Tutorial on Conditional Random Fields for Sequence Prediction

Ariadna Quattoni
RoadMap

- Sequence Prediction Problem
- CRFs for Sequence Prediction
- Generalizations of CRFs
- Hidden Conditional Random Fields (HCRFs)
- HCRFs for Object Recognition
RoadMap

- Sequence Prediction Problem
- CRFs for Sequence Prediction
- Generalizations of CRFs
- Hidden Conditional Random Fields (HCRFs)
- HCRFs for Object Recognition
Sequence Prediction Problem

Example: Part-of-Speech Tagging

He reckons the current account deficit will narrow significantly

\[ [\text{PRP} \ \text{VB} \ \text{DT} \ \text{JJ} \ \text{NN} \ \text{NN} \ \text{MD} \ \text{VB} \ \text{RB}] \]

\[ [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9] \]
Gesture Recognition

X →

Y → [HTF] [HTF] [HTF] [HOF] [HOF] [HOS]
RoadMap

- Sequence Prediction Problem
- CRFs for Sequence Prediction
- Generalizations of CRFs
- Hidden Conditional Random Fields (HCRFs)
- HCRFs for Object Recognition
Conditional Random Fields: Modelling the Conditional Distribution

Model the Conditional Distribution:

$$P(y | x)$$

To predict a sequence compute:

$$y^* = \arg \max_y P(y | x)$$

Must be able to compute it efficiently.
Conditional Random Fields: Feature Functions

\[ f_j(y_{i-1}, y_i, x, i) \]
Feature Functions

Express some characteristic of the empirical distribution that we wish to hold in the model distribution

\[ f_j(y_{i-1}, y_i, x, i) \]

1 \( if \) \( y_{i-1} = IN \) \text{ and } \( y_i = NNP \) \text{ and } \( x_i = September \)

0 \( otherwise \)
Label sequence modelled as a normalized product of feature functions:

\[
P(y \mid x, \lambda) = \frac{1}{Z(x)} \exp \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(y_{i-1}, y_i, x, i)
\]

\[
Z(x) = \sum_{y \in Y} \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(y_{i-1}, y_i, x, i)
\]

The model is log-linear on the Feature Functions.
Parameter Estimation: Maximum Likelihood

IID training samples:

\[
D = \left[(x_1^1, y_1^1), (x_2^1, y_2^1), \ldots, (x_m^1, y_m^1)\right]
\]

(negative) Conditional Log-Likelihood:

\[
L(\lambda, D) = -\log \left( \prod_{k=1}^{m} P(y^k | x^k, \lambda) \right)
\]

\[
= -\sum_{k=1}^{m} \log \left[ \frac{1}{Z(x_m)} \exp \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(y^k_{i-1}, y^k_i, x^m, i) \right]
\]
Parameter Estimation: Maximum Likelihood

Maximum Likelihood Estimation

Set optimal parameters to be:

$$\lambda^* = \arg\min_{\lambda} L(\lambda, D) + C \frac{1}{2} \|\lambda\|^2$$

This function is convex, i.e. no local minimums
Parameter Estimation: Optimization

Let: \( F_j(y, x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i) \)

Differentiating the log-likelihood with respect to parameter \( \lambda_j \)

\[
\frac{\partial L(\lambda, D)}{\partial \lambda_j} = -\frac{1}{m} \sum_{k=1}^{m} F_j(y^{k}, x^{k}) + \sum_{k=1}^{m} E_{P(y|x^{k}, \lambda)}[F_j(y, x^{k})]
\]

Observed Mean Feature Value  
Expected Feature Value Under The Model
Parameter Estimation: Optimization

Generally, it is not possible to find an analytic solution to the previous objective.

