Non-Local Contrastive Objectives

David Vickrey DVICKREY@CS.STANFORD.EDU
Cliff Chiung-Yu Lin CHIUNGYU@CS.STANFORD.EDU
Daphne Koller KOLLER@CS.STANFORD.EDU
Stanford University, Stanford, CA 94305-9010 USA

Abstract

Pseudo-likelihood and contrastive divergence are two well-known examples of contrastive methods. These algorithms trade off the probability of the correct label with the probabilities of other "nearby" instantiations. In this paper we explore more general types of contrastive objectives, which trade off the probability of the correct label against an arbitrary set of other instantiations. We prove that a large class of contrastive objectives are consistent with maximum likelihood, even for finite amounts of data. This result generalizes asymptotic consistency for pseudo-likelihood. The proof gives significant insight into contrastive objectives, suggesting that they enforce (soft) probability-ratio constraints between pairs of instantiations. Based on this insight, we propose Contrastive Constraint Generation (CCG), an iterative constraint-generation style algorithm that allows us to learn a log-linear model using only MAP inference. We evaluate CCG on a scene classification task, showing that it significantly outperforms pseudo-likelihood, contrastive divergence, and a well-known margin-based method.

1. Introduction

Learning Markov random fields is difficult because computation of the normalization term, generally referred to as the *partition function* Z, is intractable for many types of networks. Recently there has been significant interest in *contrastive methods* such as pseudo-likelihood (Besag, 1975) and contrastive divergence (Hinton, 2002). The main idea of these algorithms is to trade off the probability of the correct assignment for each labeled example with the probabilities of other, "nearby" assignments. This means that these algorithms do not need to compute the partition function Z. Unfortunately, these algorithms can suffer when the distribution is highly multi-modal, with multiple distant regions of high probability.

LeCun & Huang (2005), Smith & Eisner (2005), and Liang & Jordan (2008) all consider the general case of contrastive objectives, where the contrasting set can consist of arbitrary assignments. However, previous work has not pursued the idea of *non-local* contrastive objectives. Rather than restrict the objective to considering assignments which are close to the correct label as in pseudo-likelihood and contrastive divergence, we allow comparison to any assignment.

We prove several results which justify the use of non-local contrastive objectives. We show that a wide class of contrastive objectives are consistent with maximum likelihood, even for finite data under certain conditions. This generalizes and is a considerably stronger result than the asymptotic consistency of pseudo-likelihood. A central idea of this result is that contrastive objectives attempt to enforce probability-ratio constraints between different assignments, based on the structure of the objective. Among other consequences, this result clearly points out cases in which pseudo-likelihood (and other local methods) may fail.

Based on this insight, we propose Contrastive Constraint Generation (CCG), a constraint-generation style algorithm that iteratively constructs a contrastive objective based only on a MAP-inference procedure. While similar in flavor to the max-margin cutting plane algorithm suggested by Tsochantaridis et al. (2005), our method has the ability to obtain accurate probability estimates. We compare CCG to pseudo-likelihood, contrastive divergence, and the cutting-plane algorithm on a real-world machine vision problem; CCG achieves a 12% error reduction over the best of these, a statistically significant improvement.

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

2. Contrastive Objectives

We consider prediction problems in which we try to predict a discrete label variable (or set of variables) $\mathbf{Y}$ given a set of features $\mathbf{X}$. We are given a data set D of m examples. The i^{th} example $d^i = (\mathbf{x}^i, \mathbf{y}^i)$ consists of observed features $\mathbf{x}^i$ and a correct label $\mathbf{y}^i$. We use $\hat{P}(\mathbf{Y}|\mathbf{X}) = \frac{|d^i:(\mathbf{x}^i,\mathbf{y}^i)=(\mathbf{X},\mathbf{Y})|}{\hat{Z}(\mathbf{X})}$ to refer to the empirical distribution observed in our data set D, where $\hat{Z}(\mathbf{X}) = |d^i : \mathbf{x}^i = \mathbf{X}|$. We are also given a set of R feature functions $f_1(\mathbf{X},\mathbf{Y}), \ldots, f_R(\mathbf{X},\mathbf{Y})$; let $\mathbf{f}(\mathbf{X},\mathbf{Y})$ be a vector containing all feature functions.

Given a vector of R weights $\boldsymbol{\theta}$, our model assigns probability $P_{\boldsymbol{\theta}}(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}) = \frac{e^{\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x},\mathbf{y})}}{Z(\mathbf{x})}$, where $Z(\mathbf{X} = \mathbf{x}) = \sum_{\mathbf{y}} e^{\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x},\mathbf{y})}$. The (log) likelihood objective is $LL(\boldsymbol{\theta}; D) = \sum_{(\mathbf{x}^i,\mathbf{y}^i)} \log P_{\boldsymbol{\theta}}(\mathbf{y}^i|\mathbf{x}^i)$.

The main idea of our approach is to define smaller terms over subsets of $\mathbf{Y}$.

Definition 1. *Let S_j be some subset of values of $\mathbf{Y}$. The (conditional) contrastive probability distribution for S_j is $P_{\boldsymbol{\theta},j}(\mathbf{y}|\mathbf{x}) = \frac{e^{\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x},\mathbf{y})}}{Z_j(\mathbf{x})}$, where Z_j is the contrastive partition function $Z_j(\mathbf{x}) = \sum_{\mathbf{y}\in S_j} e^{\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x},\mathbf{y})}$.*

We refer to this distribution as contrastive because it compares the (unnormalized) probabilities of the values of $\mathbf{Y}$ in S_j. One important property of this distribution is that it also implicitly compares *normalized* probabilities: $\frac{e^{\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x},\mathbf{y})}}{Z_j(\mathbf{x})} = \frac{P_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{\sum_{\mathbf{y}'\in S_j} P_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}$ due to cancellation of the global partition function.

