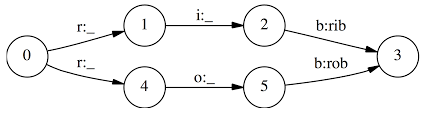
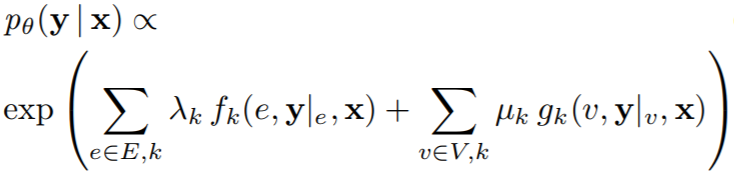
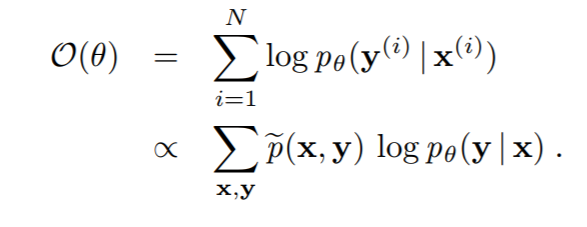
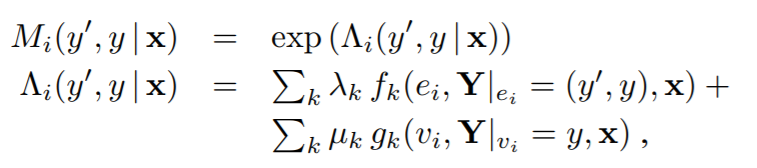
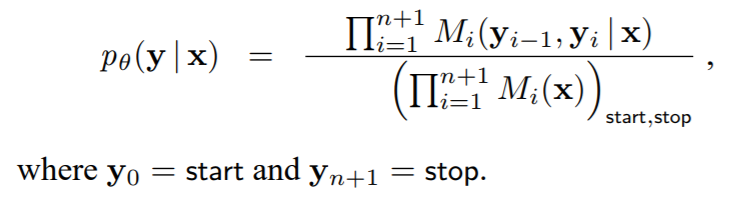
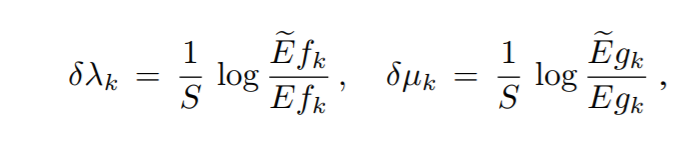
**Paper Review: “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”**

* CRF: a framework for building probabilistic models to segment and label sequence data
* Unlike HMM, can relax strong independence assumptions made in those models amongst observations
* Unlike MEMMs, tends to not be biased towards states w/ fewer successor states
* HMM / stochastic grammars = generative models
  + Assigning joint prob to paired observation and label sequences
  + Parameters trained to maximize joint likelihood of training examples
  + Not practical to represent multiple interacting features or long-range dependencies of observations (inference prob for these is intractable)
* Conditional model is alternative that specifies the probabilities of possible label sequences given observation sequence
  + Does not expend model effort on observations
  + Probs of label sequence can depend on arbitrary non-independent features of observation sequence w/o forcing to model to account for distribution of these dependencies
* Prob of transition between labels may depend not only on current observations, but also on past and future observations
  + In contrast, generative models make strict independence assumptions on the observations
* In Maximum Entropy Markov Models (MEMMs), each source state has an exponential model that takes the observation features as input and outputs a distribution over possible next states
  + MEMM increased recall and precision relative to HMMs in FAQ segmentation task
  + Non-generative, finite-state model
* MEMMs have the **label-biased problem**
  + The transitions leaving a given state compete only against each other, rather than against all other transitions in the model
* Implies a “conservation of score mass”
  + All the mass that arrives at a state must be distributed among the possible successor states
  + Observation affects the next destination state, but not how much mass it would get
  + Hence, bias towards states w/ fewer outgoing transitions
  + State w/ single outgoing transition ignores the observation
* CRFs: all advantages of MEMMs but solves the label bias problem
* MEMMs use per-state exponential models for the conditional probabilities of net stages given current state, while a CRF has a single exponential model for the join probability of the entire sequence of labels given the observation sequence
  + Therefore weights of different features at different states can be traded off against each other
* CRF: finite state model w/ **unnormalized** transition probabilities
* Trained by ML or MAP and the loss function in convex
* CRFs perform better than HMMs and MEMMs when true data distribution has higher-order dependencies than the model, as is often the case in practice
* In simple finite state model, states w/ low-entropy next state distributions (e.g. 3 states of prob 0.981, 0.01, 0.01) take little notice of observations
  + W/ just one state, effectively ignore the observations
* 
* Note the ‘o:\_’ is not generating the observation but instead only conditioning on it
* Given observation sequence ‘r i b’, equal chance of going to ‘1’ or ‘4’ and from here only choices for each are ‘I’ or ‘b’, respectively, so equal chance of predicting ‘rob’ or ‘rib’
* If one of ‘rib’ or ‘rob’ is more popular in training set, the transitions out of the start state will slightly prefer its corresponding transition and that word’s state sequence will always win
