p,1) - \ell_H(q,1) > 1 - \frac{p_1-\epsilon}{p_1+\epsilon} = \frac{2\epsilon}{p_1+\epsilon}$. And so

$$\ell_\alpha(p,y) - \ell_\alpha(q,y) > \epsilon\left[\frac{\alpha}{p_1} + \frac{2(1-\alpha)}{p_1+\epsilon}\right] \tag{7}$$

Finally, when $y = 2$ we have $\ell_L(p,2) - \ell_L(q,2) = -\ln\frac{p_2}{q_2} > \frac{q_2 - p_2}{p_2} = \frac{-\epsilon}{p_1}$ and $\ell_H(p,2) - \ell_H(q,2) = (1 - \ln\frac{p_2}{p_1}) - (1 - \ln\frac{q_2}{q_1}) = \ln\frac{q_2}{q_1} > 1 - \frac{q_1}{q_2} = \frac{-2\epsilon}{p_1+\epsilon}$. Thus,

$$\ell_\alpha(p,2) - \ell_\alpha(q,2) > -\epsilon\left[\frac{\alpha}{p_1} + \frac{2(1-\alpha)}{p_1+\epsilon}\right]. \tag{8}$$

Putting the inequalities (6), (7) and (8) together yields

$$\begin{aligned} &\lim_{\epsilon\to 0}\frac{L_\alpha(p,D) - L_\alpha(q,D)}{\epsilon} \\ &> \lim_{\epsilon\to 0}(D_1 - D_2)\left[\frac{\alpha}{p_1} + \frac{2(1-\alpha)}{p_1+\epsilon}\right] - \sum_{y=3}^{k} D_y\frac{1-\alpha}{p_1} \\ &= \frac{D_1 - D_2}{p_1}(2-\alpha) - \frac{1 - D_1 - D_2}{p_1}(1-\alpha) \\ &= \frac{1}{p_1}(D_1 - D_2 + (1-\alpha)(2D_1 - 1)). \end{aligned}$$

Observing that since $D_1 > D_2$, when $D_1 > \frac{1}{2}$ the final term is positive without any constraint on α and when $D_1 < \frac{1}{2}$ the difference in risks is positive whenever

$$\alpha > 1 - \frac{D_1 - D_2}{1 - 2D_1} \tag{9}$$

completes the proof. □

Theorem 1 can be inverted and interpreted as a constraint on the conditional distributions of some data distribution D such that a hybrid loss with parameter α will yield consistent predictions. Specifically, the hybrid loss will be consistent if, for all $x \in \mathcal{X}$ such that $q = D(x)$ has no dominant label (*i.e.*, $D_y(x) \leq \frac{1}{2}$ for all $y \in \mathcal{Y}$), the gap $D_{y_1}(x) - D_{y_2}(x)$ between the top two probabilities is larger than $(1-\alpha)(1-2D_{y_1}(x))$. When this is not the case for some x, the classification problem for that instance is, in some sense, too difficult to disambiguate. In this sense, the bound can be seen as a property on distributions akin to Tsybakov's noise condition [6]. Making this analogy precise is the focus of ongoing work.

3.2 Parametric Consistency

Since Fisher consistency is defined point-wise on observations, it is not directly applicable to parametric models as these enforce inter-observational constraints (*e.g.* smoothness). Abstractly, assuming parametric hypotheses can be seen as a restriction over the space of allowable scoring functions. When learning parametric models, risks are minimised over some subset $\mathcal{F}$ of functions from $\mathcal{X} \to \mathbb{R}^{\mathcal{Y}}$ instead of all possible functions. We now show that, given some weak assumptions on the hypothesis class $\mathcal{F}$, a loss being FCC is a necessary condition if the loss is also to be $\mathcal{F}$-consistent.

We say a loss ℓ is $\mathcal{F}$-consistent if, for any distribution, minimising its associated risk over $\mathcal{F}$ yields a hypothesis with minimal 0-1 loss in $\mathcal{F}$.[2] Recall that the risk of a hypothesis $f \in \mathcal{F}$ associated with a loss ℓ and distribution D over $\mathcal{X} \times \mathcal{Y}$ is $L_D(f) = \mathbb{E}_D\left[\ell(y, f(x))\right]$ and its 0-1 risk or misclassification error is $e_D(f) = \mathbb{P}_D\left[y \neq \operatorname{argmax}_{y' \in \mathcal{Y}} f_{y'}(x)\right]$. Formally then, given a function class $\mathcal{F}$ we say *ℓ is $\mathcal{F}$-consistent* if, for all distributions D,

$$L_D(f^*) = \inf_{f \in \mathcal{F}} L_D(f) \implies e_D(f^*) = \inf_{f \in \mathcal{F}} e_D(f). \qquad (10)$$

We need a relatively weak condition on function classes $\mathcal{F}$ to state our theorem. We say a class $\mathcal{F}$ is *regular* if the follow two properties hold: 1) For any $g \in \mathbb{R}^{\mathcal{Y}}$ there exists an $x \in \mathcal{X}$ and an $f \in \mathcal{F}$ so that $f(x) = g$; and 2) For any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ there exists an $f \in \mathcal{F}$ so that $y = \operatorname{argmax}_{y' \in \mathcal{Y}} f_{y'}(x)$. Intuitively, the first condition says that for any distribution over labels there must be a function in the class which models it perfectly on some point in the input space. The second condition requires that any mode can be modelled on any input. Importantly, these properties are fairly weak in that they do not say anything about the constraints a function class might put on relationships between distributions modelled on different inputs.

Theorem 2: For regular function classes $\mathcal{F}$ any loss that is $\mathcal{F}$-consistent is necessarily also Fisher Consistent for Classification (FCC).

Proof: The proof is by contradiction. We assume we have a regular function class $\mathcal{F}$ and a loss ℓ which is $\mathcal{F}$-consistent but not FCC. That is, (10) holds for ℓ but there exists a distribution p over $\mathcal{Y}$ such that there is a $g \in \mathbb{R}^{\mathcal{Y}}$ which minimises the conditional risk $L_q(g)$ but $\operatorname{argmax}_{y \in \mathcal{Y}} g_y \neq \operatorname{argmax}_{y \in \mathcal{Y}} q_y$.

By the assumption of the regularity of $\mathcal{F}$ there is an $x \in \mathcal{X}$ and a $f \in \mathcal{F}$ so that $f(x) = g$. We now define a distribution D over $\mathcal{X} \times \mathcal{Y}$ that puts all its mass on the set $\{x\} \times \mathcal{Y}$ so that $D(x, y) = p_y$. Since this distribution is concentrated on a single x its full risk and conditional risk on x are the same. That is, $L_D(\cdot) = L_p(\cdot)$. Thus,

$$L_D(f) = L_p(f) = \inf_{f' \in \mathcal{F}} L_p(f') = \inf_{f' \in \mathcal{F}} L_D(f')$$

By the assumption of $\mathcal{F}$-consistency, since f is a minimiser of L_D it must also minimise e_D. Once again, the construction of D means that $e_D(f) = e_p(g) = \mathbb{P}_{y \sim p}\left[y \neq \operatorname{argmax}_{y' \in \mathcal{Y}} g_y\right] = 1 - p_{y_g}$ where $y_g = \operatorname{argmax}_y g_y$ is the label predicted by g. However,

$$e_D(f) = e_p(g) = 1 - p_{y_g} > 1 - p_{y^*}$$

since $y_* = \operatorname{argmax}_y p_y \neq \operatorname{argmax} g_y = y_g$.

