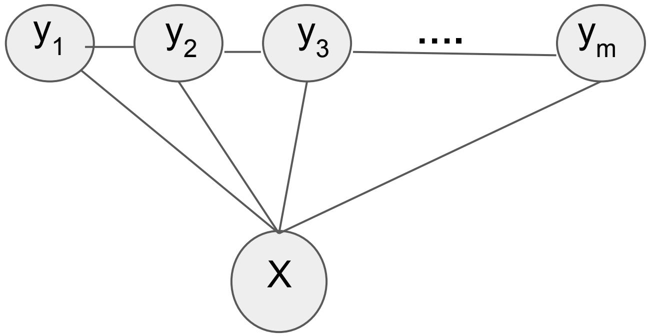
Intuitions for Conditional Random Field

Shaukat Abidi - CSIRO DATA61

Famous Classification of ML-Algorithms

1. Generative
    a. Naive Bayes
        i. Document Classification (Filtering junk-emails)
    b. Hidden Markov Model
        i. Sequence Tagging (Parts Of Speech tagging)
2. Discriminative
    a. Logistic Regression (Regression is confusing here)
        i. Text classification (Sentiment Analysis)
    b. Conditional Random Field
        i. Sequence Tagging (Named Entity Recognition)
    c. Support Vector Machines
        i. Available for IID and Structured data (Text Classification, Sequence tagging and more)
    d. Neural Networks (Connectionist models, Deep Learning, and so on)
        i. No need of introduction (Needs a wealth of data given model parameters)

Conditional Random Field (Order one - Linear Chain)



Acknowledgment

Equations now on are taken from Michael Collins' Slides
(http://www.cs.columbia.edu/~mcollins/crf.pdf)

Example Sentence

France became Fifa champions at Luzhniki

Example Sentence

France became Fifa champions at Luzhniki

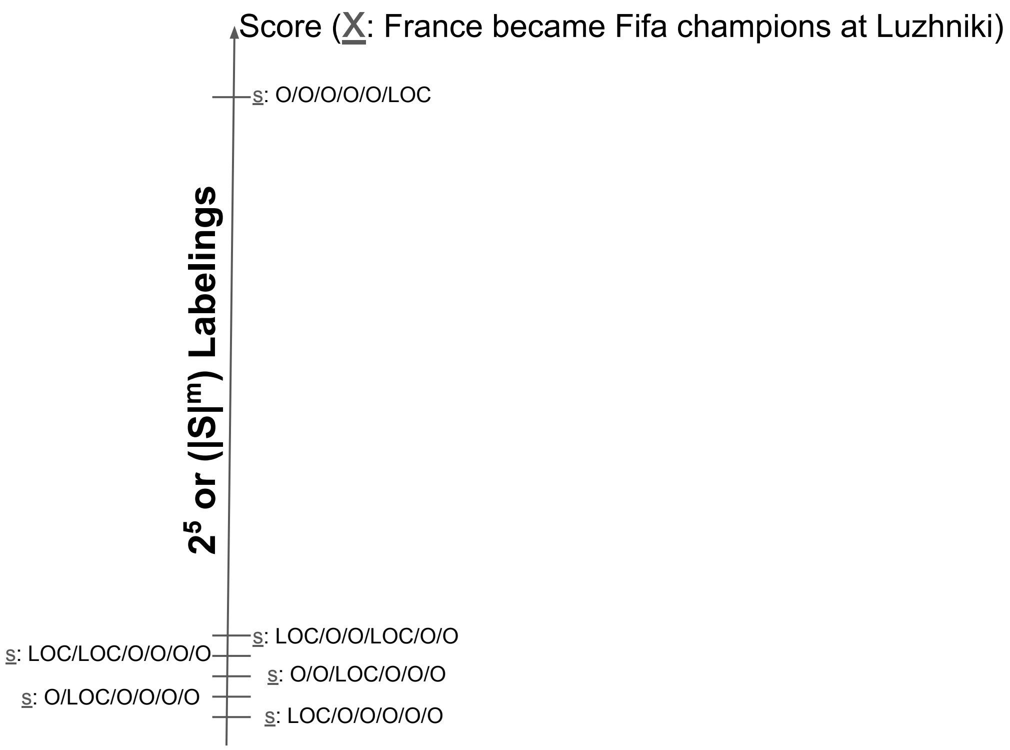
1) m=6 (number of tokens)
2) $\underline{x}$ = {France , became, Fifa, champions, at, Luzhniki}
3) $\underline{x} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$

