Journal of Machine Learning Research 11 (2010) 2001-2049 Submitted 12/09; Published 6/10

Posterior Regularization for Structured Latent Variable Models

Kuzman Ganchev KUZMAN@CIS.UPENN.EDU
Department of Computer and Information Science
University of Pennsylvania
Levine 302, 3330 Walnut St
Philadelphia PA, 19104, USA

João Graça JOAO.GRACA@L2F.INESC-ID.PT
L2F Inesc-ID Spoken Language Systems Lab
R. Alves Redol, 9
1000-029 Lisboa, Portugal

Jennifer Gillenwater JENGI@CIS.UPENN.EDU
Ben Taskar TASKAR@CIS.UPENN.EDU
Department of Computer and Information Science
University of Pennsylvania
Levine 302, 3330 Walnut St
Philadelphia PA, 19104, USA

Editor: Michael Collins

Abstract

We present posterior regularization, a probabilistic framework for structured, weakly supervised learning. Our framework efficiently incorporates indirect supervision via constraints on posterior distributions of probabilistic models with latent variables. Posterior regularization *separates* model complexity from the complexity of structural constraints it is desired to satisfy. By directly imposing decomposable regularization on the posterior moments of latent variables during learning, we retain the computational efficiency of the unconstrained model while ensuring desired constraints hold in expectation. We present an efficient algorithm for learning with posterior regularization and illustrate its versatility on a diverse set of structural constraints such as bijectivity, symmetry and group sparsity in several large scale experiments, including multi-view learning, cross-lingual dependency grammar induction, unsupervised part-of-speech induction, and bitext word alignment.[1]

Keywords: posterior regularization framework, unsupervised learning, latent variables models, prior knowledge, natural language processing

1. Introduction

In unsupervised problems where data has sequential, recursive, spatial, relational, and other kinds of structure, we often employ structured statistical models with latent variables to tease apart the underlying dependencies and induce meaningful semantic categories. Unsupervised part-of-speech and grammar induction, and word and phrase alignment for statistical machine translation in natural language processing are examples of such aims. Generative models (probabilistic grammars,

1. A preliminary version of the PR framework appeared in Graça et al. (2007). Various extensions and applications appeared in Ganchev et al. (2008a), Ganchev et al. (2008b), Ganchev et al. (2009), Graça et al. (2009a) and Graça et al. (2010).

graphical models, etc.) are usually estimated by maximizing the likelihood of the observed data by marginalizing over the hidden variables, typically via the Expectation Maximization (EM) algorithm. Because of computational and statistical concerns, generative models used in practice are very simplistic models of the underlying phenomena; for example, the syntactic structure of language or the language translation process. A pernicious problem with such models is that marginal likelihood may not guide the model towards the intended role for the latent variables, instead focusing on explaining irrelevant but common correlations in the data. Since we are mostly interested in the distribution of the latent variables in the hope that they capture *intended* regularities without direct supervision, controlling this latent distribution is critical. Less direct methods such as clever initialization, ad hoc procedural modifications, and complex data transformations are often used to affect the posteriors of latent variables in a desired manner.

A key challenge for structured, weakly supervised learning is developing a flexible, declarative framework for expressing structural constraints on latent variables arising from prior knowledge and indirect supervision. Structured models have the ability to capture a very rich array of possible relationships, but adding complexity to the model often leads to intractable inference. In this article, we present the posterior regularization (PR) framework (Graça et al., 2007), which *separates* model complexity from the complexity of structural constraints it is desired to satisfy. Unlike parametric regularization in a Bayesian framework, our approach incorporates data-dependent constraints that are easy to encode as information about model *posteriors* on the observed data, but may be difficult to encode as information about model parameters through Bayesian *priors*. In Sections 5-8 we describe a variety of such useful prior knowledge constraints in several application domains.

The contributions of this paper are:

- A flexible, declarative framework for structured, weakly supervised learning via posterior regularization.
- An efficient algorithm for model estimation with posterior regularization.
- An extensive evaluation of different types of constraints in several domains: multi-view learning, cross-lingual dependency grammar induction, unsupervised part-of-speech induction, and bitext word alignment.
- A detailed explanation of the connections between several other recent proposals for weak supervision, including structured constraint-driven learning (Chang et al., 2007), generalized expectation criteria (Mann and McCallum, 2008, 2007) and Bayesian measurements (Liang et al., 2009).

The rest of this paper is organized as follows. Section 2 describes the posterior regularization framework and Section 3 illustrates the range of different types of weak supervision constraints representable in our framework. Section 4 describes the relationship between posterior regularization and other related frameworks. Sections 5-8 describe applications of PR to several problems: word alignment (§5), multi-view learning (§6), cross-lingual projection (§7) and inducing sparsity structure (§8). Section 9 concludes the paper and presents areas for future work.

2. Posterior Regularization Framework

In this section we describe the posterior regularization framework, which incorporates side-information into parameter estimation in the form of linear constraints on posterior expectations. As we will

show, this allows tractable learning and inference even when the constraints would be intractable to encode directly in the model parameters. By defining a flexible language for specifying diverse types of problem-specific prior knowledge, we make the framework applicable to a wide variety of probabilistic models, both generative and discriminative. In Sections 2.1-2.7 we will focus on generative models, and describe the case of discriminative models in Section 2.8. We will use a problem from natural language processing as a running example in the exposition:

> **Running Example** *The task is part-of-speech (POS) tagging with limited or no training data. Suppose we know that each sentence should have at least one verb and at least one noun, and would like our model to capture this constraint on the unlabeled sentences. The model we will be using is a first-order hidden Markov model (HMM).*

We describe four other applications with empirical results in Sections 5-8, but it will be easier to illustrate key concepts using this simple example.

2.1 Preliminaries and Notation

We assume that there is a natural division of variables into "input" variables $\mathbf{x}$ and "target" variables $\mathbf{y}$ for each data instance, where $\mathbf{x}$'s are always observed. We denote the set of all instances of unlabeled data as $\mathbf{X}$. In case of semi-supervised learning, we have some labeled data as well, and we will use the notation $(\mathbf{X}_L, \mathbf{Y}_L)$ to denote all the labeled instances.

The starting point for using the PR framework is a probabilistic model. Let θ be the parameters of the model. For now we assume a generative model $p_\theta(\mathbf{x}, \mathbf{y})$, and we use $\mathcal{L}(\theta) = \log p_\theta(\mathbf{X}_L, \mathbf{Y}_L) + \log \sum_{\mathbf{Y}} p_\theta(\mathbf{X}, \mathbf{Y}) + \log p(\theta)$ to denote the parameter-regularized log-likelihood of the data.

> **Running Example** *In the POS tagging example from above, we would use* $\mathbf{x} = \{x_1, x_2, \ldots x_{|\mathbf{x}|}\}$ *to denote a sentence (i.e., a sequence of words* x_i*) and* $\mathbf{y} = \{y_1, y_2, \ldots y_{|\mathbf{x}|}\}$ *to denote a possible POS assignment. Using an HMM, it is defined in the normal way as:*
>
> $$p_\theta(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{|\mathbf{x}|} \; p_\theta(y_i | y_{i-1}) \;\; p_\theta(x_i | y_i),$$
>
> *with* θ *representing the multinomial distributions directly, and where* $p_\theta(y_1 | y_0) = p_\theta(y_1)$ *represents a set of initial probabilities. Suppose we have a small labeled corpus and a larger unlabeled corpus. For a generative model such as an HMM, the log-likelihood (+ log-prior) is:*
>
> $$\mathcal{L}(\theta) = \log p_\theta(\mathbf{X}_L, \mathbf{Y}_L) + \log \sum_{\mathbf{Y}} p_\theta(\mathbf{X}, \mathbf{Y}) + \log p(\theta),$$
>
> *where corpus probabilities are products over instances:* $p_\theta(\mathbf{x}, \mathbf{y}) = \prod p_\theta(\mathbf{x}, \mathbf{y})$ *and analogously for* $\mathbf{X}_L, \mathbf{Y}_L$*; and where* $p(\theta)$ *is a prior distribution over the parameters* θ*.*

2.2 Regularization via Posterior Constraints

The goal of the posterior regularization framework is to restrict the space of the model posteriors on unlabeled data as a way to guide the model towards desired behavior. In this section we describe a version of PR specified with respect to a set of constraints. In this case, posterior information is

specified with sets $\mathcal{Q}$ of allowed distributions over the hidden variables $\mathbf{y}$. We will define $\mathcal{Q}$ in terms of *constraint* features $\phi(\mathbf{X}, \mathbf{Y})$ and their expectations.[2]

> **Running Example** *Recall that in our running example, we want to bias learning so that each sentence is labeled to contain at least one verb. To encode this formally, we define a feature* $\phi(\mathbf{x}, \mathbf{y}) =$ *"number of verbs in* $\mathbf{y}$*", and require that this feature has expectation at least 1. For consistency with the rest of the exposition and standard optimization literature, we will use the equivalent* $\phi(\mathbf{x}, \mathbf{y}) =$ *"negative number of verbs in* $\mathbf{y}$*" and require this has expectation at most -1:*[3]
>
> $$\mathcal{Q}_{\mathbf{x}} = \{q_{\mathbf{x}}(\mathbf{y}) \ : \ \mathbf{E}_q[\phi(\mathbf{x}, \mathbf{y})] \leq -1\}.$$
>
> *Note that we enforce the constraint only in expectation, so there might be a labeling with non-zero probability that does not contain a verb. To actually enforce this constraint in the model would break the first-order Markov property of the distribution.*[4] *In order to also require at least one noun per sentence in expectation, we would add another constraint feature, so that* ϕ *would be a function from* $\mathbf{x}, \mathbf{y}$ *pairs to* $\mathbb{R}^2$.

We define $\mathcal{Q}$, the set of valid distributions, with respect to the *expectations* of constraint features, rather than their probabilities, so that our objective leads to an efficient algorithm. As we will see later in this section, we also require that the constraint features decompose as a sum in order to ensure an efficient algorithm. More generally than in the running example, we will define constraints over an entire corpus:

$$\textbf{Constrained Posterior Set}: \qquad \mathcal{Q} = \{q(\mathbf{Y}) : \mathbf{E}_q[\phi(\mathbf{X}, \mathbf{Y})] \leq \mathbf{b}\}.$$

In words, $\mathcal{Q}$ denotes the region where constraint feature expectations are bounded by $\mathbf{b}$. Additionally, it is often useful to allow small violations whose norm is bounded by $\varepsilon \geq 0$:

$$\textbf{Constrained Set (with slack)}: \quad \mathcal{Q} = \{q(\mathbf{Y}) : \exists \xi, \ \mathbf{E}_q[\phi(\mathbf{X}, \mathbf{Y})] - \mathbf{b} \leq \xi; \ ||\xi||_\beta \leq \varepsilon\}. \qquad (1)$$

Here ξ is a vector of slack variables and $||\cdot||_\beta$ denotes some norm. Note that the PR method we describe will only be useful if $\mathcal{Q}$ is non-empty:

Assumption 2.1 *$\mathcal{Q}$ is non-empty.*

We explore several types of constraints in Sections 5-8, including: constraints similar to the running example, where each hidden state is constrained to appear at most once in expectation; constraints that bias two models to agree on latent variables in expectation; constraints that enforce a particular group-sparsity of the posterior moments. The constraint set defined in Equation 1 is usually referred to as inequality constraints with slack, since setting $\varepsilon = 0$ enforces inequality constraints strictly. The derivations for equality constraints are very similar to the derivations for inequality so we leave them out in the interest of space. Note also that we can encode equality

2. Note: the constraint features do not appear anywhere in the model. If the model has a log-linear form, then it would be defined with respect to a different set of *model* features, not related to the *constraint* features we consider here.
3. Note that the distribution $q_{\mathbf{x}}$ and $\mathcal{Q}_{\mathbf{x}}$ depend on $\mathbf{x}$ because the features $\phi(\mathbf{x}, \mathbf{y})$ might depend on the particular example $\mathbf{x}$.
4. At every position in the sentence, we would need to know whether a verb was used at any other position.

Symbol	Meaning
$\mathbf{x}$	(observed) input variables for a particular example
$\mathbf{y}$	(usually hidden) output variables for a particular example
$\mathbf{X}, \mathbf{Y}$	$\mathbf{x}$ and $\mathbf{y}$ for the entire unlabeled portion of the corpus
$\mathbf{X}_L, \mathbf{Y}_L$	$\mathbf{x}$ and $\mathbf{y}$ for the entire labeled portion of the corpus (possibly empty)
$p_\theta(\mathbf{x}, \mathbf{y})$	a generative, joint model with parameters θ
$\mathcal{L}(\theta)$	data log-likelihood and parameter prior: $\log p_\theta(\mathbf{X}_L, \mathbf{Y}_L) + \log \sum_{\mathbf{Y}} p_\theta(\mathbf{X}, \mathbf{Y}) + \log p(\theta)$
$\mathcal{Q}_\mathbf{x}, \mathcal{Q}$	posterior regularization set: constrained set of desired data-conditional distributions
$\phi(\mathbf{x}, \mathbf{y})$	constraint features: used to encode posterior regularization
$\mathbf{b}$	bounds on the desired expected values of constraint features
ξ	slack variables used to allow small violations of constraints
$J_\mathcal{Q}(\theta)$	posterior regularized likelihood: $\mathcal{L}(\theta) - \mathbf{KL}(\mathcal{Q} \parallel p_\theta(\mathbf{Y}\|\mathbf{X}))$

Table 1: Summary of notation used.

constraints by adding two inequality constraints, although this will leave us with twice as many variables in the dual. The assumption of linearity of the constraints is computationally important, as we will show below. For now, we do not make any assumptions about the features $\phi(\mathbf{x}, \mathbf{y})$, but if they factor in the same way as the model, then we can use the same inference algorithms in PR training as we use for the original model (see Proposition 2.2). In PR, the log-likelihood of a model is penalized with the KL-divergence between the desired distribution space $\mathcal{Q}$ and the model posteriors,

$$\mathbf{KL}(\mathcal{Q} \parallel p_\theta(\mathbf{Y}|\mathbf{X})) = \min_{q \in \mathcal{Q}} \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})).$$

The posterior-regularized objective is:

$$\textbf{Posterior Regularized Likelihood}: \quad J_\mathcal{Q}(\theta) = \mathcal{L}(\theta) - \mathbf{KL}(\mathcal{Q} \parallel p_\theta(\mathbf{Y}|\mathbf{X})). \tag{2}$$

The objective trades off likelihood and distance to the desired posterior subspace (modulo getting stuck in local maxima) and provides an effective means of controlling the posteriors. In many cases, prior knowledge is easy to specify in terms of posteriors, and much more difficult to specify as priors on model parameters or by explicitly adding constraints to the model. A key advantage of using regularization on posteriors is that the learned model itself remains simple and tractable, while during learning it is driven to obey the constraints through setting appropriate parameters θ. The advantage of imposing the constraints via KL-divergence from the posteriors is that the objective above can be optimized using a simple EM scheme described in Section 2.6. It is also possible to use a similar algorithm to maximize $\mathcal{L}(\theta) - \alpha\mathbf{KL}(\mathcal{Q} \parallel p_\theta(\mathbf{Y} \mid \mathbf{X}))$, for $\alpha \in [0,1]$. See Appendix A for details. Note that the algorithm we will present in Section 2.6 will not allow us to optimize an objective with $\alpha > 1$, and this leads us to have both a KL-penalty term in Equation 2 and also to potentially have slack in the definition of the constraint set $\mathcal{Q}$. We do not need to allow slack in the objective, as long as we are sure that the constraint set $\mathcal{Q}$ is non-empty. At increased computational cost, it is also possible to eliminate the KL-penalty portion of the objective, instead directly constraining the model's posterior distribution to be inside the constraint set $p_\theta(\mathbf{Y}|\mathbf{X}) \in \mathcal{Q}$. See Section 4 for details. Figure 1 illustrates the objective in Equation 2. Normal maximum likelihood training is equivalent to minimizing the KL distance between the distribution concentrated on $\mathbf{X}$ and the set of distributions representable by the model. Any particular setting of the model parameters results in

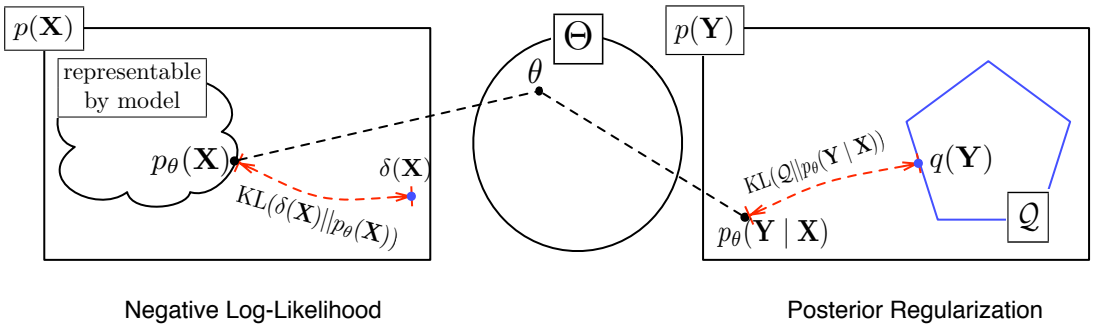


Figure 1: An illustration of the PR objective for generative models, as a sum of two KL terms. The symbol Θ represents the set of possible model parameters, $\delta(\mathbf{X})$ is a distribution that puts probability 1 on $\mathbf{X}$ and 0 on all other assignments. Consequently $\mathbf{KL}(\delta(\mathbf{X})||p_\theta(\mathbf{X})) = \mathcal{L}(\theta)$. (We ignore the parameter prior and additional labeled data in this figure for clarity.)

the posterior distribution $p_\theta(\mathbf{Y}|\mathbf{X})$. PR adds to the maximum likelihood objective a corresponding KL distance for this distribution. If $\mathcal{Q}$ has only one distribution, then we recover labeled maximum likelihood training. This is one of the justifications for the use and the particular direction of the KL distance in the penalty term.

> **Running Example** *In order to represent a corpus-wide constraint set $\mathcal{Q}$ for our POS problem, we stack the constraint features into a function from $\mathbf{X},\mathbf{Y}$ pairs (sentences, part-of-speech sequences) to $\mathbb{R}^{2|\mathbf{X}|}$, where $|\mathbf{X}|$ is the number of sentences in our unlabeled corpus. For the POS tagging example, the PR objective penalizes parameters that do not assign each sentence at least one verb and one noun in expectation.*

For PR to be successful, the model $p_\theta(\mathbf{Y}|\mathbf{X})$ has to be expressive enough to ensure that the learned model has posteriors $p_\theta(\mathbf{Y}|\mathbf{X})$ in or nearly in $\mathcal{Q}$. Even if that is the case, the same parameters might not ensure that the constraints are satisfied on a test corpus, so we could also use $q(\mathbf{Y}) = \arg\min_{q'\in\mathcal{Q}} \mathbf{KL}(q'(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X}))$ for prediction instead of $p_\theta(\mathbf{Y}|\mathbf{X})$. We will see in Sections 5 and 7 that this sometimes results in improved performance. Chang et al. (2007) report similar results for their constraint-driven learning framework.

2.3 Slack Constraints vs. Penalty

In order for our objective to be well defined, $\mathcal{Q}$ must be non-empty. When there are a large number of constraints, or when the constraint features ϕ are defined by some instance-specific process, it might not be easy to choose constraint values $\mathbf{b}$ and slack ε that lead to satisfiable constraints. It is sometimes easier to penalize slack variables instead of setting a bound ε on their norm. In these cases, we add a slack penalty to the regularized likelihood objective in Equation 2:

$$\begin{aligned} \mathcal{L}(\theta) \quad - \min_{q,\xi} \quad & \mathbf{KL}\left(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})\right) + \sigma||\xi||_\beta \\ & \text{s.t.} \quad \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} \leq \xi. \end{aligned} \tag{3}$$

The slack-constrained and slack-penalized versions of the objectives are equivalent in the sense that they follow the same regularization path: for every ε there exists some σ that results in identical parameters θ. Note that while we have used a norm $||\cdot||_\beta$ to impose a cost on violations of the constraints, we could have used any arbitrary convex penalty function, for which the minimal q is easily computable.