By the second regularity property, there must also be an $\hat{f} \in \mathcal{F}$ such that $\operatorname{argmax}_y \hat{f}_y(x) = y^*$ so that $e_D(f) > \inf_{f' \in \mathcal{F}} e_D(f') = e_D(\hat{f}) = 1 - p_{y^*}$. Thus, we have shown that there exists a distribution D so $f \in \mathcal{F}$ is a minimiser of the risk L_D but is not a minimiser of the misclassification rate e_D which contradicts the assumption of the $\mathcal{F}$-consistency of ℓ. Therefore, ℓ must be FCC.

□

4 Generalisation Bound

We now give a PAC-Bayesian bound [18] for the generalisation error e_D of the hybrid model that can be specialised to recover a bound for the multiclass hinge loss.

Theorem 3 (Generalisation Margin Bound): For any data distribution D, for any prior P over w, for any w, any $\delta \in (0, 1]$ and for any $\gamma > 0$ and any $\alpha \in [0, 1)$, with probability at least $1 - \delta$ over random samples S from D with m instances, there exists a constant c, such that

$$\begin{aligned} e_D \leq & \mathbb{P}_{(x,y) \sim S}(\mathbb{E}_Q(M(w', y)) \leq \gamma) \\ & + O\left(\sqrt{\frac{\frac{||w||^2}{2c(1-\alpha)\gamma^2} \ln(m|\mathcal{Y}|) + \ln m + \ln \delta^{-1}}{m}}\right). \end{aligned}$$

Proof: By choosing the weight prior $P(w) = \frac{1}{Z}\exp(-\frac{\|w\|^2}{2})$ and the posterior $Q(w') = \frac{1}{Z}\exp(-\frac{\|w' - w\|^2}{2})$, one can show $e_D = \mathbb{P}_D(\mathbb{E}_Q M(w', y) \leq 0)$ by symmetry argument proposed in [14, 17]. Applying the PAC-Bayes margin bound [13, 27] and knowing the margin threshold $\gamma' \leq c(1-\alpha)\gamma$ and $\mathrm{KL}(Q||P) = \frac{||w||^2}{2}$ yields the theorem.

□

Setting $\alpha = 0$ in the above bound recovers a margin bound for SVMs (see [13] for an averaging classifiers of SVMs, and [27] for structured case). Unfortunately, one cannot set $\alpha = 1$ to achieve a PAC-Bayes bound for a

[2] While this is simpler and stronger than the usual asymptotic notation of consistency [16] it most readily relates to FCC and suffices for our discussion since we are only establishing that FCC is a necessary condition.

pure log loss classifier in this manner due the the $(1-\alpha)^{-1}$ dependence. However, to our knowledge, we are not aware of any PAC-Bayes bound on the generalisation error for log loss.

A similar, alternative bound to capture how much the margin violates for the hybrid loss is as follows:

Theorem 4 (Margin Violation Bound): For any data distribution D, for any prior P over w, for any $\delta \in (0,1]$ and $\alpha \in [0,1)$ and for any $\gamma \geq 0$, for any w, with probability at least $1-\delta$ over random samples S from D with m instances, we have

$$\mathbb{E}_D\left[\Big(\gamma - M(w,y)\Big)_+\right] \leq \frac{1}{m}\sum_{i=1}^{m}\Big(\gamma - M(w,y_i)\Big)_+$$
$$+\frac{1}{(1-\alpha)}\left(\alpha\sqrt{\frac{1}{m}}+\sqrt{\frac{\ln\frac{1}{P(w)}+\ln A(\alpha,w)+\ln\frac{1}{\delta(1-e^{-2})}}{2m}}\right)$$

where

$$\begin{aligned}R(\alpha,w) =&\alpha\,\mathbb{E}_D\Big[-\ln p(y|x;w)\Big]\\ &+(1-\alpha)\,\mathbb{E}_D\left[\Big(\gamma - M(w,y)\Big)_+\right],\\ R_S(\alpha,w) =&\Bigg[\alpha\frac{\sum_{i=1}^{m}-\ln p(y_i|x_i;w)}{m}\\ &+(1-\alpha)\frac{\sum_{i=1}^{m}\Big(\gamma - M(w,y_i)\Big)_+}{m}\Bigg],\\ A(\alpha,w) =&\mathbb{E}_{s\sim D^m}\, e^{2m(R(\alpha,w)-R_S(\alpha,w))^2}.\end{aligned}$$

Here A is upper bounded independently of D. For example, for a zero-one loss, it is upper bounded by $m+1$ (see [9]). The theorem is a special case of Theorem 6 in Appendix A.

The theorem gives a bound on the true margin violation of the hybrid model. It can be used to access how reliable a large margin threshold is — larger margin threshold implies a tighter error bound in Theorem 4.

5 Experiments

The analysis of the hybrid loss suggests it should be able to outperform the hinge loss due to its improved consistency on distributions with non-dominant labels. Furthermore, it should also make more efficient use of data than log loss on distributions with dominant labels. These hypotheses were confirmed by applying the hybrid, log and hinge losses to a number of synthetic multiclass data sets in which the data set size and proportion of examples with non-dominant labels are carefully controlled.

We also compared the hybrid loss with the log and hinge losses on several real structured estimation problems and observed that the hybrid loss regularly outperforms the other losses and consistently performs at least as well as the better of the log and hinge losses on any problem.

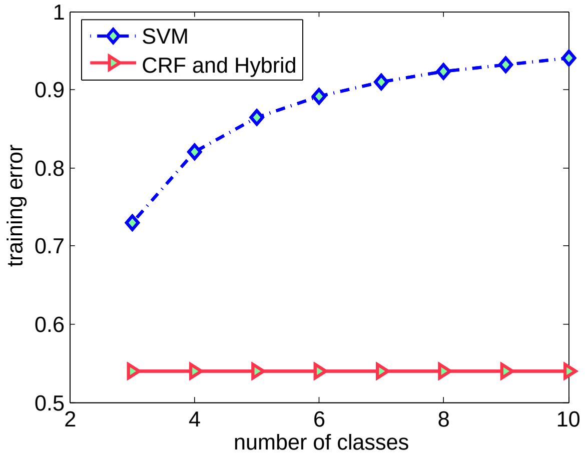


Fig. 1. Training Error with various number of classes. $\alpha = 0.5$ for the hybrid loss.

5.1 Multiclass Classification

Two types of multiclass simulations were performed. The first examined the performances of the hybrid, log and hinge losses when no observations had a dominant label. That is all observations were drawn from a D with $D_y(x) < 1/2$ for all labels y. The second experiment considered distributions with a controlled mixture of observations with dominant and non-dominant labels.