Iterative techniques, i.e. gradient based methods.
Maximum Entropy Interpretation

Notice that at the optimal solution of:

\[ \lambda^* = \arg \min_{\lambda} \ L(\lambda, D) + C \frac{1}{2} \|\lambda\|^2 \]

We must have that:

\[ \frac{1}{m} \sum_{k=1}^{m} F_j(y^k, x^k) = \sum_{k=1}^{m} E_{P(y|x^k, \lambda)} \left[ F_j(y, x^k) \right] \]

Maximizing log-likelihood \: \approx \: \text{Finding max-entropy distribution that satisfies the set of constraints defined by the feature functions}
CRF’s Inference

Given a model, i.e. parameter values

Can we compute the following efficiently?

**Best Label Sequence**

\[ y^* = \arg \max_y P(y | x, \lambda^*) \]

**Expected Values**

\[ \sum_{k=1}^{m} E_{P(y|x^k, \lambda)} [F_j(y, x^k)] = \sum_{k=1}^{m} \sum_y p(y | x^k, \lambda) F_j(y, x^k) \]

\[ = \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{y : y_{i-1} = a, y_i = b} p(y_{i-1} = a, y_i = b | x^k, \lambda) f_j(a, b, x^k, i) \]

Both can be computed using dynamic programming
RoadMap

- Sequence Prediction Problem
- CRFs for Sequence Prediction

Generalizations of CRFs

- Hidden Conditional Random Fields (HCRFs)
- HCRFs for Object Recognition
Predicting Trees: Application Constituent Parsing

$$P(y \mid x, \lambda) = \frac{1}{Z(x)} \exp \sum_{\langle A \rightarrow BC, i, l, k \rangle \in y} \sum_j \lambda_j f_j (A \rightarrow BC, i, l, k, x)$$
Generalization II: Factorized Linear Models

To predict a sequence compute:

\[ y^* = \arg \max_y \frac{1}{Z(x)} \exp \sum_{i=1}^{n} \sum_j \lambda_j f_j (y_{i-1}, y_i, x, i) \]

\[ = \arg \max_y \sum_{i=1}^{n} \sum_j \lambda_j f_j (y_{i-1}, y_i, x, i) \]

Objective: making accurate predictions on unseen data

The parameters of the linear model can be optimized using other loss functions
Generalization II: Factorized Linear Models

Structured Hinge Loss

Let \( z \) be the correct label sequence:

\[
l(x, z, \lambda) = \begin{cases} 
0 & \text{if} \quad \sum_{i=1}^{n} \sum_{j} \lambda_{j} f_{j}(z_{i-1}, z_{i}, x, i) > \arg \max_{y \neq z} \sum_{i=1}^{n} \sum_{j} \lambda_{j} f_{j}(y_{i-1}, y_{i}, x, i) + 1 \\
\arg \max_{y \neq z} \sum_{i=1}^{n} \sum_{j} \lambda_{j} f_{j}(y_{i-1}, y_{i}, x, i) - \sum_{i=1}^{n} \sum_{j} \lambda_{j} f_{j}(z_{i-1}, z_{i}, x, i) - 1 & \text{otherwise}
\end{cases}
\]

Structured SVM

\[
\lambda^{*} = \arg \min_{\lambda} \sum_{k=1}^{m} l(x^{k}, y^{k}, \lambda) + C \frac{1}{2} \| \lambda \|^2
\]
RoadMap

- Sequence Prediction Problem
- CRFs for Sequence Prediction
- Generalizations of CRFs
- Hidden Conditional Random Fields (HCRFs)
- HCRFs for Object Recognition
Hidden Conditional Random Fields

Sentiment Detection

\[
\begin{bmatrix}
  x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12}
\end{bmatrix}
\]

This movie greatly appealed to me for many reasons - I loved it

+1 Positive Review

As dumb as history gets

-1 Negative Review
Hidden Conditional Random Fields

Object Recognition

\[ G(V, E) \]

A training sample \[ (x^i, y^i, G^i) \]
Hidden Conditional Random Fields

Model the conditional probability: \( P(y \mid x, G) \)

We introduce hidden variables: \( h = \{h_1, h_2, \ldots, h_m\} \quad h \in H \)