Example Sentence

France became Fifa champions at Luzhniki

1) m=6 (number of tokens)
2) $\underline{x}$ = {France , became, Fifa, champions, at, Luzhniki}
3) S = {O, Location} (Set of States -- |S| = 2)
4) $\underline{s}$= {o,o,o,o,o,Location}
5) $\underline{s}= \{s_1,s_2,s_3,s_4,s_5,s_6\}$

Intuitions for Scoring Function

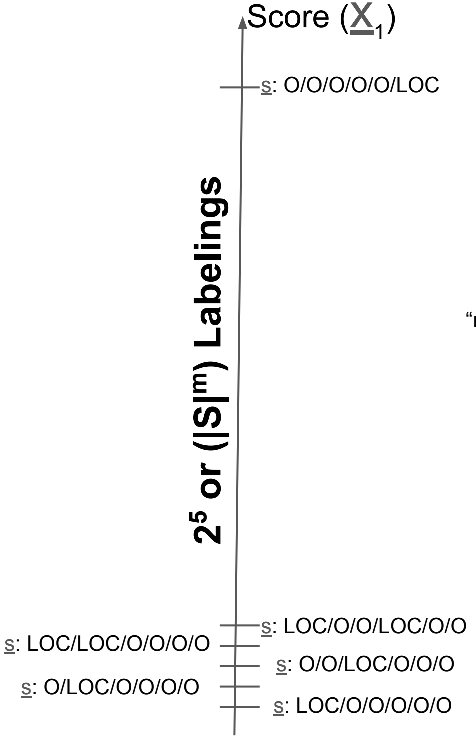


Intuition for Training Objective

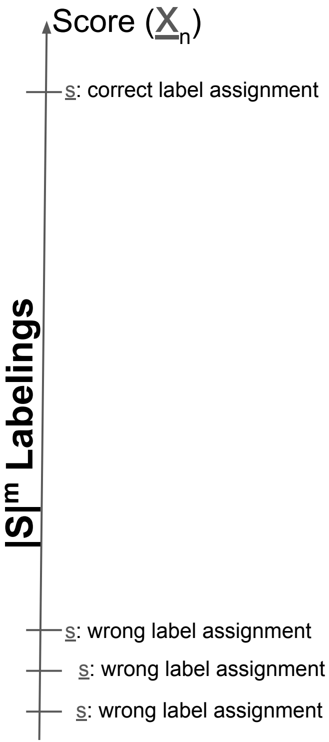
Get as many correct labelings as possible on top on the following scale