5.1.1 Non-dominant Distributions

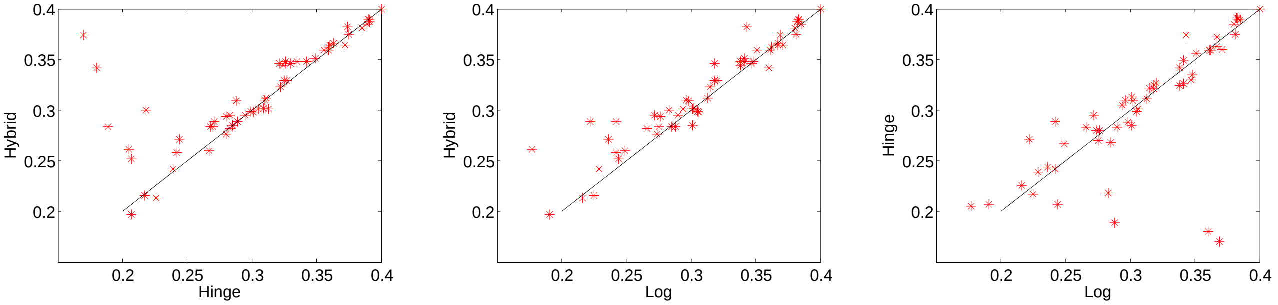
To make the experiment as simple as possible, we considered an observation space of size $|\mathcal{X}| = 1$ and focused on varying the number of labels and their probabilities. The label set $\mathcal{Y}$ took the sizes $|\mathcal{Y}| = 3,4,5,\ldots,10$. One label $y^* \in \mathcal{Y}$ was assigned probability $D_{y^*}(x) = 0.46$ and the remainder are given an equal portion of 0.54 (*e.g.*, in the 3 class case the other labels each have probability 0.27, and in the 10 class case, 0.06). Note that this means for all the label set sizes, the gap $D_{y^*}(x) - D_y(x)$ is at least 0.19 which is always greater than $(1-\alpha)(1-2D_{y^*}(x)) = 0.04$ so the hybrid consistency condition (5) is always met.

Features were a constant value in $\mathbb{R}^2$ as were the parameter vectors $w_y \in \mathbb{R}^2$ for $y \in \mathcal{Y}$. Models were found using LBFGS [5]. The resulting training errors for hinge, log and hybrid losses are plotted in Figure 1 as a function of the number of labels. As we can clearly see, the hinge loss error increases as the number of classes increases, whereas the errors for the log and the hybrid losses remain a constant $(1 - D_{y^*}(x))$, in concordance with the consistency analysis.

5.1.2 Mix of Non-dominant and Dominant Distributions

The second synthetic experiment examined how the three losses performed given various training set sizes (denoted by m) and various proportions of instances with non-dominant distributions (denoted by ρ).

We generated 60 different data sets, all with $\mathcal{Y} = \{1,2,3,4,5\}$, in the following manner: Instances came



(a) Hybrid v.s. Hinge (31/15)

(b) Hybrid v.s. Log (34/15)

(c) Hinge v.s. Log (30/23)

Fig. 2. Performance of the hybrid, hinge, and log losses on non-dominant/dominant mixtures. Points denote pairs of test accuracies for models trained on one of 60 data sets using the losses named on the axes. Score (a/b) denotes the vertical loss with a wins and b losses (ties not counted).

from either a non-dominant class distribution or a dominant class distribution. In the non-dominant class case, $x \in \mathbb{R}^1 00$ is set to a predefined, constant, non-zero vector and its label distribution is $D_1(x) = 0.4$ and $D_y(x) = 0.15$ for $y > 1$. In the dominant case, each dimension x_i was drawn from a normal distribution $N(\mu = 1 + y, \sigma = 0.6)$ depending on the class $y = 1, \ldots, 5$. The proportion ρ ranged over 10 values $\rho = 0.1, 0.2, 0.3, \ldots, 1$ and for each ρ, test and validation sets of size 1000 were generated. Training set sizes of $m = 30, 60, 100, 300, 600, 1000$ were used for each ρ value for a total of 60 training sets. The optimal regularisation parameter λ and hybrid loss parameter α were selected using the validation set for each loss on each training set. Then models with parameters $w_y \in \mathbb{R}^{100}$ for $y \in \mathcal{Y}$ were found using LBFGS [5] for each of the three losses on each of the 60 training sets and then assessed using the test set.

The results are summarised in Figure 2. Each point shows the test accuracy for a pair of losses. The predominance of points above the diagonal lines in a) and b) show that the hybrid loss outperforms the hinge loss and the log loss in most of the data sets. while the log and hinge losses perform competitively against each other.