2.4 Computing the Posterior Regularizer

In this subsection, we describe how to compute the objective we have introduced for fixed parameters θ. The regularization term is stated in Equations 2 and 3 in terms of an optimization problem. We assume that we have algorithms to do inference[5] in the statistical model of interest, p_θ. We describe the computation of the regularization term for the inequality constraints:

$$\min_{q,\xi} \quad \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) \qquad \text{s.t.} \quad \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} \leq \xi; \quad ||\xi||_\beta \leq \varepsilon. \tag{4}$$

Proposition 2.1 *The regularization problems for PR with inequality constraints in Equation 4 can be solved efficiently in its dual form. The primal solution q^* is unique since KL divergence is strictly convex and is given in terms of the dual solution λ^* by:*

$$q^*(\mathbf{Y}) = \frac{p_\theta(\mathbf{Y}|\mathbf{X}) \exp\{-\lambda^* \cdot \phi(\mathbf{X},\mathbf{Y})\}}{Z(\lambda^*)} \tag{5}$$

where $Z(\lambda^*) = \sum_{\mathbf{Y}} p_\theta(\mathbf{Y}|\mathbf{X}) \exp\{-\lambda^* \cdot \phi(\mathbf{X},\mathbf{Y})\}$. Define $||\cdot||_{\beta^*}$ as the dual norm of $||\cdot||_\beta$. The dual of the problem in Equation 4 is:

$$\max_{\lambda \geq 0} \quad -\mathbf{b} \cdot \lambda - \log Z(\lambda) \; - \varepsilon \, ||\lambda||_{\beta^*} . \tag{6}$$

The proof is included in Appendix B using standard Lagrangian duality results and strict convexity of KL (e.g., Bertsekas, 1999). The dual form in Equation 6 is typically computationally more tractable than the primal form (Equation 4) because there is one dual variable per expectation constraint, while there is one primal variable per labeling $\mathbf{Y}$. For structured models, this is typically intractable. An analogous proposition can be proven for the objective with penalties (Equation 3), with almost identical proof. We omit this for brevity.

2.5 Factored $q(\mathbf{Y})$ for Factored Constraints

The form of the optimal q with respect to $p_\theta(\mathbf{Y}|\mathbf{X})$ and ϕ has important computational implications.

Proposition 2.2 *If $p_\theta(\mathbf{Y}|\mathbf{X})$ factors as a product of clique potentials over a set of cliques $\mathcal{C}$, and $\phi(\mathbf{X},\mathbf{Y})$ factors as a sum over some subset of those cliques, then the optimizer $q^*(\mathbf{Y})$ of Equation 5 will also factor as a product of potentials of cliques in $\mathcal{C}$.*

This is easy to show. Our assumptions are a factorization for p_θ:

$$\textbf{Factored Posteriors}: \quad p(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_{c \in \mathcal{C}} \psi(\mathbf{X}, \mathbf{Y}_c)$$

5. Specifically, we need to be able to compute marginal distributions efficiently.

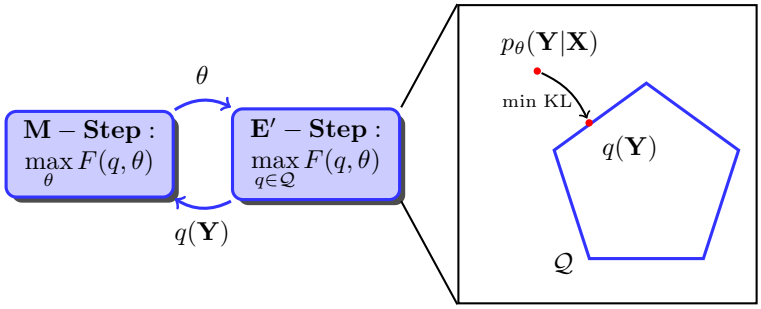


Figure 2: Modified EM for optimizing generative PR objective $\mathcal{L}(\theta) - \mathbf{KL}(Q \parallel p_\theta(\mathbf{Y}|\mathbf{X}))$.

and the same factorization for ϕ:

$$\textbf{Factored Features}: \quad \phi(\mathbf{X},\mathbf{Y}) = \sum_{c \in \mathcal{C}} \phi(\mathbf{X},\mathbf{Y}_c)$$

which imply that $q^*(\mathbf{Y})$ will also factor as a product over the cliques $\mathcal{C}$:

$$\begin{aligned} \textbf{Factored Solution}: \quad q^*(\mathbf{Y}) &= \frac{1}{Z(\mathbf{X})Z(\lambda)} \prod_{c \in \mathcal{C}} \psi(\mathbf{X},\mathbf{Y}_c) \ \exp\{-\lambda \cdot \phi(\mathbf{X},\mathbf{Y}_c)\} \\ &= \frac{1}{Z'(\mathbf{X})} \prod_{c \in \mathcal{C}} \psi'(\mathbf{X},\mathbf{Y}_c), \end{aligned}$$

where $\psi'(\mathbf{X},\mathbf{Y}_c) = \psi(\mathbf{X},\mathbf{Y}_c)\exp\{-\lambda \cdot \phi(\mathbf{X},\mathbf{Y}_c)\}$ and $Z'(\mathbf{X}) = Z(\mathbf{X})Z(\lambda)$.

2.6 Generative Posterior Regularization via Expectation Maximization

This section presents an optimization algorithm for the PR objective. The algorithm we present is a minorization-maximization algorithm akin to EM, and both slack-constrained and slack-penalized formulations can be optimized using it. To simplify the exposition, we focus first on slack-constrained version, and leave a treatment of optimization of the slack-penalized version to Section 2.7.

Recall the standard expectation maximization (EM) algorithm used to optimize marginal likelihood $\mathcal{L}(\theta) = \log \sum_{\mathbf{Y}} p_\theta(\mathbf{X},\mathbf{Y})$. Again, for clarity of exposition, we ignore $\log p(\theta)$, the prior on θ, as well as $\log p_\theta(\mathbf{X}_L,\mathbf{Y}_L)$, the labeled data term, as they are simple to incorporate, just as in regular EM. Neal and Hinton (1998) describe an interpretation of the EM algorithm as block coordinate ascent on a function that lower-bounds $\mathcal{L}(\theta)$, which we also use below. By Jensen's inequality, we define a lower-bound $F(q,\theta)$ as

$$\mathcal{L}(\theta) = \log \sum_{\mathbf{Y}} q(\mathbf{Y}) \frac{p_\theta(\mathbf{X},\mathbf{Y})}{q(\mathbf{Y})} \geq \sum_{\mathbf{Y}} q(\mathbf{Y}) \log \frac{p_\theta(\mathbf{X},\mathbf{Y})}{q(\mathbf{Y})} = F(q,\theta).$$

we can re-write $F(q,\theta)$ as

$$\begin{aligned} F(q,\theta) &= \sum_{\mathbf{Y}} q(\mathbf{Y}) \log(p_\theta(\mathbf{X}) p_\theta(\mathbf{Y}|\mathbf{X})) - \sum_{\mathbf{Y}} q(\mathbf{Y}) \log q(\mathbf{Y}) \\ &= \mathcal{L}(\theta) - \sum_{\mathbf{Y}} q(\mathbf{Y}) \log \frac{q(\mathbf{Y})}{p_\theta(\mathbf{Y}|\mathbf{X})} \\ &= \mathcal{L}(\theta) - \mathbf{KL}(q(\mathbf{Y}) || p_\theta(\mathbf{Y}|\mathbf{X})). \end{aligned}$$

Using this interpretation, we can view EM as performing coordinate ascent on $F(q,\theta)$. Starting from an initial parameter estimate θ^0, the algorithm iterates two block-coordinate ascent steps until a convergence criterion is reached:

$$\begin{aligned} \mathbf{E}: q^{t+1} &= \arg\max_q F(q,\theta^t) = \arg\min_q \mathbf{KL}(q(\mathbf{Y}) \parallel p_{\theta^t}(\mathbf{Y} \mid \mathbf{X})), \\ \mathbf{M}: \theta^{t+1} &= \arg\max_\theta F(q^{t+1},\theta) = \arg\max_\theta \mathbf{E}_{q^{t+1}}[\log p_\theta(\mathbf{X},\mathbf{Y})]. \end{aligned} \tag{7}$$

It is easy to see that the E-step sets $q^{t+1}(\mathbf{Y}) = p_{\theta^t}(\mathbf{Y}|\mathbf{X})$.

The PR objective (Equation 2) is

$$J_\mathcal{Q}(\theta) = \max_{q \in \mathcal{Q}} F(q,\theta) = \mathcal{L}(\theta) - \min_{q(\mathbf{Y}) \in \mathcal{Q}} \mathbf{KL}(q(\mathbf{Y}) || p_\theta(\mathbf{Y}|\mathbf{X})),$$

where $\mathcal{Q} = \{q(\mathbf{Y}) : \exists \xi,\ \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} \leq \xi;\ ||\xi||_\beta \leq \varepsilon\}$. In order to optimize this objective, it suffices to modify the E-step to include the constraints:

$$\mathbf{E}': q^{t+1} = \arg\max_{q \in \mathcal{Q}} F(q,\theta^t) = \arg\min_{q \in \mathcal{Q}} \mathbf{KL}(q(\mathbf{Y}) \parallel p_{\theta^t}(\mathbf{Y}|\mathbf{X})). \tag{8}$$

The projected posteriors $q^{t+1}(\mathbf{Y})$ are then used to compute sufficient statistics and update the model's parameters in the M-step, which remains unchanged, as in Equation 7. This scheme is illustrated in Figure 2.

Proposition 2.3 *The modified EM algorithm illustrated in Figure 2, which iterates the modified E-step (Equation 8) with the normal M-step (Equation 7), monotonically increases the PR objective:* $J_\mathcal{Q}(\theta^{t+1}) \geq J_\mathcal{Q}(\theta^t)$.

Proof: The proof is analogous to the proof of monotonic increase of the standard EM objective. Essentially,

$$J_\mathcal{Q}(\theta^{t+1}) = F(q^{t+2},\theta^{t+1}) \geq F(q^{t+1},\theta^{t+1}) \geq F(q^{t+1},\theta^t) = J_\mathcal{Q}(\theta^t).$$

The two inequalities are ensured by the $\mathbf{E}'$-step and M-step. $\mathbf{E}'$-step sets $q^{t+1} = \arg\max_{q \in \mathcal{Q}} F(q,\theta^t)$, hence $J_\mathcal{Q}(\theta^t) = F(q^{t+1},\theta^t)$. The M-step sets $\theta^{t+1} = \arg\max_\theta F(q^{t+1},\theta)$, hence $F(q^{t+1},\theta^{t+1}) \geq F(q^{t+1},\theta^t)$. Finally, $J_\mathcal{Q}(\theta^{t+1}) = \max_{q \in \mathcal{Q}} F(q,\theta^{t+1}) \geq F(q^{t+1},\theta^{t+1})$ ■

Note that the proposition is only meaningful when $\mathcal{Q}$ is non-empty and $J_\mathcal{Q}$ is well-defined. As for standard EM, to prove that coordinate ascent on $F(q,\theta)$ converges to stationary points of $J_\mathcal{Q}(\theta)$, we need to make additional assumptions on the regularity of the likelihood function and boundedness of the parameter space as in Tseng (2004). This analysis can be easily extended to our setting, but is beyond the scope of the current paper.

We can use the dual formulation of Proposition 2.1 to perform the projection. Proposition 2.2 implies that we can use the same algorithms to perform inference in our projected model q as we did in our original model p_θ. We illustrate this with the running example.

Running Example *For the POS tagging example with zero slack, the optimization problem we need to solve is:*

$$\arg\min_{q} \quad \mathbf{KL}(q(\mathbf{Y}) \,||\, p_\theta(\mathbf{Y}|\mathbf{X})) \qquad \text{s.t.} \quad \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] \leq -\mathbf{1}$$

where $\mathbf{1}$ *is a vector of with 1 in each entry. The dual formulation is given by*

$$\arg\max_{\lambda \geq 0} \quad \mathbf{1}\cdot\lambda - \log Z(\lambda) \quad \textit{with} \quad q^*(\mathbf{Y}) = \frac{p_\theta(\mathbf{Y}|\mathbf{X})\exp\{-\lambda^*\cdot\phi(\mathbf{X},\mathbf{Y})\}}{Z(\lambda^*)}. \tag{9}$$

We can solve the dual optimization problem by projected gradient ascent. The HMM model can be factored as products over sentences, and each sentence as a product of emission probabilities and transition probabilities.

$$p_\theta(\mathbf{y} \mid \mathbf{x}) = \frac{\prod_{i=1}^{|\mathbf{x}|} p_\theta(y_i|y_{i-1})p_\theta(x_i|y_i)}{p_\theta(\mathbf{x})} \tag{10}$$

where $p_\theta(y_1|y_0) = p_\theta(y_1)$ *are the initial probabilities of our HMM. The constraint features* ϕ *can be represented as a sum over sentences and further as a sum over positions in the sentence:*

$$\phi(\mathbf{x},\mathbf{y}) \quad = \quad \sum_{i=1}^{|\mathbf{x}|}\phi_i(\mathbf{x},y_i) \quad = \quad \sum_{i=1}^{|\mathbf{x}|}\begin{cases}(-1,0)^\top & \textit{if } y_i \textit{ is a verb in sentence } \mathbf{x}\\ (0,-1)^\top & \textit{if } y_i \textit{ is a noun in sentence } \mathbf{x}\\ (0,0)^\top & \textit{otherwise}\end{cases} \tag{11}$$

combining the factored Equations 10 and 11 with the definition of $q(\mathbf{Y})$ *we see that* $q(\mathbf{Y})$ *must also factor as a first-order Markov model for each sentence:*

$$q^*(\mathbf{Y}) \propto \quad \prod_{\mathbf{x}\in\mathbf{X}}\prod_{i=1}^{|\mathbf{x}|} p_\theta(y_i|y_{i-1})p_\theta(x_i|y_i)e^{-\lambda^*\cdot\phi_i(\mathbf{x},y_i)}.$$

Hence $q^*(\mathbf{Y})$ *is just a first-order Markov model for each sentence, and we can compute the normalizer* $Z(\lambda^*)$ *and marginals* $q(y_i)$ *for each example using forward-backward. This allows computation of the dual objective in Equation 9 as well as its gradient efficiently. The gradient of the dual objective is* $1 - \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})]$*. We can use projected gradient (Bertsekas, 1999) to perform the optimization, and the projection can be done sentence-by-sentence allowing for online optimization such as stochastic gradient. Optimization for non-zero slack case can be done using projected subgradient (since the norm is not smooth).*

Note that on *unseen* unlabeled data, the learned parameters θ might not satisfy the constraints on posteriors exactly, although typically they are fairly close if the model has enough capacity.

2.7 Penalized Slack via Expectation Maximization

If our objective is specified using slack-penalty such as in Equation 3, then we need a slightly different E-step. Instead of restricting $q \in \mathcal{Q}$, the modified $\mathbf{E}'$-step adds a cost for violating the constraints

$$\begin{aligned}\mathbf{E}' : \min_{q,\xi} \quad & \mathbf{KL}\left(q(\mathbf{Y}) \,||\, p_\theta(\mathbf{Y}|\mathbf{X})\right) + \sigma||\xi||_\beta \\ \text{s.t.} \quad & \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} \leq \xi.\end{aligned} \tag{12}$$

An analogous monotonic improvement of modified EM can be shown for the slack-penalized objective. The dual of Equation 12 is

$$\max_{\lambda \geq 0} \; -\mathbf{b} \cdot \lambda - \log Z(\lambda) \qquad \text{s.t.} \quad ||\lambda||_{\beta^*} \leq \sigma.$$

2.8 PR for Discriminative Models

The PR framework can be used to guide learning in discriminative models as well as generative models. In the case of a discriminative model, we only have $p_\theta(\mathbf{y}|\mathbf{x})$, and the likelihood does not depend on unlabeled data. Specifically,

$$\mathcal{L}^D(\theta) = \log p_\theta(\mathbf{Y}_L|\mathbf{X}_L) + \log p(\theta),$$

where $(\mathbf{Y}_L, \mathbf{X}_L)$ are any available labeled data and $\log p(\theta)$ is a prior on the model parameters. With this definition of $\mathcal{L}(\theta)$ for discriminative models we will optimize the discriminative PR objective (zero-slack case):

$$\textbf{Discriminative PR Likelihood}: \quad J^D_Q(\theta) = \mathcal{L}^D(\theta) - \mathbf{KL}(Q \,\|\, p_\theta(\mathbf{Y}|\mathbf{X})). \tag{13}$$

In the absence of both labeled data and a prior on parameters $p(\theta)$, the objective in Equation 2 is optimized (equal to zero) for any $p_\theta(\mathbf{Y} \mid \mathbf{X}) \in Q$. If we employ a parametric prior on θ, then we will prefer parameters that come close to satisfying the constraints, where proximity is measured by KL-divergence.

> **Running Example** *For the POS tagging example, our discriminative model might be a first order conditional random field. In this case we model:*
>
> $$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{\exp\{\theta \cdot \mathbf{f}(\mathbf{x}, \mathbf{y})\}}{Z_\theta(\mathbf{x})}$$
>
> *where* $Z_\theta(\mathbf{x}) = \sum_{\mathbf{y}} \exp\{\theta \cdot \mathbf{f}(\mathbf{x}, \mathbf{y})\}$ *is a normalization constant and* $\mathbf{f}(\mathbf{x}, \mathbf{y})$ *are the* model *features. We will use the same constraint features as in the generative case:* $\phi(\mathbf{x}, \mathbf{y}) =$ *"negative number of verbs in* $\mathbf{y}$*", and define* $Q_\mathbf{x}$ *and* $q_\mathbf{x}$ *also as before. Note that* $\mathbf{f}$ *are features used to define the model and do not appear anywhere in the constraints while* ϕ *are constraint features that do not appear anywhere in the model.*

Traditionally, the EM algorithm is used for learning generative models (the model can condition on a subset of observed variables, but it must define a distribution over some observed variables). The reason for this is that EM optimizes marginal log-likelihood ($\mathcal{L}$ in our notation) of the observed data $\mathbf{X}$ according to the model. In the case of a discriminative model, $p_\theta(\mathbf{Y}|\mathbf{X})$, we do not model the distribution of the observed data, the value of $\mathcal{L}^D$ as a function of θ depends only on the parametric prior $p(\theta)$ and the labeled data. By contrast, the PR objective uses the KL term and the corresponding constraints to bias the model parameters. These constraints depend on the observed data $\mathbf{X}$ and if they are sufficiently rich and informative, they can be used to train a discriminative model. In the extreme case, consider a constraint set Q that contains only a single distribution q, with $q(\mathbf{Y}^*) = 1$. So, q is concentrated on a particular labeling $\mathbf{Y}^*$. In this case, the PR objective in Equation 13 reduces to

$$J^D_Q(\theta) = \mathcal{L}^D(\theta) + \log p_\theta(\mathbf{Y}^*|\mathbf{X}) = \log p(\theta) + \log p_\theta(\mathbf{Y}_L|\mathbf{X}_L) + \log p_\theta(\mathbf{Y}^*|\mathbf{X}).$$

§#	Problem	Gen/Disc	p/q	Summary of Structural Constraints
§5	Word Alignment	G	q	Translation process is symmetric and bijective
§6	Multi-view learning	D	q	Multiple views should agree on label distribution
§7	Dependency Parsing	G+D	p	Noisy, partially observed labels encoded in ϕ and $\mathbf{b}$
§8	Part-of-speech induction	G	p	Sparsity structure independent of model parameters: each word should be generated by a small number of POS tags

Table 2: Summary of applications of Posterior Regularization described in this paper. Gen/Disc refers to generative or discriminative models. The p/q column shows whether we use the original model p or the projected distribution q at decode time.

Thus, if Q is informative enough to uniquely specify a labeling of the unlabeled data, the PR objective reduces to the supervised likelihood objective. When Q specifies a range of distributions, such as the one for multi view learning (Section 6), PR biases the discriminative model to have $p_\theta(\mathbf{Y}|\mathbf{X})$ close to Q.

Equation 13 can also be optimized with a block-coordinate ascent, leading to an EM style algorithm very similar to the one presented in Section 2.6. We define a lower bounding function:

$$F'(q,\theta) = -\mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) = \sum_{\mathbf{Y}} q(\mathbf{Y}) \log \frac{p_\theta(\mathbf{Y}|\mathbf{X})}{q(\mathbf{Y})}.$$

Clearly, $\max_{q \in Q} F'(q,\theta) = -\mathbf{KL}(Q \parallel p_\theta(\mathbf{Y}|\mathbf{X}))$ so $F'(q,\theta) \leq -\mathbf{KL}(Q \parallel p_\theta(\mathbf{Y}|\mathbf{X}))$ for $q \in Q$.