Intuition for Training Objective

BUT at the same time, avoid overfitting.
REGULARIZE your weight vector

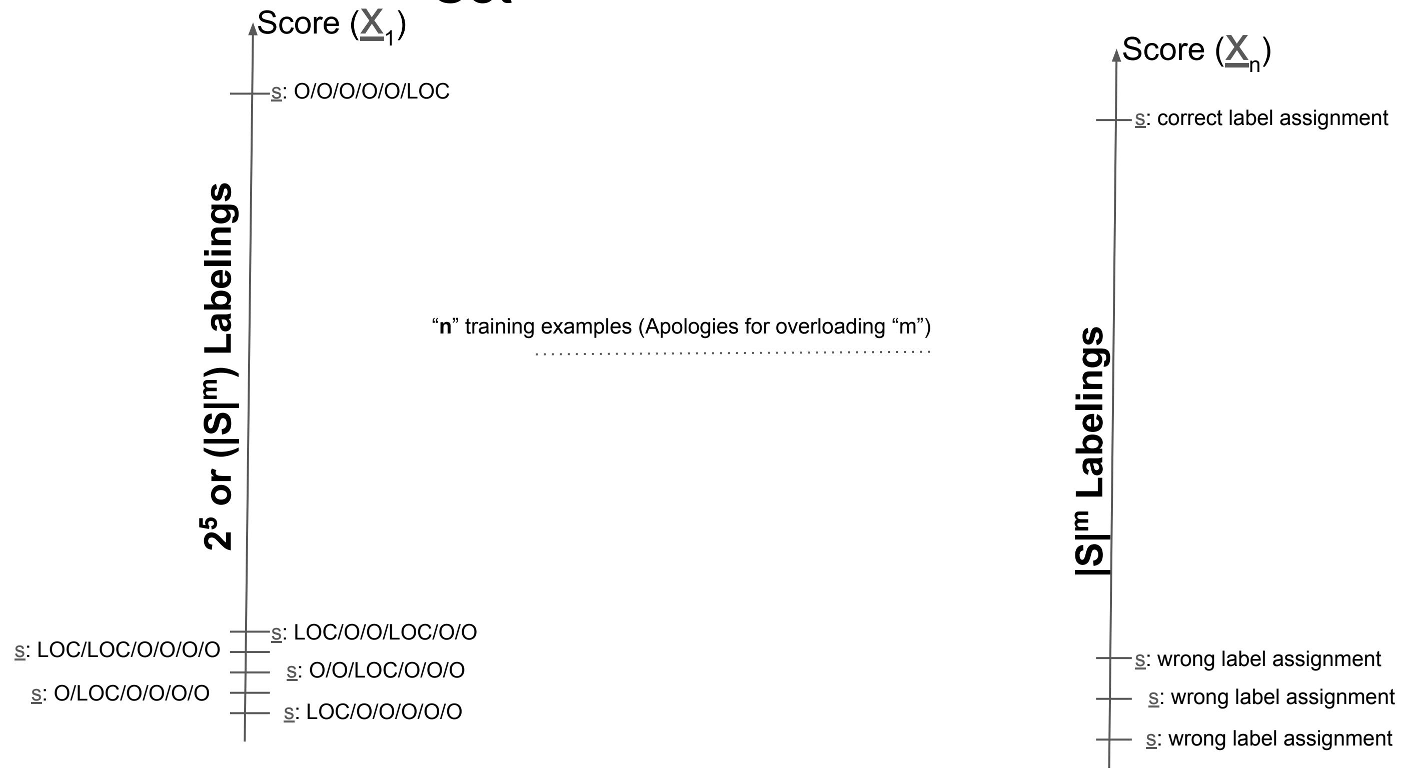


"**n**" training examples (Apologies for overloading "m")



