Empirical Risk Minimization of Graphical Model Parameters Given Approximate Inference, Decoding, and Model Structure

Veselin Stoyanov, Alexander Ropson and Jason Eisner
Johns Hopkins University, Center for Language and Speech Processing

Motivation

In practice, Probabilistic Graphical Models are used with several approximations:
- •Mis-specified model structure
- •MAP training
- •Approximate inference
- •Approximate decisions ("decoding")

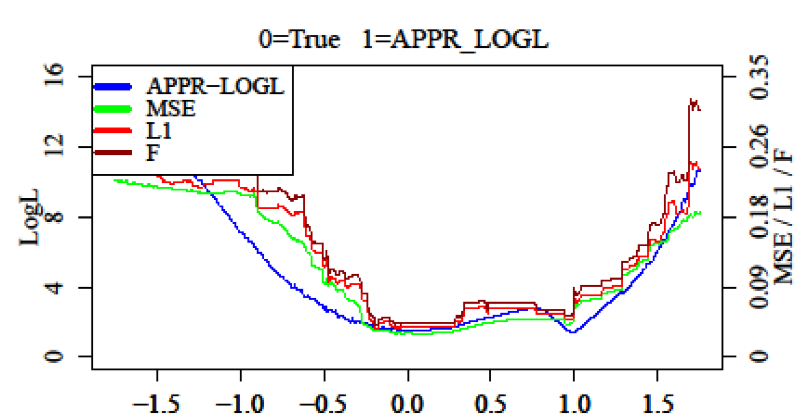
How to learn in the presence of these approximations? We could use the same equations as in the exact case, and plug in approximate inference. However, doing this is not theoretically sound:
- •It can lead to degenerate settings and divergence of the learner (Kulesza and Pereira, 2008)
- •In the presence of approximations, it can be beneficial to learn an inconsistent model (Wainwright, 2006)
- •It can be beneficial to calibrate the learned parameters with respect to loss on the decision task (Lacoste-Julien et al.,2011)
- •Even when exact inference is tractable, exact loss-based decoding may not be

Our Approach

Consider the whole system (approximations and all) to be a decision rule tuned by parameters ϴ.

Use the following method to find ϴ that minimizes **the empirical risk** (like minimizing error in a neural network):
- •Compute gradients of ϴ with respect of the loss using back propagation.
- •Use a local optimization method such as Stochastic Meta Descent (Schradoulph, 1999) to minimize loss on training data
- •In practice, we also use pre-training and a continuation method to deal with non-convexity

Objective Function Landscape



Experiments

Synthetic data (this paper):

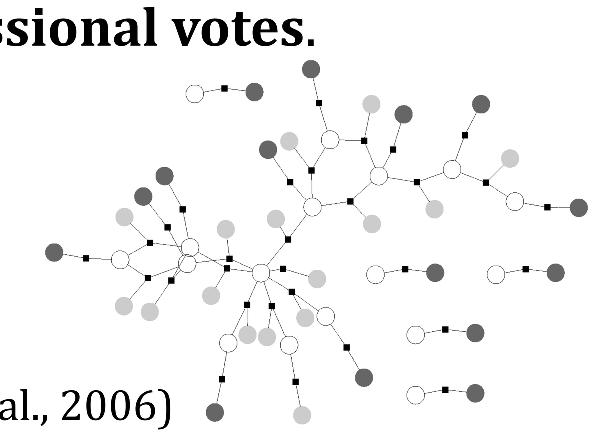
Shows significant improvements across a controlled *range* of conditions (see bottom). 12 randomly generated CRFs with known structure and parameters. (Up to 200 binary input/output/latent variables; Erdos-Renyi random topology; parameters sampled from a Gaussian.) Train and test using different loss functions: L1, MSE, F-score.

Real-world loopy graphs (follow-up NLP paper):

Jointly modeling congressional votes.