The modified $\mathbf{E}'$ and $\mathbf{M}'$ steps are:[6]

$$\begin{aligned} \mathbf{E}' : q^{t+1} &= \arg\max_{q \in Q} F'(q, \theta^t) = \arg\min_{q \in Q} \mathbf{KL}(q(\mathbf{Y}) \parallel p_{\theta^t}(\mathbf{Y}|\mathbf{X})), \\ \mathbf{M}' : \theta^{t+1} &= \arg\max_{\theta} F'(q^{t+1}, \theta) = \arg\max_{\theta} \mathbf{E}_{q^{t+1}} [\log p_\theta(\mathbf{Y}|\mathbf{X})]. \end{aligned} \tag{14}$$

Here the difference between Equation 7 and Equation 14 is that now there is no generative component in the lower-bound $F'(q,\theta)$ and hence we have a discriminative update to the model parameters in Equation 14.

3. Summary of Applications

Because the PR framework allows very diverse prior information to be specified in a single formalism, the application Sections (§5-§8) are very diverse in nature. This section attempts to summarize their similarities and differences without getting into the details of the problem applications and intuition behind the constraints. Table 2 summarizes the applications and constraints described in the rest of the paper while Table 3 summarizes the meanings of the variables $\mathbf{x}$, $\mathbf{y}$ and $\phi(\mathbf{X},\mathbf{Y})$ as well as the optimization procedures used for the applications presented in the sequel.

In the statistical word alignment application described in Section 5, the goal is to identify pairs or sets of words that are direct translations of each other. The statistical models used suffer from what

6. As with the **M-step** in Equation 7 we have ignored the prior $p(\theta)$ on model parameters and the labeled data terms, which can be easily incorporated in the $\mathbf{M}'$ step.

Application	Symbol	Meaning
Word Alignment	$\mathbf{x}$	Pair of sentences that are translations of each other.
	$\mathbf{y}$	Set of alignment links between words in the sentences (exponentially many possibilities).
	$\phi(\mathbf{x},\mathbf{y})$	Bijective: number of times each source word is used in the alignment $\mathbf{y}$. Symmetric: expected difference in number of times each link is used by source$\rightarrow$target model and target $\rightarrow$source model.
	OPT	Bijective: projected gradient. Symmetric: L-BFGS.
Multi-view learning	$\mathbf{x}$	Varies by application.
	$\mathbf{y}$	Varies by application.
	$\phi(\mathbf{x},\mathbf{y})$	Indexed by label; ± 1 if only one model predicts the label, and 0 if both or none predict the label.
	OPT	Closed form.
Dependency Parsing	$\mathbf{x}$	Natural language sentence as a sequence of words.
	$\mathbf{y}$	Dependency parse tree (set of edges from one word to another, forming a tree).
	$\phi(\mathbf{x},\mathbf{y})$	Number of edges in $\mathbf{y}$ that are in the set of translated edges.
	OPT	Line search.
Part-of-speech induction	$\mathbf{x}$	Natural language sentence as a sequence of words.
	$\mathbf{y}$	Sequence of syntactic categories, one for each word.
	$\phi(\mathbf{X},\mathbf{Y})$	Indexed by w,i,s; 1 if the i^{th} occurrence of word w in the corpus is tagged with syntactic category s, and 0 otherwise.
	OPT	Projected gradient.

Table 3: Summary of input and output variable meanings as well as meanings of constraint features and optimization methods used (OPT) for the applications summarized in Table 2.

is known as a garbage collector effect: the likelihood function of the simplistic translation models used prefers to align sections that are not literal translations to rare words, rather than leaving them unaligned (Brown et al., 1993). This results in each rare word in a source language being aligned to 4 or 5 words in the target language. To alleviate this problem, we introduce constraint features that count how many target words are aligned to each source word, and use PR to encourage models where this number is small in expectation. Modifying the model itself to include such a preference would break independence and make it intractable.

The multi-view learning application described in Section 6 leverages two or more sources of input ("views") along with unlabeled data. The requirement is to train two models, one for each view, such that they usually agree on the labeling of the unlabeled data. We can do this using PR, and we recover the Bhattacharyya distance as a regularizer. The PR approach also extends naturally to structured problems, and cases where we only want partial agreement.

The grammar induction application of Section 7 takes advantage of an existing parser for a resource-rich language to train a comparable resource for a resource-poor language. Because the two languages have different syntactic structures, and errors in word alignment abound, using such out-of-language information requires the ability to train with missing and noisy labeling. This is

achieved using PR constraints that guide learning to prefer models that tend to agree with the noisy labeling wherever it is provided, while standard regularization guides learning to be self-consistent.

Finally, Section 8 describes an application of PR to ensure a particular sparsity structure, which can be independent of the structure of the model. Section 8 focuses on the problem of unsupervised part-of-speech induction, where we are given a sample of text and are required to specify a syntactic category for each token in the text. A well-known but difficult to capture piece of prior knowledge for this problem is that each word type should only occur with a small number of syntactic categories, even though there are some syntactic categories that occur with many different word types. By using an ℓ_1/ℓ_∞ norm on constraint features we are able to encourage the model to have precisely this kind of sparsity structure, and greatly increase agreement with human-generated syntactic categories.

Table 2 also shows for each application whether we use the distribution over hidden variables given by the model parameters $p_\theta(\mathbf{Y}|\mathbf{X})$ to decode, or whether we first project the distribution to the constraint set and use $q(\mathbf{Y})$ to decode. In general we found that when applying the constraints on the labeled data is sensible, performing the projection before decoding tends to improve performance. For the word alignment application and the multi-view learning application we found decoding with the projected distribution improved performance. By contrast, for dependency parsing, we do not have the English translations at test time and so we cannot perform a projection. For part-of-speech induction the constraints are over the entire corpus, and different regularization strengths might be needed for the training and test sets. Since we did not want to tune a second hyperparameter, we instead decoded with p.

4. Related Frameworks

The work related to learning with constraints on posterior distributions is described in chronological order in the following three subsections. An overall summary is most easily understood in reverse chronological order though, so we begin with a few sentences detailing the connections to it in that order. Liang et al. (2009) describe how we can view constraints on posterior distributions as measurements in a Bayesian setting, and note that inference using such information is intractable. By approximating this problem, we recover either the generalized expectation constraints framework of Mann and McCallum (2007), or with a further approximation we recover a special case of the posterior regularization framework presented in Section 2. Finally, a different approximation recovers the constraint driven learning framework of Chang et al. (2007). To the best of our knowledge, we are the first to describe all these connections.

4.1 Constraint Driven Learning

Chang et al. (2007, 2008) describe a framework called constraint driven learning (CODL) that can be viewed as an approximation to optimizing the slack-penalized version of the PR objective (Equation 3). Chang et al. (2007) are motivated by hard-EM, where the distribution q is approximated by a single sample at the mode of $\log p_\theta(\mathbf{Y}|\mathbf{X})$. Chang et al. (2007) propose to augment $\log p_\theta(\mathbf{Y}|\mathbf{X})$ by adding to it a penalty term based on some domain knowledge. When the penalty terms are well-behaved, we can view them as adding a cost for violating expectations of constraint features ϕ. In such a case, CODL can be viewed as a "hard" approximation to the PR objective:

$$\arg\max_\theta \ \mathcal{L}(\theta) \quad -\min_{q \in M} \Big(\mathbf{KL}(q(\mathbf{Y})||p_\theta(\mathbf{Y}|\mathbf{X})) \ + \ \sigma \big|\big|\mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b}\big|\big|_\beta \Big)$$

where M is the set of distributions concentrated on a single $\mathbf{Y}$. The modified E-Step becomes:

$$\text{CODL } \mathbf{E}'\text{-step} : \max_{\mathbf{Y}} \log p_\theta(\mathbf{Y}|\mathbf{X}) \; - \; \sigma ||\phi(\mathbf{X},\mathbf{Y}) - \mathbf{b}||_\beta .$$

Because the constraints used by Chang et al. (2007) do not allow tractable inference, they use a beam search procedure to optimize the min-KL problem. Additionally they consider a K-best variant where instead of restricting themselves to a single point estimate for q, they use a uniform distribution over the top K samples.

Carlson et al. (2010) train several named entity and relation extractors concurrently in order to satisfy type constraints and mutual exclusion constraints. Their algorithm is related to CODL in that hard assignments are made in a way that guarantees the constraints are satisfied. However, their algorithm is motivated by adding these constraints to algorithms that learn pattern extractors: at each iteration, they make assignments only to the highest confidence entities, which are then used to extract high confidence patterns for use in subsequent iterations. By contrast hard EM and CODL would make assignments to every instance and change these assignments over time. Daumé III (2008) also use constraints to filter out examples for self-training and also do not change the labels.

4.2 Generalized Expectation Criteria

Generalized expectation criteria (GE) allow a user to specify preferences about model expectations in the form of linear constraints on some feature expectations (Mann and McCallum, 2007, 2008). As with PR, a set of constraint features ϕ are introduced, and a penalty term is added to the log-likelihood objective. The GE objective is

$$\max_{\theta} \; \mathcal{L}(\theta) - \sigma ||\mathbf{E}_{p_\theta}[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b}||_\beta . \tag{15}$$

where $||\cdot||_\beta$ is typically the l_2 norm (Druck et al., 2009 use l_2^2) or a distance based on KL divergence (Mann and McCallum, 2008), and the model is a log-linear model such as maximum entropy or a CRF.

The idea leading to this objective is the following: Suppose that we only had enough resources to make a very small number of measurements when collecting statistics about the true distribution $p^*(\mathbf{y}|\mathbf{x})$. If we try to create a maximum entropy model using these statistics we will end up with a very impoverished model. It will only use a small number of features and consequently will fail to generalize to instances where these features cannot occur. In order to train a more feature-rich model, GE defines a wider set of *model* features $\mathbf{f}$ and uses the small number of estimates based on constraint features ϕ to guide learning. By using l_2 regularization on model parameters, we can ensure that a richer set of model features are used to explain the desired expectations.

Druck et al. (2009) use a gradient ascent method to optimize the objective. Unfortunately, because the second term in the GE objective (Equation 15) couples the constraint features ϕ and the model parameters θ, the gradient requires computing the covariance between model features $\mathbf{f}$ and the constraint features ϕ under p_θ:

$$\frac{\partial \mathbf{E}_{p_\theta}[\phi(\mathbf{X},\mathbf{Y})]}{\partial \theta} = \mathbf{E}_{p_\theta}[\mathbf{f}(\mathbf{X},\mathbf{Y})\phi(\mathbf{X},\mathbf{Y})] - \mathbf{E}_{p_\theta}[\phi(\mathbf{X},\mathbf{Y})]\mathbf{E}_{p_\theta}[\mathbf{f}(\mathbf{X},\mathbf{Y})].$$

Because of this coupling, the complexity of the dynamic program needed to compute the gradient is higher than the complexity of the dynamic program for the model. In the case of graphical models

where $\mathbf{f}$ and ϕ have the same Markov dependencies, computing this gradient usually squares the running time of the dynamic program. A more efficient dynamic program might be possible (Li and Eisner, 2009; Pauls et al., 2009), however current implementations are prohibitively slower than PR when there are many constraint features.

In order to avoid the costly optimization procedure described above, Bellare et al. (2009) propose a variational approximation. Recall that at a high level, the difficulty in optimizing Equation 15 is because the last term couples the constraint features ϕ with the model parameters θ. In order to separate out these quantities, Bellare et al. (2009) introduce an auxiliary distribution $q(\mathbf{Y}) \approx p_\theta(\mathbf{Y}|\mathbf{X})$, which is used to approximate the last term in Equation 15. The variational objective contains three terms instead of two:

$$\arg\max_{\theta} \ \mathcal{L}(\theta) \quad - \min_{q} \left(\mathbf{KL}(q(\mathbf{Y})||p_\theta(\mathbf{Y}|\mathbf{X})) \ + \ \sigma \left|\left| \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} \right|\right|_\beta \right). \tag{16}$$

This formulation is identical to the slack-penalized version of PR, and Bellare et al. (2009) use the same optimization procedure (described in Section 2). Because both the minorization and the maximization steps implement minimum Kullback-Leibler projections, Bellare et al. (2009) refer to this algorithm as alternating projections. Note that PR can also be trained in an online fashion, and Ganchev et al. (2009) use an online optimization for this objective to train a dependency parser. These experiments are described in Section 7.

Closely related to GE, is the work of Quadrianto et al. (2009). The authors describe a setting where the constraint values, $\mathbf{b}$, are chosen as the empirical estimates on some labeled data. They then train a model to have high likelihood on the labeled data, but also match the constraint features on unlabeled data. They show that for appropriately chosen constraint features, the estimated constraint values should be close to the true means, and show good experimental improvements on an image retrieval task.

4.3 Measurements in a Bayesian Framework

Liang et al. (2009) approach the problem of incorporating prior information about model posteriors from a Bayesian point of view. They motivate their approach using the following caricature. Suppose we have log-linear model $p_\theta(\mathbf{y}|\mathbf{x}) \propto \exp(\theta \cdot \mathbf{f}(\mathbf{y},\mathbf{x}))$. In addition to any labeled data $(\mathbf{X}_L, \mathbf{Y}_L)$, we also have performed some additional experiments.[7] In particular, we have observed the expected values of some constraint features $\phi(\mathbf{X},\mathbf{Y})$ on some unlabeled data $\mathbf{X}$. Because there is error in measurement, they observe $\mathbf{b} \approx \phi(\mathbf{X},\mathbf{Y})$. Figure 3 illustrates this setting. The leftmost nodes represent $(\mathbf{x},\mathbf{y})$ pairs from the labeled data $(\mathbf{X}_L, \mathbf{Y}_L)$. The nodes directly to the right of θ represent unlabeled $(\mathbf{x},\mathbf{y})$ pairs from the unlabeled data $\mathbf{X}$. All the data are tied together by the dependence on the model parameters θ. The constraint features take as input the unlabeled data set $\mathbf{X}$ as well as a full labeling $\mathbf{Y}$, and produce some value $\phi(\mathbf{X},\mathbf{Y})$, which is never observed directly. Instead, we observe some noisy version $\mathbf{b} \approx \phi(\mathbf{X},\mathbf{Y})$. The measured values $\mathbf{b}$ are distributed according to some noise model $p_N(\mathbf{b}|\phi(\mathbf{X},\mathbf{Y}))$. Liang et al. (2009) note that the optimization is convex for log-concave noise and use box noise in their experiments, giving $\mathbf{b}$ uniform probability in some range near $\phi(\mathbf{X},\mathbf{Y})$.

In the Bayesian setting, the model parameters θ as well as the observed measurement values $\mathbf{b}$ are random variables. Liang et al. (2009) use the mode of $p(\theta|\mathbf{X}_L, \mathbf{Y}_L, \mathbf{X}, \mathbf{b})$ as a point estimate

7. In their exposition, Liang et al. (2009) incorporate labeled data by including the labels among experiments. We prefer to separate these types of observations because full label observations do not require approximations.

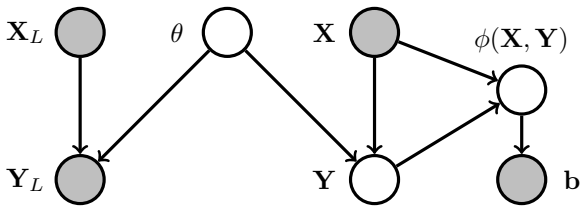


Figure 3: The model used by Liang et al. (2009), using our notation. We have separated treatment of the labeled data $(\mathbf{X}_L, \mathbf{Y}_L)$ from treatment of the unlabeled data $\mathbf{X}$.

for θ:

$$\arg\max_{\theta} \; p(\theta|\mathbf{X}_L, \mathbf{Y}_L, \mathbf{X}, \mathbf{b}) \quad = \quad \arg\max_{\theta} \; \sum_{\mathbf{Y}} p(\theta, \mathbf{Y}, \mathbf{b}|\mathbf{X}, \mathbf{X}_L, \mathbf{Y}_L),$$

with equality because $p(\theta|\mathbf{X}_L, \mathbf{Y}_L, \mathbf{X}, \mathbf{b}) \propto p(\theta, \mathbf{b}|\mathbf{X}_L, \mathbf{Y}_L, \mathbf{X}) = \sum_{\mathbf{Y}} p(\theta, \mathbf{Y}, \mathbf{b}|\mathbf{X}, \mathbf{X}_L, \mathbf{Y}_L)$. Liang et al. (2009) focus on computing $p(\theta, \mathbf{Y}, \mathbf{b}|\mathbf{X}, \mathbf{X}_L, \mathbf{Y}_L)$. They define their model for this quantity as follows:

$$p(\theta, \mathbf{Y}, \mathbf{b}|\mathbf{X}, \mathbf{X}_L, \mathbf{Y}_L) \quad = \quad p(\theta|\mathbf{X}_L, \mathbf{Y}_L) \;\; p_\theta(\mathbf{Y}|\mathbf{X}) \;\; p_N(\mathbf{b}|\phi(\mathbf{X}, \mathbf{Y})) \tag{17}$$

where the $\mathbf{Y}$ and $\mathbf{X}$ are particular instantiations of the random variables in the entire unlabeled corpus $\mathbf{X}$. Equation 17 is a product of three terms: a prior on θ, the model probability $p_\theta(\mathbf{Y}|\mathbf{X})$, and a noise model $p_N(\mathbf{b}|\phi)$. The noise model is the probability that we observe a value, $\mathbf{b}$, of the measurement features ϕ, given that its actual value was $\phi(\mathbf{X}, \mathbf{Y})$. The idea is that we model errors in the estimation of the posterior probabilities as noise in the measurement process. Liang et al. (2009) use a uniform distribution over $\phi(\mathbf{X}, \mathbf{Y}) \pm \varepsilon$, which they call "box noise". Under this model, observing $\mathbf{b}$ farther than ε from $\phi(\mathbf{X}, \mathbf{Y})$ has zero probability. In log space, the exact MAP objective, becomes:

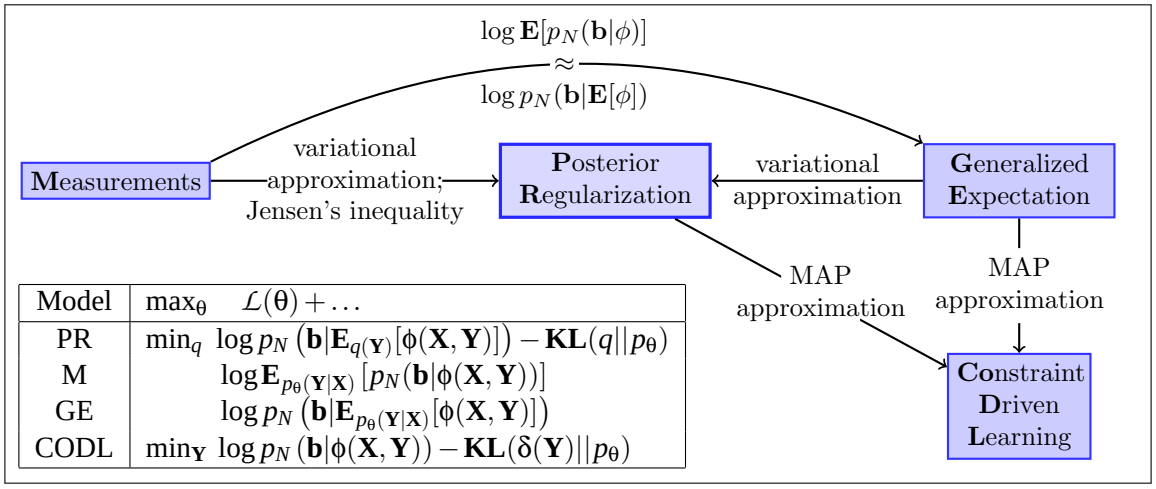
$$\max_{\theta} \quad \mathcal{L}(\theta) \quad + \log \mathbf{E}_{p_\theta(\mathbf{Y}|\mathbf{X})} \left[p_N(\mathbf{b}|\phi(\mathbf{X}, \mathbf{Y})) \right]. \tag{18}$$

Unfortunately with almost all noise models (including no noise), and box noise in particular, the second term in Equation 18 makes the optimization problem intractable.[8] Liang et al. (2009) use a variational approximation as well as a further approximation based on Jensen's inequality to reach the PR objective, which they ultimately optimize for their experiments. We also relate their framework to GE and CODL. If we approximate the last term in Equation 18 by moving the expectation inside the probability:

$$\mathbf{E}_{p_\theta(\mathbf{Y}|\mathbf{X})} \left[p_N(\mathbf{b} \mid \phi(\mathbf{X}, \mathbf{Y})) \right] \approx p_N \left(\mathbf{b} \mid \mathbf{E}_{p_\theta(\mathbf{Y}|\mathbf{X})} [\phi(\mathbf{X}, \mathbf{Y})] \right),$$

we end up with an objective equivalent to GE for appropriate noise models. In particular Gaussian noise corresponds to l_2^2 regularization in GE, since the log of a Gaussian is squared Euclidean distance (up to scaling). This approximation can be motivated by the case when $p_\theta(\mathbf{Y}|\mathbf{X})$ is concentrated on a particular labeling $\mathbf{Y}^*$: $p_\theta(\mathbf{Y}|\mathbf{X}) = \delta(\mathbf{Y}^*)$. In this special case the $\approx$ is an equality.