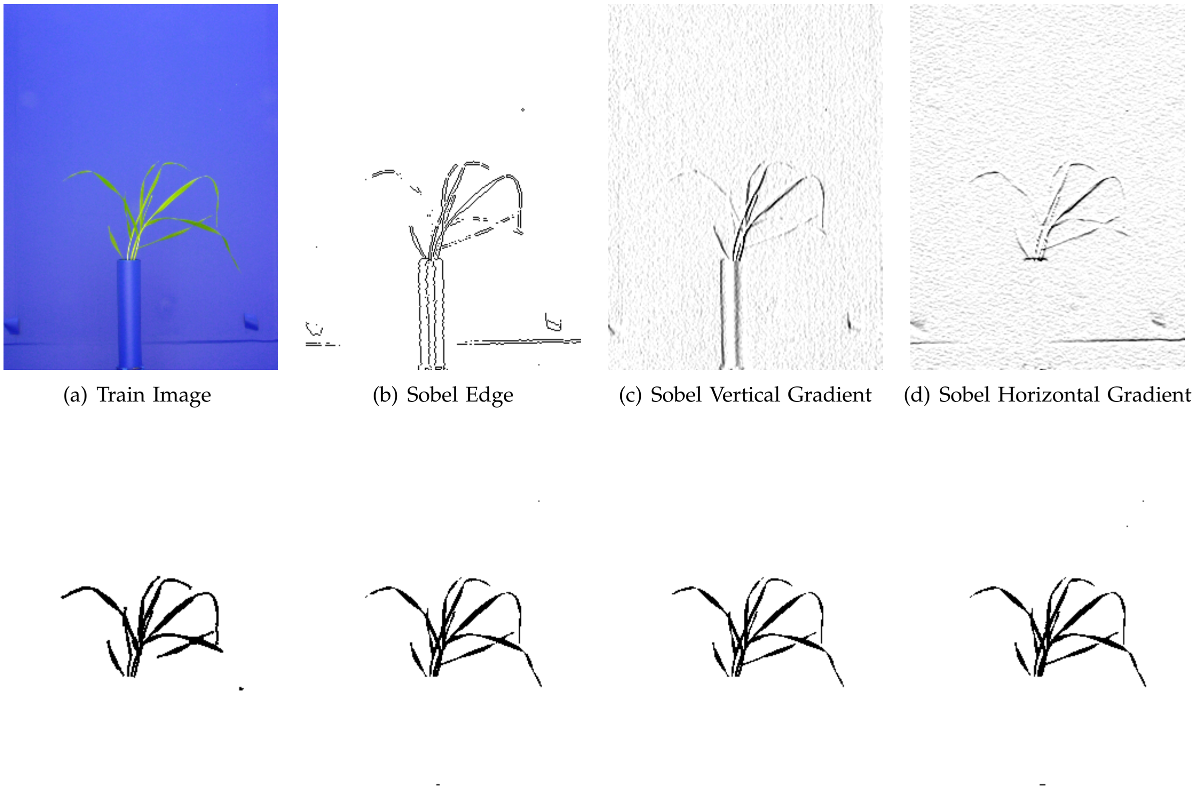
5.1.3 Plant Segmentation

Here we consider a real world problem for monitoring plants in a controlled botany laboratory environment where botanists want to record and track the details of the growth of plants in an automated manner. Plants are rotating around several cameras, thus pictures from different angles are obtained at different times. Segmenting plants from their background is a fundamental step for subsequent tasks such as automatic matching, tracking, measuring and 3D modelling. Plant segmentation is essentially a classification problem — an image pixel is either on the plant (labelled as 1) or not (labelled as -1). Assigning labels $\{\pm 1\}$ to pixels can be done via energy minimisation such as graph cut [4, 3, 10], which often requires to set parameters manually. To reduce human interference and make use of the large amount of available data, we will use supervised learning that can automatically determine these parameters.

The botany laboratory provided 10 manually labelled images with a resolution of 1280×960 pixels which we down-sampled to 320×240. For each pixel of the down-sampled images, we extracted features $x \in \mathbb{R}^9$ where $x = (r, g, b, gv_r, gh_r, gv_g, gh_g, gv_b, gh_b)$. Here, r, g, b are the pixel values in RGB channels, gv_r, gh_r are vertical and horizontal gradients determined by a 3×3 Sobel operator [20] on the R (red) channel (see Figure 3(c) and 3(d)). Likewise, gv_g, gh_g, gv_b, gh_b are vertical and horizontal gradients on the other channels. The 10 images were split into 3 sets: training (4 images), validation (3 images) and test sets (3 images). All three losses (hinge, log and hybrid) were trained on 4 different data sets with features derived from 1 image, 2 images, 3 images and 4 images before being run on the the validation set (3 images) and the test set (3 images). A example of one of the training images is shown in Figure 3(a) with its label depicted in Figure 3(e). Here we see that some thin parts of plants are not labelled as the plant — *e.g.* part of the right bottom leaf in Figure 3(a) is mislabelled as background. The mislabelled pixels are considered noise. We expect an ideal classifier to be robust to a certain degree of noise since labelling every pixel accurately is impractical due to human error. As seen in Figure 3(f),3(g),3(h), the supervised learning classifiers for the three losses recover the mislabelled leaf with the correct prediction.

As an alternative to supervised learning, we consider the widely used method of graph cut energy minimisation for classification [4, 3, 10]. This approach is used to rectify shortcomings of traditional edge detectors such as the Sobel operator which may not be able to distinguish the edge of plants and the edges of the wall and the vase (see Figure 3(b)). This technique assigns labels to each pixel from a predefined set of classes of size k. We ran graph cut[3] with and and have presented the results in Figure 4(b), 4(c) and 4(d). Ideally, one should let $k = 2$, because there are two classes — a pixel either belongs to the plant or to the background. However,

[3] the code can be downloaded from http://www.wisdom.weizmann.ac.il/~bagon/matlab_code/GCmex1.3.tar.gz


(a) Train Image (b) Sobel Edge (c) Sobel Vertical Gradient (d) Sobel Horizontal Gradient

(e) Train Label (f) Hinge Segmentation (g) Log Segmentation (h) Hybrid Segmentation

Fig. 3. Illustration of the supervised learning for plant segmentation. The results of sobel operator are done on on the red channel. Similar results are obtained on other channels. The gradients have been enhanced by a factor of 10 for visualisation purpose. A 9-dimensional vector consisting of RGB pixels, sobel gradients on all channels represents the feature of each pixel. The features are used for training and predicting labels for all pixels. The segmentation results by 3 losses are shown in (f),(g),(h).

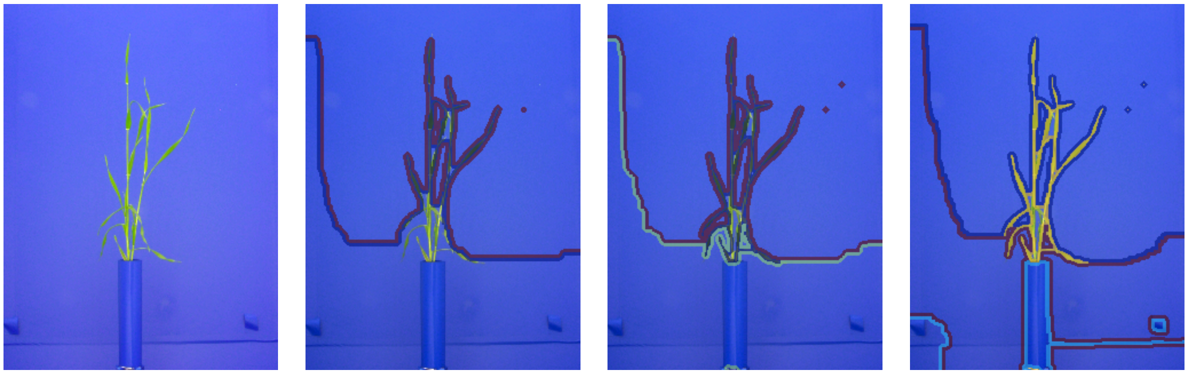

(a) Validation Image (b) Graph Cut k=2 (c) Graph Cut k = 3 (d) Graph Cut k =4

Fig. 4. Graph Cut results on a plant image. One needs to pre-specify the number of classes denoted by k for graph cut. Here we let $k = 2, 3, 4$. Ideally $k = 2$ for there are only plant and the background. However, graph cut yields poor result when $k = 2$. Even increasing k up to 4, there still are some isolated segments are mis-classified as plant (*e.g.* see the bottom of the vase and the two small dots in the right top area).

graph cut may yield poor result with $k = 2$ (see Figure 4(b)). However, to make the graph cut approach more competitive, we increased the number of classes to $k = 3$ and 4. Unfortunately, even with $k = 4$, there still are some isolated segments which are misclassified as plant — for example, the bottom of the vase and the two small dots in the right top area. As well as these shortcomings in its performance, using graph cut for inference has other undesirable properties relative to supervised classification. Choosing the best k for good segmentation and merging the results requires some care as does the definition of the energy function. Also, unlike

TABLE 1
Accuracy on plant segmentation. I: the number of training images. m: the number of training examples/pixels. Loss: choice of loss functions. Acc Valid: the accuracy on the validation set. Acc Test i: the accuracy on i-th test image. The largest accuracies among 3 losses are in boldface.

