Conditional Random Fields

(SNLP tutorial)

Vilém Zouhar

March 12, 2021

Overview

- Sequence Labelling / Entity Recognition
-     - Rule-based
-     - HMM
-     - Bayesian Network
-     - Log-linear 1st Order Sequential Model
-     - Linear Chain CRF / CRF
- Model comparison
- Code
- Homework

Sequence Labelling / Entity Recognition

- `My name is V. Zouhar, I live in Saarbrücken and my matriculation number is 1234.`

Sequence Labelling / Entity Recognition

- `My name is V. Zouhar, I live in Saarbrücken and my matriculation number is 1234.`
- `My name is [V. Zouhar:person], I live in [Saarbrücken:loc] and my matriculation number is [1234:mat-num].`

Sequence Labelling / Entity Recognition

- `My name is V. Zouhar, I live in Saarbrücken and my matriculation number is 1234.`
- `My name is [V. Zouhar:person], I live in [Saarbrücken:loc] and my matriculation number is [1234:mat-num].`
- NER as Sequence labeling:
  X: sequence of words
  Y: labels {`mat-num, person, location, none`}

Rule-based

- Regex substitute:
  `matriculation (number)? (is)? (\d+)` $\rightarrow$ `[\3:mat-num]`

Rule-based

- Regex substitute:
  `matriculation (number)? (is)? (\d+)` → `[\3:mat-num]`
- Gets out of hand quickly:
  `(am|name (is)?) (.*?) (and|\s[.,?])?` → `[\3:person]`

Rule-based

- Regex substitute:
  `matriculation (number)? (is)? (\d+)` $\rightarrow$ `[\3:mat-num]`
- Gets out of hand quickly:
  `(am|name (is)?) (.*?) (and|\s[.,?])?` $\rightarrow$ `[\3:person]`
- No automated learning

HMM

- Hidden states: {`mat-num`, `person`, `location`, `none`}

HMM

- Hidden states: {`mat-num`, `person`, `location`, `none`}
- Better hidden states: {`mat-num`, `START+person`, `INTERNAL+person`, `END+person`, `location`, `none`, `...`}

HMM

- Hidden states: {`mat-num, person, location, none`}
- Better hidden states: {`mat-num, START+person, INTERNAL+person, END+person, location, none, ...`}
- Transitions: MLE from annotated data

HMM

- Hidden states: {mat-num, person, location, none}
- Better hidden states: {mat-num, START+person, INTERNAL+person, END+person, location, none, ...}
- Transitions: MLE from annotated data
- Emission probabilities: MLE from annotated data (+ smoothing)

HMM

- Hidden states: {`mat-num`, `person`, `location`, `none`}
- Better hidden states: {`mat-num`, `START+person`, `INTERNAL+person`, `END+person`, `location`, `none`, `...`}
- Transitions: MLE from annotated data
- Emission probabilities: MLE from annotated data (+ smoothing)
- $p(x, y) = \prod_i a(y_{i-1}, y_i) \cdot o(y_i, x_i)$

HMM

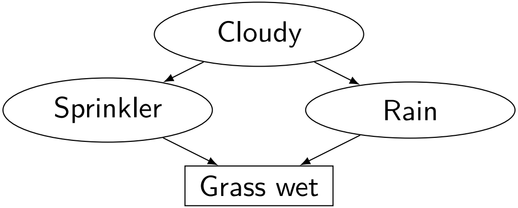
- Hidden states: {`mat-num`, `person`, `location`, `none`}
- Better hidden states: {`mat-num`, `START+person`, `INTERNAL+person`, `END+person`, `location`, `none`, `...`}
- Transitions: MLE from annotated data
- Emission probabilities: MLE from annotated data (+ smoothing)
- $p(x, y) = \prod_i a(y_{i-1}, y_i) \cdot o(y_i, x_i)$
- Optimizes $p(x, y|\theta)$, though we are interested in $p(y|x, \theta)$