8. For very special noise, such as noise that completely obscures the signal, we can compute the second term in Equation 18.



Model	$\max_\theta \quad \mathcal{L}(\theta) + \ldots$
PR	$\min_q \ \log p_N\left(\mathbf{b}\middle\vert\mathbf{E}_{q(\mathbf{Y})}[\phi(\mathbf{X},\mathbf{Y})]\right) - \mathbf{KL}(q\vert\vert p_\theta)$
M	$\log \mathbf{E}_{p_\theta(\mathbf{Y}\vert\mathbf{X})}\left[p_N(\mathbf{b}\vert\phi(\mathbf{X},\mathbf{Y}))\right]$
GE	$\log p_N\left(\mathbf{b}\middle\vert\mathbf{E}_{p_\theta(\mathbf{Y}\vert\mathbf{X})}[\phi(\mathbf{X},\mathbf{Y})]\right)$
CODL	$\min_\mathbf{Y} \ \log p_N\left(\mathbf{b}\vert\phi(\mathbf{X},\mathbf{Y})\right) - \mathbf{KL}(\delta(\mathbf{Y})\vert\vert p_\theta)$

Figure 4: A summary of the different models. We use $p_\theta(\mathbf{Y}|\mathbf{X})$ to denote the model probability, $q(\mathbf{Y})$ to denote a proposal distribution, and p_N for the noise model. The symbol $\delta(\mathbf{Y})$ denotes a distribution concentrated on $\mathbf{Y}$. The approximations are described in the text: M→GE near Equation 19, GE→PR near Equation 16, PR→CODL at the end of Section 4.3.

This approximation is also used in Liang et al. (2009). This provides an interpretation of GE as an approximation to the Bayesian framework proposed by Liang et al. (2009):

$$\max_\theta \quad \mathcal{L}(\theta) \quad + \log p_N\left(\mathbf{b}\middle|\mathbf{E}_{p_\theta(\mathbf{Y}|\mathbf{X})}[\phi(\mathbf{X},\mathbf{Y})]\right). \tag{19}$$

Note that the objective in Equation 19 is a reasonable objective in and of itself, essentially stating that the measured values $\mathbf{b}$ are not dependent on any particular instantiation of the hidden variables, but rather represent the integral over all their possible assignments. Liang et al. (2009) also use a variational approximation similar to the one of Bellare et al. (2009) so that the objective they optimize is exactly the PR objective, although their optimization algorithm is slightly different from the one presented in Section 2. Finally, if we restrict the set of allowable distributions further to be concentrated on a single labeling $\mathbf{Y}$, we recover the CODL algorithm. Figure 4 summarizes the relationships.

5. Statistical Word Alignments

Word alignments, introduced by Brown et al. (1994) as hidden variables in probabilistic models for statistical machine translation (IBM models 1-5), describe the correspondence between words in source and target sentences. We will denote each target sentence as $\mathbf{x}^t = (x_1^t, \ldots, x_i^t, \ldots, x_I^t)$ and each source sentence as $\mathbf{x}^s = (x_1^s, \ldots, x_j^s, \ldots, x_J^s)$. A word alignment will be represented as a matrix with entries y_{ij} indicating that target word i is a translation of source word j. Although the original IBM models are no longer competitive for machine translation, the resulting word alignments are still a valuable resource. Word alignments are used primarily for extracting minimal translation units for machine translation, for example, phrases in phrase-based translation systems (Koehn et al., 2003)

and rules in syntax-based machine translation (Galley et al., 2004; Chiang et al., 2005), as well as for MT system combination (Matusov et al., 2006). But their importance has grown far beyond machine translation: for instance, transferring annotations between languages by projecting POS taggers, NP chunkers and parsers through word alignment (Yarowsky and Ngai, 2001; Rogati et al., 2003; Hwa et al., 2005; Ganchev et al., 2009); discovery of paraphrases (Bannard and Callison-Burch, 2005; Callison-Burch, 2007, 2008); and joint unsupervised POS and parser induction across languages (Snyder and Barzilay, 2008; Snyder et al., 2009).

Here we describe two types of prior knowledge that when introduced as constraints in different word alignment models significantly boost their performance. The two constraints are: (i) bijectivity: "one word should not translate to many words"; and (ii) symmetry: "directional alignments of one model should agree with those of another model". A more extensive description of these constraints applied to the task of word alignments and the quality of the resulting alignments can be found in Graça et al. (2010).

5.1 Models

We consider two models below: IBM Model 1 proposed by Brown et al. (1994) and the HMM model proposed by Vogel et al. (1996). Both models can be expressed as:

$$p(\mathbf{x}^t, \mathbf{y} \mid \mathbf{x}^s) = \prod_j p_d(y_j \mid j, y_{j-1}) p_t(\mathbf{x}^t_j \mid \mathbf{x}^s_{y_j}),$$

where $\mathbf{y}$ is the alignment and y_j is the index of the hidden state (source language index) generating the target language word at index j. The models differ in their definition of the distortion probability $p_d(y_j \mid j, y_{j-1})$. Model 1 assumes that the target words are generated independently and assigns uniform distortion probability. The HMM model assumes that only the distance between the current and previous source word index is important $p_d(y_j \mid j, y_{j-1}) = p_d(y_j \mid y_j - y_{j-1})$. Both models are augmented by adding a special "null" word to the source sentence.

The likelihood of the corpus, marginalized over possible alignments, is concave for Model 1, but not for the HMM model (Brown et al., 1994; Vogel et al., 1996). For both models though, standard training using the Expectation Maximization algorithm (Dempster et al., 1977) seeks model parameters θ that maximize the log-likelihood of the parallel corpus.

On the positive side, both models are simple and complexity of inference is $O(I \times J)$ for IBM Model 1 and $O(I \times J^2)$ for the HMM. However there are several problems with the models that arise from their directionality.

- **Non-bijective:** Multiple target words can align to a single source word with no penalty.
- **Asymmetric:** Swapping the source and target languages can produce very different alignments, since only constraints and correlations between consecutive positions on one side are enforced by the models.

The top row of Figure 5 shows an example of the posterior distribution for the alignment between an English and a French sentence using the HMM model. The left figure shows the alignment in the English to French direction where the rows are source words and columns are target words, while the right figure shows the alignment posteriors of the opposite direction. The first observation we make is that the posteriors are concentrated around particular source words (rare words

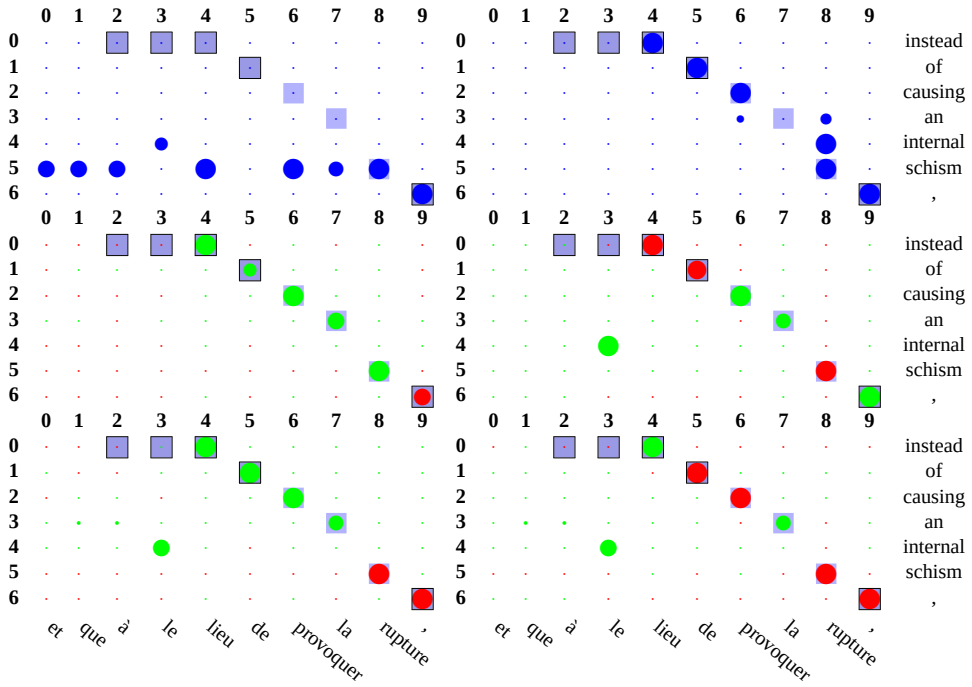


Figure 5: Posterior distributions on an English to French sentence using the HMM model. **Left:** EN→FR model. **Right:** FR→ EN model. **Top:** Regular EM posteriors. **Middle:** After applying bijective constraint. **Bottom:** After applying symmetric constraint. *Sure* alignments are squares with borders; *possible* alignments are squares without borders. Circle size indicates probability value. See Graça et al. (2010) for a description of the difference between sure and possible alignments. Circle color in the middle and bottom rows indicates difference in posterior from the top row. Green (light gray) - higher probability, red (dark gray) - lower probability.

occurring less than 5 times in the corpus) in both directions, instead of being spread across different words. This is a well known problem when training using EM, called the "garbage collector effect" (Brown et al., 1993). That is, rare words in the source language end up aligned to too many words in the target language because the generative model has to distribute translation probability for each source word among all candidate target words. Since the rare source word occurs in only a few sentences it needs to spread its probability mass over fewer competing target words. In this case, choosing to align the rare word to all of these target words leads to higher likelihood than correctly aligning them or aligning them to the special *null* word, since it increases the likelihood of this sentence without lowering the likelihood of many other sentences. Moreover, by correcting the garbage collector effect we increase the overall performance on the common words, since now these common words can be aligned to the correct words. For this particular corpus, 6.5% of the English tokens and 7.7% of the French tokens are rare.

5.2 Bijectivity Constraints

Bijectivity constraints are based on the observation that in most gold alignments, words are aligned one-to-one (98% for the sure alignments in the Hansard corpus). We would like to introduce this

trend into the model, but adding it directly requires large factors (breaking the Markov property). In fact, summing over one-to-one or near one-to-one weighted matchings is a classical #P-Complete problem (Valiant, 1979). However, introducing alignment degree constraints *in expectation* in the PR framework is easy and tractable. We simply add inequality constraints $\mathbf{E}[\phi(\mathbf{x},\mathbf{y})] \leq 1$ where we have one feature for each source word j that counts how many times it is aligned to a target word in the alignment $\mathbf{y}$:

$$\textbf{Bijective Features}: \quad \phi_j(\mathbf{x},\mathbf{y}) = \sum_i \mathbf{1}(y_i = j).$$

For example, in the alignment at the top right of Figure 5, the posteriors over the source word *schism* clearly sum to more than 1. The effect of applying PR constraints to the posteriors is shown in the second row. Enforcing the one to (at most) one constraint clearly alleviates the garbage collector effect. Moreover, when distributing the probability mass to the other words, most of the probability mass goes into the correct positions (as can be seen by comparison to the gold alignments). Note that the bijectivity constraints only hold approximately in practice and so we expect that having a KL-based penalty for them might be better than ensuring that the model satisfies them, since we can violate bijectivity in order to achieve higher likelihood. Another way to understand what is going on is to see how the parameters are affected by the bijectivity constraint. In particular the translation table becomes much cleaner, having a much lower entropy. The average entropy for the translation probability, averaged over all source language words for EM training is 2.0-2.6 bits, depending on the direction. When we train using PR with the bijectivity constraint, this entropy drops to 0.6 for both directions. We see that while the model does not have any particular parameter that can enforce bijectivity, in practice the model is expressive enough to learn a preference for bijectivity by appropriately setting the existing model parameters.

5.3 Symmetry Constraints

Word alignment should not depend on translation direction, but this principle is clearly violated by the directional models. In fact, each directional model makes different mistakes. The standard approach is to train two models independently and then intersect their predictions (Och and Ney, 2003).[9] However, we show that it is much better to train two directional models concurrently, coupling their posterior distributions over alignments with constraints that force them to approximately agree. The idea of training jointly has also been explored by Matusov et al. (2004) and Liang et al. (2006), although their formalization is quite different.

Let the directional models be defined as: $\overrightarrow{p}_\theta(\overrightarrow{\mathbf{y}})$ (forward) and $\overleftarrow{p}_\theta(\overleftarrow{\mathbf{y}})$ (backward). We suppress dependence on $\mathbf{x}^s$ and $\mathbf{x}^t$ for brevity. Define $\mathbf{y}$ to range over the union of all possible directional alignments $\overrightarrow{\mathbf{y}} \cup \overleftarrow{\mathbf{y}}$. We then define a mixture model $p_\theta(\mathbf{y}) = \frac{1}{2}\overrightarrow{p}_\theta(\mathbf{y}) + \frac{1}{2}\overleftarrow{p}_\theta(\mathbf{y})$ where $\overleftarrow{p}_\theta(\overrightarrow{\mathbf{y}}) = 0$ and vice-versa (i.e., the alignment of one directional model has probability zero according to the other model). We then define the following feature for each target-source position pair i, j:

$$\textbf{Symmetric Features}: \quad \phi_{ij}(\mathbf{x},\mathbf{y}) = \begin{cases} +1 & \mathbf{y} \in \overrightarrow{\mathbf{y}} \text{ and } \overrightarrow{y_i} = j \\ -1 & \mathbf{y} \in \overleftarrow{\mathbf{y}} \text{ and } \overleftarrow{y_j} = i \\ 0 & \text{otherwise.} \end{cases}$$

The feature takes the value zero in expectation if a word pair i, j is aligned with equal probability in both directions. So the constraint we want to impose is $\mathbf{E}_q[\phi_{ij}(\mathbf{x},\mathbf{y})] = 0$ (possibly with some small

9. Sometimes union or a heuristic is used instead of intersection.

violation). Note that this constraint is only feasible if the posteriors are bijective. Clearly these features are fully factored, so to compute expectations of these features under the model q we only need to be able to compute them under each directional model, as we show below. To see this, we have by the definition of q_λ and p_θ,

$$q_\lambda(\mathbf{y} \mid \mathbf{x}) = \frac{\overrightarrow{p}_\theta(\mathbf{y} \mid \mathbf{x}) + \overleftarrow{p}_\theta(\mathbf{y} \mid \mathbf{x})}{2} \frac{\exp\{-\lambda \cdot \phi(\mathbf{x}, \mathbf{y})\}}{Z} = \frac{\overrightarrow{q}(\mathbf{y} \mid \mathbf{x}) \frac{Z_{\overrightarrow{q}}}{\overrightarrow{p}_\theta(\mathbf{x})} + \overleftarrow{q}(\mathbf{y} \mid \mathbf{x}) \frac{Z_{\overleftarrow{q}}}{\overleftarrow{p}_\theta(\mathbf{x})}}{2Z},$$

where we have defined:

$$\begin{aligned}
\overrightarrow{q}(\mathbf{y} \mid \mathbf{x}) = &\quad \frac{1}{Z_{\overrightarrow{q}}} \overrightarrow{p}_\theta(\mathbf{y}, \mathbf{x}) \exp\{-\lambda \cdot \phi(\mathbf{x}, \mathbf{y})\} \quad \text{with} \quad Z_{\overrightarrow{q}} = \sum_{\mathbf{y}} \overrightarrow{p}_\theta(\mathbf{y}, \mathbf{x}) \exp\{-\lambda \cdot \phi(\mathbf{x}, \mathbf{y})\}, \\
\overleftarrow{q}(\mathbf{y} \mid \mathbf{x}) = &\quad \frac{1}{Z_{\overleftarrow{q}}} \overleftarrow{p}_\theta(\mathbf{y}, \mathbf{x}) \exp\{-\lambda \cdot \phi(\mathbf{x}, \mathbf{y})\} \quad \text{with} \quad Z_{\overleftarrow{q}} = \sum_{\mathbf{y}} \overleftarrow{p}_\theta(\mathbf{y}, \mathbf{x}) \exp\{-\lambda \cdot \phi(\mathbf{x}, \mathbf{y})\}, \\
Z = &\quad \frac{1}{2} \left(\frac{Z_{\overrightarrow{q}}}{\overrightarrow{p}_\theta(\mathbf{x})} + \frac{Z_{\overleftarrow{q}}}{\overleftarrow{p}_\theta(\mathbf{x})} \right).
\end{aligned}$$

All these quantities can be computed separately in each model.

The last row in Figure 5 shows both directional posteriors after imposing the symmetric constraint. Note that the projected posteriors are equal in the two models. Also, one can see that in most cases the probability mass was moved to the correct place with the exception of the word pair *internal/le*; this is because the word *internal* does not appear on the French side, but the model still has to spread around the probability mass for that word. In this case the model decided to accumulate it on the word *le* instead of moving it to the *null* word.

5.4 Algorithm for Projection

Because both the symmetric and bijective constraints decompose over sentences, and the model distribution $p(\mathbf{Y}|\mathbf{X})$ decomposes as a product distribution over instances, the projected distribution $q(\mathbf{Y})$ will also decompose as a product over instances. Furthermore, because the constraints for different instances do not interact, we can project the sentences one at a time and we do not need to store the posteriors for the whole corpus in memory at once. We compute $q(\mathbf{y})$ for each sentence pair $\mathbf{x}$ using projected gradient ascent for the bijective constraints and L-BFGS for the symmetric constraints.

5.5 Results

We evaluated the constraints using the Hansard corpus (Och and Ney, 2000) of English/French. Following prior work by Och and Ney (2003), we initialize the Model 1 translation table with uniform probabilities over word pairs that occur together in same sentence. The HMM is initialized with the translation probabilities from Model 1 and with uniform distortion probabilities. We train M1 for 5 iterations and train the HMM model until no further improvement on precision and recall is seen on standard (small) development set for this corpus. We note that when using regular EM training this requires around 4 iterations, while just 2 iterations suffices when using PR. This is likely due to the added information that the constraints provide. We use a 40 word maximum length cutoff for training sentences and train all models on $100,000$ sentences, testing precision and recall on the standard test set.

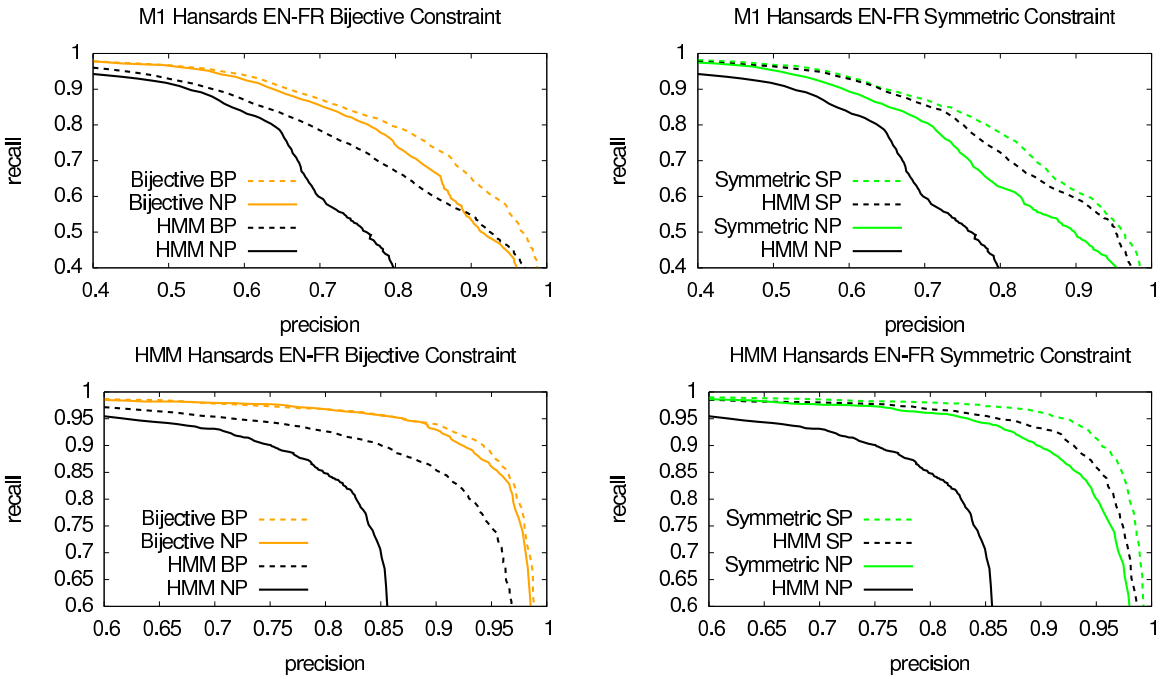


Figure 6: Precision vs Recall curves of both models using standard EM training (Regular) versus PR with bijective constraints (Bijective) and symmetry constraints (Symmetric) and different decoding types: decoding without any projection (NP), doing bijective projection before decoding (BP), and doing symmetric projection before decoding (SP). Data is 100k sentences of the Hansard corpus. Highest label in the legend corresponds to highest line in the graph, second highest label to second highest line, and so on.