Definition 2. *A contrastive sub-objective $C_j(\boldsymbol{\theta}; D)$ is a weighted maximum-likelihood objective with the model distribution $P_{\boldsymbol{\theta}}$ replaced by the contrastive distribution $P_{\boldsymbol{\theta},j}$ for some subset S_j:*

$$\sum_{(\mathbf{x}^i,\mathbf{y}^i):\mathbf{y}^i\in S_j} w_j(\mathbf{x}^i)\left(\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x}^i,\mathbf{y}^i) - \log Z_j(\mathbf{x}^i)\right).$$

$w_j(\mathbf{x})$ is a parameter of the sub-objective that determines the overall strength of the sub-objective as well as the relative importance of each value of $\mathbf{x}$.

A *contrastive objective* $C(\boldsymbol{\theta}; D)$ is a sum of J sub-objectives C_j, each with a different subset S_j and set of weights $w_j(\mathbf{x})$. C is tractable to compute (and optimize) if the contrastive partition functions are tractable to compute. In some cases, we can compute the contrastive partition function $Z_j(\mathbf{x})$ even if S_j contains an exponential number of values, e.g., using dynamic programming for tractable sub-structures.

We say that sub-objective C_j is *active* for example $(\mathbf{x}^i, \mathbf{y}^i)$ if $\mathbf{y}^i \in S_j$ and $w_j(\mathbf{x}^i) > 0$. The number of active sub-objectives for a particular data set D may be much smaller than the total number of sub-objectives.

For now, we assume that C is given. We will discuss how to construct contrastive objectives in Section 5.

2.1. Related Learning Methods

The log-likelihood objective $LL(\boldsymbol{\theta}; D)$ is a contrastive objective with one sub-objective C_1, where S_1 contains all values in $\mathbf{Y}$ and $w_1(\mathbf{x}) = 1$ for all $\mathbf{x}$.

If $\mathbf{Y}$ is an MRF (or CRF), contrastive objectives are a generalization of pseudo-likelihood (PL). Let y_l be the value of node l in our network, $\mathbf{y}_{-l}$ be the value of all nodes *except* node l, and $(\mathbf{y}_{-l}, y_l)$ be a combined instantiation to $\mathbf{y}$ which matches y_l for node l and $\mathbf{y}_{-l}$ for all other nodes. Let $dom(Y_l)$ denote the set of possible values of Y_l. For each l, for every possible instantiation $\mathbf{y}_{-l}$, we have one sub-objective $S_{\mathbf{y}_{-l}}$ that contains exactly the set of instantiations consistent with $\mathbf{y}_{-l}$, i.e., $(\mathbf{y}_{-l}, Y_l = a)$ for all $a \in dom(Y_l)$. All sub-objective weights $w_{\mathbf{y}_{-l}}(\mathbf{x})$ are set to 1. Since a sub-objective $C_{\mathbf{y}_{-l}}$ is only active for examples where $\mathbf{y}^i \in S_{\mathbf{y}_{-l}}$, it follows that each example participates in n sub-objectives, where n is the number of variables in the network. This yields the contrastive objective

$$\sum_{(\mathbf{x}^i,\mathbf{y}^i)} \sum_{l} \left(\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x}^i,\mathbf{y}^i) - \log \sum_{a\in dom(Y_l)} e^{\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x}^i,(\mathbf{y}^i_{-l},a))}\right),$$

which is the definition of pseudo-likelihood. All of the sub-objectives in PL are *local*; they only involve instantiations that differ on a single node. Generalized pseudo-likelihood (GPL) can also easily be expressed in this framework. In GPL two or more variables are allowed to vary. This can lead to large, potentially exponential sub-objectives. In some cases, dynamic programming can render inference tractable within sub-objectives. Unlike GPL, our framework allows us to vary multiple variables at a time *without* including all possible combinations of these variables, giving us considerably more flexibility.

Another related learning method is contrastive divergence (CD), which approximates the gradient of maximum likelihood using a non-mixed Markov chain, initialized at the label of the current example. CD is generally defined by the update rule

$$\Delta\boldsymbol{\theta}_t = \sum_{(\mathbf{x}^i,\mathbf{y}^i)} f_t(\mathbf{x}^i,\mathbf{y}^i) - E_{P^k_{\boldsymbol{\theta}}}(f_t(\mathbf{x}^i,\mathbf{y}^i)),$$

where $P^k_{\boldsymbol{\theta}}$ is the distribution over $\mathbf{y}$ obtained by initializing some MCMC procedure at $\mathbf{x}^i$ and running for k steps.[1] CD cannot be expressed as a contrastive objective, because CD uses $P^k_{\boldsymbol{\theta}}$ to compute expectations rather than P_θ. This means that the probability-ratio matching intuitions in the next section do not hold for

[1] In practice, it is intractable to compute the expectation over $P^k_{\boldsymbol{\theta}}$ exactly; instead, it is estimated through sampling.

CD. In fact, CD does not optimize *any* objective. This means that CD requires using stochastic gradient for optimization, whereas a contrastive objective can be optimized using a variety of methods (in this paper, BFGS). Furthermore, similar to PL, standard implementations of CD are local: they compare the correct label $\mathbf{y}^i$ only to nearby values $\mathbf{y}'$.

3. Theoretical Results

The main results of this section show the finite and asymptotic consistency of contrastive objectives, under suitable conditions on the model distribution $P_{\boldsymbol{\theta}}$ and sub-objectives C_j. The proofs of these theoretical results also illustrate a key feature of contrastive objectives: if two label values $\mathbf{y}, \mathbf{y}'$ are connected through a series of sub-objectives, then the objective will encourage $\frac{P_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{P_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}$ to match $\frac{\hat{P}(\mathbf{y}|\mathbf{x})}{\hat{P}(\mathbf{y}'|\mathbf{x})}$. This will be the main motivation for the methods we propose in Section 5 for choosing non-local sub-objectives.