Bayesian Network

- DAG, $(x \to y) \in E : y$ dependent on x

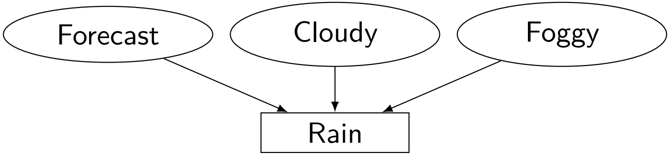
Local Markov Property

Node is conditionally independent of its nondescendants given its parents.
$p(\text{Sprinkler}|\text{Cloudy}, \text{Rain}) = p(\text{Sprinkler}|\text{Cloudy})$

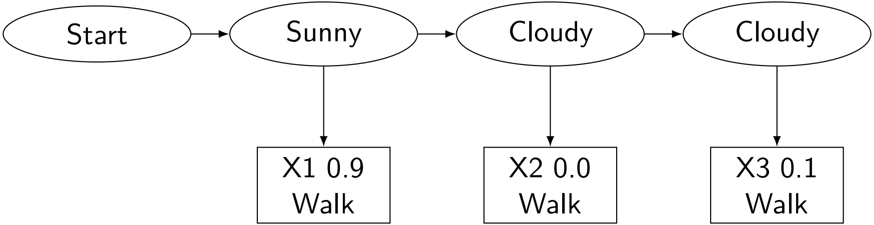


Naïve Bayes

- Assume absolute independence except for the one observed variable
- $p(y = \text{Yes}|x) = p(y_j|x) = \frac{p(x|y_j)p(y_j)}{p(x)} \propto p(x|y_j)p(y_j) \approx p(y_j)\prod_i p(x_i|y_j)$

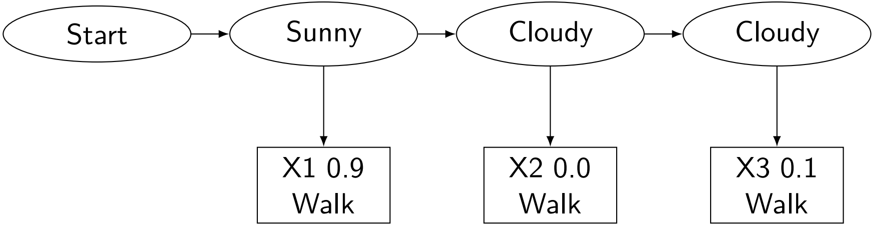


HMM



Sketch of HMM structure
observed variable *Walk duration*, latent variable: $Weather \in \{Sunny, Cloudy\}$

HMM



Sketch of HMM structure
observed variable *Walk duration*, latent variable: $Weather \in \{Sunny, \ Cloudy\}$

$$p(y|x) = \prod_i p(y_i) \cdot o(y_i, x_i) \text{ (Naïve Bayes)}$$

$$\Rightarrow$$

$$p(y|x) = \prod_i a(y_{i-1}, y_i) \cdot o(y_i, x_i) \text{ (HMM)}$$

Logistic Regression

$$p(y|x) = \frac{\exp(\Phi(y,x))}{\sum_{y'} \exp(\Phi(y',x))}$$

$$\arg\max_y \frac{\exp(\Phi(y,x))}{\sum_{y'} \exp(\Phi(y',x))} = \arg\max_y \exp(\Phi(y,x))$$

Log-linear 1st Order Sequential Model

- Sequence of hidden states: y, {`mat-num`, `person`, `location`, `none`}

Log-linear 1st Order Sequential Model

- Sequence of hidden states: y, {`mat-num`, `person`, `location`, `none`}
- Observed sequence of variables: x (words)

Log-linear 1st Order Sequential Model

- Sequence of hidden states: y, `{mat-num, person, location, none}`
- Observed sequence of variables: x (words)
- $p(y|x) \propto \exp\{\sum_j \log a(y_{j-1}, y_j) + \log o(y_j, x_j)\}$