I	m	Loss	Acc Valid	Acc Test 1	Acc Test 2	Acc Test 3
1	76800	Hinge	99.12	99.43	98.26	**99.10**
		Log	99.12	**99.45**	**98.28**	**99.10**
		Hybrid	**99.13**	99.42	98.24	99.08
2	153600	Hinge	99.12	99.37	98.14	**99.11**
		Log	99.12	99.36	98.09	**99.11**
		Hybrid	**99.13**	**99.40**	**98.21**	**99.11**
3	230400	Hinge	99.12	99.41	98.21	**99.10**
		Log	99.12	99.39	98.17	**99.10**
		Hybrid	**99.13**	**99.43**	**98.28**	**99.10**
4	307200	Hinge	99.12	99.46	98.29	**99.10**
		Log	99.11	99.38	98.14	**99.10**
		Hybrid	**99.13**	**99.48**	**98.34**	99.09

supervised learning, graph cut does not fully exploit the label information available in the training data.

We assume (x, y) pairs are drawn i.i.d. from a fixed but unknown distribution $D(x, y)$ and model $p_y(x; f)$ as (3) where $f_y(x; w) = \langle w, \phi(x, y)\rangle$. The task is now to learn a good parameter w that can assign probabilities consistent with the underlying distribution D. We train 3 losses on 4 training sets (*i.e.* with 1 image, ... , 4 images) by LBFGS [5]. Once again, the optimal regularisation parameter λ and hybrid loss parameter α were selected using the validation set for each loss on each training set. We report the results in Table 1. As can been seen, when we train on 1 image (*i.e.* 76800 examples), the log loss slightly outperforms the hinge loss and the hybrid loss. As we increase the training data size the hybrid loss starts outperforming the other two losses. Also, with increased training size, the overall test accuracy of the hybrid loss is improving. This suggests when we have more training images the performance could be further improved. Since the reported accuracy is quite high for all methods, large additive improvements in accuracy are impossible. However, if we consider when accuracy goes from 99.36% to 99.40%, the error goes from 0.64% to 0.60% which is a relative decrease (*i.e.* improvement) of $0.04/0.64 = 6.25\%$.

From Theorem 3 we know that, with probability at least 99.9% (*i.e.* $\delta = 0.001$), the term within the big O notation is tight — it is $0.0198, 0.0143, 0.0118, 0.0103$ for 1 to 4 images respectively, evaluated with $\gamma = 1, \|w\| = 1, c = 1, \alpha = 0.5, k = 2$. This may explain why the test accuracies on different sets are quite stable (around 99%).

Note that graph cut uses the correlation between neighbouring pixels, whereas the supervised learning method treats each pixel i.i.d. ignoring the within image dependency. To model the within image dependency, we could model an entire image as x instead of per pixel. However, this model would be difficult to learn as the higher order parameter space would be very large. Moreover, it will require a much larger amount of training images to estimate the model with a tight generalisation bound (see Theorem 3) since $k = 2^{76800}$ — the number of possible pixel labelling for a single image — is enormous. Tractable structured estimation for the plant segmentation problem is ongoing research.

5.2 Structured Estimation

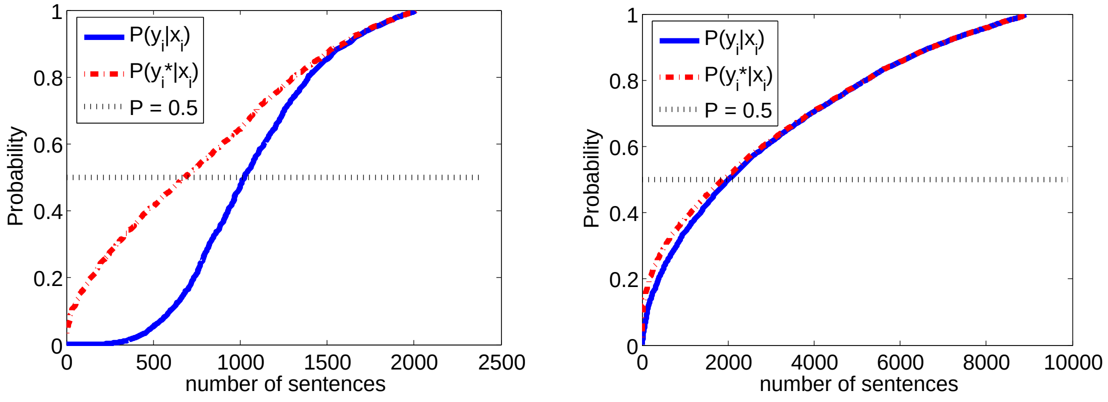
Unlike the general multiclass case, structured estimation problems have a higher chance of non-dominant distributions because of the very large number of labels as well as ties or ambiguity regarding those labels. For example, in text chunking, changing the tag one phrase while leaving the rest unchanged should not drastically change the probability predictions – especially when there are ambiguities. Because of the prevalence of non-dominant distributions, we expect that training models using a hinge loss to perform poorly on these problems relative to training with hybrid or log losses.

5.2.1 CONLL2000 Text Chunking

Our first structured estimation experiment is carried out on the CONLL2000 text chunking task [7]. The data set has 8936 training sentences and 2012 testing sentences with 106978 and 23852 phrases (a.k.a. chunks) respectively. The task is to divide a text into syntactically correlated parts of words such as noun phrases, verb phrases, and so on. For a sentence with L chunks, its label consists of the tagging sequence of all its chunks, *i.e.* $y = (y^1, y^2, \ldots, y^L)$, where y^i is the chunking tag for chunk i. As commonly used in this task, the label y is modelled as a 1D Markov chain to account for the dependency between adjacent chunking tags (y_i^j, y_i^{j+1}) given observation x_i. Clearly, the model has exponentially many possible labels, which suggests there are many non-dominant classes.

Since the true underlying distribution is unknown, we train a CRF[4] on the training set and then apply the trained model to both testing and training datasets to get an estimate of the conditional distributions for each instance. We sort the sentences x_i from highest

[4] using the feature template from the CRF++ toolkit [11], and the CRF code from Leon Bottou [2].



(a) the testing set

(b) the training set

Fig. 5. Estimated probabilities of the true label $D_{y_i}(x_i)$ and most likely label $D_{y_i^*}(x_i)$. Sentences are sorted according to $D_{y_i}(x_i)$ and $D_{y_i^*}(x_i)$ respectively in ascending order. $D = 1/2$ is shown as the straight black dot line. About 700 sentences out of 2012 in the testing set and 2000 sentences out of 8936 in the training set have no dominant class.

TABLE 2
Accuracy, precision, recall and F1 Score on the CONLL2000 text chunking task.

Train Portion	Loss	Accuracy	Precision	Recall	F1 Score
0.1	Hinge	91.14	85.31	85.52	85.41
	Log	92.05	87.04	87.01	87.02
	Hybrid	**92.07**	**87.17**	**86.93**	**87.05**
1	Hinge	94.61	91.23	91.37	91.30
	Log	95.10	92.32	91.97	92.15
	Hybrid	**95.11**	**92.35**	**92.00**	**92.17**

to lowest estimated probability on the true chunking label y_i given x_i. The result is plotted in Figure 5, from which we observe the existence of many non-dominant distributions — about 1/3 of the testing sentences and about 1/4 of the training sentences.