3.1. Finite Consistency

Let Θ be the set of all possible parameter vectors $\boldsymbol{\theta}$, and let P_Θ denote the set of all possible models $P_{\boldsymbol{\theta}}$ obtainable using $\boldsymbol{\theta} \in \Theta$. We say that P_Θ can *represent* probability distribution $P'(\mathbf{Y}|\mathbf{X})$ if there exists parameters $\boldsymbol{\theta}' \in \Theta$ such that $P_{\boldsymbol{\theta}'}(\mathbf{Y}|\mathbf{x}) = P'(\mathbf{Y}|\mathbf{x})$ for all $\mathbf{x}$. Let $\Theta[P']$ denote the set of such $\boldsymbol{\theta}'$.

While asymptotic representability (i.e., P_Θ being able to represent $P^*(\mathbf{Y}|\mathbf{X})$) is a standard concept in analysis of learning algorithms, finite representability (P_Θ being able to represent $\hat{P}(\mathbf{Y}|\mathbf{X})$) is less common. Suppose P_Θ can represent $\hat{P}(\mathbf{Y}|\mathbf{X})$. In many cases, we only see each $\mathbf{X} = \mathbf{x}$ one time in our data set D, which means that $\hat{P}(\mathbf{Y}|\mathbf{X})$ will have a point estimate on the correct label $\mathbf{y}^i$ for each $\mathbf{x}^i$. Thus, it can be a fairly strong condition on our model class P_Θ.

However, if we expand the set of allowed weight vectors Θ to include a certain type of infinite length weight vectors (defined by a direction in weight space), it is possible to show that representability is actually a weaker condition than *separability*, a commonly-used condition in analysis of learning algorithms. Even without infinite-length weight vectors, if D is separable then we can obtain an arbitrarily close approximate to $\hat{P}$ using finite-length weight vectors. We will assume exact representability for the remainder of this section, but replacing with "near-exact" representability would only slightly weaken the results (specifically, it would add an arbitrarily small approximation factor).

Let $\hat{P}_j$ be the contrastive observed data distribution relative to S_j: $\hat{P}_j(\mathbf{y}|\mathbf{x}) = \frac{|(\mathbf{x}^i,\mathbf{y}^i)=(\mathbf{x},\mathbf{y})|}{\hat{Z}_j(\mathbf{x})} = \frac{\hat{P}(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{y}\in S_j}\hat{P}(\mathbf{x},\mathbf{y})}$ where $\hat{Z}_j(\mathbf{x}) = |(\mathbf{x}^i, \mathbf{y}^i) : \mathbf{x}^i = \mathbf{x}, \mathbf{y}^i \in S_j|$.

Lemma 1. *Suppose that P_Θ can represent $\hat{P}(\mathbf{y}|\mathbf{x})$. Then for* any *contrastive objective $C(\boldsymbol{\theta}; D)$,*

i. If $\hat{\boldsymbol{\theta}} \in \Theta[\hat{P}]$, $C(\boldsymbol{\theta}; D)$ has a global optimum at $\hat{\boldsymbol{\theta}}$.
ii. If $\boldsymbol{\theta}'$ optimizes $C(\boldsymbol{\theta}; D)$, then for any $j, \mathbf{x}$ such that $\hat{P}(\mathbf{x})w_j(\mathbf{x})(\sum_{\mathbf{y}\in S_j}\hat{P}(\mathbf{y}|\mathbf{x})) > 0$, we have $P_{\boldsymbol{\theta}',j}(\mathbf{y}|\mathbf{x}) = \hat{P}_j(\mathbf{y}|\mathbf{x})$.

We omit proofs of this and subsequent results for lack of space.

Corollary 1. *Suppose that P_Θ can represent $\hat{P}(\mathbf{y}|\mathbf{x})$ and $\boldsymbol{\theta}'$ optimizes $C(\boldsymbol{\theta}; D)$. Also suppose $\mathbf{y}_1, \mathbf{y}_2 \in S_j$, $\hat{P}(\mathbf{x}) > 0$, $\hat{P}(\mathbf{y}_1|\mathbf{x}) > 0$, and $w_j(\mathbf{x}) > 0$. Then $\frac{P_{\boldsymbol{\theta}'}(\mathbf{y}_2|\mathbf{x})}{P_{\boldsymbol{\theta}'}(\mathbf{y}_1|\mathbf{x})} = \frac{\hat{P}(\mathbf{y}_2|\mathbf{x})}{\hat{P}(\mathbf{y}_1|\mathbf{x})}$.*

Thus, probability ratios according to $P_{\boldsymbol{\theta}'}$ match those according to $\hat{P}$ within a sub-objective set.

Definition 3. *We are given a set of sub-objectives C_j with weights $w_j(\mathbf{x})$. For a fixed feature value $\mathbf{x}$, we say that there is a* path *from $\mathbf{y}_1$ to $\mathbf{y}_2$ relative to probability distribution $P(\mathbf{Y}|\mathbf{x})$ if there is a sequence $S_{j_b} : b = 1, ..., k$ such that*

i. $\mathbf{y}_1 \in S_{j_1}$ and $\mathbf{y}_2 \in S_{j_k}$
ii. for every pair $S_{j_b}, S_{j_{b+1}}$, there exists $\mathbf{z}_b \in dom(\mathbf{Y})$ s.t. $\mathbf{z}_b \in S_{j_b}$, $\mathbf{z}_b \in S_{j_{b+1}}$, and $P(\mathbf{z}_b|\mathbf{x}) > 0$
iii. $w_{j_b}(\mathbf{x}) > 0$ for all j_b

Intuitively, this definition means that it is possible to "walk" from $\mathbf{y}_1$ to $\mathbf{y}_2$: if you are currently at value $\mathbf{y}$, you are allowed to move to any other value $\mathbf{y}'$ if $\mathbf{y}$ and $\mathbf{y}'$ both are contained in some sub-objective set S_j with positive weight $w_j(\mathbf{x})$. If $P(\mathbf{y}'|\mathbf{x}) = 0$, the walk must stop; otherwise it can continue.