Figure 6 shows the precision vs recall curves of both models (EN-FR, FR-EN independently) when training using standard EM versus PR with both constraints, and the results of additionally applying the constraints at decode time in order to tease apart the effect of the constraints during training vs. during testing. The first observation is that training with PR significantly boosts the performance of each model. Moreover using the projection at decode time always increases performance. Comparing both constraints, it seems that the bijective constraint is more useful at training time. Note that using this constraint at decode time with regular training yields worse results than just training with the same constraint using PR. On the other hand, the symmetric constraint is stronger at decode time.

A comprehensive comparison of the word alignment application is presented in Graça et al. (2010), where in six different languages the use of these constraints always significantly outperforms the simpler unconstrained HMM model. Moreover, in 9 out of 12 times using these constraints outperforms the more complex model IBM M4 (Brown et al., 1994). The alignments are also evaluated in the task of statistical machine translation where they are used as an initial step to extract parallel phrases used for translation. Using these constraints leads to better translation quality. Finally, the different alignments are tested on the task of transferring syntactic annotations, which

is described in the next section. The new alignments increase the number of correctly transferred edges.

6. Multi-view learning

Multi-view learning refers to a set of semi-supervised methods which exploit redundant views of the same input data (Blum and Mitchell, 1998; Collins and Singer, 1999; Brefeld et al., 2005; Sindhwani et al., 2005). These multiple views can come in the form of context and spelling features in the case of text processing and segmentation, hypertext link text and document contents for document classification, and multiple cameras or microphones in the case of speech and vision. Multi-view methods typically begin by assuming that each view alone can yield a good predictor. Under this assumption, we can regularize the models from each view by constraining the amount by which we permit them to disagree on unlabeled instances. This regularization can lead to better convergence by significantly decreasing the effective size of our hypothesis class (Balcan and Blum, 2005; Kakade and Foster, 2007; Rosenberg and Bartlett, 2007). This idea is related to the symmetry constraints described in Section 5.

In this section, we use PR to derive a multi-view learning algorithm. The idea is very simple: train a model for each view, and use constraints that the models should agree on the label distribution. Where our work is most similar to co-regularization schemes, a minimum Kullbeck-Leibler (KL) distance projection can be computed in closed form resulting in an algorithm that performs better than both CoBoosting and two view Perceptron on several natural language processing tasks. In this case, the resulting regularizer is identical to adding a penalty term based on the Bhattacharyya distance (Kailath, 1967) between models trained using different views.

In addition, this framework allows us to use different labeled training sets for the two classifiers, in the case where they have different label sets. That is, we don't require that our two views are both on the same labeled corpus. In that case, we can reduce the hypothesis space by preferring pairs of models that agree on *compatible* labeling of some additional unlabeled data rather than on *identical* labeling, while still minimizing KL in closed form. When the two views come from models that differ not only in the label set but also in the model structure of the output space, our framework can still encourage agreement, but the KL minimization cannot be computed in closed form. Finally, this method uses soft assignments to latent variables resulting in a more stable optimization procedure.

6.1 Stochastic Agreement

Note that the constraint in this section is similar to the one described in Section 5, but here we focus on discriminative learning and the formulation is slightly different. For notational convenience, we focus on two view learning in this exposition, however the generalization to more than two views is fairly straightforward. Also, we focus on two discriminative log-linear models and start by considering the setting of complete agreement. In this setting we have a common desired output for the two models and we believe that each of the two views is sufficiently rich to predict labels accurately. We can leverage this knowledge by restricting our search to model pairs p_1, p_2 that satisfy $p_1(\mathbf{y} \mid \mathbf{x}) \approx p_2(\mathbf{y} \mid \mathbf{x})$. Since p_1 and p_2 each define a distribution over labels, we will consider the product distribution $p_1(\mathbf{y}_1)p_2(\mathbf{y}_2)$ and define constraint features such that our proposal distribution $q(\mathbf{y}_1, \mathbf{y}_2)$ will have the same marginal for $\mathbf{y}_1$ and $\mathbf{y}_2$. In particular, we will have one constraint feature for each label $\mathbf{y}$:

$$\phi_{\mathbf{y}}(\mathbf{y}_1, \mathbf{y}_2) = \delta(\mathbf{y}_1 = \mathbf{y}) - \delta(\mathbf{y}_2 = \mathbf{y}).$$

Where $\delta(\cdot)$ is the 0-1 indicator function. The constraint set $Q = \{q : \mathbf{E}_q[\phi] = 0\}$ will require that the marginals over the two output variables are identical $q(\mathbf{y}_1) = q(\mathbf{y}_2)$. It will be useful in the sequel to define an agreement between two models agree(p_1, p_2) as

$$\text{agree}(p_1, p_2) = \arg\min_q \ \mathbf{KL}(q(\mathbf{y}_1, \mathbf{y}_2) || p_1(\mathbf{y}_1) p_2(\mathbf{y}_2)) \quad \text{s.t.} \quad \mathbf{E}_q[\phi] = 0. \tag{20}$$

Proposition 6.1 relates the Bhattacharyya regularization term to the value of the optimization problem in Equation 20. The Bhattacharyya distance is a very natural, symmetric measure of difference between distributions which has been used in many signal detection applications (Kailath, 1967). It is also related to the well-known Hellinger distance.

Proposition 6.1 *The Bhattacharyya distance* $-\log \sum_{\mathbf{y}} \sqrt{p_1(\mathbf{y}) p_2(\mathbf{y})}$ *is equal to* $\frac{1}{2}$ *of the value of the convex optimization problem*

$$\begin{array}{ll} \min\limits_{q \in Q} & \mathbf{KL}(q(\mathbf{y}_1, \mathbf{y}_2) || p_1(\mathbf{y}_1) p_2(\mathbf{y}_2)) \\ \text{where} & Q = \left\{ q : \mathbf{E}_q[\delta(\mathbf{y}_1 = \mathbf{y}) - \delta(\mathbf{y}_2 = \mathbf{y})] = 0 \ \forall y \right\}, \end{array} \tag{21}$$

and where $\delta(cond)$ *is* 1 *if cond is true and* 0 *otherwise. Furthermore, the minimizer decomposes as* $q(\mathbf{y}_1, \mathbf{y}_2) = q_1(\mathbf{y}_1) q_2(\mathbf{y}_2)$ *and is given by* $q_i(y) \propto \sqrt{p_1(\mathbf{y}) p_2(\mathbf{y})}$.

Proof Taking the dual of the optimization problem in Equation 21 we get

$$\arg\max_{\lambda} -\log \sum_{\mathbf{y}_1, \mathbf{y}_2} p(\mathbf{y}_1, \mathbf{y}_2) \exp(\lambda \cdot \phi)$$

with $q(\mathbf{y}_1, \mathbf{y}_2) \propto p(\mathbf{y}_1, \mathbf{y}_2) \exp(\lambda \cdot \phi(\mathbf{y}_1, \mathbf{y}_2))$. Where $\phi(\mathbf{y}_1, \mathbf{y}_2)$ is a vector of features of the form $\delta(\mathbf{y}_1 = \mathbf{y}) - \delta(\mathbf{y}_2 = \mathbf{y})$ with one entry for each possible label $\mathbf{y}$. Noting that the features decompose into $\phi'(\mathbf{y}_1) - \phi'(\mathbf{y}_2)$, we know that $q(\mathbf{y}_1, \mathbf{y}_2)$ decomposes as $q_1(\mathbf{y}_1) q_2(\mathbf{y}_2)$. Furthermore, our constraints require that $q_1(\mathbf{y}) = q_2(\mathbf{y}) \forall \mathbf{y}$ so we have $q(\mathbf{y}_1) q(\mathbf{y}_2) \propto p_1(\mathbf{y}_1) \exp(\lambda \cdot \phi'(\mathbf{y}_1)) p_2(\mathbf{y}_2) \exp(-\lambda \cdot \phi'(\mathbf{y}_2))$. Letting $\mathbf{y}_1 = \mathbf{y}_2$ we have $q(\mathbf{y})^2 = p_1(\mathbf{y}) p_2(\mathbf{y})$ which gives us a closed form computation of agree$(p_1, p_2) \propto \sqrt{p_1(\mathbf{y}) p_2(\mathbf{y})}$. Substituting this solution into the problem of Proposition 6.1, and performing algebraic simplification yields the desired result. ■

Replacing the minimum KL term in Equation 2 with a Bhattacharyya regularization term yields the objective

$$\min_{\theta} \mathcal{L}_1(\theta_1) + \mathcal{L}_2(\theta_2) + c \mathbf{E}_U[B(p_1(\theta_1), p_2(\theta_2))]$$

where $\mathcal{L}_i = \mathbf{E}[-\log(p_i(\mathbf{y}_i | \mathbf{x}; \theta_i))] + \frac{1}{\sigma_i^2} ||\theta_i||^2$ for $i = 1, 2$ are the standard regularized log likelihood losses of the models p_1 and p_2, $\mathbf{E}_U[B(p_1, p_2)]$ is the expected Bhattacharyya distance (Kailath, 1967) between the predictions of the two models on the unlabeled data, and c is a constant defining the relative weight of the unlabeled data relative to the labeled data.

Our regularizer extends to full agreement for undirected graphical models. In the case where p_1 and p_2 have the same structure, $q = \text{agree}(p_1, p_2)$ will share this structure and the projection can be computed in closed form.

Proposition 6.2 *Suppose $p_i(\mathbf{Y}|\mathbf{X}), i \in \{1,2\}$ factor as a set of clique potentials from a set of cliques $\mathcal{C}$:*

$$p_i(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z_i(\mathbf{X})} \prod_{c \in \mathcal{C}} \psi_i(\mathbf{X}, \mathbf{Y}_c),$$

then $q_i(\mathbf{Y})$ also factors as a product over clique potentials in $\mathcal{C}$, and can be computed in closed form modulo normalization as $q(\mathbf{Y}_1, \mathbf{Y}_2) = q_1(\mathbf{Y}_1) q_2(\mathbf{Y}_2)$ with

$$q_i(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z'(\mathbf{X})} \prod_{c \in \mathcal{C}} \sqrt{\psi_1(\mathbf{X}, \mathbf{Y}_c) \psi_2(\mathbf{X}, \mathbf{Y}_c)}.$$

Proof The proof is simple algebraic manipulation. We start with Equation 22, an application of Proposition 6.1.

$$\begin{aligned} q_i(\mathbf{Y})^2 &\propto p_1(\mathbf{Y}|\mathbf{X}) p_2(\mathbf{Y}|\mathbf{X}) \\ &= Z_1^{-1} Z_2^{-1} \prod_c \psi_1(\mathbf{X}, \mathbf{Y}_c) \psi_2(\mathbf{X}, \mathbf{Y}_c) \\ &= \left(\frac{1}{Z'(\mathbf{X})} \prod_c \sqrt{\psi_1(\mathbf{X}, \mathbf{Y}_c) \psi_2(\mathbf{X}, \mathbf{Y}_c)} \right)^2. \end{aligned} \tag{22}$$

■

Note that Proposition 6.2 is not a special case of Proposition 2.2 because we have defined one constraint feature $\phi_{\mathbf{y}}$ for each possible labeling $\mathbf{y}$, and these do not decompose according to $\mathcal{C}$. We could alternatively have proven that ensuring agreement on clique potentials is identical to ensuring agreement on labelings. In the case of log-linear Markov random fields, the clique potentials are stored in log space so computing q corresponds to averaging the values before computing normalization.

6.2 Partial Agreement and Hierarchical Labels

Our method extends naturally to partial-agreement scenarios. For example we can encourage two part-of-speech taggers with different tag sets to produce compatible parts of speech, such as noun in tag set one and singular-noun in tag set 2, as opposed to noun in tag set 1 and verb in tag set 2. In particular, suppose we have a mapping from both label sets into a common space where it makes sense to encourage agreement. For the part-of-speech tagging example, this could mean mapping all nouns from both tag sets into a single class, all verbs into another class and so on. In general suppose we have functions $g_1(\mathbf{y}_1)$ and $g_2(\mathbf{y}_2)$ that map variables for the two models onto the same space $\{\mathbf{z}\}$. Then, $p_i(\mathbf{y}_i)$ and g_i induce a distribution:

$$p_i(\mathbf{z}) = \sum_{\mathbf{y} \,:\, g_i(\mathbf{y}) = \mathbf{z}} p_i(\mathbf{y}) \quad \text{and} \quad p_i(\mathbf{y}_i|\mathbf{z}) = p_i(\mathbf{y}_i)/p_i(\mathbf{z}).$$

We can encourage $p_1(\mathbf{z}) \approx p_2(\mathbf{z})$ by adding a feature for each label in the joint space:

$$\phi_{\mathbf{z}}(\mathbf{y}_i) = \begin{cases} 1 & \text{if } i = 1 \text{ and } g_1(\mathbf{y}_1) = \mathbf{z} \\ -1 & \text{if } i = 2 \text{ and } g_2(\mathbf{y}_2) = \mathbf{z} \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

In this case our objective becomes:

$$\min_{\theta} \mathcal{L}_1(\theta_1) + \mathcal{L}_2(\theta_2) + c\mathbf{E}_U[B(p_1(\mathbf{z}), p_2(\mathbf{z}))] \ .$$

In the special case where some labels are identical for the two models and others are incompatible, we have $g_1(\mathbf{z}_1)$ mapping the incompatible labels into one bin and the others into their own special bins. Proposition 6.3 along with the optimization algorithm described in Section 2.6 allows us to optimize this objective.

Proposition 6.3 *The Bhattacharyya distance* $-\log\sum_{\mathbf{z}}\sqrt{p_1(\mathbf{z})p_2(\mathbf{z})}$ *is* $\frac{1}{2}$ *the value of the convex optimization problem*

$$\begin{aligned} \min_{q} \quad & \mathbf{KL}(q(\mathbf{Y}_1,\mathbf{Y}_2)||p_1(\mathbf{Y}_1)p_2(\mathbf{Y}_2)) \\ \text{s.t.} \quad & \mathbf{E}_q(\phi) = 0, \end{aligned}$$

where the constraint features ϕ *are defined as in Equation 23. Furthermore, the minimizer decomposes as* $q(\mathbf{Y}_1,\mathbf{Y}_2) = q_1(\mathbf{Y}_1|\mathbf{z}_1)q_1(\mathbf{z}_1)q_2(\mathbf{Y}_2|\mathbf{z}_2)q_2(\mathbf{z}_2)$*, where* $q_1(\mathbf{z}_1) = q_2(\mathbf{z}_2) \propto \sqrt{p_1(\mathbf{z}_1)p_2(\mathbf{z}_2)}$ *and* $q_i(\mathbf{Y}_i|\mathbf{z}_i) = p_i(\mathbf{Y}_i|\mathbf{z}_i)$ $i \in \{1,2\}$.

Note that the computation of agree(p_1,p_2) is still in closed form if our models are unstructured.

Unfortunately, if we collapse some labels for structured models, $p(\mathbf{Y})$ might not have the same Markov properties as $p(\mathbf{z})$. For example, consider the case where p is a distribution over three states (1,2,3) that assigns probability 1 to the sequence (1,2,3,1,2,3,...) and probability zero to other sequences. This is a first-order Markov chain. If the mapping is $1 \mapsto 1$ and $2,3 \mapsto 0$ then $p(\mathbf{y})$ assigns probability 1 to (1,0,0,1,0,0,...), which cannot be represented as a first-order Markov chain. Essentially, the original chain relied on being able to distinguish between the allowable transition (2,3) and the disallowed transition (3,2). When we collapse the states, both of these transitions map to (0,0) and cannot be distinguished. Consequently, the closed form solution given in Proposition 6.3 is not usable. Potentially, we could compute some approximation to $p(\mathbf{y})$ and from that compute an approximation to q. Instead, we re-formulate our constraints to require only that the marginals of each clique in p_1 and p_2 match each other rather than requiring the joint to have the same probability:

$$\phi_{c,\mathbf{z}_c}(\mathbf{Y}_1,\mathbf{Y}_2) = \begin{cases} 1 & \text{if } i=1 \text{ and } g_1(\mathbf{y}_1)_c = \mathbf{z}_c \\ -1 & \text{if } i=2 \text{ and } g_2(\mathbf{y}_2)_c = \mathbf{z}_c \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

By Proposition 2.2, the features in Equation 24 lead to a q that respects the Markov properties of the original models.

6.3 Relation to Other Multi-View Learning

To avoid a long detour from PR, we describe here only CoBoosting (Collins and Singer, 1999) and two view Perceptron (Brefeld et al., 2005), the frameworks with which we empirically compare our method in the next section. Since these methods are based on different objective functions from ours it is worth examining where each one works best. Altun et al. (2003) compare log-loss and exp-loss for sequential problems. They find that the loss function does not have as great an effect on performance as the feature choice. However, they also note that exp-loss is expected to

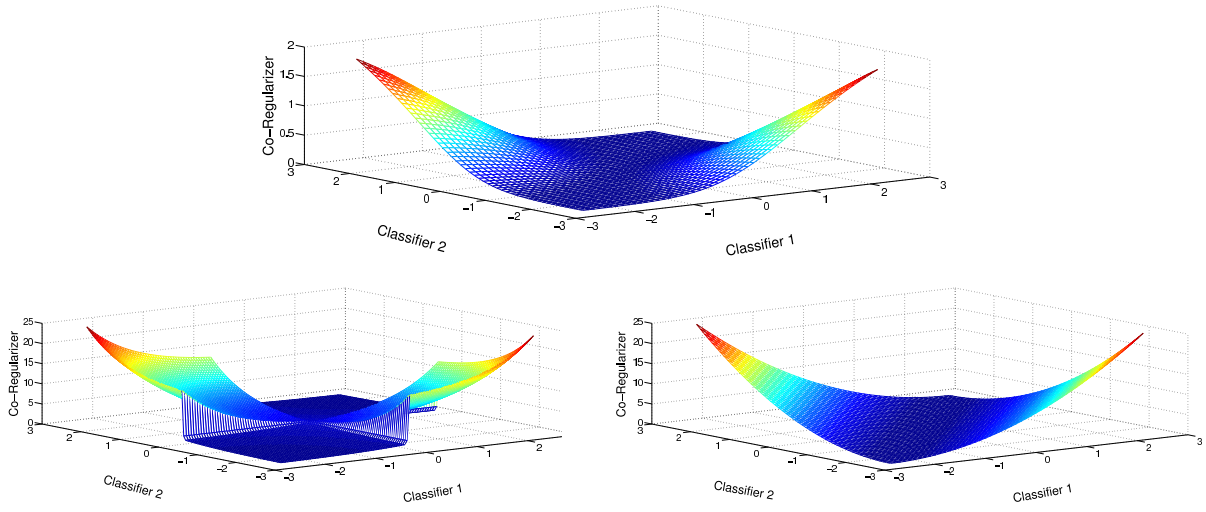


Figure 7: Different Loss Functions. **Top**: Bhattacharyya distance regularization. **Bottom left**: Exp-loss regularization. **Bottom right**: Least squares regularization.

perform better for clean data, while log-loss is expected to perform better when there is label noise. The intuition behind this is in the rate of growth of the loss functions. Exp-loss grows exponentially with misclassification margin while log-loss grows linearly. Consequently when there is label noise, AdaBoost focuses more on modeling the noise. Since CoBoosting optimizes a co-regularized exp-loss while our work optimizes a co-regularized log-loss we expect to do better on problems where the labels are noisy.