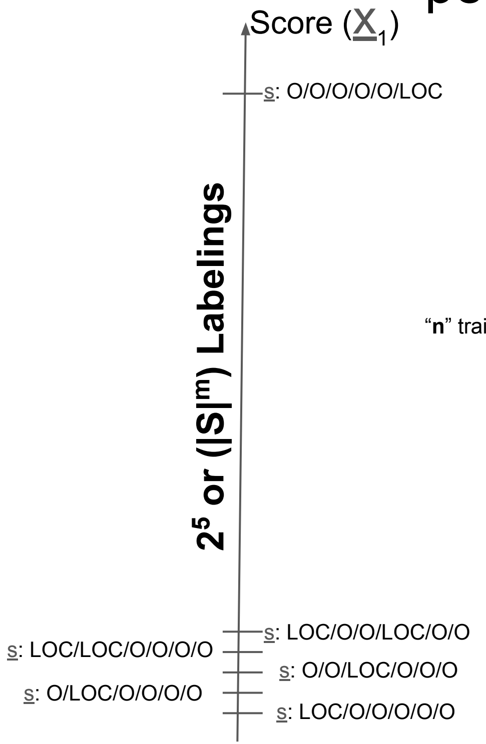
Intuition for Training Objective

The hope is you will get similar effect on Test Set

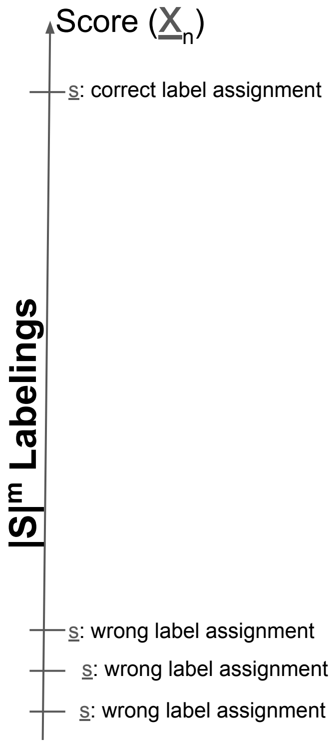


Intuition for Training Objective

And the ambitious goal is to get such performance in real-time systems

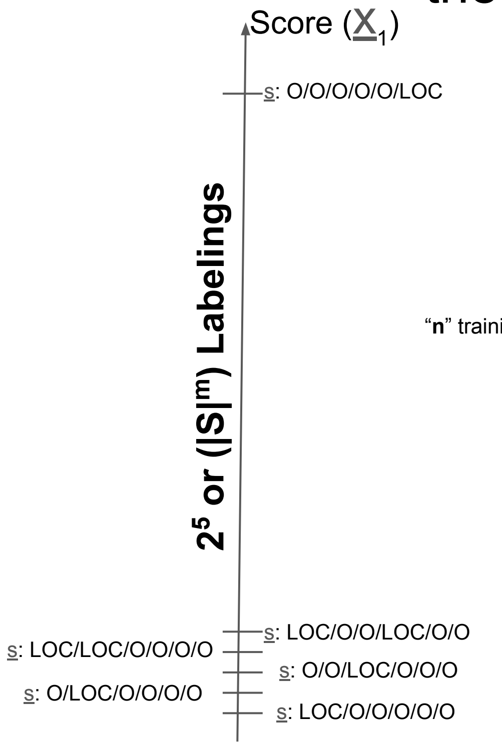


“**n**” training examples (Apologies for overloading “m”)

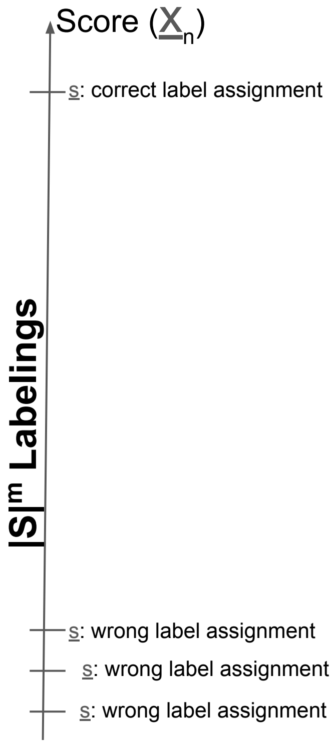


Intuition for Training Objective

We will stick to the performance on Train set for the sake of simplicity



"**n**" training examples (Apologies for overloading "m")



Terminologies Ahead

- Exponential Families
- Log-linear models
- Partition Function
- Feature Function
- Decoding and parameter estimation

Feature Function

Feature function converts **INPUT** of arbitrary length vector to the **OUTPUT** of fixed length vector

Feature Function

Feature function converts **INPUT** of arbitrary length vector to the **OUTPUT** of fixed length vector

$$\underline{\Phi}(\underline{x}, \underline{s}) \in \mathbb{R}^d$$

Feature Function

Feature function converts **INPUT** of arbitrary length vector to the **OUTPUT** of fixed length vector

$$\Phi(\underline{x}, \underline{s}) \in \mathbb{R}^d$$

Input sequence

Sequence of States

Feature Function

Feature function converts **INPUT** of arbitrary length vector to the **OUTPUT** of fixed length vector

Types of Features (you will come across these names in literature)

- State or Emission Features
- Transition Features

$$\underline{\Phi}(\underline{x}, \underline{s}) = \begin{bmatrix} \uparrow \\ \mathbb{R}^D \\ \downarrow \end{bmatrix}$$

Feature Function

Feature function converts **INPUT** of arbitrary length vector to the **OUTPUT** of fixed length vector

Types of Features (you will come across these names in literature)

- State or Emission Features
- Transition Features

$$\underline{\Phi}(\underline{x}, \underline{s}) = \begin{bmatrix} \uparrow \\ R^D \\ \downarrow \end{bmatrix}$$

DIMENSION OF FEATURE FUNCTION WILL DETERMINE THE DIMENSION OF WEIGHT VECTOR

Feature Function

Input sequence is converted to a vector of fixed length

$$\underline{\Phi}(\underline{x}, \underline{s}) = \begin{bmatrix} \\ \mathbb{R}^d \\ \\ \end{bmatrix}$$