Lemma 2. *Suppose that P_Θ can represent $\hat{P}(\mathbf{y}|\mathbf{x})$ and $\boldsymbol{\theta}'$ optimizes $C(\boldsymbol{\theta}; D)$. Also, suppose that $\hat{P}(\mathbf{x}) > 0$, $\hat{P}(\mathbf{y}_1|\mathbf{x}) > 0$, and there is a path from $\mathbf{y}_1$ to $\mathbf{y}_2$ relative to $\hat{P}(\mathbf{Y}|\mathbf{x})$. Then $\frac{P_{\boldsymbol{\theta}'}(\mathbf{y}_2|\mathbf{x})}{P_{\boldsymbol{\theta}'}(\mathbf{y}_1|\mathbf{x})} = \frac{\hat{P}(\mathbf{y}_2|\mathbf{x})}{\hat{P}(\mathbf{y}_1|\mathbf{x})}$.*

This result follows from applying Corollary 1 to each sub-objective along the path from $\mathbf{y}_1$ to $\mathbf{y}_2$. We now have that the probability ratios according to $P_{\boldsymbol{\theta}'}(\mathbf{y}|\mathbf{x})$ match those according to $\hat{P}(\mathbf{y}|\mathbf{x})$ for any pair of values $\mathbf{y}_1, \mathbf{y}_2$ connected by a path (relative to $\hat{P}(\mathbf{Y}|\mathbf{x})$).

Definition 4. *For fixed $\mathbf{x}$, a set of sub-objectives S_j with weights $w_j(\mathbf{x})$* span *$\mathbf{Y}$ relative to a probability distribution $P(\mathbf{Y}|\mathbf{x})$ if for every pair of values $\mathbf{y}_1, \mathbf{y}_2$ there is a path from $\mathbf{y}_1$ to $\mathbf{y}_2$ relative to $P(\mathbf{Y}|\mathbf{x})$.*

Note that this condition requires the total size of our sub-objectives to be at least the cardinality of $\mathbf{Y}$.

Let Θ' be the set of $\boldsymbol{\theta}'$ that optimize $C(\boldsymbol{\theta}; D)$.

Theorem 1. *Suppose that P_Θ can represent $\hat{P}(\mathbf{y}|\mathbf{x})$. Furthermore, suppose that for every $\mathbf{x}^i$ (i.e., every value of $\mathbf{X}$ observed in D), our sub-objectives S_j with weights $w_j(\mathbf{x}^i)$ span $\mathbf{Y}$ relative to $\hat{P}(\mathbf{Y}|\mathbf{x}^i)$. Then $\Theta' = \Theta[\hat{P}]$. That is, $\boldsymbol{\theta}'$ optimizes $C(\boldsymbol{\theta}; D)$ if and only if $P_{\boldsymbol{\theta}'}(\mathbf{Y}|\mathbf{x}^i) = \hat{P}(\mathbf{Y}|\mathbf{x}^i)$ for all $\mathbf{x}^i$.*

This result follows directly from Lemma 2 and the definition of span.

The optima of the log-likelihood objective are exactly $\Theta[\hat{P}]$ (provided P_Θ can represent $\hat{P}(\mathbf{y}|\mathbf{x})$). Thus, the optima of any contrastive objective fulfilling the conditions of the previous theorem are exactly the same as those of the log-likelihood objective.

3.2. Asymptotic Consistency

Asymptotic consistency also holds for certain contrastive objectives. If the true data distribution P^* is in our model class (with parameters $\boldsymbol{\theta}^* \in \Theta[P^*]$), then in the limit of infinite data, our objective will recover the correct parameters provided that it spans $\mathbf{Y}$. Asymptotic representability is a considerably weaker condition than finite representability, because the model does not need to capture noise in the data.

Let $\{d^1, d^2, \ldots\}$ be an infinite sequence of examples drawn i.i.d from $P^*(\mathbf{X}, \mathbf{Y})$. We refer to the data set composed of the first n of these as D^n, and its empirical distribution as $\hat{P}(\mathbf{y}|\mathbf{x}; D^n)$. By the strong law of large numbers, $P(\lim_{n\to\infty} \hat{P}(\mathbf{y}|\mathbf{x}; D^n) = P^*(\mathbf{X}, \mathbf{Y})) = 1$, or in short-hand $\hat{P}$ converges to P^* almost surely $\hat{P}(\mathbf{y}|\mathbf{x}; D^n) \overset{a.s.}{\to} P^*(\mathbf{y}|\mathbf{x})$ as $n \to \infty$.

Let $\boldsymbol{\theta}^{(n)}$ be a sequence of weight vectors $\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots\}$ such that for all n, $\boldsymbol{\theta}^n$ optimizes $C(\boldsymbol{\theta}; D^n)$.

Theorem 2. *Suppose P_Θ can represent $P^*(\mathbf{y}|\mathbf{x})$. Furthermore, suppose that for every $\mathbf{x}$ such that $P^*(\mathbf{x}) > 0$, our sub-objectives S_j with weights $w_j(\mathbf{x})$ span $\mathbf{Y}$ relative to $P^*(\mathbf{Y}|\mathbf{x})$. Then for any $\boldsymbol{\theta}^{(n)}$ and any $\mathbf{x}$ such that $P^*(\mathbf{x}) > 0$, $P_{\boldsymbol{\theta}^n}(\mathbf{Y}|\mathbf{x}) \overset{a.s.}{\to} P^*(\mathbf{Y}|\mathbf{x})$ as $n \to \infty$.*

The proof of this theorem follows the same lines as finite consistency, with the additional use of the law of large numbers. All of the intermediate lemmas hold in the infinite data case; in particular, values $\mathbf{y}, \mathbf{y}'$ that are connected through a series of sub-objectives have calibrated probability ratios. Note that this theorem does not depend on the previous one: asymptotic consistency can hold even if finite consistency does not.

From this result we can derive consistency of pseudo-likelihood for strictly positive data distributions $P_{\boldsymbol{\theta}^*}$ (i.e., $P_{\boldsymbol{\theta}^*}(\mathbf{y}|\mathbf{x}) > 0$ for all $\mathbf{y}, \mathbf{x}$), simply by noting that in the limit of infinite data, the set of active PL sub-objectives will span $\mathbf{Y}$. It is also easy to see why PL is usually not consistent with finite data (it is unlikely to span the space), and why it may not be consistent for non-positive data distributions (again, because it may not span the space).

