Brief Summary of work on Constrained Conditional Models (CCMs) done by the Cognitive Computation Group (as of Spring 2010)

This note summarizes only work done in CCG; the tutorial discusses a lot of work by other authors that make use of this model.

Our first work on this framework was done by Scott Yih and Dan Roth in CoNLL'04, in the context of jointly working on Entities and Relations; the comparison is between pipeline modeling and inference with global constraints and it shows the gain from using global constraints.
The CoNLL’04 paper is on ``A Linear Programming Formulation for Global Inference in Natural Language Tasks”. A longer version of this paper appears in the Statistical Relational Learning collection: Global Inference for Entity and Relation Identification via a Linear Programming Formulation.

A series of works has shown how to use CCMs for Semantic Role Labeling. Starting with CoNLL'05, CONLING'05 and then IJCAI'05.

A longer version of this paper is in Computational Linguistics: ``The importance of Syntactic Parsing and Inference in Semantic Role Labeling.”

In additional to explaining how to use the global inference for SRL, global inference is used there also to combine output of several systems (based on difference syntactic parses).
Note that all the applications mentioned so far, the CCM was completely decomposed to classifiers f(x), no dependence on y.

Training paradigms:

An IJCAI'05 paper on ``Learning and Inference over Constrained Output” studies two training paradigms that we call L+I (learning+inference) and IBT (Inference based training) and shows (some bounds and experimental results) that while IBT (joint training) is best in the limit, under some conditions (basically, "good" components") L+I generalizes better.
On the same line as in the previous point, an ICML'05 paper on ``Integer Linear Programming Inference for Conditional Random Fields” shows how to train CRFs using the CCM formulation and shows experimentally the advantages of L+I.

More training issues are studied in an ACL'07 paper on ``Guiding Semi-Supervision with Constraint-Driven Learning” and an ICML'08 workshop paper on ``Constraints as Prior Knowledge”. This work introduces Constraints Driven Learning (CODL) and focuses on semi-supervised learning; the learned model is HMM, but similar training issues (L+I and IBT) are discussed, with results that are in agreement with previous observations.

The work on CCM so far is summarized in a AAAI'08 Nectar paper: ``Learning and Inference with Constraints”.

More training issues are discussed in an AI&Stat'09 paper, ``Sequential Learning of Classifiers for Structured Prediction Problems” where we look at pipelines, or "sequential learning", and propose a sequential-joint training paradigm.

Using CCMs with a latent layer is done in a recent NAACL’09 paper in the context of an unsupervised model for transliteration: ``Unsupervised Constraint Driven Learning for Transliteration Discovery”.

Semi-Supervised Training and Training with Indirect Supervision:

The Constrains Driven Learning (CODL) paradigm deals with semi-supervised learning guided by constraints. In recent work we have introduced the notion of **Indirect Supervision.** We show how to improve structured prediction using an easy to get “companion” binary supervision signal (Structured Output Learning with Indirect Supervision, ICML’10) and how to improve on binary classification by learning a latent structure for which we do not get supervision (Discriminative Learning over Constrained Latent Representations, NAACL’10). In both cases the CCM supports joint learning of the target task and the intermediate structured representation, and inference in the intermediate level is supported by incorporating expressive declarative constraints.
In Driving Semantic Parsing from the World's Response (CoNLL’10) we provide another instance of Indirect Supervision, and develop joint learning algorithm that get supervision directly from the response to the model’s actions in the world.

Learning based Java:

The most recent version of our programming language for CCMs, Learning Based Java (LBJ), is described in an LREC’10 paper “Learning Based Java for Rapid Development of NLP Systems”. An updated version is under progress; you can see and download it at: http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=LBJ