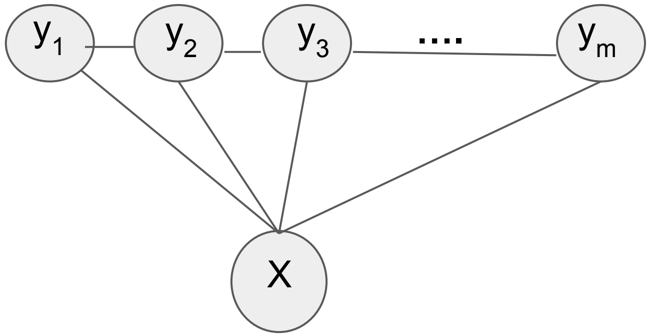
$\underline{x} \rightarrow$ {France, became, Fifa, Champions, at, Luzhniki}

$\underline{s} \rightarrow$ {0,0,0,0,0,Loc}

Emission or state features

Transition features

Conditional Random Field
(Order one - Linear Chain)



SCORING A SEQUENCE - Score Function

$$\underline{w} \cdot \underline{\Phi}(\underline{x}, \underline{s})$$

OR

$$\underline{w}^T \underline{\Phi}(x, s)$$

weight vector

our feature function

SCORING A SEQUENCE - Score Function

For some reasons, lets make score positive (ADD Picture)

We are introducing the notion of exponential families here

Applying exp( )

$$\exp\left(w^T \underline{\Phi}(\underline{x}, \underline{s}) \right)$$

Transition to VALID PROBABILITY DISTRIBUTION

- We need NORMALIZING CONSTANT (PARTITION FUNCTION: Z(x))

TURNING EVERYTHING DOWN TO A
VALID PROBABILITY DISTRIBUTION

$$Z(\underline{x}) = \sum_{s' \in S^m} \exp(\underline{w} \cdot \underline{\Phi}(\underline{x}, s'))$$

GIANT LOG-LINEAR MODEL

$$\underbrace{p(\underline{s}|\underline{x};\underline{w})}_{\substack{\text{CONDITIONAL}\\\text{DISTRIBUTION}\\\text{"states given observations"}}} = \frac{\exp(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{s}))}{Z(\underline{x}) = \sum_{s'\in S^m}\exp(\underline{w}\cdot\underline{\Phi}(\underline{x},\underline{s}'))}$$

And current model (w)

$$\underbrace{p(\underline{s}|\underline{x};\underline{w})\in[0,1]}_{\text{for a single sequence}}$$

Likelihood Score for a sequence

$\underline{x}$ = {France , became, Fifa, champions, at, Luzhniki}

$\underline{s}$= {o,o,o,o,o,Location}

P($_{\underline{s}\text{= \{o,o,o,o,o,Location\}}}$ | $_{\underline{x}\text{=\{France , became, Fifa, champions, at, Luzhniki\}; w}}$) = ??

Likelihood Score for a sequence

For our example sequence, we can have 64 possible labelings

$$P(\underline{s} = \{o, o, o, o, o, loc\} \mid \underline{x}; w) = \frac{\exp(\underline{w} \cdot \underline{\Phi}(\underline{x}, \underline{s} = \{o, o, o, o, o, loc\}))}{\exp(\underline{w} \cdot \bar{\Phi}(\underline{x}, \underline{s}^1)) + \exp(\underline{w} \cdot \bar{\Phi}(\underline{x}, \underline{s}^2)) + \cdots + \exp(\underline{w} \cdot \bar{\Phi}(\underline{x}, \underline{s}^{64}))}$$

Solving the issue of Tractability

Feature function depends on current and previous states only (ORDER-ONE CRF)

$$\underline{\Phi}(\underline{x}, s) = \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

Solving the issue of Tractability

Feature function depends on current and previous states only (ORDER-ONE CRF)

tot tokens in a sequence

$$\underline{\Phi}(\underline{x}, s) = \sum_{j=1}^{m} \underline{\phi}(\underline{x}, j, s_{j-1}, s_j)$$

state of "j-1"th token

state of "j"th token

CRF Likelihood Function

Training Likelihood

PROBLEM: UNDERFLOW (Could happen)

$$\text{Likelihood}(w) = \underbrace{p(\underline{s}^1 / \underline{x}^1 ; w)}_{\substack{\text{likelihood} \\ \text{score} \\ \text{for} \\ 1^{st} \text{ sequence}}} \times \ldots \times \underbrace{p(\underline{s}^n / \underline{x}^n ; w)}_{\substack{\text{likelihood} \\ \text{score} \\ \text{for} \\ n^{th} \text{ sequence}}}$$