To get more intuition about this, Figure 7 shows the co-regularization loss functions for our method, CoBoosting, and co-regularized least squares (Sindhwani et al., 2005). For two underlying binary linear classifiers, $\hat{y}_1 = \text{sign}(w_1 \cdot x)$ and $\hat{y}_2 = \text{sign}(w_2 \cdot x)$, the horizontal axes represent the values $\hat{y}_1$ and $\hat{y}_2$, while the vertical axis is the loss. If we consider the plane parallel to the page, we see how the different co-regularizers penalize the classifiers when they disagree and are equally confident in their decision. Restricted to this plane, all three co-regularizers grow at the same asymptotic rate as the loss functions for the individual models: Linearly for our work, exponentially for CoBoosting and quadratically for co-RLS. If we look at the area where the two models agree (the flat part of the CoBoosting graph) we see what the penalty is when the classifiers agree but have different confidence. In this case co-RLS is harshest since it penalizes differences in the dot product equally regardless of the absolute value of the dot product. Intuitively, this is a problem. If one model predicts 1 with confidence 0.5 and the other predicts -1 with confidence 0.5 they are disagreeing while if they both predict 1 with confidence 1000 and 1001 respectively, they are agreeing on the label and are very close in their confidence estimates. At the other extreme, CoBoosting imposes almost no penalty whenever the two classifiers agree, regardless of their confidence. The Bhattacharyya distance co-regularizer lies between these extremes, penalizing differences in confidence near the origin but is more lenient when the classifiers are both very confident and agree.

Finally, if we have labeled data from one domain but want to apply it to another domain we can use any of the co-training frameworks mentioned earlier, including our own, to perform do-

Domains	MIRA	Boost	Perc	mx-ent	SCL	CoBoost	coPerc	PR
books→dvds	77.2	72.0	74	78.5	75.8	78.8	75.5	**79.8**
dvds→books	72.8	74.8	74.5	80.3	79.7	79.8	74.5	**81.3**
books→electr	70.8	70.3	73.3	72.5	75.9	**77.0**	69.3	75.5
electr→books	70.7	62.5	73	72.8	**75.4**	71.0	67.5	74.3
books→kitchn	74.5	76.3	73.5	77.8	78.9	78.0	76.5	**81.0**
kitchn→books	70.9	66.5	67.3	70.3	68.6	69.8	66	**72.8**
dvds→electr	73.0	73.2	73.5	75.5	74.1	75.3	71.2	**76.5**
electr→dvds	70.6	66.3	64.8	69.3	**76.2**	73.5	63.3	73.0
dvds→kitchn	74.0	75.5	78.3	80.5	81.4	79.0	78.25	**82.8**
kitchn→dvds	72.7	61.8	64	69.5	**76.9**	70.1	60.5	72.8
electr→kitchn	84.0	73.2	81	**86.5**	85.9	85.0	83.3	85.8
kitchn→electr	82.7	66.3	81	82.8	**86.8**	83.0	80.5	85.5
Average improvement	-1.87	-6.47	-3.18	N/A	1.61	0.33	-4.16	**2.07**
Standard deviation	3.04	4.73	2.21	N/A	3.31	2.03	2.12	**1.27**

Table 4: Performance of several methods on a sentiment classification transfer learning task. Reviews of objects of one type are used to train a classifier for reviews of objects of another type. The abbreviations in the column names are as follows. Boost: AdaBoost algorithm, Perc: Perceptron, mx-ent: maximum entropy, SCL: structural correspondence learning, CoBoost: CoBoosting, coPerc: two view Perceptron, PR: this work. The best accuracy is shown in bold for each task. The last two rows of the table show the average improvement over maximum entropy (the best performing supervised method), and also the standard deviation of the improvement.

main transfer. For sentiment classification we will see that our method performs comparably with Structural Correspondence Learning (Blitzer et al., 2006), which is based on Alternating Structure Optimization (Ando and Zhang, 2005).

6.4 Experiments

Our first set of experiments is for transfer learning for sentiment classification. We use the data from Blitzer et al. (2007). The two views are generated from a random split of the features. We compare our method to several supervised methods as well as CoBoosting (Collins and Singer, 1999), two view Perceptron (Brefeld et al., 2005) and structural correspondence learning (Blitzer et al., 2007). Results are in Table 4. The column labeled "SCL" contains the best results from Blitzer et al. (2007), and is not directly comparable with the other methods since it uses some extra knowledge about the transfer task to choose auxiliary problems. For all the two-view methods we weigh the total labeled data equally with the total unlabeled data. We regularize the maximum entropy classifiers with a unit variance Gaussian prior. Out of the 12 transfer learning tasks, our method performs best in 6 cases, SCL in 4, while CoBoosting performs best only once. Two view Perceptron never outperforms all other methods. One important reason for the success of our method is the relative strength of the maximum entropy classifier relative to the other supervised methods for this

particular task. We expect that CoBoosting will perform better than our method in situations where Boosting significantly out-performs maximum entropy.

The next set of our experiments are on named entity disambiguation. Given a set of already segmented named entities, we want to predict what type of named entity each one is. We use the training data from the 2003 CoNLL shared task (Sang and Meulder, 2003). The two views comprise content versus context features. The content features are words, POS tags and character n-grams of length 3 for all tokens in the named entity, while context features the same but for three words before and after the named entity. We used 2000 examples as test data and roughly 30,000 as unlabeled (train) data. Table 5 shows the results for different amounts of labeled train data. For this data, we choose the variance of the Gaussian prior as well as the relative weighting of the labeled and unlabeled data by cross validation on the train set. In order to test whether the advantage our method gets is from the joint objective or from the use of agree(p_1, p_2), which is an instance of logarithmic opinion pools, we also report the performance of using agree(p_1, p_2) when the two views p_1 and p_2 have been trained only on the labeled data. In the column labeled "agree$_0$" we see that for this data set the benefit of our method comes from the joint objective function rather than from the use of logarithmic opinion pools.

Data size	mx-ent	agree$_0$	PR	RRE
500	74.0	74.4	76.4	9.2%
1000	80.0	80.0	81.7	8.5%
2000	83.4	83.4	84.8	8.4%

Table 5: Named entity disambiguation. Prior variance and c chosen by cross validation. agree$_0$ refers to performance of two view model before first iteration of EM. RRE is reduction in error relative to error of MaxEnt model.

In order to investigate the applicability of our method to structured learning we apply it to the shallow parsing task of noun phrase chunking. We our experiments are on the English training portion of the CoNLL 2000 shared task (Sang and Buchholz, 2000). We select 500 sentences as test data and varying amounts of data for training; the remainder was used as unlabeled (train) data. We use content and context views, where the content view is the current word and POS tag while the context view is the previous and next words and POS tags. We regularize the CRFs with a variance 10 Gaussian prior and weigh the unlabeled data so that it has the same total weight as the labeled data. The variance value was chosen based on preliminary experiments with the data. Table 6 shows the F-1 scores of the different models. We compare our method to a monolithic CRF as well as averaged Perceptron the two view Perceptron of Brefeld et al. (2005) with averaging. The Perceptron models were trained for 20 iterations. Preliminary experiments show that performance on held out data does not change after 10 iterations so we believe the models have converged. Both two view semi-supervised methods show gains over the corresponding fully-supervised method for 10-100 sentences of training data, but do not improve further as the amount of labeled data increases. The method presented in this paper out-performs two view Perceptron when the amount of labeled data is very small, probably because regularized CRFs perform better than Perceptron for small amounts of data. As the number of training sentences increases, two view Perceptron performs as well as our method, but at this point it has little or no improvement over the fully-supervised Perceptron.

size	CRF	SAR(RRE)	Perc	coPerc
10	73.2	**78.2** (19%)	69.4	71.2
20	79.4	**84.2** (23%)	74.4	76.8
50	86.3	**86.9** (4%)	80.1	84.1
100	88.5	**88.9** (3%)	86.1	88.1
200	89.6	89.6 (0%)	89.3	**89.7**
500	**91.3**	90.6 (-8%)	90.8	90.9
1000	91.6	91.1 (-6%)	91.5	**91.8**

Table 6: F-1 scores for noun phrase chunking with context/content views. Test data comprises 500 sentences, with 8436 sentences divided among labeled and unlabeled train data. The best score is shown in bold for each train data size.

7. Cross Lingual Projection

For English and a handful of other languages, there are large, well-annotated corpora with a variety of linguistic information ranging from named entity to discourse structure. Unfortunately, for the vast majority of languages very few linguistic resources are available. This situation is likely to persist because of the expense of creating annotated corpora that require linguistic expertise (Abeillé, 2003). On the other hand, parallel corpora between many resource-poor languages and resource-rich languages are ample, motivating recent interest in transferring linguistic resources from one language to another via parallel text.

Dependency grammars are one such resource. They are useful for language modeling, textual entailment and machine translation (Haghighi et al., 2005; Chelba et al., 1997; Quirk et al., 2005; Shen et al., 2008), to name a few tasks. Dependency grammars are arguably more robust to transfer than constituent grammars, since syntactic relations between aligned words of parallel sentences are better conserved in translation than phrase structure (Fox, 2002; Hwa et al., 2005). The two main challenges to accurate training and evaluation from aligned bitext are: (1) errors in word alignments and source language parses, (2) unaligned words due to non-literal or distant translation.

Hwa et al. (2005) proposed to learn generative dependency grammars using Collins' parser (Collins, 1999) by constructing full target parses via projected dependencies. To address challenge (1), they introduced on the order of one to two dozen language-specific transformation rules. To address challenge (2), they used a set of tree-completion rules. We present here an alternative approach to dependency grammar transfer. Our approach uses a single, intuitive PR constraint to guide grammar learning. With this constraint, we avoid the need for complex tree completion rules and many language-specific rules, yet still achieve acceptable parsing accuracy.

It should be noted that while our source of supervision, a bitext, is the same as that of Hwa et al. (2005), our learning method is more closely related to that of Druck et al. (2009). They use the GE framework to train a dependency parser. Their source of supervision comes in the form of corpus-wide expected values of linguistic rules provided by a linguistic informant.

In what follows, X will indicate parallel part-of-speech tagged sentences in a bitext corpus, along with a dependency parse of the source language. Y will indicate the dependency parses for the target language sentences.

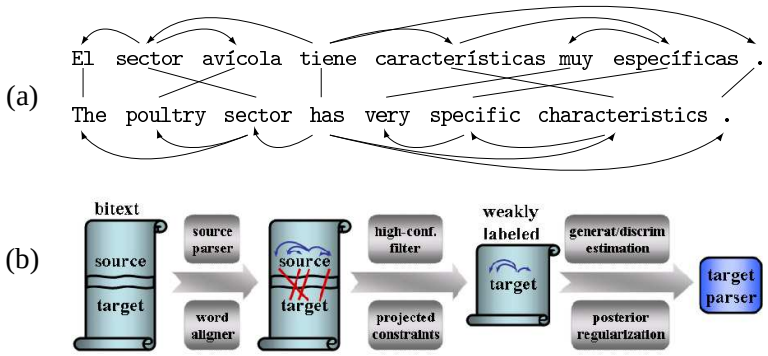


Figure 8: (a) An example word-aligned sentence pair with perfectly projected dependencies. (All dependency edges are conserved in this case.) (b) Overview of our grammar induction approach via bitext: the source (English) is parsed and word-aligned with target; after filtering, projected dependencies define constraints over target parse tree space, providing weak supervision for learning a target grammar.

7.1 Approach

Figure 8(a) shows an aligned sentence pair example where dependencies are perfectly "conserved" across the alignment. An edge from English parent p to child c is called conserved if word p aligns to word p' in the second language, c aligns to c' in the second language, and p' is the parent of c'. Note that we are not restricting ourselves to one-to-one alignments here; p, c, p', and c' can all also align to other words. Unfortunately the sentence in Figure 8(a) is highly unusual in its amount of dependency conservation, so we need to do more than directly transfer conserved edges to get good parsing accuracy.

The key to our approach is a single PR constraint, which ensures that the expected proportion of conserved edges in a sentence pair is at least η (the exact proportion we used was 0.9, which was determined using unlabeled data as described in the experiments section). Specifically, let $C_{\mathbf{x}}$ be the set of directed edges projected from English for a given sentence $\mathbf{x}$. Then given a parse $\mathbf{y}$, the proportion of conserved edges is $\phi(\mathbf{x},\mathbf{y}) = \frac{1}{|C_{\mathbf{x}}|}\sum_{y\in\mathbf{y}} \mathbf{1}(y \in C_{\mathbf{x}})$ and the expected proportion of conserved edges under distribution $p(\mathbf{y} \mid \mathbf{x})$ is

$$\mathbf{E}_p[\phi(\mathbf{x},\mathbf{y})] = \frac{1}{|C_{\mathbf{x}}|}\sum_{y\in C_{\mathbf{x}}} p(y \mid \mathbf{x}).$$

Consider how this constraint addresses errors in word alignment and source language parses, challenge (1) from above. First, note that we are constraining groups of edges rather than a single edge. For example, in some sentence pair we might find 10 edges that have both end points aligned and can be transferred. Rather than requiring our target language parse to contain each of the 10 edges, we require that the expected number of edges from this set is at least 10η. This gives the parser freedom to have some uncertainty about which edges to include, or alternatively to choose to exclude some of the transferred edges.

Our constraint does not address unaligned words due to non-literal or distant translation, challenge (2), as directly. Yet, we find that the constraint sufficiently limits the distribution over possible parses of unaligned words, such that the parser still makes reasonable choices for them. It is also

Basic Uni-gram Features

x_i-word, x_i-pos
x_i-word
x_i-pos
x_j-word, x_j-pos
x_j-word
x_j-pos

Basic Bi-gram Features

x_i-word, x_i-pos, x_j-word, x_j-pos
x_i-pos, x_j-word, x_j-pos
x_i-word, x_j-word, x_j-pos
x_i-word, x_i-pos, x_j-pos
x_i-word, x_i-pos, x_j-word
x_i-word, x_j-word
x_i-pos, x_j-pos

In Between POS Features

x_i-pos, b-pos, x_j-pos

Surrounding Word POS Features

x_i-pos, x_i-pos+1, x_j-pos-1, x_j-pos
x_i-pos-1, x_i-pos, x_j-pos-1, x_j-pos
x_i-pos, x_i-pos+1, x_j-pos, x_j-pos+1
x_i-pos-1, x_i-pos, x_j-pos, x_j-pos+1

Table 7: Features used by the MSTParser. For each edge (i, j), x_i-word is the parent word and x_j-word is the child word, analogously for POS tags. The +1 and -1 denote preceeding and following tokens in the sentence, while b denotes tokens between x_i and x_j.

worth noting that if we wished to more directly address challenge (2), we could add additional constraint features to the PR framework. For example, it seems intuitive that unaligned words might tend to be leaves (e.g., articles that are dropped in some languages but not in others). Thus, one constraint we could enforce would be to restrict the number of children of unaligned words to fall below some threshold.

For both models, we compute the projection onto the constraint set using a simple line search. This is possible since there is only one constraint per sentence and the constraints do not interact.

At a high level our approach is illustrated in Figure 8(b). A parallel corpus is word-level aligned using the method described in Section 5, where we train an alignment model via PR using symmetry constraints. The source (English) is parsed using a dependency parser (McDonald et al., 2005). Then, the filtering stage eliminates low-confidence alignments such as noun-to-verb alignments, restricts the training set to exclude possible sentence fragments, and follows the method of Klein and Manning (2004) in stripping out punctuation. We then learn a probabilistic parsing model using PR. In our experiments we evaluate the learned models on dependency treebanks (Nivre et al., 2007).

7.2 Parsing Models

We explored two parsing models: a generative model used by several authors for unsupervised induction and a discriminative model previously used for fully supervised training.

The discriminative parser is based on the edge-factored model and features of the MSTParser (McDonald et al., 2005). The parsing model defines a conditional distribution $p_\theta(\mathbf{y} \mid \mathbf{x})$ over each projective parse tree $\mathbf{y}$ for a particular sentence $\mathbf{x}$, parameterized by a vector θ. The probability of any particular parse is

$$p_\theta(\mathbf{y} \mid \mathbf{x}) \propto \prod_{y \in \mathbf{y}} e^{\theta \cdot \mathbf{f}(y, \mathbf{x})},$$

where y is a directed edge contained in the parse tree $\mathbf{y}$ and $\mathbf{f}$ is a feature function. In the fully supervised experiments we run for comparison, parameter estimation is performed by stochastic gradient ascent on the conditional likelihood function, similar to maximum entropy models or conditional random fields. One needs to be able to compute expectations of the features $\mathbf{f}(y, \mathbf{x})$ under the distribution $p_\theta(y \mid \mathbf{x})$. A version of the inside-outside algorithm (Lee and Choi, 1997) performs this computation. Viterbi decoding is done using Eisner's algorithm (Eisner, 1996).

We also used a generative model based on dependency model with valence (Klein and Manning, 2004). Under this model, the probability of a particular parse $\mathbf{y}$ and a sentence with part-of-speech tags $\mathbf{x}$ is given by

$$p_\theta(\mathbf{y},\mathbf{x}) = p_{\text{root}}(r(\mathbf{x})) \cdot \Big(\prod_{y\in\mathbf{y}} p_{\neg\text{stop}}(y_p,y_d,v_y)\, p_{\text{child}}(y_p,y_d,y_c)\Big) \cdot \Big(\prod_{x\in\mathbf{x}} p_{\text{stop}}(x,\text{left},v_l)\, p_{\text{stop}}(x,\text{right},v_r)\Big)$$

where $r(\mathbf{x})$ is the part-of-speech tag of the root of the parse tree $\mathbf{y}$, y is an edge from parent y_p to child y_c in direction y_d, either left or right, and v_y indicates valency—false if y_p has no other children further from it in direction y_d than y_c, true otherwise. The valencies v_r/v_l are marked as true if x has any children on the left/right in $\mathbf{y}$, false otherwise.

We regularize the models by parameter prior $-\log p(\theta) = R(\theta)$, where $p(\theta)$ is Gaussian for the discriminative model and Dirichlet for the generative.

7.3 Experiments

We evaluate our approach by transferring from an English parser trained on the Penn treebank to Bulgarian and Spanish. We evaluate our results on the Bulgarian and Spanish corpora from the CoNLL X shared task. The Bulgarian experiments transfer a parser from English to Bulgarian, using the OpenSubtitles corpus (Tiedemann, 2007). The Spanish experiments transfer from English to Spanish using the Spanish portion of the Europarl corpus (Koehn, 2005). For both corpora, we performed word alignments with the open source PostCAT (Graça et al., 2009b) toolkit. We used the Tokyo tagger (Tsuruoka and Tsujii, 2005) to POS tag the English tokens, and generated parses using the first-order model of McDonald et al. (2005) with projective decoding, trained on sections 2-21 of the Penn treebank with dependencies extracted using the head rules of Yamada and Matsumoto (2003). For Bulgarian we trained the Stanford POS tagger (Toutanova et al., 2003) on the Bulgtreebank corpus from CoNLL X. The Spanish Europarl data was POS tagged with the FreeLing language analyzer (Atserias et al., 2006). The discriminative model used the same features as MSTParser, summarized in Table 7. Our model uses constraints of the form: the expected proportion of conserved edges in a sentence pair is at least $\eta = 90\%$.[10]

In order to better evaluate our method, we construct a baseline inspired by Hwa et al. (2005). The baseline creates a full parse tree from the incomplete and possibly conflicting transferred edges using a simple random process. We start with no edges and try to add edges one at a time verifying at each step that it is possible to complete the tree. We first try to add the transferred edges in random order, then for each orphan node we try all possible parents (both in random order). We then use this full labeling as supervision for a parser. Note that this baseline is very similar to the first iteration of our model, since for a large corpus the different random choices made in different sentences tend to smooth each other out. We also tried to create rules for the adoption of orphans, but the simple rules we tried added bias and performed worse than the baseline we report.

10. We chose η in the following way: We split the unlabeled parallel text into two portions. We trained models with different η on one portion and ran it on the other portion. We chose the model with the highest fraction of conserved constraints on the second portion.

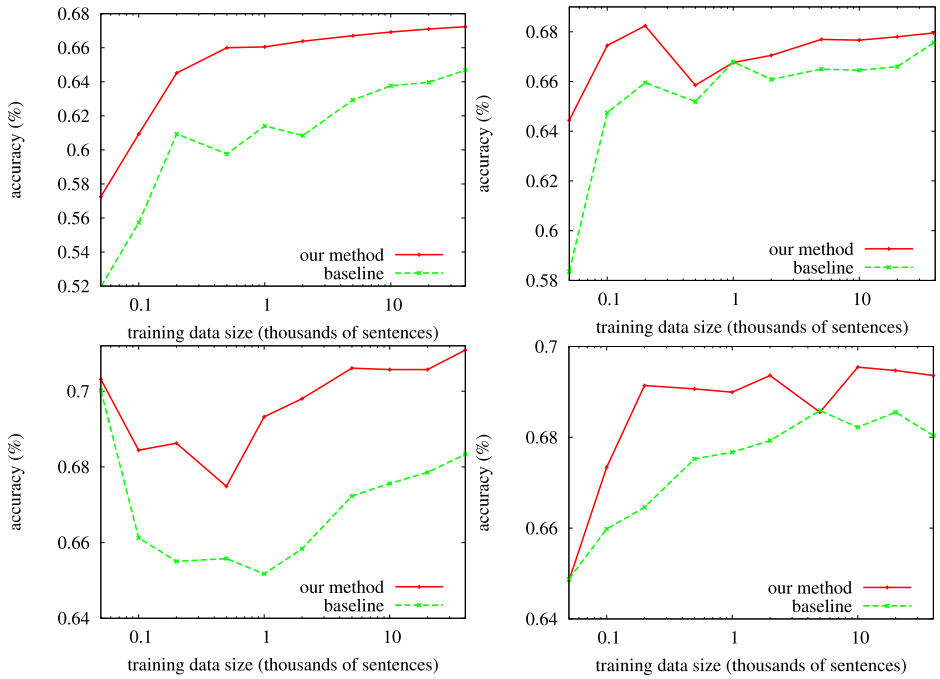


Figure 9: Learning curves. Each graph compares transferring a single tree of edges (baseline) and transferring all possible projected edges (our method). The models were trained on sentences of length up to 20 and tested on CoNLL train sentences of length up to 10. Punctuation was stripped at train time. **Top**: Bulgarian. **Bottom**: Spanish. **Left**: Discriminative model. **Right**: Generative model. The non-monotonicity of the (such as bottom left) is because each point is based on a single random sample of sentences. This random selection can greatly affect performance when the number of sentences is small.