The most important practical implication from this section is that a contrastive objective attempts to calibrate probabilities within connected components of sub-objectives, but cannot calibrate probabilities between disconnected components. This has important implications for the performance of PL (and other local contrastive objectives), as we will see in Section 6.

4. Analyzing the Weights

So far we only considered whether $w_j(\mathbf{x}) > 0$. To better understand the effect of the weights on the objective, we write the weights as a combination of three terms, $w_j(\mathbf{x}) = w(\mathbf{x}) * \sum_{\mathbf{y}} P(C_j|\mathbf{y}, \mathbf{x}) Q(\mathbf{y}|\mathbf{x})$, each chosen by the designer of the contrastive objective.

$w(\mathbf{x})$ allows the designer to reweight the relative importance of terms in the objective corresponding to different values of $\mathbf{x}$. This could be useful if we believe that the empirical distribution $\hat{P}(\mathbf{x})$ observed in our data does not match the true (or desired) distribution over $\mathbf{x}$. $P(C_j|\mathbf{y}, \mathbf{x})$ allows the designer to choose the relative importance of different sub-objectives for a particular value of the label variable $\mathbf{y}$ (and also relative to a feature value $\mathbf{x}$). $P(C_j|\mathbf{y}, \mathbf{x})$ is constrained to be a probability distribution such that $P(C_j|\mathbf{y}, \mathbf{x}) > 0$ only when $\mathbf{y} \in S_j$. $Q(\mathbf{y}|\mathbf{x})$ is an auxiliary probability distribution that allows the designer to choose how important each $\mathbf{y}$ is to the objective.

Since P and Q are probability distributions, the sum of all weights $w_j(\mathbf{x})$ for a given $\mathbf{x}$ is $w(\mathbf{x})$. Thus, P and Q do not affect the relative influence of different values of $\mathbf{x}$. This decomposition of the weights $w_j(\mathbf{x})$ is overparametrized; it is not hard to show that we can write any choice of $w_j(\mathbf{x})$ in this form.

We now state a strong relationship between a particular contrastive objective and $LL(\boldsymbol{\theta}; D)$:

Lemma 3. *Let C contain a sub-objective S_{jk} for every pair of instantiations $\mathbf{y}_j, \mathbf{y}_k$ (including singleton sub-objectives where $\mathbf{y}_j = \mathbf{y}_k$). Let $w(\mathbf{x}) = |\mathbf{Y}|$ for all $\mathbf{x}$, let $P(C_{jk}|\mathbf{y}, \mathbf{x}) = \frac{1}{|\mathbf{Y}|}$ if $\mathbf{y} \in C_{jk}$, 0 otherwise (i.e., P is uniform over sub-objectives containing $\mathbf{y}$), and $Q(\mathbf{y}|\mathbf{x}) = P_{\boldsymbol{\theta}_0}(\mathbf{y}|\mathbf{x})$ for some fixed parameter vector $\boldsymbol{\theta}_0$. Then $\frac{dC}{d\boldsymbol{\theta}}|_{\boldsymbol{\theta}_0} = \frac{dLL}{d\boldsymbol{\theta}}|_{\boldsymbol{\theta}_0}$.*

Thus, at any point $\boldsymbol{\theta}$ in weight space, we can construct a contrastive objective tangent to log-likelihood at $\boldsymbol{\theta}$. As a result, we can optimize LL using an EM-like algorithm. We initialize with weight vector $\boldsymbol{\theta}_0$. During the i^{th} Expectation step, we update $Q(\mathbf{y}|\mathbf{x}) = P_{\boldsymbol{\theta}_{i-1}}(\mathbf{y}|\mathbf{x})$. During the i^{th} Maximization step, we fix Q and use C to compute the gradient of log-likelihood at $\boldsymbol{\theta}_{i-1}$. We then take a step in the direction of the gradient, obtaining a new weight vector $\boldsymbol{\theta}_i$.

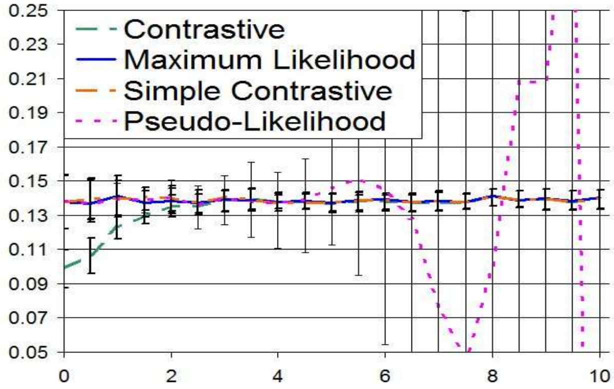

Figure 1. Estimate of λ_0 (y-axis) vs. λ_1 used to generate the data (x-axis). The plot shows median learned parameter value over 100 synthetic data sets, each with 1000 instances. Error bars indicate 25^{th} and 75^{th} percentile estimates. Correct $\lambda_0 = .139$.

This algorithm has some similarities to an algorithm proposed by Hoefling & Tibshirani (2009) for optimizing log-likelihood using a series of pseudo-likelihoods. We omit further discussion for lack of space.

5. Selecting Sub-objectives

In practice we are not going to be able to span $\mathbf{Y}$ with our sub-objectives. In this section, we propose several techniques for constructing tractable objectives and examine some of their properties.

5.1. Basic Methods

One simple way to construct sub-objectives is to use expert knowledge to determine useful values to compare. In pseudo-likelihood, for example, each sub-objective corresponds to instantiations which differ on a (particular) single variable. In generalized PL, sub-objectives contain all instantiations which differ on a particular subset of variables.