Log-linear 1st Order Sequential Model

- Sequence of hidden states: y, {`mat-num`, `person`, `location`, `none`}
- Observed sequence of variables: x (words)
- $p(y|x) \propto \exp\left\{\sum_j \log a(y_{j-1}, y_j) + \log o(y_j, x_j)\right\}$
- $p(y|x) = \frac{1}{Z(x)} \cdot \exp\left\{\sum_j \log a(y_{j-1}, y_j) + \log o(y_j, x_j)\right\}$

Log-linear 1st Order Sequential Model

- Sequence of hidden states: y, `{mat-num, person, location, none}`
- Observed sequence of variables: x (words)
- $p(y|x) \propto \exp\left\{ \sum_j \log a(y_{j-1}, y_j) + \log o(y_j, x_j) \right\}$
- $p(y|x) = \frac{1}{Z(x)} \cdot \exp\left\{ \sum_j \log a(y_{j-1}, y_j) + \log o(y_j, x_j) \right\}$
- $p(y|x) = \frac{1}{Z(x)} \cdot \prod_j \{ a(y_{j-1}, y_j) o(y_j, x_j) \}$

Log-linear 1st Order Sequential Model

- Sequence of hidden states: y, `{mat-num, person, location, none}`
- Observed sequence of variables: x (words)
- $p(y|x) \propto \exp\left\{\sum_j \log a(y_{j-1}, y_j) + \log o(y_j, x_j)\right\}$
- $p(y|x) = \frac{1}{Z(x)} \cdot \exp\left\{\sum_j \log a(y_{j-1}, y_j) + \log o(y_j, x_j)\right\}$
- $p(y|x) = \frac{1}{Z(x)} \cdot \prod_j \{a(y_{j-1}, y_j) o(y_j, x_j)\}$
- $argmax\ p(y|x) \ldots$

Log-linear 1st Order Sequential Model

Viterbi:

$$argmax\ \ p(y|x) = argmax\ \ \log p(y|x) = argmax\ \ F(y,x) - \log \sum_{y'} \exp F(y',x)$$

$$= argmax\ \ F(y,x)$$

$$\alpha_t(y_j) = \max_i \exp \left(\log \alpha_{t-1}(y_i) + a(y_j,y_i) + o(y_j,x_t) \right)$$

$$\alpha'_t(y_j) = argmax_i\ \ \alpha_{t-1}(y_i) + \exp\left(a(y_j,y_i) + o(y_j,x_t)\right)$$

$O(|Y|^2 \cdot T)$

Log-linear 1st Order Sequential Model

Forward:

$$\log fw_t(y_j) = \log \sum_i \exp\left(\log fw_{t-1}(y_i) + a(y_j, y_i) + o(y_j, x_t) \right)$$

$$Z(X) = \sum_i \exp\left(\log fw_{|T|-1}(y_i) + a(y_j, y_i) + o(y_j, x_t) \right)$$

$$\rightarrow$$

$$p(y|x) = \frac{\alpha_{|T|}(y_{:-1})}{Z(x)}$$

$O(|Y|^2 \cdot T)$

Log-linear 1st Order Sequential Model

- Replace $o(y_j, x_t)$ with $\theta_1 h_1(y_j, x_t) + \theta_2 h_2(y_j, x_t) + \ldots$

Log-linear 1st Order Sequential Model

- Replace $o(y_j, x_t)$ with $\theta_1 h_1(y_j, x_t) + \theta_2 h_2(y_j, x_t) + \ldots$
- Same with $a(y_j, y_i) = \theta'_1 g_1(y_j, y_i) + \theta'_2 g_2(y_j, y_i) + \ldots$

Log-linear 1st Order Sequential Model

- Replace $o(y_j, x_t)$ with $\theta_1 h_1(y_j, x_t) + \theta_2 h_2(y_j, x_t) + \ldots$
- Same with $a(y_j, y_i) = \theta'_1 g_1(y_j, y_i) + \theta'_2 g_2(y_j, y_i) + \ldots$
- Why not just $\sum_{\text{feature } f} \theta_i f_i(y_i, y_j, x_t)$?