7.3.1 RESULTS

Models are evaluated based on attachment accuracy—the fraction of words assigned the correct parent. Figure 9 shows that models generally improve with more transfer-type data. It also shows our method consistently outperforms the baseline. Note that each point in these graphs is based on a single random subsample of the data, which leads to some non-monotonicity in the left-half of some of the curves. The exact accuracy numbers for the 10k training sentences point of Figure 9 are given in Table 8. Link-left baselines for these corpora are much lower: 33.8% and 27.9% for Bulgarian and Spanish respectively.

7.3.2 GENERATIVE PARSER

The generative model we use is a state of the art model for unsupervised parsing. Before evaluating, we smooth the resulting models by adding e^{-10} to each learned parameter, merely to remove the chance of zero probabilities for unseen events. (We did not bother to tune this value at all as it makes very little difference for final parses.) Unfortunately, we found generative model performance was disappointing in the unsupervised setting. Using the initialization procedure from Klein and Man-

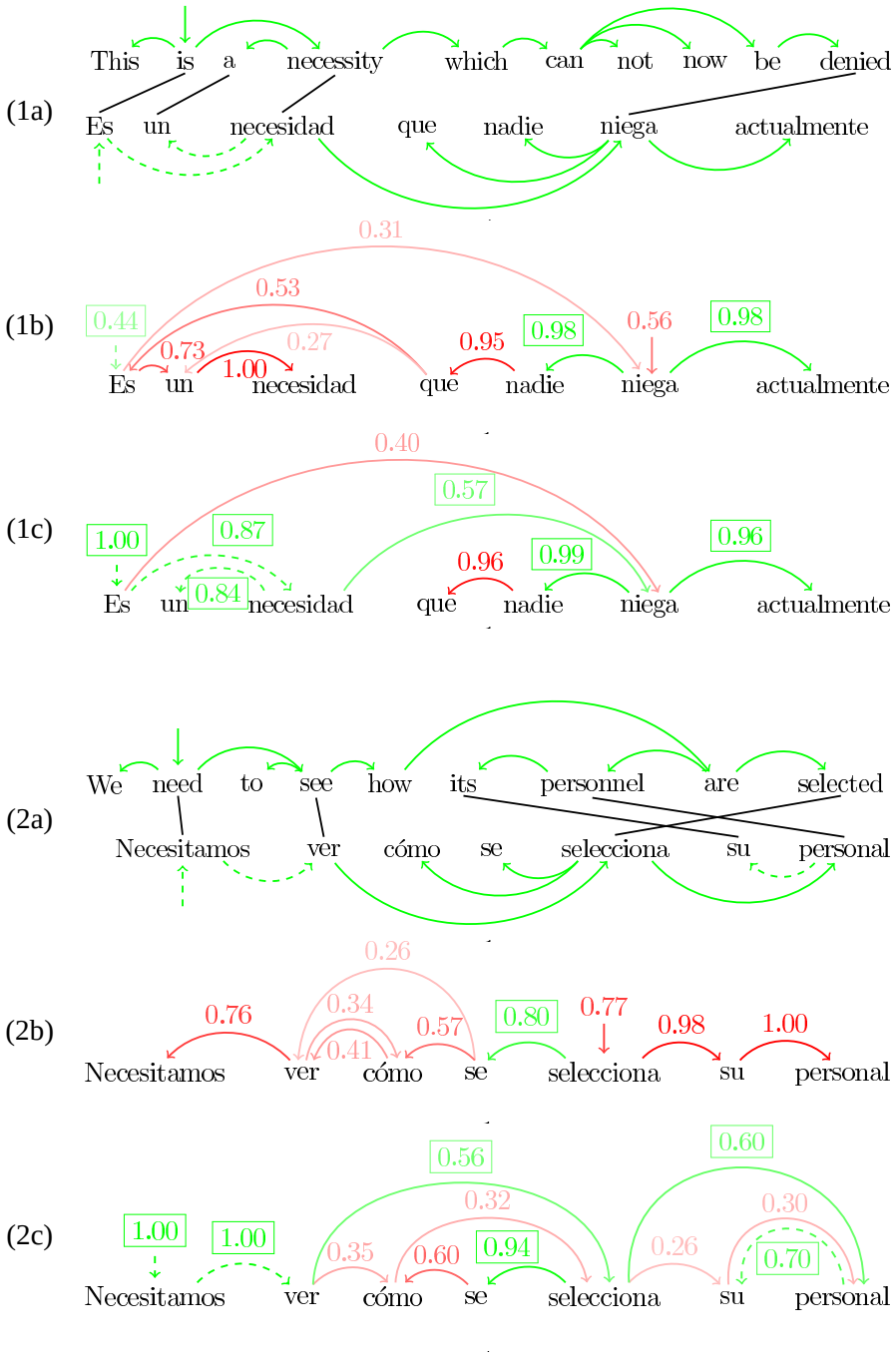


Figure 10: Posteriors of two Spanish sentences from Europarl. The number on each edge indicates the edge's posterior probability. Edges with probability less than 0.25 are not shown. Darker (more saturated) edges are higher probability. Green (with boxed number) indicates a correct edge, red (no box) an incorrect. Dotted edges are conserved. **(a)** The gold source and target parses and their alignment. **(b)** Unsupervised model initialized as per Klein and Manning (2004) and trained for 100 EM iterations. **(c)** PR projection applied to the posteriors of the middle figure, forcing $\mathbf{E}_p[C] \geq C * \eta = 3 * 0.9$, where C is the number of conserved edges.

	Discriminative		Generative	
	Bulgarian	Spanish	Bulgarian	Spanish
Baseline	63.8	67.6	66.5	68.2
Post.Reg.	**66.9**	**70.6**	**67.8**	**69.5**

Table 8: Accuracy values at the 10k training sentences point of Figure 9.

ning (2004), the maximum unsupervised accuracy it achieves is 55.4% for Bulgarian and 41.7% for Spanish, and these results are not stable. Changing the initialization parameters or training sample drastically affects the results, even for samples with several thousand sentences. But when we use the transferred information to constrain the learning, EM stabilizes and achieves much better performance, also beating the Hwa et al. (2005)-inspired baseline. With the transferred information, even setting all parameters equal at the outset does not prevent the model from learning the dependency structure of the aligned language. Figure 10 shows an example of how PR projection helps better estimate posteriors of two example sentences.

7.3.3 DISCRIMINATIVE PARSER

We trained our discriminative parser for 100 iterations of online EM with a Gaussian prior variance of 100. The transfer system performs better than the unsupervised generative model and the baseline model for both Bulgarian and Spanish. We observed another desirable property of the discriminative model: While the generative model can get confused and perform poorly when the training data contains very long sentences, the discriminative parser does not appear to have this drawback. In fact we observed that as the maximum training sentence length increased, the parsing performance also improved.

8. Enforcing Sparsity Structure

Many important NLP tasks (e.g., tagging, parsing, named-entity recognition) involve word classification. Often, we know a priori that a word type might belong to a small set of classes (where the class of a specific instance depends on its context) and should never belong to any of the many possible classes outside this small set. The part-of-speech tagging task, described in the running example, is one instance of this phenomenon. For example, consider the word type "run". It might belong to the verb class in some instances and the noun class in others, but it will never be an adjective, adverb, conjunction, determiner, etc. Learning algorithms typically assume that each word type can be associated with any existing tag, even though in reality each word type is only ever associated with a few tags.

Unsupervised induction of this latent structure is normally performed using the EM algorithm, but it has exhibited disappointing performance in previous work. One well-known reason for this is that EM tends to allow each word to be generated by most POS tags some of the time. In reality, we would like most words to have a small number of possible POS tags. Previous work has attempted to solve this problem by applying the Bayesian approach, using a prior to encourage sparsity in the model *parameters* (Gao and Johnson, 2008; Johnson, 2007; Goldwater and Griffiths, 2007). However, this approach has the drawback of enforcing sparsity in the wrong direction; sparsity at the parameter level encodes a preference that each POS tag should generate only a few words,

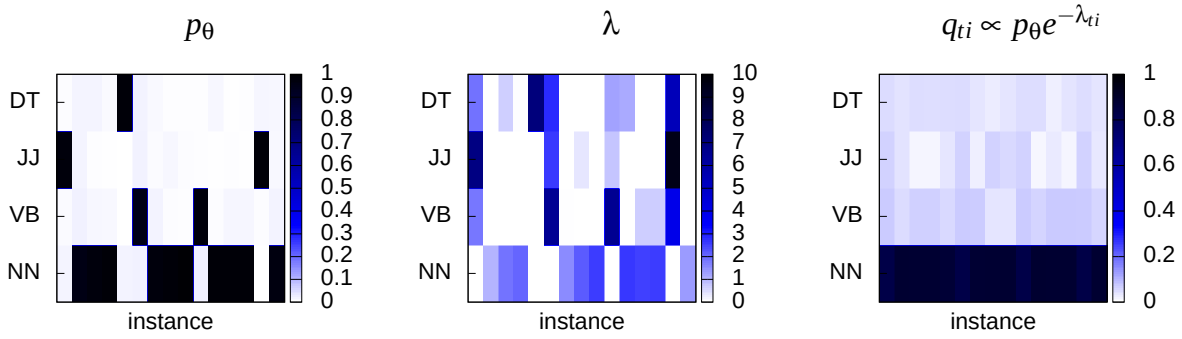


Figure 11: An illustration of ℓ_1/ℓ_∞ regularization. Left panel: initial tag distributions (columns) for 15 instances of a word. Middle panel: optimal regularization parameters λ, each row sums to $\sigma = 20$. Right panel: q concentrates the posteriors for all instances on the NN tag, reducing the ℓ_1/ℓ_∞ norm from just under 4 to a little over 1.

instead of encoding that each word should generate only a few POS tags. Here we explore the problem of biasing unsupervised models to favor the correct sparsity by encouraging the model to achieve *posterior* sparsity on unlabeled training data.

8.1 ℓ_1/ℓ_∞ Regularization

We focus on the slack-penalized formulation of Section 2.3 for this task. We choose the PR constraint to encourage each word to be associated with only a few parts of speech. Let the constraint feature $\phi_{wti}(\mathbf{X}, \mathbf{Y})$ have value 1 whenever the i^{th} occurrence of word w has part-of-speech tag t, and value 0 otherwise. For every word w, we would like there to be only a few POS tags t such that there are occurrences i where t has nonzero probability. This can be achieved if it "costs" a lot the first time an occurrence of a word takes a particular tag, but afterwards future occurrences of the word can receive that same tag for free. More precisely, for each word type w, we would like the sum (ℓ_1 norm), over tags t, of the maximum (ℓ_∞ norm), over all occurrences w_i of w, of $p(w_i \mid t)$, to be small; we want $\sum_{t,w} \max_i p(w_i \mid t)$ to be small. Note that in contrast to previous applications, these constraints are corpus-wide instead of instance-specific. For notational simplicity we will write $\phi_{wti}(\mathbf{X}, \mathbf{Y}) = \phi_{wti}(\mathbf{Y})$, since any dependence on $\mathbf{X}$ is captured in the subscripts of ϕ_{wti}.

Formally, this objective is an example of the slack-penalized formulation (as in Equation 3), but for simplicity we will use the notation:

$$\min_{q, c_{wt}} \ \mathbf{KL}(q||p_\theta) + \sigma \sum_{wt} c_{wt} \quad \text{s.t.} \quad \mathbf{E}_q[\phi_{wti}] \leq c_{wt}.$$

Mapping this notation to the original equation we have: $\mathbf{b} = 0$ and regularization strength σ. The constraints on the features ϕ_{wti} and the summation over c_{wt} together encode the ℓ_1/ℓ_∞ norm. The variables c_{wt} represent the ℓ_∞ norm of ϕ_{wti}, $c_{wt} = ||\phi_{wti}||_{\ell_\infty}$, while the summation is the ℓ_1 norm of c_{wt}. The dual of this objective has a very simple form:

$$\max_{\lambda \geq 0} \ -\log \left(\sum_y p_\theta(\mathbf{Y}) \exp(-\boldsymbol{\lambda} \cdot \phi(\mathbf{Y})) \right) \quad \text{s.t.} \quad \sum_i \lambda_{wti} \leq \sigma, \tag{25}$$

where $\mathbf{Y}$ ranges over assignments to the hidden tag variables for all of the occurrences in the training data, $\phi(\mathbf{Y})$ is the vector of ϕ_{wti} constraint feature values for assignment $\mathbf{Y}$, $\boldsymbol{\lambda}$ is the vector of dual parameters λ_{wti}, and the primal parameters are $q(\mathbf{Y}) \propto p_\theta(\mathbf{Y}) \exp(-\boldsymbol{\lambda} \cdot \phi(\mathbf{Y}))$.

An advantage of using slack penalties in this case is that ℓ_1/ℓ_∞ as a slack constraint in the primal would lead to a non-differentiable dual penalty term, which somewhat complicates optimization. Using a slack penalty makes sparsity regularization a primal penalty, yielding dual simplex constraints, solvable efficiently via projected gradient, as described by Bertsekas (1999). Note that the simplex constraints in Equation 25 can be interpreted as an ℓ_∞/ℓ_1 norm, which is the dual of the ℓ_1/ℓ_∞.

Figure 11 illustrates how the ℓ_1/ℓ_∞ norm operates on a toy example. For simplicity suppose we are only regularizing one word and our model p_θ is just a product distribution over 15 instances of the word. The left panel in Figure 11 shows the posteriors under p_θ. We would like to concentrate the posteriors on a small subset of rows. The center panel of the figure shows the $\boldsymbol{\lambda}$ values determined by Equation 25, and the right panel shows the projected distribution q, which concentrates most of the posterior on the bottom row. Note that we are not requiring the posteriors to be sparse, which would be equivalent to preferring that the distribution is peaked; rather, we want a word to concentrate its tag posterior on a few tags across all instances of the word. Indeed, most of the instances (columns) become less peaked than in the original posterior to allow posterior mass to be redistributed away from the outlier tags. Since they are more numerous than the outliers, they moved less. This also justifies only regularizing relatively frequent events in our model.

8.2 Results

In this section we present an empirical comparison of first-order HMMs trained with three different methods: classic EM (EM), ℓ_1/ℓ_∞ PR (Sparse), and Bayesian estimation using a variational approximation described in Johnson (2007) and Gao and Johnson (2008) (VEM). Models are trained and tested on three different corpora: the Wall Street Journal portion of the Penn treebank (Marcus et al., 1993) using a reduced set of 17 tags (Smith et al., 2005) (PTB17); the Bosque subset of the Portuguese Floresta Sinta(c)tica Treebank (Afonso et al., 2002)[11] used for the ConLL X shared task on dependency parsing (PT-CoNLL)[12]; and the Bulgarian BulTreeBank (Simov et al., 2002) (BulTree) with 12 coarse tags. All words that occurred only once were replaced by the token "unk". To measure model sparsity, we compute the average ℓ_1/ℓ_∞ norm over words occurring more than 10 times; the label 'L1LMax' denotes this measure in figures. Table 9 gives statistics for each corpus as well as the sparsity for a first-order HMM trained on the labeled data.

Following Gao and Johnson (2008), the parameters were initialized with a "pseudo E-step" as follows: we filled the expected count matrices with numbers $1 + X \times U(0,1)$, where $U(0,1)$ is a random number between 0 and 1 and X is a parameter. These matrices are then fed to the M-step; the resulting "random" transition and emission probabilities are used for the first real E step. For VEM X was set to 0.0001 (almost uniform) since this showed a significant improvement in performance. On the other hand EM showed less sensitivity to initialization, and we used $X = 1$ which resulted in the best results. The models were trained for 200 iterations as longer runs did not significantly change the results. For VEM we tested 4 different prior combinations based on the results of Johnson (2007); in later work Gao and Johnson (2008) considered a wider range of values

11. The subset can be found at `http://www.linguateca.pt/Floresta/`.
12. The task can be found at `http://nextens.uvt.nl/~conll/`.

	Types	Tokens	Unk	Tags	ℓ_1/ℓ_∞
PTB17	23768	950028	2%	17	1.23
PT-Conll	11293	206678	8.5%	22	1.14
BulTree	12177	174160	10%	12	1.04

Table 9: Corpus statistics. All words with only one occurrence where replaced by the 'unk' token. The third column shows the percentage of tokens replaced. ℓ_1/ℓ_∞ is the value of the sparsity for a fully supervised HMM trained in all available data.

but did not identify definitely better choices. Sparse was initialized with the parameters obtained by running EM for 30 iterations, followed by 170 iterations of the new training procedure. Predictions were obtained using posterior decoding since this consistently showed small improvements over Viterbi decoding.

We compare the models by measuring the mutual information between the distribution of hidden states and the distribution of the truth. Ideally, a perfect method would have mutual information equal to the entropy of both distributions. The farther the distribution that a method produces is from the truth the smaller the information gain is. We also evaluate the accuracy of the models using two established mappings between hidden states and POS tags: **(1-Many**) maps each hidden state to the tag with which it co-occurs the most; **1-1** (Haghighi and Klein, 2006) greedily picks a tag for each state under the constraint of never using the same tag twice. This results in an approximation of the optimal 1-1 mapping. If the numbers of hidden states and tags are not the same, some hidden states will be unassigned (and hence always wrong) or some tags not used. In all our experiments the number of hidden states is the same as the number of POS tags.

Figure 12 (Top Left) shows mutual information between the hidden state distribution of each method and the truth. The entropy of the true distribution are: BulTree 3.05, PT-CoNLL 3.49 and PTB17 3.22. Sparse is the method that achieves the biggest information gain across all corpora, and is not particularly sensitive to the strength of regularization used. Interestingly, VEM often has the smallest ℓ_1/ℓ_∞, even though mutual information is often worst than EM.

Figure 12 (Top Right) shows the different average values of the L1LMax statistics for each method across corpora. We see that both VEM and Sparse achieve values of ℓ_1/ℓ_∞ close to the gold standard, on the other hand EM as expected as bigger values which confirms the intuition that EM allows each word to be generated by most of the possible POS tags.

Figure 12 (Bottom Left) shows errors for all methods on the different corpora after 10 random initializations using the 1-Many mapping. For both VEM and Sparse we pick parameter settings resulting in the best average performance. A first conclusion is that using the ℓ_1/ℓ_∞ constraint consistently and significantly improves the results when compared with the other two methods.