As a concrete example of a sub-objective which is not possible using (generalized) PL, we consider a binary chain MRF ($\mathbf{x}$ is empty) with 10 nodes and two parameters: a single bias term specifying the relative weight of 0 vs. 1; and a single affinity term specifying how likely two neighboring nodes are to have the same value. The log-score of an instantiation is $\lambda_0 * |y_i = 1| + \lambda_1 * |y_i = y_{i+1}|$. For large λ_1, the instantiations $\{000000000\}$ and $\{1111111111\}$ have much higher probability than any other instantiations; we expect PL to have trouble fitting λ_0 in this case, since it does not directly compare the probabilities of these two instantiations. However, we can augment the PL objective with an additional sub-objective containing exactly these two values — we refer to this objective as Simple Contrastive. Figure 1 shows the error in the estimate of λ_0 as we vary λ_1. Simple Contrastive accurately reconstructs λ_0 for all values of λ_1, while PL does not. Contrastive objectives constructed in this way can be quite powerful, but are somewhat difficult to design. We omit further discussion for lack of space.

A different approach is to use the data D to guide the selection of sub-objectives. For unconditioned Markov networks, a simple approach is to construct sub-objectives which compare different observed values of the label variables $\mathbf{y}^i$. We employed this strategy for the MRF described above, augmenting PL with a single sub-objective containing all values observed in D. This objective is referred to as Contrastive. As shown in Figure 1, Contrastive is also effective at recovering λ_0, although for low values of λ_1, the estimate is slightly inaccurate.

Simple Contrastive used a *static* method to choose sub-objectives: the weights $w_j(\mathbf{x})$ do not depend on the examples observed in D. Contrastive used a *dynamic* method.

Lemma 4. *Suppose the $w_j(\mathbf{x})$ do not depend on D. Then $E_{P^*}[\frac{dC(\boldsymbol{\theta};D)}{d\boldsymbol{\theta}}|_{\boldsymbol{\theta}_0}] = \frac{dC(\boldsymbol{\theta};D^*)}{d\boldsymbol{\theta}}|_{\boldsymbol{\theta}_0}$ for all $\boldsymbol{\theta}_0$, where D^* is a data set (of possibly infinite size) such that $\hat{P}(\mathbf{y}|\mathbf{x};D^*) = P^*(\mathbf{y}|\mathbf{x})$.*

This lemma shows that for a static method, we get an unbiased estimate of the gradient at any point $\boldsymbol{\theta}_0$. Suppose that P_Θ can represent $P^*(\mathbf{y}|\mathbf{x})$. In this case, we can apply this lemma at $\boldsymbol{\theta}_0 = \boldsymbol{\theta}^*$ to get that the expectation D of the gradient at $\boldsymbol{\theta}^*$ is 0. Loosely speaking, this means that the learned parameters for different D sampled from P^* will be centered around $\boldsymbol{\theta}^*$. Dynamic methods have no such guarantee. This bias in the gradient is precisely the reason why Contrastive in Figure 1 gives an inaccurate estimate for λ_0 for small values of λ_1. However, since dynamic contrastive objectives are more flexible than static ones, this bias may often be acceptable in practice.

5.2. Contrastive Constraint Generation

For many problems, the basic approaches presented above are not sufficiently powerful. We propose a (dynamic) method for constructing contrastive objectives called Contrastive Constraint Generation (CCG).

In CCG, we begin by building an initial contrastive objective C_0 containing relatively few sub-objectives. During iteration t, we first optimize C_{t-1} to obtain a new weight vector $\boldsymbol{\theta}_t$. Next, for each example d^i, we find one or more "interesting" instantiations based on the current model $P_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})$. Finally, we construct a new contrastive objective C_t that incorporates these new instantiations into C_{t-1}. We repeat this process until convergence (or until we decide to stop). We now describe the details of each of these steps.

Initialization. We consider two simple initializations: empty (no sub-objectives); and adding all sub-objectives from Pseudo-Likelihood.

Optimization. This step is straight-forward. We simply optimize C_{t-1} using a method such as BFGS.

Finding Interesting Instantiations. We consider two general methods for finding new instantiations. For simplicity, we assume that only one new instantiation is generated per round per example, referred to as $\mathbf{y}^i_t$.

The first general method is to use a maximum a-posteriori (MAP) inference algorithm in order to find the highest probability $\mathbf{y}$ according to $P_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}^i)$. In practice, the MAP algorithm will be approximate, i.e., $\mathbf{y}$ will not be guaranteed to actually maximize $P_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}^i)$. We considered two methods in this paper. The first, iterated conditional modes (ICM), proposed by Besag (1986), is a simple greedy ascent algorithm. At each round, a region is chosen at random; the label of this region is then changed to the value that gives the highest score . This is repeated until a local maximum is reached. The second is max-product belief propagation (MP) (Pearl, 1988); we use the variant known as residual belief propagation, proposed by Elidan et al. (2006). ICM and MP are both randomly initialized to encourage finding different local optima from iteration to iteration. We also tried a third inference method based on dual decomposition, proposed by Komodakis et al. (2007), but this method obtained similar results to MP while being significantly slower; we do not present results for dual decomposition in this paper.

The second general method uses a sampling algorithm such as Gibbs sampling to generate one or more instantiations. Contrastive divergence takes this second approach, with the sampling algorithm initialized at $\mathbf{y}^i$ and run for only a few steps. If we use this approximate sampling procedure, we end up with an algorithm that has many similarities with CD. The main differences are that CCG uses $P_{\boldsymbol{\theta}}$ to score the instantiations while CD uses $P^k_{\boldsymbol{\theta}}$; and CD can only use stochastic gradient methods for optimization.

Building a New Objective. There are many possible ways to construct a new contrastive objective incorporating the new instantiations; we consider one simple option. For each example d^i, we construct a sub-objective C_{d^i} such that, at iteration t, S_{d^i} contains the correct label $\mathbf{y}^i$ as well as $\mathbf{y}^i_{t'}$ for all $t' \leq t$ (since S_{d^i} is a set, duplicate values are ignored). All weights $w_j(\mathbf{x})$ are set to 1.