Log-linear 1st Order Sequential Model

- Replace $o(y_j, x_t)$ with $\theta_1 h_1(y_j, x_t) + \theta_2 h_2(y_j, x_t) + \dots$
- Same with $a(y_j, y_i) = \theta'_1 g_1(y_j, y_i) + \theta'_2 g_2(y_j, y_i) + \dots$
- Why not just $\sum_{\text{feature } f} \theta_i f_i(y_i, y_j, x_t)$?
- Why not allow $\sum_{\text{feature } f} \theta_i f_i(y_i, y_j, x, t)$?

Model overview

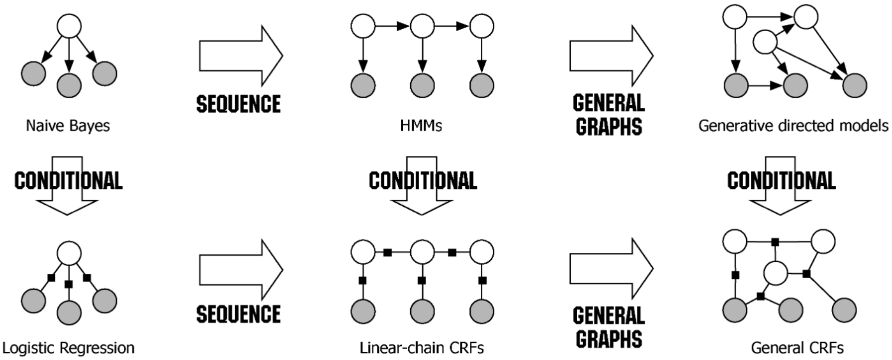


Figure 1: CRF in relation to other models; Source [2]

HMM → Linear Chain CRF

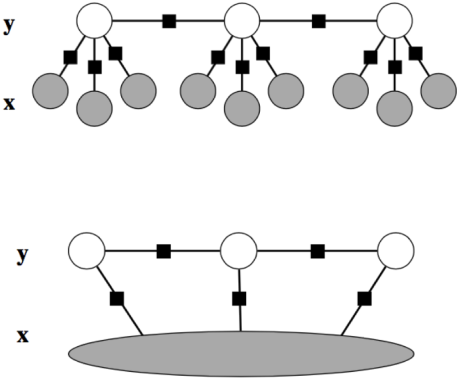


Figure 2: HMM vs. Linear Chain CRF; Source [12]

Model overview

- Multinomial logistic regression:
  $p(y_j|x) = \frac{exp(Z_j \cdot x)}{\sum_i exp(Z_i \cdot x)}$
- Multiclass naïve Bayes:
  $p(y_j|x) = \frac{p(x|y_j)p(y_j)}{p(x)} \propto p(x|y_j)p(y_j) \approx p(y_j) \prod_i p(x_i|y_j)$

Linear Chain CRF

- Sequence of hidden states: y, `{mat-num, person, location, none}`
- Observed sequence of variables: x (words)
- $p(y|x) \propto \prod_t \exp\{\sum_{\text{feature } f} \theta_i f_i(y_{t-1}, y_t, x, t)\}$
- $p(y|x) = \frac{1}{Z(x)} \prod_t \exp\{\sum_{\text{feature } f_i} \theta_i f_i(y_{t-1}, y_t, x, t)\}$
- Features: $f_i(y_{t-1}, y_t, x, t) \geq 0$
- Parameters: θ

Linear Chain CRF - Features

$$f_i(y_{t-1}, y_t, x, t) = \begin{cases} 1 & \text{if } \text{cond}_f(y_{t-1}, y_t, x, t) \\ 0 & \text{else} \end{cases}$$