CRF Likelihood Function

We would like to Maximize Training Likelihood; ideally, “w” should give the highest score to the correct labeling

$$\text{Likelihood}(w) = \underbrace{p(\underline{s}^1 / \underline{x}^1; w)}_{\substack{\text{likelihood} \\ \text{score} \\ \text{for} \\ 1^{st}\ \text{sequence}}} \times \cdots \times \underbrace{p(\underline{s}^n / \underline{x}^n; w)}_{\substack{\text{likelihood} \\ \text{score} \\ \text{for} \\ n^{th}\ \text{sequence}}}$$

CRF LOG Likelihood Function

Log Likelihood for training set ("n" sequences)

$$\mathcal{L}(\underline{w}) = \log P(\underline{s}^1 | \underline{x}^1; w) + \log (\underline{s}^2 | \underline{x}^2; w) + \cdots + \log P(\underline{s}^n | \underline{x}^n; w)$$

$$\mathcal{L}(\underline{w}) = \sum_{i=1}^{n} \log P(\underline{s}^i | \underline{x}^i; \underline{w})$$

CRF LOG Likelihood Function with Regularizer

Log Likelihood for training set ("n" sequences)

$$\underline{w}^* = \underset{\underline{w} \in \mathbb{R}^d}{\operatorname{argmax}} \sum_{i=1}^{n} \log P(\underline{s}^i \mid \underline{x}^i ; w) - \frac{\lambda}{2} \|w\|^2$$

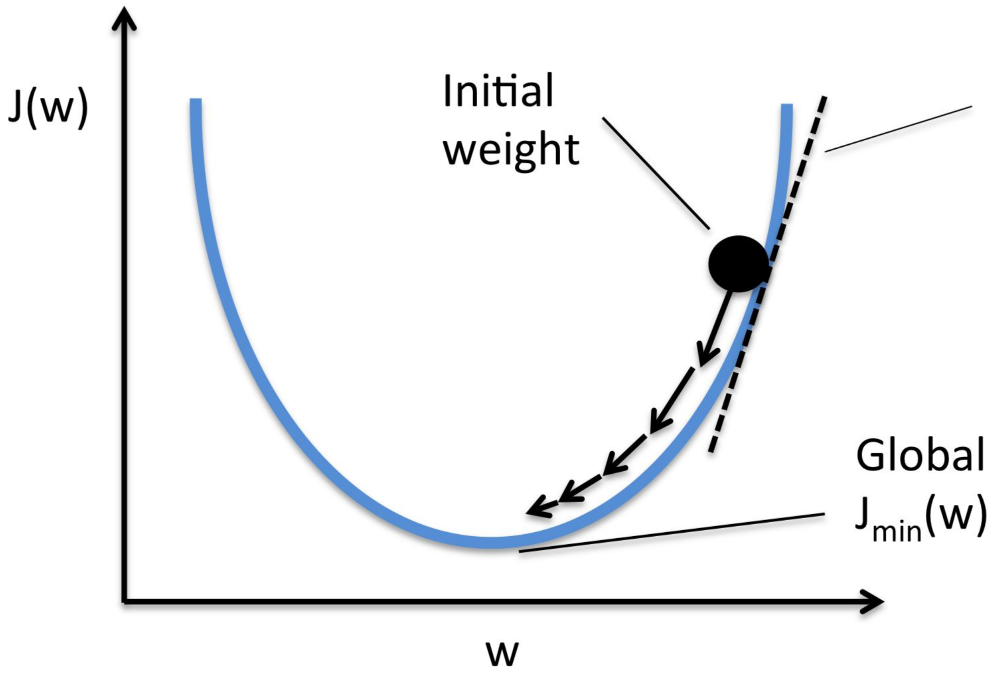
$\underline{w}^*$ is the model/weight vector we would ~~to~~ like to find.