Convergence. Convergence is reached when, for all examples, $\mathbf{y}^i_t$ has already been seen.

6. Experimental Results

In this section, we apply CCG to a real-world machine vision problem. We use the street scenes image data set described by Gould et al. (2009), consisting of 715 images. Every pixel in each image is labeled with one of 8 classes. To reduce the computational burden and to have access to more coherent features, we took as input the regions as predicted in Gould et al. (2009). This limits the maximum pixel-wise accuracy: the best-possible labeling of regions for this data obtains pixel-wise error of 12.0% (Lower Bound in Table 1). Our model is a CRF using intra-region (single node) and inter-region (pairwise) features, also taken from (Gould et al., 2009). We tested the following learning algorithms:

Independent (I). Only the singleton potentials are used during training. Equivalent to logistic regression with individual regions as training examples.

Pseudo-Likelihood (PL). See Section 2.1.

Contrastive Divergence (CD). Each iteration, we generated a single sample for each data example d^i and use it to compute a stochastic approximation to the gradient. We used Gibbs sampling to generate the samples, following standard practice for CD (see, for example, (Bengio, 2009)). We tested three variants: CD-1, CD-10, and CD-100, which generate samples using 1, 10, and 100 round of Gibbs, respectively.[2] We ran each variant for 10,000 seconds, corresponding to 50k, 16.5k, and 2k iterations, respectively.

Max-Margin Cutting Planes (MM). This is a constraint-generation algorithm proposed by Tsochantaridis et al. (2005). It uses a margin-based objective, which tries to find a weight vector $\boldsymbol{\theta}$ such that for every d^i, the score $\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x}^i, \mathbf{y}^i)$ of the correct label $\mathbf{y}^i$ is larger than $\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}^i)$ for any other $\mathbf{y}$, where $\Delta(\mathbf{y}, \mathbf{y}^i)$ is a loss function which measures how much $\mathbf{y}$ and $\mathbf{y}^i$ differ. For these experiments, we used pixel-wise error as our loss function. The cutting plane algorithm finds at each step the most violated constraint, which corresponds to, for each d^i, finding the $\mathbf{y}$ which maximizes $\boldsymbol{\theta}^T\mathbf{f}(\mathbf{x}^i, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}^i)$; it then adds a new constraint based on these values. To find the most violated constraint, we tried using (appropriately modified versions of) both ICM and MP, which we refer to as ICM-MV and MP-MV. For our reported results, we initialized with an empty constraint set; initializing with constraints corresponding to PL instantiations did not improve performance. This method has a hyper-parameter C which we chose to maximize performance on a small subset of the test set. This method usually converged in about 150-170 iterations.

[2] In one round of Gibbs sampling, each node is resampled once, in random order.

Table 1. Pixel-wise ICM Test Error

Learning Method	Test Error	Std Dev
Lower Bound	.120	.005
Independent	.225	.014
PL	.461	.044
CD-1	.225	.016
CD-10	.219	.015
CD-100	.225	.014
MM(ICM-MV)	.217	.009
MM(MP-MV)	.218	.007
CCG(Gibbs-1+PL)	.225	.016
CCG(Gibbs-10+PL)	.218	.015
CCG(Gibbs-100+PL)	.217	.015
CCG(ICM+PL)	.200	.013
CCG(MP+PL)	.198	.015
CCG(ICM-MV+PL)	.192	.011
CCG(MP-MV+PL)	**.190**	.013

Contrastive Constraint Generation (CCG). Our method as described in Section 5.2. For initialization, we tried empty initialization and using the PL sub-objectives. We tried seven total ways of generating instantiations. First, we used the same sampling procedures as the CD variants – Gibbs-1, Gibbs-10, and Gibbs-100. Next, we used the approximate MAP procedures ICM and MP to generate instantiations. Finally, we used the most-violated constraint procedures ICM-MV and MP-MV. We refer to, for example, CCG with MP instantiations and empty initialization as CCG(MP); with PL initialization, CCG(MP+PL). The number of iterations required to reach convergence varied based on the initialization and instantiation method, from about 20 iterations for CCG(ICM) to 85 with CCG(MP+PL), while for the Gibbs variants, convergence is not reached within 100 iterations (we stopped at this point).

To evaluate the learned weights, we needed a maximum a-posterior (MAP) inference algorithm to produce the most likely labeling at test time. We found that ICM consistently outperformed MP as a test-time inference algorithm, so we only report results using ICM for test-time inference. Results were generated using 10-fold cross-validation on the 715 images, reporting pixel-wise error. Standard deviations are computed based on the individual results for each fold.

Table 1 shows all tested algorithms except for CCG with empty initialization. Based on the computed standard deviations, a difference in error of about .01 is statistically significant according to an unpaired t-test. There are two important things to note in this table. First, methods using non-local instantiations clearly outperform methods using the more local instantiations generated by Gibbs sampling (even when run for 100 rounds). The best non-local method, CCG(MP-MV+PL), decreases absolute error over the best local method, CCG(Gibbs-100+PL), by 2.7%, a 12% relative reduction in error. Second, CCG significantly outperforms the other non-local method, MM; CCG(MP-MV+PL) reduces absolute error from MM(ICM-MV) by 2.7% (12% relative error reduction).

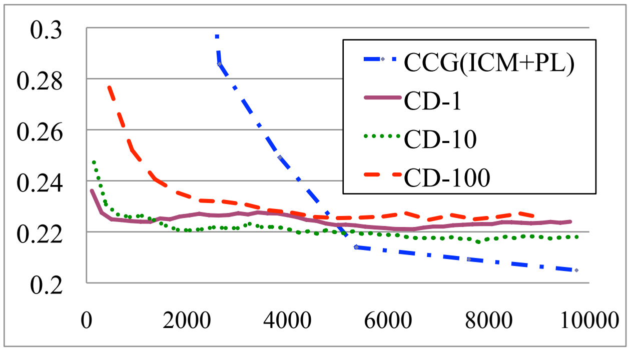

Figure 2. Test Error vs. Running Time (in seconds)

CCG is the only algorithm to improve substantially over Independent. PL more than doubles the pixel-wise error rate. This is because labels of neighboring regions are highly correlated — PL relies heavily on this during training, but at test time, the neighbors are no longer given. The strong locality of PL is a significant disadvantage for this problem.