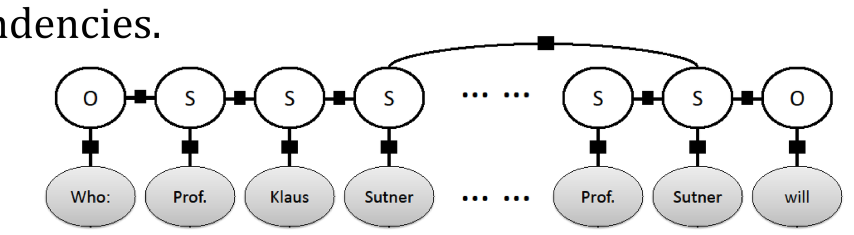
Binary variables for the votes.
Conditioned on:
- •Text of the speeches the representative gave
- •Context of his/her references to other representatives

The ConVote corpus (Thomas et al., 2006)



Info Extraction from Semi-Structured Text

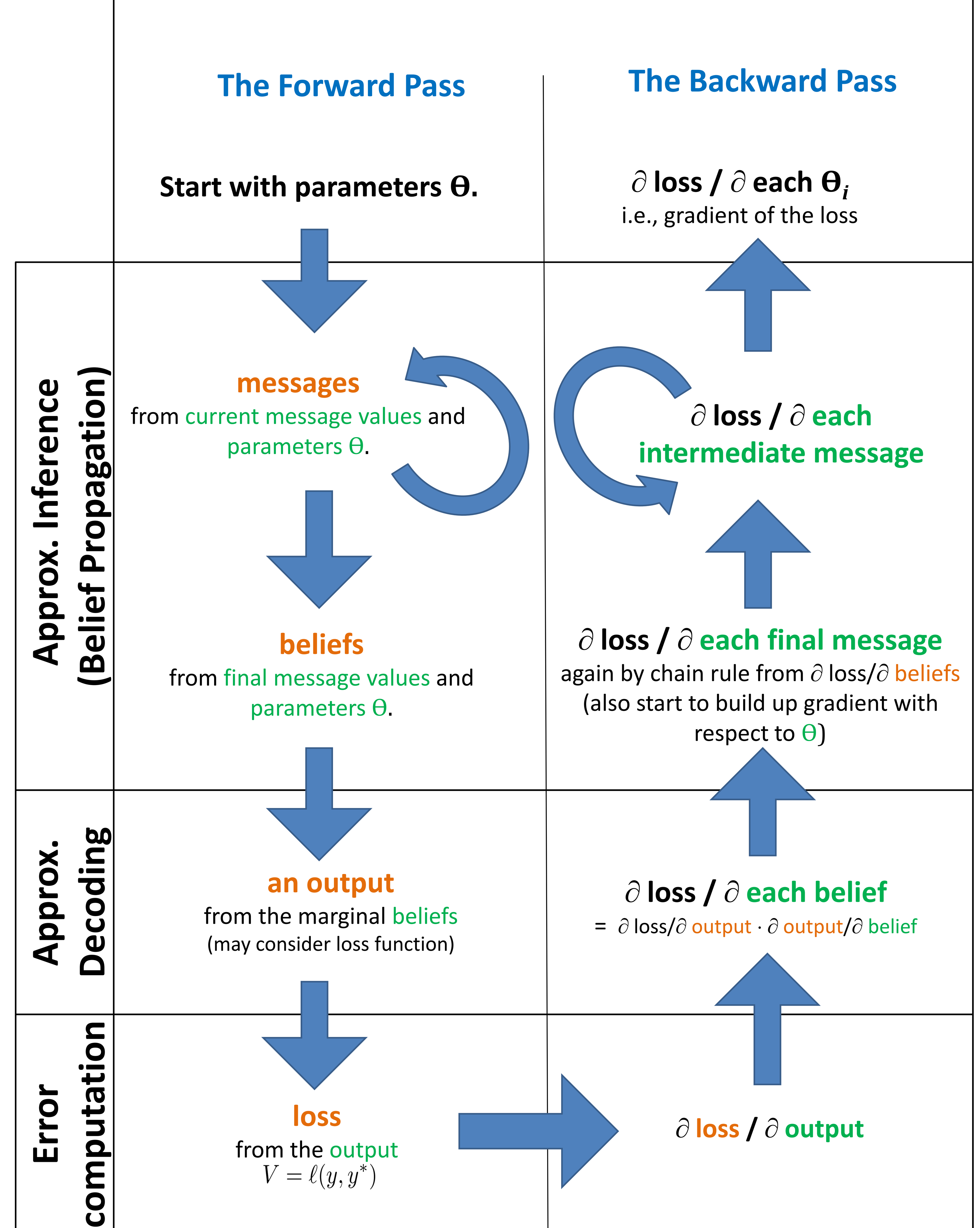
Extract *speaker, location, start time* and *end-time* from seminar announcement emails (Freitag, 1999). Skip-chain CRF models non-local dependencies.