* One solution is to collapse the states ‘1’ and ‘4’ and delay branching until we get to a discriminating observation
* Proper solutions require models that account for whole state sequences at once by letting some transitions ‘vote’ more strongly than others depending on corresponding observations
  + This is where the weighting of feature functions come in
  + Hence, score mass is not conserved and individual transitions can amplify or dampen the mass they receive
* In above example, transitions from start state would have very weak effect on path score (i.e. low weighting on feature vectors at that time step), while transitions from ‘1’ and ‘4’ would have stronger effects depending on observations (where ‘w’ depends on this next time step increment and also which label sequence it is computing) and proportionally higher contribution to selection of Viterbi path
* X = data sequences to be labelled (observations), Y = label sequences, Yi of Y are in a range of finite label alphabet ‘y’ (e.g. X might range over NL sentences, while Y might range over Po Stages)
* P(Y|X): discriminative framework from paired observations and label sequences; does not explicitly model the marginal P(X)
* Let G=(V,E) be a graph such that Y=(Yv), v belongs to V, so that Y is indexed by the vertices of G; then (X, Y) is a conditional random field in the case, when conditioned on X, the random variables ‘Yv’ obey the Markov property with respect to the graph:
  + P(Yv | X, Yw, w != v) = P(Yv | X, Yw, w ~ v), where w ~ v means that ‘w’ and ‘v’ are neighbours in G
* CRF is random field globally conditioned on the observation X
* Joint distribution over the label sequence Y given X has the form:
  + 
  + Where X = data sequence, y = label sequence, and y|s = set of components of ‘y’ associated w/ vertices in subgraph ‘s’
* Param estimation problem is to determine parameters:
  + ϴ = (λ1, λ2, …; μ1, μ2, …) from train data D = {(x(i), y(i))}i=1Nw/ empirical distribution p~(x,y)
* Log likelihood objective function
  + 
* CRFs much more expressive than HMM-like models (which it encompasses) because it allows arbitrary dependencies of the observation sequence
* Suppose pϴ(Y|X) is a CRF
  + For each position ‘I’ in the observation sequence ‘X’, we define the |y| x |y| (where |y| = # of possible labels) matrix random variable Mi(x) = [Mi(y; y|x)] by:
  + 
  + Where ‘ei’ is the edge with labels (Yi-1, Yi) and vi is the vertex with label Y­i
* CRFs don’t need to enumerate over all possible sequences ‘x’ and therefore these matrices can be computed directly as needed from a given train or test observation sequence ‘x’ and parameter vector ‘ϴ’
* Normalization Zϴ(x) is the (start, stop) entry of the product of these matrices:
  + Zϴ(x) = (Mi(x), M2(x), …., M­n+1(x))start, stop
* Conditional prob of the label sequence ‘y’ is written as:
  + 
* Iterative scaling algorithms updates the weights as:
  + Λk 🡨 λ­k + δλk and μk 🡨 μk + δμk
* Update equations are:
  + 
* The constant ‘S’ can be quite large since in practice it is proportional to the length of the longest training observation sequence
  + Hence, algorithm may converge slowly w/ small steps
* Single iteration of update algorithms has roughly the same time/space complexity as well known Baum-Welch algorithms for HMMs
* The results of papers’ experiments indicate clearly that even when the models are parameterized in exactly the same way, CRFs are more robust to inaccurate modelling assumptions than MEMMs or HMMs and resolves the label bias problem
* Experiment 1: data is generated from simple HMM and train both an MEMM and a CRF w/ same topologies on the data generate by the HMM