Initializing CCG using PL resulted in small but noticeable gains when using ICM at test time (absolute difference ranged from .004 to .007). It also significantly reduced the number of iterations required to reach convergence. When using MP as the test-time inference method, we get very bad results with empty initialization (difference ranged from .145 to .295).

Each algorithm performs a differing amount of work at each iteration, ranging from CD (least) to CCG (most). Table 2 shows test accuracy vs. running time for the CD variants as well as for CCG(ICM+PL). The number of iterations pictured is 50k, 16.5k, 2k, and 6, for CD-1, CD-10, CD-100, and CCG(ICM+PL). CD-1 has converged, CD-10 probably has, while CD-100 has not. Despite the very small number of iterations for CCG, it is already significantly outperforming CD at this point. This shows that the non-local instantiations generated by ICM are much more informative than the instantiations generated by Gibbs sampling.

In fact, the difference between local and non-local methods is even greater than this graph suggests. After six iterations, CCG(Gibbs-1+PL), CCG(Gibbs-10+PL), and CCG(Gibbs-100+PL) have error rates .234, .227, and .229, vs. .205 for CCG(ICM+PL); all have similar running times. CD-n is much faster than CCG(Gibbs-n+PL), with comparable results at convergence. The main reason for this is that batch optimization is much slower than stochastic optimization, at least initially. In future work, we plan to implement an SGD version of CCG and compare it to CD.

7. Discussion and Related Work

LeCun & Huang (2005) and Smith & Eisner (2005) present frameworks for learning energy functions by comparing scores of sets of instantiations. The latter framework, called *contrastive estimation*, has the same functional form as one of our sub-objectives. However, Smith & Eisner (2005) primarily focus on unsupervised learning tasks, while this work is mostly aimed at supervised learning. More importantly, these two works focus on local contrastive objectives, whereas we propose non-local contrastive objectives, which address weakness in methods such as PL and CD.

Hyvärinen (2007) proposes an objective for learning binary Markov networks by trying to match ratios of probabilities between the model and the observed data. This objective minimizes squared-loss instead of log-loss; the advantage of log-loss is that contrastive objectives are a direct generalization of both log-likelihood and PL. Additionally, Hyvärinen (2007) only proposes matching local probability ratios.

Recently, Gutmann & Hyvärinen (2010) proposed a method based on learning probability ratios between the data distribution and some hand-constructed noise distribution. Similar to our method, it does not require computation of the global partition function. Unlike our method, it looks at probability ratios between different *distributions*, while our method looks at probability ratios between different *instantiations*.

Liang & Jordan (2008) provide an asymptotic analysis of contrastive objectives. They show that under certain conditions, the more different assignments are covered by the objective, the more efficient it is as an estimator. They apply this result to compare the efficiency of pseudo-likelihood to that of maximum-likelihood. Their results suggest that increasing the number of assignments covered by the contrastive objective leads to improved learning efficiency.

Max-margin-based methods, such as those proposed by Taskar et al. (2003), also do not need to compute the global partition function. As discussed above, the cutting-plane algorithm proposed by Tsochantaridis et al. (2005) is similar in spirit CCG. Like max-margin methods, CCG can learn using only MAP inference. The primary advantage of contrastive objectives over margin-based methods is that they can calibrate probabilities between instantiations.

Hinton et al. (2004) and Tieleman (2008) improve CD by adding non-local contrastive terms. Like CD, these methods do not correspond to an objective. Our analysis gives a theoretical grounding to non-local contrastive learning, including a well-defined objective.

References

Bengio, Y. Learning deep architectures for AI. *Found. and Trends in Mach. Learn.*, pp. 1–127, 2009.

Besag, J. Statistical analysis of non-lattice data. *The Statistician*, pp. 179–195, 1975.

Besag, J. On the statistical analysis of dirty pictures. *J. Royal Stat. Soc. B*, pp. 259–302, 1986.

Elidan, G., McGraw, I., and Koller, D. Residual belief propagation: Informed scheduling for asynchronous message passing. *Uncertainty in AI*, 2006.

Gould, S., Gao, T., and Koller, D. Region-based segmentation and object detection. *Neural Information Processing Systems*, 2009.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A New Estimation Principle for Unnormalized Statistical Models. *AI Stat.*, pp. 297–304, 2010.

Hinton, G. Training products of experts by minimizing contrastive divergence. *Neural Computation*, pp. 1771–1800, 2002.

Hinton, G. E., Welling, M., and Mnih, A. Wormholes improve contrastive divergence. *Neural Information Processing Systems*, 2004.

Hoefling, H. and Tibshirani, R. Estimation of sparse binary pairwise markov networks using psuedo-likelihoods. *J. Mach. Learn. R.*, pp. 883–906, 2009.

Hyvärinen, A. Some extensions of score matching. *Comp. Stat. & Data Analysis*, pp. 2499–2512, 2007.

Komodakis, N., Paragios, N., and Tziritas, G. MRF optimization via dual decomposition: Message-passing revisited. *Comp. Vis. & Pat. Recog.*, 2007.

LeCun, Y. and Huang, F.J. Loss functions for discriminitive training of energy-based models. *AI Statistics*, 2005.

Liang, P. and Jordan, M. I. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. *Int. Conf. on Mach. Learn.*, 2008.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Francisco, 1988.

Smith, N. and Eisner, J. Contrastive estimation: Training log-linear models on unlabeled data. *Assoc. for Comp. Ling.*, pp. 354–362, 2005.

Taskar, B., Guestrin, C., and Koller, D. Max-margin markov networks. *Neural Info. Proc. Sys.*, 2003.

Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. *Int. Conf. on Mach. Learn.*, pp. 1064–1071, 2008.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, pp. 1435–1484, 2005.