We split the data into 3 parts: training (20%), testing (40%) and validation (40%). The regularisation parameter λ and the weight α were determined via parameter selection using the validation set. To see the performance with different training sizes, we took part of the training data to learn the model and gathered statistics on the test set. The accuracy, precision, recall and F1 Score on test set are reported in Table 3 when using 10% and 100% of the training set. The hybrid loss outperforms both the hinge loss and the log loss (albeit marginally).

5.2.2 baseNP Chunking

A similar methodology to the previous experiment is applied to the BaseNP data set [11]. It has 900 sentences in total and the task is to automatically classify a chunking phrase is as baseNP or not. We split the data into 3 parts: training (20%), testing (40%) and validation (40%). Once again, λ and α are determined via model selection on the validation set. We report the test accuracy, precision, recall and F1 Score in Table 3 for training on increasing proportion of the training set. The hybrid outperforms the other two losses on all measures.

5.2.3 Japanese named entity recognition

Finally, we used a multiclass data set containing 716 Japanese sentences and 17 annotated named entities [11]. The task is to locate and classify proper nouns and numerical information in a document into certain classes of named entities such as names of persons, organizations, and locations. We train all 3 models on 216 sentences and test on 500 sentences with the default parameters found in Bottou's CRF code. The extra parameter α is selected for the smallest test error. The result is reported in Table 4. Once again, the hybrid loss outperforms the others two losses.

5.2.4 Joint Image Object Categorization

Our final experiment is joint image categorization. The task is to categorize pre-segmented image areas by considering their dependency across the image segments. We use the well-known Corel dataset [21], which has 100 images and 7 classes: *hippo*, *polar bear*, *water*, *snow*, *vegetation*, *ground*, and *sky*. This is a very challenging task since there are 7^n many possible labels for an image with n segments.

The ground truth segmentations provided from the dataset are used as pre-segmented object regions. Therefore each image contains one or multiple objects/regions. We use 56 images for training, and 20 images for testing. The rest images are excluded for too small or too many objects. The graphical model of an images is shown in Figure 6, where each segment is a node and edges

TABLE 3
Accuracy, precision, recall and F1 Score on the baseNP chunking task for training on increasing portions of training set.

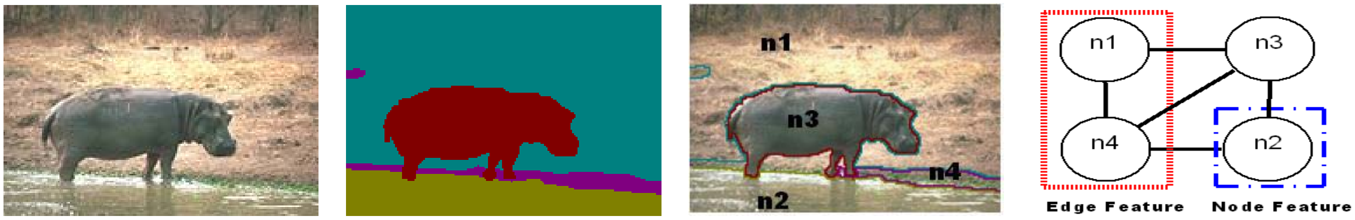
Train Portion	Loss	Accuracy	Precision	Recall	F1 Score
0.1	Hinge	88.48	71.70	75.96	73.77
	Log	90.86	81.09	78.96	80.01
	Hybrid	**90.90**	**81.23**	**79.09**	**80.15**
1	Hinge	94.64	87.58	88.30	87.94
	Log	95.21	90.07	88.89	89.48
	Hybrid	**95.24**	**90.12**	**88.98**	**89.55**

TABLE 4
Accuracy, precision, recall and F1 Score on the Japanese named entity recognition task.

Loss	Accuracy	Precision	Recall	F1 Score
Hinge	95.63	73.24	64.37	68.52
Log	95.92	78.22	64.85	70.91
Hybrid	**95.95**	**79.02**	**65.32**	**71.52**

TABLE 5
Image object categorization. SVM-Linear: non-structured SVM using Linear kernel. SVM-RBF: non-structured SVM using RBF kernel. SVM-Struct: structured SVM using Linear kernel; CRF: CRF on sparse graph using MAP estimator with LBP inference. Hybrid: our structured hybrid model on sparse graph.

Dataset	$\lvert\mathcal{Y}\rvert$	SVM-linear	SVM-RBF	SVM-Struct	CRF	Hybrid
Corel	7^n	58.62	65.52	**89.66**	86.21	**89.66**



(a) Raw image (b) Segmentation (c) Objects (d) Features

Fig. 6. An illustration of the image objects, graph and features. **(a)** The raw hippo image. **(b)** The segmentation result. **(c)** The objects. **(d)** Node and edge features: node feature encodes the object characteristics, while the edge feature encodes the interaction between objects.

capture the adjacency dependency.

5.2.4.1 Features: Any image with n segments and labels is represented as $(x, y) = \{(x^i, y^i)\}_{i=1}^n$, where x^i and y^i are the i-th segment and corresponding label. We assume that global feature $\phi(x, y)$ is decomposed over singleton terms $\phi_i(x^i, y^i)$, $\forall i, 1 \le i \le n$, as well as over pairwise terms $\phi_{ij}(x^i, y^i)$, $\forall (i, j) \in \mathcal{A}_x$, where $\mathcal{A}_x$ is the set of adjacent segments in x

$$\phi(x, y) = \sum_i \phi_i(x^i, y^i) + \sum_{(i,j)\in\mathcal{A}} \phi_{ij}(x^i, y^i, y^j). \qquad (11)$$

We assume ϕ_i is composed by a tensor product of instance and label feature functions, given by $\phi_i(x, y) = \varphi_i(x) \otimes y_i$ where φ_i is the raw node feature depending only on the observed segmented image. Similarly $\phi_{ij}(x, y) = \varphi_{ij}(x) \otimes y_{ij}$, where φ_{ij} is the raw edge feature depending only on the observation as well, and $y^{ij} := [y^i\ y^j]$. φ_i and φ_{ij} are assembled from

φ_1 We extract a well known texton feature vector [23] from each patch, hence every pixel is represented by a texton vector. The node feature for an object is the empirical mean of the texton vector of pixels. The raw node feature $\varphi_i(x) = [1\ \varphi_1(x^i)]$.

φ_2 We use the mean of the boosted texton probability density of all interior and boundary pixels of the objects as their edge feature. The raw edge feature $\varphi_{ij}(x) = [1\ \varphi_2(x^i)\ \varphi_2(x^j)]$.

As shown in Figure 6, the graph model is very general. Exactly computing CRF gradient involving computing an expectation is NP hard. The common way is to run approximation such as Loopy Belief Propagation (LBP) or sampling. We use LBP for CRF here. All structured algorithms use the same node and edge features. Non-structured algorithms use the node feature only. As shown in Table 4, structured algorithms outperform the non-structured ones as expected. And it is interesting to see that structured SVM outperforms CRF. We conjecture that is because CRF decision hyperplane becomes less accurate due to approximated gradient (*i.e.*, the expectation of the feature). Whereas structured SVM needs only an argmax operation which is more efficient and perhaps more reliable.