Linear Chain CRF - Features

$$f_i(y_{t-1}, y_t, x, t) = \begin{cases} 1 & \text{if } \mathrm{cond}_f(y_{t-1}, y_t, x, t) \\ 0 & \text{else} \end{cases}$$

$$f_1(y_{t-1}, y_t, x, t) = \begin{cases} 1 & \text{if } x_{t-2} \text{ is capitalized} \\ 0 & \text{else} \end{cases}$$

$$f_a(y_{t-1}, y_t, x, t) = \begin{cases} 1 & \text{if } y_{t-1} = \texttt{number} \land y_t = \texttt{none} \\ 0 & \text{else} \end{cases}$$

$$\theta_a = a(\text{number}, \texttt{none})$$

$$f_o(y_{t-1}, y_t, x, t) = \begin{cases} 1 & \text{if } y_t = \texttt{number} \land x_t = \texttt{<num>} \\ 0 & \text{else} \end{cases}$$

$$\theta_o = o(\text{number}, \texttt{<num>})$$

Linear Chain CRF - Features

$f_w(y_{t-1}, y_t, x, t) = x_t$ word length
$f_s(y_{t-1}, y_t, x, t) = x_t$ number of non-alphabetic characters

CRF - Operations

Inference:

$$argmax_y \ p(y|x, \theta)$$

CRF - Operations

Inference:

$$argmax_y \ p(y|x,\theta)$$

Decoding:

$$p(y|x,\theta)$$

CRF - Operations

Inference:

$$argmax_y \; p(y|x,\theta)$$

Decoding:

$$p(y|x,\theta)$$

Training:

$$argmax_\theta \; p(y_D|x_D,\theta)$$

Linear Chain CRF - Estimating θ

Gradient descent (ascent):

$$\frac{\partial \log p(y|x,\theta)}{\partial \theta_i} = \sum_{t=1}^{T} f_i(y_{t-1}, y_t, x, t) - \sum_{y'} \sum_{t=1}^{T} f_i(y'_{t-1}, y'_t, x, t) \cdot p(y'|x)$$

$$\theta_f \leftarrow \theta_f + \epsilon \left[\sum_{t=1}^{T} F(y_{t-1}, y_t, x, t) - \sum_{y'} \sum_{t=1}^{T} F(y'_{t-1}, y'_t, x, t) \cdot p(y'|x,\theta) \right]$$

Linear Chain CRF - Estimating θ

Gradient descent (ascent):

$$\frac{\partial \log p(y|x,\theta)}{\partial \theta_i} = \sum_{t=1}^{T} f_i(y_{t-1}, y_t, x, t) - \sum_{y'} \sum_{t=1}^{T} f_i(y'_{t-1}, y'_t, x, t) \cdot p(y'|x)$$

$$\theta_f \leftarrow \theta_f + \epsilon \left[\sum_{t=1}^{T} F(y_{t-1}, y_t, x, t) - \sum_{y'} \sum_{t=1}^{T} F(y'_{t-1}, y'_t, x, t) \cdot p(y'|x,\theta) \right]$$

Limited-memory BFGS (quasi-Newton method)

Linear Chain CRF - Regularization

Objective function:

$\mathcal{L} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta)$

Linear Chain CRF - Regularization

Objective function:

$\mathcal{L} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta)$

LASSO:

$\mathcal{L}_{+lasso} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta) - \lambda_1 \sum_i |\theta_i|$

Linear Chain CRF - Regularization

Objective function:

$\mathcal{L} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta)$

LASSO:

$\mathcal{L}_{+lasso} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta) - \lambda_1 \sum_i |\theta_i|$

Ridge:

$\mathcal{L}_{+ridge} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta) - \frac{\lambda_2}{2} \sum_i \theta_i^2$

Linear Chain CRF - Regularization

Objective function:

$\mathcal{L} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta)$

LASSO:

$\mathcal{L}_{+lasso} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta) - \lambda_1 \sum_i |\theta_i|$

Ridge:

$\mathcal{L}_{+ridge} = \sum_s \log p(y^{(s)}|x^{(s)}, \theta) - \frac{\lambda_2}{2} \sum_i \theta_i^2$

Elastic net:

$\mathcal{L}_{+elastic} = \sum_s \log p(y^{(s)}|x^{(s)}\theta) - \frac{\lambda_2}{2} \sum_i \theta_i^2 - \lambda_1 \sum_i |\theta_i|$

General CRF

- Factorization to maximal clicques.
- Allow access to a whole clicque

Clique

$G = (V, E) \quad C \subseteq V : \forall x, y \in C : (x, y) \in E$

Maximal Clique

$C \subseteq C' \Rightarrow C = C'$

CRF

$p(Y|X) = \frac{1}{Z(X)} \prod_{C \in Y} \Psi_C(X_C)$
$\Psi_C(Y, X) \sum_i \theta_i f_i(Y_{i-1}, Y_i, X, i) \geq 0$

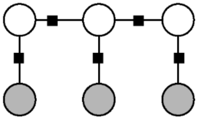


Figure 3: Linear Chain CRF [2]

Code

```
from sklearn_crfsuite import CRF

X_train = [
    [word2features(s, i) for i in range(len(s))]
    for s in train_sents]
y_train = [
    [label for token, postag, label in s]
    for s in train_sents]

crf = sklearn_crfsuite.CRF(
    algorithm='lbfgs',
    c1=0.1, c2=0.1,
    max_iterations=100,
)
crf.fit(X_train, y_train)
```

Notes

Feature selection:

1. Start with all features.
2. a. If there exists a feature removing which worsens the performance by $< t$, remove it. Repeat 2.
3. b. If not, exit.

Notes

Feature selection:

1. Start with all features.
2. a. If there exists a feature removing which worsens the performance by $< t$, remove it. Repeat 2.
3. b. If not, exit.

1. Start with no features.
2. a. If there exists a feature adding which improves the performance by $> t$, add it. Repeat 2.
3. b. If not, exit.

Notes

Feature selection:

1. Start with all features.
2. a. If there exists a feature removing which worsens the performance by $< t$, remove it. Repeat 2.
3. b. If not, exit.

1. Start with no features.
2. a. If there exists a feature adding which improves the performance by $> t$, add it. Repeat 2.
3. b. If not, exit.

Properties

- Hard to setup & train
- Fast inference

Homework

TBD

Resources

1. Overview: https://www.analyticsvidhya.com/blog/2018/08/nlp-guide-conditional-random-fields-text-classification
2. Very detailed: http://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf
3. NER using CRF: https://medium.com/data-science-in-your-pocket/named-entity-recognition-ner-using-conditional-random-fields-in-nlp-3660df22e95c
4. Forward-backward for CRF: https://www.cs.cornell.edu/courses/cs5740/2016sp/resources/collins_fb.pdf
5. Academic-level introduction to CRF: https://www.youtube.com/watch?v=7L0MKKfqe98
6. Generalized CRF: https://people.cs.umass.edu/~wallach/technical_reports/wallach04conditional.pdf
7. Accessible introduction: http://pages.cs.wisc.edu/~jerryzhu/cs769/CRF.pdf
8. Python code: https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html#let-s-use-conll-2002-data-to-build-a-ner-system

Resources

9. Fast Linear Chain CRFs (C): http://www.chokkan.org/software/crfsuite/
10. Fast Linear Chain CRFs (C++): https://taku910.github.io/crfpp/
11. Bayesian Networks: https://www.ics.uci.edu/~rickl/courses/cs-171/0-ihler-2016-fq/Lectures/Ihler-final/09b-BayesNet.pdf
12. Naïve Bayes to HMM to CRF: http://cnyah.com/2017/08/26/from-naive-bayes-to-linear-chain-CRF/