CRF TRAINING

- Gradient Descent (or Ascent) is used for minimizing (or maximizing) training objective

- The only thing we need is the partial derivative of weight vector

Gradient Descent Illustration for Convex Objective



https://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/

For thorough introduction to Convex Optimization, please see **Boyd, Stephen, and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.**

CRF TRAINING

- Gradient Descent (or Ascent) is used for minimizing (or maximizing) training objective
- The only thing we need is the partial derivative of weight vector

$$\frac{d}{dw_K} L(\underline{w}) = \sum_i \Phi_K(\underline{x}^i, \underline{s}^i) - \sum_i \sum_{\underline{s} \in S^m} p(\underline{s} \mid \underline{x}^i; w)\, \Phi_K(\underline{x}^i, \underline{s}) - \lambda w_K$$

CRF TRAINING

- Gradient Descent (or Ascent) is used for minimizing (or maximizing) training objective
- The only thing we need is the partial derivative of weight vector

$$\frac{d}{dw_K} L(\underline{w}) = \sum_i \Phi_K(\underline{x}^i, \underline{s}^i) - \sum_i \sum_{\underline{s} \in S^m} p(\underline{s} \mid \underline{x}^i; w)\, \Phi_K(\underline{x}^i, \underline{s}) - \lambda w_K$$

Not hard to impute

Simplify it - Backward-Forward Algorithm will help us imputing it

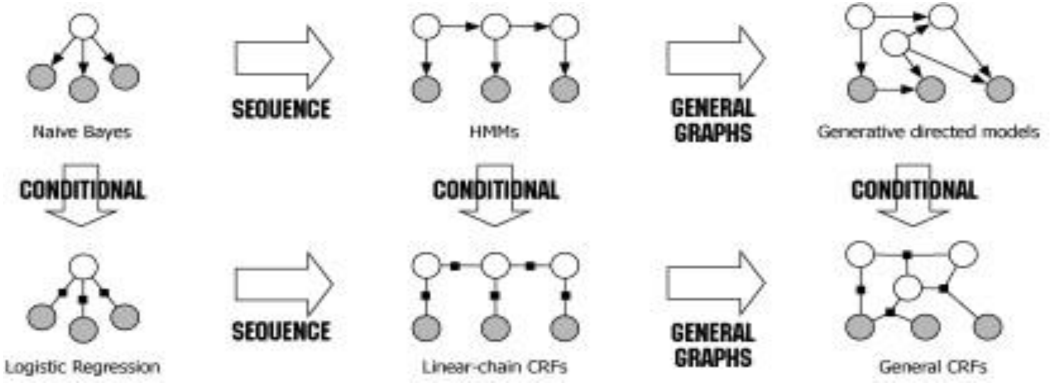
CRF DECODING (Linear-Chain)

- Once we get the weight vector, we can now use Viterbi Algorithm for finding out the best state assignment (s) given observations (x) and weight vector (w)

- Viterbi finds out the best sequence (for linear-chain CRFs) in $O(mk^2)$

Facts

- Conditional Random Field (Linear Chain) is a sequential version of Logistic Regression
- HMM and CRF are generative-discriminative pairs



Sutton, Charles, and Andrew McCallum. "An introduction to conditional random fields." Foundations and Trends® in Machine Learning 4.4 (2012): 267-373.

Conclusion

- We have seen the intuitions for maximizing training objective for linear-chain CRF
- We have seen how to score a single sequence
- We get an idea for training and decoding a linear-chain CRF
- We will now move onto the practical demonstration where we will use **sklearn-crfsuite** for CRF training and decoding

REFERENCES

- Michael Collins. “Log-Linear Models, MEMMs, and CRFs.” (http://www.cs.columbia.edu/~mcollins/crf.pdf)

- Sutton, Charles, and Andrew McCallum. "An introduction to conditional random fields." Foundations and Trends® in Machine Learning 4.4 (2012): 267-373.

- Stephen Boyd, Lieven Vandenberghe, Convex Optimization, Cambridge University Press,2004, http://www.stanford.edu/~boyd/cvxbook

- Massimo Piccardi. “A vademecum of statistical pattern recognition techniques with applications to image and video analysis” (Lecture Series at UTS)

QUESTIONS