Analogous to the standard CRF we define:

\[
P(y, h \mid x, G, \lambda) = \frac{\exp^{\psi(y, h, x, G, \lambda)}}{\sum_{y', h} \exp^{\psi(y', h, x, G, \lambda)}}
\]

\[
P(y \mid x, G, \lambda) = \sum_h P(y, h \mid x, G, \lambda) = \frac{\sum_h \exp^{\psi(y, h, x, \lambda)}}{\sum_{h, y'} \exp^{\psi(y', h, x, \lambda)}}
\]

\( \psi(y, h, x, G, \lambda) \) Maps a configuration to the reals.
Hidden Conditional Random Fields

Feature Functions

\[ \psi(y, h, x, G, \lambda) = \sum_{k \in V} \sum_{j} \lambda^1_j f^1_j(k, y, h_k, x) + \sum_{(k,l) \in E} \sum_{j} \lambda^2_j f^2_j(k, l, y, h_k, h_l, x) \]
Parameter Estimation

Maximum Likelihood:

\[ L(\lambda, D) = -\sum_i \log P(y^i \mid x^i, G^i, \lambda) \]

Find optimal parameters:

\[ \lambda^* = \arg \min_{\lambda} L(\lambda, D) + C \frac{1}{2} \|\lambda\|^2 \]

Iterative techniques, i.e. gradient based methods.
But now the function is not convex!!!

At test time make prediction:

\[ y^* = \arg \max_y P(y \mid x, G, \lambda^*) \]
The derivative of the loss function is given by:

\[
\frac{\partial L_i(x^i, G^i, y)}{\partial \lambda_j} - \sum_{y \in Y, k \in V^i, a \in H} P(h_k = a, y | x^i, G^i, \lambda) f_j^1(k, y, a, x^i) + \sum_{k \in V^i, a \in H} P(h_k = a | y^i, x^i, G^i, \lambda) f_j^1(k, y', a, x^i)
\]

The derivative can be expressed in terms of components:

\[
P(h_j = a | x, G, \lambda) \quad P(h_k = a, h_l = b | x, G, \lambda) \quad P(y | x, G, \lambda)
\]

that can be calculated using dynamic programming. Similarly the argmax can also be computed efficiently.
RoadMap

- Sequence Prediction Problem
- CRFs for Sequence Prediction
- Generalizations of CRFs
- Hidden Conditional Random Fields (HCRFs)
- HCRFs for Object Recognition
Application :: Object Recognition

SemiSupervised Part-based Models

\[ x = \{ x_1, \ldots, x_m \} \]

\[ \phi(x_i) \in \mathbb{R}^d \]
Motivation

- Use a discriminative model.
- Spatial dependencies between parts.
- It is convenient to use an intermediate discrete hidden variable.
- Potential of learning semantically-meaningful parts.
- Framework for investigating which part structures emerge.
Graph Structure

$h_i \rightarrow G(V, E)$
Feature Functions

$G(V,E)$ is a minimum spanning tree. Weight $(i,j) = \text{distance between patches } x_i \text{ and } x_j$

Compatibility between a pair of part labels and a category

Compatibility between a part label and a category

Compatibility between a patch and a part label

$f^2(j,k,y,h_j,h_k,x,\theta) = \theta(y,h_j,h_k)$

$f^1(j,y,h_j,x,\theta) = \phi(x_j) \bullet \theta(h_j) + \theta(y,h_j)$

obtained with Lowe’s detector (textured regions)
SIFT features (describes the texture of the image region).
Patch description also includes relative location.
Viterbi Configuration
Learning Shape

Car Side Shape Model
Conclusions

- Factorized Linear Models generalize linear prediction models to the setting of structure prediction.

- In standard linear prediction, finding the argmax and computing gradients is trivial. In structure prediction it involves inference.

- Factored representations allow for efficient inference algorithms (most times based on dynamic programming)

- Conditional Random Fields are an instance of this framework

Future Work

- Better Algorithms for training HCRFs