6 Conclusion and Discussion

We have provided theoretical and empirical motivation for the use of a novel hybrid loss for multiclass and structured prediction problems which can be used in place of the more common log loss or multiclass hinge loss. This new loss attempts to blend the strength of purely discriminative approaches to classification, such as Support Vector machines, with probabilistic approaches, such as Conditional Random Fields. Theoretically, the hybrid loss enjoys better consistency guarantees than the hinge loss while experimentally we have seen that the addition of a purely discriminative component can improve accuracy when data is less prevalent.

6.1 Future Work

Theoretically, we expect that some stronger sufficient conditions on α are possible since the bounds used to establish Theorem 1 are not tight. Our conjecture is that a necessary and sufficient condition would include a dependency on the number of classes. We are also investigating connections between α and the multiclass Tsybakov noise condition [6].

To our knowledge, the notion of a regular function class for the purposes of consistency analysis is a novel one. Characterisations of this property for various existing parametric models would make testing for regularity easier.

One current limitation of the hybrid model is the use of a single, fixed α for all observations in a training set. One interesting avenue to explore would be trying to dynamically estimate a good value of α on a per-observation basis. This may further improve the efficacy of the hybrid loss by exploiting the robustness of SVMs (low α) when the label distribution for an observation has a dominant class but switching to probability estimation via CRFs (high α) when this is not the case.

For the future work of the plant task, we would like to classify finer parts of the plant such as stem, leaf, leaf tip and so on. The labels of these parts have strong dependency of the label assignments on their neighbouring parts. Hence treating it as a structured prediction problem might produce superior result. How to efficiently learn such a complex model with a stable and consistent prediction is future work.

Acknowledgments

The main work of this paper is done when Q. Shi is at NICTA. NICTA is funded by the Australian Government as represented by the Department of Broad- band, Communications and the Digital Economy and the Australian Research Council through the ICT Center of Ex- cellence program.

This research was partly supported under Australian Research Council Discovery Projects funding scheme (project DP0988439) and supported by The Australian Centre for Visual Technologies and The Computer Vision group of The University of Adelaide.

References

[1] G. Bakir, T. Hofmann, B. Schölkopf, A. Smola, B. Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data*. MIT Press, Cambridge, Massachusetts, 2007.

[2] Leon Bottou. Stochastic gradient descent for conditional random fields (crfs), 2010. v1.3 http://leon.bottou.org/projects/sgd.

[3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[5] Richard H. Byrd, Jorge Nocedal, and Robert B. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 1994.

[6] Di-Rong Chen and Tao Sun. Consistency of multiclass empirical risk minimization based on convex loss. *JMLR*, 7:2435–2447, 2006.

[7] CoNLL. Shared task for conference on computational natural language learning (conll-2000), 2000. http://www.cnts.ua.ac.be/conll2000/chunking/.

[8] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. In N. Cesa-Bianchi and S. Goldman, editors, *Proc. Annual Conf. Computational Learning Theory*, pages 35–46, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[9] Pascal Germain, Alexandre Lacasse, Francois Laviolette, and Mario Marchand. Pac-bayesian learning of linear classifiers. In *ICML*, 2008.

[10] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

[11] Taku Kudo. Crf++: Yet another crf toolkit, 2010. v0.53 http://crfpp.sourceforge.net/.

[12] J. D. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic modeling for segmenting and labeling sequence data. In *Proc. Intl. Conf. Machine Learning*, volume 18, pages 282–289, San Francisco, CA, 2001. Morgan Kaufmann.

[13] John Langford, Matthias Seeger, and Nimrod Megiddo. An improved predictive accuracy bound for averaging classifiers. In *ICML*, 2001.

[14] John Langford and John Shawe-Taylor. Pac-bayes and margin. In *Neural Information Processing Systems*. MIT Press, 2003.

[15] Yufeng Liu. Fisher consistency of multicategory support vector machines. In *Proc. Intl. Conf. Machine Learning*, 2007.

[16] G. Lugosi and N. Vayatis. On the bayes-risk consistency of regularized boosting methods. *The Annals of Statistics*, 32(1):30–55, 2004.

[17] David McAllester. Generalization bounds and consistency for structured labeling. In *Predicting Structured Data*, Cambridge, Massachusetts, 2007. MIT Press.

[18] David A. McAllester. Some PAC Bayesian theorems. In *Proc. Annual Conf. Computational Learning Theory*, pages 230–234, Madison, Wisconsin, 1998. ACM Press.

[19] David A. McAllester. Pac-bayesian stochastic model selection. *ML*, 2001.

[20] J. R. Parker. *Algorithms for Image Processing and Computer Vision*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[21] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proc. 9th Int'l. Conf. Computer Vision*, volume 1, pages 10–17, 2003.

[22] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL*, pages 213–220, Edmonton, Canada, 2003. Association for Computational Linguistics.

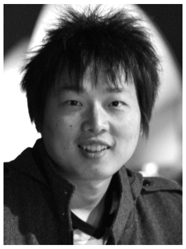
[23] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *In ECCV*, pages 1–15, 2006.

[24] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 25–32, Cambridge, MA, 2004. MIT Press.

[25] A. Tewari and P.L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

[26] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005.

[27] Jun Zhu and Eric P. Xing. Maximum entropy discrimination markov networks. *J. Mach. Learn. Res.*, 2009.



Qinfeng Shi is a senior research associate in the School of Computer Science in The University of Adelaide. He received B. Sc. and M. Sc. in computer science from The Northwestern Polytechnical University, in 2003 and 2006 respectively. He is currently finishing the PhD degree with the thesis titled “Hash Kernels and Structured Learning (submitted in April 2010)” in The Australian National University.

Mark Reid is a Research Fellow at The Australian National University in Canberra. He received a PhD in machine learning in 2007 from the University of New South Wales after completing a Bachelor of Science with honours in Pure Mathematics and Computer Science in 1996 from the same institution. In between, he worked as a research scientist at various companies including IBM and Canon.

Tiberio Caetano received the BSc degree in electrical engineering (with research in physics) and the PhD degree in computer science, (with highest distinction) from the Universidade Federal do Rio Grande do Sul (UFRGS), Brazil. The research part of the PhD program was undertaken at the Computing Science Department at the University of Alberta, Canada. He held a postdoctoral research position at the Alberta Ingenuity Centre for Machine Learning and is currently a senior researcher with the Statistical Machine Learning Group at NICTA. He is also an adjunct research fellow at the Research School of Information Sciences and Engineering, Australian National University. His research interests include pattern recognition, machine learning, and computer vision.

Anton van den Hengel Prof van den Hengel is the founding Director of The Australian Centre for Visual Technologies (ACVT). A/Prof van den Hengel received a PhD in Computer Vision in 2000, a Masters Degree in Computer Science in 1994, a Bachelor of Laws in 1993, and a Bachelor of Mathematical Science in 1991, all from The University of Adelaide.

APPENDIX A
PROOF FOR PAC-BAYES BOUNDS

For explicitly, we rewrite M and p_y as $M(x,y;w)$ and $p(y|x;w)$ when they are parameterized by w.