Collective Multi-Label Classification

Assign multiple labels to each document. Use a fully connected CRF with binary edges to model label dependencies.
We use Reuters corpus, version 2 (Lewis et al., 2004).

The Error-Back Propagation Algorithm



Notes:
Each step of the backward pass, also uses intermediate quantities, such as the beliefs, and intermediate BP messages. Thus, we need to record BP messages sent at each time step.

Our algorithm does not require that BP is run to convergence.

The paper includes equations for computing the gradients. It also includes equations for efficiently computing Hessian vector products needed by Stochastic Meta Descent.

Time complexity of the backward pass is similar to the complexity of the forward pass.

Experimental Results

Synthetic Data

Training for Different Losses

test setting	train setting	Δloss	wins
frac-MSE (.04610)	APPR-LOGL	.00710	
	frac-MSE	.00482	5·0·7
	frac-MSE-in	**.00057**	12·0·0
int-F (.06425)	APPR-LOGL	.01170	
	int-F-hyb	.00411	7·0·5
	int-F-in	**.00115**	10·1·1
	int-F-hyb-in	**.00081**	11·0·1
int-L1 (.06385)	APPR-LOGL	.00751	
	int-L1-hyb	.00398	5·1·6
	int-L1-in	**.00137**	10·2·0
	int-L1-hyb-in	**.00079**	10·2·0
APPR-LOGL	APPR-LOGL	-.31618	

Training with Wrong Model Structure

test setting	train setting	Perturbation 10%	20%	30%	40%
frac-MSE	APPR-LOGL	.00352	.00642	.00622	.01118
	frac-MSE	**.00101** 12·0·0	**.00316** 11·0·1	**.00312** 11·0·1	**.00534** 10·0·2
int-F	APPR-LOGL	.01042	.01928	.01026	.02123
	int-F	**.00095** 11·0·1	**.00472** 10·1·1	**.00473** 11·0·1	**.00969** 9·0·3
int-L1	APPR-LOGL	.00452	.00748	.00569	.01173
	int-L1	**.00147** 9·2·1	**.00442** 9·0·3	.00602 9·0·3	.00945 9·0·3
APPR-LOGL	APPR-LOGL	-.3096	-.0180	-.0373	-.1169

Training for Poor BP Approximation

test setting	train setting	Num. of BP iterations 100	30	20	10
frac-MSE	APPR-LOGL	.00710	.00301	.00816	.02461
	frac-MSE	**.00057** 12·0·0	**.00072** 11·0·1	**.00063** 12·0·0	**.00064** 12·0·0
int-F	APPR-LOGL	.01170	.00476	.01276	.03085
	int-F	**.00081** 11·0·1	**.00126** 12·0·0	**.00058** 10·1·1	**.00091** 11·0·1
int-L1	APPR-LOGL	.00751	.00344	.01087	.02984
	int-L1	**.00079** 10·2·0	**.00101** 10·0·2	**.00078** 10·2·0	**.00096** 12·0·0
APPR-LOGL	APPR-LOGL	-.3161	-.1823	-.2422	-.1104

Real-World Data

Congressional Votes

Method	Accuracy
Majority baseline	58.37
supp-opp baseline	62.67
Thomas et al. (2006)	71.25
Greene (2007)	74.19
CRF models	
APPR-LOGL	**79.42**
LOSS-BASED	**84.42**

Multi-Label Classification

Method	Accuracy	F-measure
MaxEnt	96.32	81.62
CRF	96.42	84.04
CRF-Accuracy	**96.50**	83.20
CRF-F-measure	**96.50**	**84.60**

Information Extraction

Method	F-measure spkr	loc	stime	etime	combined
CRF	77.64	87.44	95.21	92.96	**87.25**
CRF-F	78.17	88.36	95.21	92.96	**87.60**
SC-CRF	84.68	89.68	96.80	96.80	**90.99**
SC-CRF-F	85.99	90.62	96.84	96.80	**91.67**