Figure 12 (Bottom Right) shows the same errors for the 1-1 mapping. In this case Sparse still beats the EM but does not always outperform VEM. One reason for this behavior is that this metric is very sensitive to the number of word types associated with each hidden state. VEM tends to encourage some large hidden states with many word types, which is preferable using the 1-1 mapping for large word categories such as nouns. On the other hand Sparse tends to spread the nouns over 4 different hidden states. This difference is particularly pronounced for the condensed tag sets (PTB17, PT-CoNLL) where different kinds of nouns are joined into one large tag. Also this

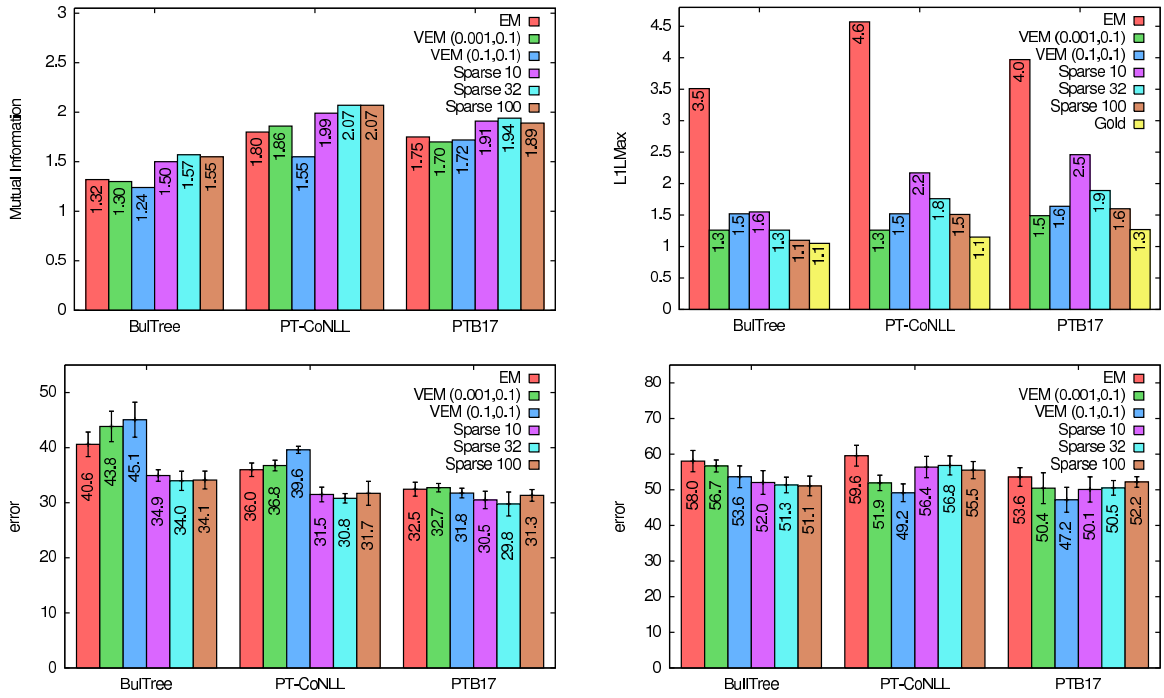


Figure 12: (Top Left) Mutual information in bits between gold tag distribution and hidden state distribution. The maximum is the entropy of the gold set (BulTree 3.05, PT-CoNLL 3.49 and PTB17 3.22), (Top Right) ℓ_1/ℓ_∞ value, and average (Bottom Left) 1-Many error, (Bottom Right) 1-1 error over 10 different runs (same seeds used for each model) for 200 iterations. Error bars are standard deviation for the 10 runs. All models are first order HMMs: EM trained using expectation maximization, VEM trained using variational EM using 0.1 state to state prior and (01,0.0001) observation prior; Sparse trained using PR with constraint strength $\sigma = 10, 32, 100$.

difference is bigger for VEM when the observation prior is set to bigger values (0.1), leading at the same time to worse results in 1-Many mapping.

9. Conclusion

In this paper we have presented posterior regularization (PR), a technique for regularizing models by encoding prior knowledge in constraints on model posteriors. On the algorithmic side, we have shown that PR can be solved efficiently in dual form, and that the regularization can be easily incorporated into a variant of the classical EM optimization method. In relating PR to similar frameworks, we have clarified its main advantages: faster optimization speed with respect to generalized expectation (Mann and McCallum, 2007, 2008), and greater distributional estimation accuracy with respect to constraint-driven learning (Chang et al., 2007). To the best of our knowledge, we are the first to link all these learning frameworks by explicitly stating a sense in which they all approximate the Bayesian perspective that motivates Liang et al. (2009). An interesting avenue for future work

includes an exploration of the tradeoff between computational complexity and accuracy of the different approximations presented in this and related work (Figure 4). For example, is there a large performance drop as we go from GE to PR and from PR to CODL, or are the variational and MAP approximations accurate in practice?

In addition to discussing PR's theoretical potential, we have demonstrated that it lives up to this potential in a wide variety realistic applications. The applications we focus on in this paper are word alignment, multi-view learning, dependency parsing, and part-of-speech tagging. Yet PR can express such a wide variety of prior knowledge that can be encoded by functions of model posteriors, and there remains a vast array of unexplored possible applications for this technique.

In addition to using PR in other applications, we would like to investigate alternative optimization methods. The main optimization bottleneck that PR implementations encounter is the extensive time required for projecting the posterior distribution into the constrained posterior space. Each evaluation of the objective or its gradient requires inference in the original model. One direction for exploration is the use of second order or approximate second-order optimization methods. Another potential direction is to use approximate inference in some parts of the optimization, for example fully factored variational inference. Finally, for applications where some constraints span multiple instances, but others do not not, it would be interesting to combine online and batch methods.

A last key extension to the current PR work is to explore the case where the constraint set Q is not easily specified using linear constraints on some constraint features ϕ. Thus far we have only developed theory and applications for linear constraints. It would be interesting to explore applications and derive efficient learning methods when the constraints are not linear, for example, applications with semi-definite or polynomial constraints.

Acknowledgments

The authors would like to thank Gregory Druck, Gideon Mann, Percy Liang, Fernando Pereira, Umar Syed and Mitch Marcus for helpful comments and discussion on drafts of this paper. J. V. Graça was supported by a fellowship from Fundação para a Ciência e Tecnologia (SFRH/ BD/ 27528/ 2006) and by FCT project CMU-PT/HuMach/0039/2008. K. Ganchev was supported by ARO MURI SUBTLE W911NF-07-1-0216. J. Gillenwater work was partially supported by NSF-IGERT 0504487. B. Taskar was partially supported by DARPA CSSG and ONR Young Investigator Award N000141010746.

Appendix A. Scaling the Strength of PR

This appendix describes how to optimize a version of our objective with scaled posterior regularization strength. In this case, we will use a modified EM algorithm that maximizes:

$$F'(q,\theta) = \mathcal{L}(\theta) - \alpha \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y} \mid \mathbf{X})) \qquad \text{s.t.}\, q \in Q$$

where $\alpha \in [0,1]$. The optimization procedure closely follows the one in Section 2.6. When performing the M-step, we use a mixture of the projected posteriors q and the model posteriors $p_\theta(\mathbf{Y}|\mathbf{X})$ to

update the model parameters. The updated EM algorithm is:

$$\mathbf{E}' - \textbf{step}: \max_q F'(q,\theta) = \min_{q \in Q} \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})),$$
$$\mathbf{M}' - \textbf{step}: \max_\theta F'(q,\theta) = \max_\theta (1-\alpha)\mathbf{E}_{p_{\theta'}}[\log p_\theta(\mathbf{X},\mathbf{Y})] + \alpha \mathbf{E}_q[\log p_\theta(\mathbf{X},\mathbf{Y})].$$

Note that the $\mathbf{E}'$-step is identical to the one in Equation 8.

Appendix B. Proof of Proposition 2.1

The modified E-step involves a projection step that minimizes the Kullback-Leibler divergence:

$$\arg\min_{q,\xi} \quad \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) \qquad \text{s.t.} \quad \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} \leq \xi; \quad ||\xi||_\beta \leq \varepsilon.$$

Assuming the set $Q = \{q(\mathbf{Y}) : \exists \xi : \mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} \leq \xi; \quad ||\xi||_\beta \leq \varepsilon\}$ is non-empty, the corresponding Lagrangian is

$$\max_{\lambda \geq 0, \alpha \geq 0, \gamma} \min_{q(\mathbf{Y}), \xi} L(q(\mathbf{Y}), \xi, \lambda, \alpha, \gamma),$$

where

$$L(q,\xi,\lambda,\alpha,\gamma) = \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) \quad + \lambda \cdot (\mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} - \xi) \\ + \alpha(||\xi||_\beta - \varepsilon) \quad + \gamma \left(\sum_{\mathbf{Y}} q(\mathbf{Y}) - 1 \right).$$

In order to compute the dual of this Lagrangian, we first represent

$$\alpha ||\xi||_\beta = \max_\eta \xi \cdot \eta \quad \text{s.t.} \quad ||\eta||_{\beta^*} \leq \alpha.$$

This results in a variational Lagrangian

$$\max_{\lambda \geq 0, \alpha \geq 0, \gamma} \max_{||\eta||_{\beta^*} \leq \alpha} \min_{q(\mathbf{Y}), \xi} L(q(\mathbf{Y}), \xi, \lambda, \alpha, \gamma, \eta),$$

with $L(q(\mathbf{Y}), \xi, \lambda, \alpha, \gamma, \eta)$ defined as

$$L(q,\xi,\lambda,\alpha,\gamma,\eta) = \quad \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) \quad + \lambda \cdot (\mathbf{E}_q[\phi(\mathbf{X},\mathbf{Y})] - \mathbf{b} - \xi) \\ + \xi \cdot \eta - \alpha\varepsilon \quad + \gamma \left(\sum_{\mathbf{Y}} q(\mathbf{Y}) - 1 \right),$$

$$\frac{\partial L(q(\mathbf{Y}), \xi, \lambda, \alpha, \gamma, \eta)}{\partial q(\mathbf{Y})} = \quad \log q(\mathbf{Y}) + 1 - \log p_\theta(\mathbf{Y}|\mathbf{X}) + \lambda \cdot \phi(\mathbf{X},\mathbf{Y}) + \gamma = 0$$
$$\implies \quad q(\mathbf{Y}) = \frac{p_\theta(\mathbf{Y}|\mathbf{X}) \exp(-\lambda \cdot \phi(\mathbf{X},\mathbf{Y}))}{e \exp(\gamma)},$$
$$\frac{\partial L(q(\mathbf{Y}), \xi, \lambda, \alpha, \gamma, \eta)}{\partial \xi_i} = \quad \eta_i - \lambda_i = 0 \quad \implies \quad \eta = \lambda. \qquad (26)$$

Note that Equation 26 implies that we have the constraint $||\lambda||_{\beta^*} \leq \alpha$ and also the positive and negative $\lambda \cdot \xi$ cancel each other out. Plugging $q(\mathbf{Y})$, $\eta = \lambda$ in $L(q(\mathbf{Y}), \xi, \lambda, \alpha, \gamma, \eta)$ and taking the derivative with respect to γ

$$\frac{\partial L(\lambda, \alpha, \gamma)}{\partial \gamma} = \sum_{\mathbf{Y}} \frac{p_\theta(\mathbf{Y}|\mathbf{X}) \exp(-\lambda \cdot \phi(\mathbf{X}, \mathbf{Y}))}{e \exp(\gamma)} - 1 = 0$$
$$\Longrightarrow \gamma = \log \left(\frac{\sum_{\mathbf{Y}} p_\theta(\mathbf{Y}|\mathbf{X}) \exp\left(-\lambda \cdot \phi(\mathbf{x}, \mathbf{z})\right)}{e} \right).$$

From there we can simplify $q(\mathbf{Y}) = \frac{p_\theta(\mathbf{Y}|\mathbf{X}) \exp(-\lambda \cdot \phi(\mathbf{X}, \mathbf{Y}))}{Z_\lambda}$ where $Z_\lambda = \sum_{\mathbf{Y}} p_\theta(\mathbf{Y}|\mathbf{X}) \exp(-\lambda \cdot \phi(\mathbf{X}, \mathbf{Y}))$ ensures that $q(\mathbf{Y})$ is properly normalized. Plugging γ into $L(\lambda, \alpha, \gamma)$

$$L(\lambda, \alpha) = -\log(Z_\lambda) - \mathbf{b} \cdot \lambda - \alpha\varepsilon.$$

Now our objective is:

$$\max_{\lambda \geq 0, \alpha \geq 0} \quad -\log(Z_\lambda) - \mathbf{b} \cdot \lambda - \alpha\varepsilon \quad \text{s.t.} \quad ||\lambda||_{\beta^*} \leq \alpha.$$

We can analytically see that the optimum of this objective with respect to α is $\alpha = ||\lambda||_{\beta^*}$ and placing this in $L(\lambda, \alpha)$ we get the dual objective:

$$\textbf{Dual } \mathbf{E}' : \quad \arg\max_{\lambda \geq 0} \quad -\mathbf{b} \cdot \lambda - \log(Z_\lambda) - \varepsilon ||\lambda||_{\beta^*}$$

as desired.

References

A. Abeillé. *Treebanks: Building and Using Parsed Corpora*. Springer, 2003.

S. Afonso, E. Bick, R. Haber, and D. Santos. Floresta Sinta(c)tica: a treebank for Portuguese. In *Proc. LREC*, 2002.

Y. Altun, M. Johnson, and T. Hofmann. Investigating loss functions and optimization methods for discriminative learning of label sequences. In *Proc. EMNLP*, 2003.

R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

J. Atserias, B. Casas, E. Comelles, M. González, L. Padró, and M. Padró. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *Proc. LREC*, 2006.

M. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. In *Proc. COLT*, 2005.

C. Bannard and C. Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proc. ACL*, 2005.

K. Bellare, G. Druck, and A. McCallum. Alternating projections for learning with expectation constraints. In *Proc. UAI*, 2009.

D. P. Bertsekas. *Nonlinear Programming: 2nd Edition*. Athena scientific, 1999.

J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proc. EMNLP*, 2006.

J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. ACL*, 2007.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. COLT*, 1998.

U. Brefeld, C. Büscher, and T. Scheffer. Multi-view hidden markov perceptrons. In *Proc. LWA*, 2005.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, M. J. Goldsmith, J. Hajic, R. L. Mercer, and S. Mohanty. But dictionaries are data too. In *Proc. HLT*, 1993.

P. F. Brown, S. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1994.

C. Callison-Burch. *Paraphrasing and Translation*. PhD thesis, University of Edinburgh, 2007.

C. Callison-Burch. Syntactic Constraints on Paraphrases Extracted from Parallel Corpora. In *Proc. EMNLP*, 2008.

A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr., and T. M. Mitchell. Coupled Semi-Supervised Learning for Information Extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM)*, 2010.

M. Chang, L. Ratinov, and D. Roth. Guiding semi-supervision with constraint-driven learning. In *Proc. ACL*, 2007.

M.W. Chang, L. Ratinov, N. Rizzolo, and D. Roth. Learning and inference with constraints. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. AAAI, 2008.

C. Chelba, D. Engle, F. Jelinek, V. Jimenez, S. Khudanpur, L. Mangu, H. Printz, E. Ristad, R. Rosenfeld, A. Stolcke, and D. Wu. Structure and performance of a dependency language model. In *Proc. Eurospeech*, 1997.

D. Chiang, A. Lopez, N. Madnani, C. Monz, P. Resnik, and M. Subotin. The hiero machine translation system: extensions, evaluation, and analysis. In *Proc. HLT-EMNLP*, 2005.

M. Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proc. SIGDAT-EMNLP*, 1999.

H. Daumé III. Cross-task knowledge-constrained self training. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Royal Statistical Society, Ser. B*, 39(1):1–38, 1977.

G. Druck, G. Mann, and A. McCallum. Semi-supervised learning of dependency parsers using generalized expectation criteria. In *Proc. ACL-IJCNLP*, 2009.

J. Eisner. Three new probabilistic models for dependency parsing: an exploration. In *Proc. CoLing*, 1996.

H. Fox. Phrasal cohesion and statistical machine translation. In *Proc. EMNLP*, 2002.

M. Galley, M. Hopkins, K. Knight, and D. Marcu. What's in a translation rule? In *Proc. HLT-NAACL*, 2004.

K. Ganchev, J. Graça, J. Blitzer, and B. Taskar. Multi-view learning over structured and non-identical outputs. In *Proc. UAI*, 2008a.

K. Ganchev, J. Graça, and B. Taskar. Better alignments = better translations? In *Proc. ACL*, 2008b.

K. Ganchev, J. Gillenwater, and B. Taskar. Dependency grammar induction via bitext projection constraints. In *Proc. ACL-IJCNLP*, 2009.

J. Gao and M. Johnson. A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. In *Proc. EMNLP*, 2008.

S. Goldwater and T. Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proc. ACL*, 2007.

J. Graça, K. Ganchev, and B. Taskar. Expectation maximization and posterior constraints. In *Proc. NIPS*, 2007.

J. Graça, K. Ganchev, F. Pereira, and B. Taskar. Parameter vs. posterior sparisty in latent variable models. In *Proc. NIPS*, 2009a.

J. Graça, K. Ganchev, and B. Taskar. Postcat - posterior constrained alignment toolkit. In *The Third Machine Translation Marathon*, 2009b.

J. Graça, K. Ganchev, and B. Taskar. Learning tractable word alignment models with complex constraints. *Computational Linguistics*, 36, September 2010.

A. Haghighi and D. Klein. Prototype-driven learning for sequence models. In *Proc. NAACL*, 2006.

A. Haghighi, A. Ng, and C. Manning. Robust textual inference via graph matching. In *Proc. EMNLP*, 2005.

R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:11–311, 2005.

M. Johnson. Why doesn't EM find good HMM POS-taggers. In *Proc. EMNLP-CoNLL*, 2007.

T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communications*, 15(1):52–60, 2 1967. ISSN 0096-2244.

S. Kakade and D. Foster. Multi-view regression via canonical correlation analysis. In *Proc. COLT*, 2007.

D. Klein and C. Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proc. ACL*, 2004.

P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, 2005.

P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. NAACL*, 2003.

S. Lee and K. Choi. Reestimation and best-first parsing algorithm for probabilistic dependency grammar. In *Proc. WVLC-5*, 1997.

Z. Li and J. Eisner. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 40–51, Singapore, 2009.

P. Liang, B. Taskar, and D. Klein. Alignment by agreement. In *Proc. HLT-NAACL*, 2006.

P. Liang, M. I. Jordan, and D. Klein. Learning from measurements in exponential families. In *Proc. ICML*, 2009.

G. S. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proc. ICML*, 2007.

G. S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proc. ACL*, 2008.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330, 1993.

E. Matusov, R. Zens, and H. Ney. Symmetric word alignments for statistical machine translation. In *Proc. COLING*, 2004.

E. Matusov, N. Ueffing, and H. Ney. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. In *Proc. EACL*, 2006.

R. McDonald, K. Crammer, and F. Pereira. Online large-margin training of dependency parsers. In *Proc. ACL*, 2005.

R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer, 1998.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proc. EMNLP-CoNLL*, 2007.

F. J. Och and H. Ney. Improved statistical alignment models. In *Proc. ACL*, 2000.

F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003. ISSN 0891-2017.

A. Pauls, J. Denero, and D. Klein. Consensus training for consensus decoding in machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1418–1427, Singapore, 2009. Association for Computational Linguistics.

N. Quadrianto, J. Petterson, and A. Smola. Distribution matching for transduction. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1500–1508. MIT Press, 2009.

C. Quirk, A. Menezes, and C. Cherry. Dependency treelet translation: syntactically informed phrasal smt. In *Proc. ACL*, 2005.

M. Rogati, S. McCarley, and Y. Yang. Unsupervised learning of arabic stemming using a parallel corpus. In *Proc. ACL*, 2003.

D. Rosenberg and P. Bartlett. The rademacher complexity of co-regularized kernel classes. In *Proc. AI Stats*, 2007.

E. F. Tjong Kim Sang and S. Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proc. CoNLL and LLL*, 2000.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proc. HLT-NAACL*, 2003.

L. Shen, J. Xu, and R. Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. ACL*, 2008.

K. Simov, P. Osenova, M. Slavcheva, S. Kolkovska, E. Balabanova, D. Doikoff, K. Ivanova, A. Simov, E. Simov, and M. Kouylekov. Building a linguistically interpreted corpus of bulgarian: the bultreebank. In *Proc. LREC*, 2002.

V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proc. ICML*, 2005.

A. Smith, T. Cohn, and M. Osborne. Logarithmic opinion pools for conditional random fields. In *Proc. ACL*, 2005.

B. Snyder and R. Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proc. ACL*, 2008.

B. Snyder, T. Naseem, J. Eisenstein, and R. Barzilay. Adding more languages improves unsupervised multilingual part-of-speech tagging: a bayesian non-parametric approach. In *Proc. NAACL*, 2009.

J. Tiedemann. Building a multilingual parallel subtitle corpus. In *Proc. CLIN*, 2007.

K. Toutanova, D. Klein, C. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. HLT-NAACL*, 2003.

P. Tseng. An analysis of the EM algorithm and entropy-like proximal point methods. *Mathematics of Operations Research*, 29(1):27–44, 2004.

Y. Tsuruoka and J. Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proc. HLT-EMNLP*, 2005.

L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8: 189–201, 1979.

S. Vogel, H. Ney, and C. Tillmann. Hmm-based word alignment in statistical translation. In *Proc. COLING*, 1996.

H. Yamada and Y. Matsumoto. Statistical dependency analysis with support vector machines. In *Proc. IWPT*, 2003.

D. Yarowsky and G. Ngai. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proc. NAACL*, 2001.