Lemma 5 (PAC-Bayes bound[19, 9]): For any data distribution D, for any prior P and posterior Q over w, for any $\delta \in (0,1]$, for any loss ℓ. With probability at least $1-\delta$ over random sample S from D with m instances, we have

$$R(Q,\ell) \le R_S(Q,\ell) + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln(\frac{1}{\delta}\,\mathbb{E}_{s\sim D^m}\,\mathbb{E}_{w\sim P}\, e^{2m(R(Q,\ell)-R_S(Q,\ell))^2})}{2m}},$$

where $\mathrm{KL}(Q||P) := \mathbb{E}_{w\sim Q}\ln(\frac{Q(w)}{P(w)})$ is the Kullback-Leibler divergence between Q and P, and $R(Q,\ell) = \mathbb{E}_{Q,D}[\ell(x,y;w)], R_S(Q,\ell) = \mathbb{E}_Q \frac{\sum_{i=1}^m \ell(x_i,y_i;w)}{m}$.

Theorem 6 (Bound on Averaging classifier): For any data distribution D, for any prior P and posterior Q over w, for any $\delta \in (0,1]$ and $\alpha \in [0,1)$ and for any $\gamma \ge 0$. With probability at least $1-\delta$ over random sample S from D with m instances, we have

$$\begin{aligned}\mathbb{E}_{Q,D}\Big[[\gamma - M(x,y;w)]_+\Big] &\le \frac{1}{m}\,\mathbb{E}_Q\Big[\sum_{i=1}^m[\gamma - M(x_i,y_i;w)]_+\Big]\\ &+\frac{\alpha}{1-\alpha}\sqrt{\frac{1}{m}} + \frac{1}{1-\alpha}\sqrt{\frac{\mathrm{KL}(Q||P) + \ln A(\alpha) + \ln\frac{1}{\delta(1-e^{-2})}}{2m}},\end{aligned}$$

where $\mathrm{KL}(Q||P) := \mathbb{E}_{w\sim Q}\ln(\frac{Q(w)}{P(w)})$ is the Kullback-Leibler divergence between Q and P, and

$$\begin{aligned}R(\alpha) &= \alpha\,\mathbb{E}_{Q,D}\Big[-\ln p(y|x;w)\Big] + (1-\alpha)\,\mathbb{E}_{Q,D}\Big[\Big(\gamma - M(x,y;w)\Big)_+\Big],\\ R_S(\alpha) &= \mathbb{E}_Q\Big[\alpha\frac{\sum_{i=1}^m -\ln p(y_i|x_i;w)}{m} + (1-\alpha)\frac{\sum_{i=1}^m\Big(\gamma - M(x_i,y_i;w)\Big)_+}{m}\Big],\\ A(\alpha) &= \mathbb{E}_{s\sim D^m}\,\mathbb{E}_{w\sim P}\, e^{2m(R(\alpha)-R_S(\alpha))^2}.\end{aligned}$$

Proof: Since $\mathbb{E}_D\Big(\mathbb{E}_Q\Big[\frac{\sum_{i=1}^m -\ln p(y_i|x_i;w)}{m}\Big]\Big) = \mathbb{E}_{Q,D}\Big[-\ln p(y|x;w)\Big]$, by Chernoff bound we have

$$\mathbb{P}_{S\sim D^m}\left(\mathbb{E}_Q\Big[\frac{\sum_{i=1}^m -\ln p(y_i|x_i;w)}{m}\Big] - \mathbb{E}_{Q,D}\Big[-\ln p(y|x;w)\Big] < \epsilon\right) > 1 - e^{-2m\epsilon^2}.$$

Define $B(S) := \mathbb{E}_Q\Big[\frac{\sum_{i=1}^m -\ln p(y_i|x_i;w)}{m}\Big] - \mathbb{E}_{Q,D}\Big[-\ln p(y|x;w)\Big]$.

Applying Lemma 5 for $R(\alpha)$ and $R_S(\alpha)$, we have for any P, Q

$$\begin{aligned}
\delta >& \mathbb{P}_{S\sim D^m}\left(R(\alpha) \geq R_S(\alpha) + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{1}{\delta} + \ln A(\alpha)}{2m}}\right) \\
\geq& \mathbb{P}_{S\sim D^m}\left(R(\alpha) \geq R_S(\alpha) + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{1}{\delta} + \ln A(\alpha)}{2m}}, B(S) < \epsilon\right) \\
\geq& \mathbb{P}_{S\sim D^m}\Bigg((1-\alpha)\,\mathbb{E}_{Q,D}\left[\Big(\gamma - M(x,y;w)\Big)_+\right] \geq (1-\alpha)\frac{\sum_{i=1}^m \Big(\gamma - M(x_i,y_i;w)\Big)_+}{m} \\
&+\alpha\epsilon + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{1}{\delta} + \ln A(\alpha)}{2m}}, B(S) < \epsilon\Bigg) \\
=& \mathbb{P}_{S\sim D^m}\Bigg((1-\alpha)\,\mathbb{E}_{Q,D}\left[\Big(\gamma - M(x,y;w)\Big)_+\right] \geq (1-\alpha)\frac{\sum_{i=1}^m \Big(\gamma - M(x_i,y_i;w)\Big)_+}{m} \\
&+\alpha\epsilon + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{1}{\delta} + \ln A(\alpha)}{2m}}\Big| B(S) < \epsilon\Bigg)\,\mathbb{P}_{S\sim D^m}\Big(B(S) < \epsilon\Big) \\
\geq& \mathbb{P}_{S\sim D^m}\Bigg((1-\alpha)\,\mathbb{E}_{Q,D}\left[\Big(\gamma - M(x,y;w)\Big)_+\right] \geq (1-\alpha)\frac{\sum_{i=1}^m \Big(\gamma - M(x_i,y_i;w)\Big)_+}{m} \\
&+\alpha\epsilon + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{1}{\delta} + \ln A(\alpha)}{2m}}\Bigg)\,\mathbb{P}_{S\sim D^m}\Big(B(S) < \epsilon\Big)
\end{aligned}$$

Divide two sides by $\mathbb{P}_{S\sim D^m}\Big(B(S) < \epsilon\Big)$, we get

$$\begin{aligned}
&\mathbb{P}_{S\sim D^m}\Bigg((1-\alpha)\,\mathbb{E}_{Q,D}\left[\Big(\gamma - M(x,y;w)\Big)_+\right] \geq (1-\alpha)\frac{\sum_{i=1}^m \Big(\gamma - M(x_i,y_i;w)\Big)_+}{m} \\
&+\alpha\epsilon + \sqrt{\frac{\mathrm{KL}(Q||P) + \ln\frac{1}{\delta} + \ln A(\alpha)}{2m}}\Bigg) \leq \frac{\delta}{\mathbb{P}_{S\sim D^m}\Big(B(S) < \epsilon\Big)} \leq \frac{\delta}{1 - e^{-2m\epsilon^2}}.
\end{aligned}$$

Let $\epsilon = \sqrt{\frac{1}{m}}$, and then let $\delta' = \frac{\delta}{1-e^{-2m(\epsilon^2)}} = \frac{\delta}{1-e^{-2}}$, we get $\delta = \frac{1}{\delta'(1-e^{-2})}$. The theorem follows by substituting δ with δ' and dividing by $(1-\alpha)$ on both sides of the inequality inside of the probability. □