* W/ 2000 train and 500 test samples trained to convergence, CRF error = 4.6% and MEMM error = 42%
  + Y = {a-e}, |Y| = 5
  + X = {A-Z}, |X| = 26
* Note: using 2nd order HMM, so current label yi depends on both yi-1 and yi-2, and current observation xi depends on both yi and xi-1
* For each randomly generated model, a sample of 1000 sequences of length 25 is generated for training and testing
* MEMM takes ~100 iterations to stabilize, while CRF takes ~500
* Experiment 2: trained HMM, MEMM, CRF models as in the synthetic data experiments with PoS tagging, where each word in a given input sequence must be labelled with one of 45 syntactic tags
* HMM outperforms the MEMM as a consequence of the label bias problem, while the CRF outperforms the HMM
* Also introduces small set of orthographic features (e.g. whether word begins w/ number of upper case letter, contains a hyphen, and whether it ends in a suffix)
* Both MEMM and CRF benefit significantly from use of these features
* Can also use the optimal MEMM parameter vector as a starting point for training corresponding CRF
  + Overcomes the slow start of CRF
* CRFs can be trained using the exponential loss objective function used by the AdaBoost algorithm
* One can implement efficient feature selection and feature induction algorithms for CRFs
  + Rather than specifying in advance which features of (X, Y) to use, we could start from feature-generating rules and evaluate the benefit of generated features automatically in data
* CRFs combine benefits of conditional models w/ global normalization of random field models
* CRF offer unique combination of properties
  + Discriminatively trained models for sequence segmentation and labelling
  + Combination of arbitrary, overlapping, and agglomerative observation features from both past and future
  + Efficient training + decoding based on dynamic programming
  + Parameter estimation guaranteed to find global optimum

**Significant Points and Takeaways from Paper**

* Better than HMMs in that:
  + Can relax strong independence assumptions w.r.t. the observation sequence
  + CRFs don’t need to enumerate over all possible sequences
  + Much more expressive than HMM-like models (which it encompasses) because it allows arbitrary dependencies of the observation sequence
* Better than MEMMs in that:
  + Don’t suffer from label bias problem
  + Tends to not be bias towards states w/ fewer successor states
  + CRF has single exponential model for joint prob of entire sequence of labels, while MEMMs use per-state exponential models for the conditional probabilities of net stages given current state
* CRFs perform better than HMMs and MEMMs when true data distribution has higher-order dependencies than the model, as is often the case in practice
* Trained by ML or MAP and the loss function in convex
  + Can also use the optimal MEMM parameter vector as a starting point for training corresponding CRF
  + Overcomes the slow start of CRF
* One can implement efficient feature selection and feature induction algorithms for CRFs
  + Rather than specifying in advance which features of (X, Y) to use, we could start from feature-generating rules and evaluate the benefit of generated features automatically in data
* The results of papers’ experiments indicate clearly that even when the models are parameterized in exactly the same way, CRFs are more robust to inaccurate modelling assumptions than MEMMs or HMMs and resolves the label bias problem
* CRFs combine benefits of conditional models w/ global normalization of random field models