Classification
- \( y \): random variable for prediction (output)
- \( x \): random variable for observation (input)
- Training Data = Collection of \((x, y)\) pairs
- Machine Learning = Given the training data, learn a mapping function \( f(x) = y \) that can map input variables to output variables
  - Binary classification
  - Multiclass classification

Sequence Tagging
- \( y \): A sequence of random variables for prediction (output)
- \( x \): A sequence of random variables for observation (input)
- Training Data = Collection of \((x, y)\) pairs
- Machine Learning = Given the training data, learn a mapping function \( f(x) = y \) that can map input variables to output variables
  - Binary classification
  - Multiclass classification

Hidden Markov Model (HMM) represented as a graphical model

Sequence Tagging
- Prediction using \textit{BIO} tagging

\begin{quote}
\textit{The Washington Post} reported \textit{Obama}'s view on the oil crisis.
\end{quote}

- \textbf{Classifier}
  - Maximum Entropy (MaxEnt)
  - Naïve Bayes

- \textbf{Sequence Tagger}
  - Hidden Markov Models (HMMs)
  - Maximum Entropy Markov Models (MEMMs)
  - Conditional Random Fields (CRFs)
Maximum Entropy (MaxEnt) Models

- Also known as “Log-linear” Models (linear if you take log)

\[
P(y|x, w) = \frac{\exp(w^T f(y))}{\sum_y \exp(w^T f(y'))}
\]

- The feature vector representation may include redundant and overlapping features

Training Maximum Entropy (MaxEnt)

- Maximizing the likelihood of the training data incidentally maximizes the entropy (hence “maximum entropy”)

\[
P(y|x, w) = \frac{\exp(w^T f(y))}{\sum_y \exp(w^T f(y'))} \quad \text{Make positive}
\]
\[
\text{Normalize}
\]

- Maximize the (log) conditional likelihood of training data

\[
L(w) = \log \prod_i P(y^i|x^i, w) = \sum_i \log \left( \frac{\exp(w^T f_i(y^i))}{\sum_y \exp(w^T f(y'))} \right)
\]
\[
= \sum_i \left( w^T f_i(y^i) - \log \sum_y \exp(w^T f(y)) \right)
\]

Convex Optimization for Training

- The likelihood function is convex. (can get global optimum)
- Many optimization algorithms/software available.
  - Gradient ascent (descent), Conjugate Gradient, L-BFGS, etc
- All we need are:
  1) evaluate the function at current ‘w’
  2) evaluate its derivative at current ‘w’

Training Maximum Entropy

\[
L(w) = \sum_i \left( w^T f_i(y^i) - \log \sum_y \exp(w^T f(y)) \right)
\]

\[
\frac{\partial L(w)}{\partial w_n} = \sum_i \left( f_i(y^i)_n - \sum_y P(y|x_i) f_i(y)_n \right)
\]

Expected count of feature \(n\) in predicted candidates

Total count of feature \(n\) in correct candidates

Training with Regularization

\[
L(w) = -k|w|^2 + \sum_i \left( w^T f_i(y^i) - \log \sum_y \exp(w^T f(y)) \right)
\]

\[
\frac{\partial L(w)}{\partial w_n} = -2k w_n + \sum_i \left( f_i(y^i)_n - \sum_y P(y|x_i) f_i(y)_n \right)
\]

Big weights are bad

Expected count of feature \(n\) in predicted candidates

Total count of feature \(n\) in correct candidates

(slide modified from Dan Klein's)
Graphical Representation of **MaxEnt**

\[ P(y|x, w) = \frac{\exp(w^T f(y))}{\sum_y \exp(w^T f(y))} \]

Graphical Representation of **Naïve Bayes**

\[ P(X | Y) = \prod_{j=1}^n P(x_j | Y) \]

- Consult AI text book for more details

**Naïve Bayes vs. MaxEnt**

- **Naïve Bayes Classifier**
  - "Generative" models
    - \( P(\text{input} | \text{output}) \)
    - For instance, for text categorization, \( P(\text{words} | \text{category}) \)
    - Waste energy on generating input (which we don't need to generate during test)
    - Independent assumption among input variables: Given the category, each word is generated independently from other words (too strong assumption in reality?)
    - Cannot incorporate arbitrary/redundant/overlapping features

- **MaxEnt Classifier**
  - "Discriminative" models
    - \( P(\text{output} | \text{input}) \)
    - For instance, for text categorization, \( P(\text{category} | \text{words}) \)
    - Focusing only on predicting the output
    - By conditioning on the entire input, we don't need to worry about the independent assumption among input variables
    - Can incorporate arbitrary features
    - Can handle redundant and overlapping features

**Sequence Tagging with HMM / MEMM / CRF**

**Graphical Model Basics**

- Conditional probability for each node
  - e.g. \( p(Y_3 | Y_2, X_3) \) for \( Y_3 \)
  - e.g. \( p(X_3) \) for \( X_3 \)
- Conditional independence
  - e.g. \( p(Y_1 | Y_2, X_1) = p(Y_1 | Y_2, X_2, X_1, X_2, X_3) \)
- Joint probability of the entire graph
  - \( = \) product of conditional probability of each node

**HMM v.s. MEMM (Maximum Entropy Markov Models)**

- **HMM**
  - NNP → VBZ → VBN → TO → VB → NR
  - Secretariat is expected to race tomorrow

- **MEMM**
  - NNP → VBZ → VBN → TO → VB → NR
  - Secretariat is expected to race tomorrow
Generative models

$ p(\text{words} \mid \text{tags})$

"generate" input (in addition to tags)

but we need to predict tags, not words!

Discriminative or Conditional models

$ p(\text{tags} \mid \text{words})$

"condition" on input

Focusing only on predicting tags

HMM v.s. MEMM

HMM

MEMM

MEMM v.s. CRF (Conditional Random Fields)

CRF

Undirected Graphical Model Basics

MEMM

CRF

Undirected Graphical Model Basics

Joint probability of the entire graph

$$ P(\mathcal{Y}) = \frac{1}{Z} \prod_{\text{clique } C} \phi(\mathcal{Y}_C) $$

$$ Z = \sum_{\mathcal{Y}} \prod_{\text{clique } C} \phi(\mathcal{Y}_C) $$

Secretariat is expected to race tomorrow

Secretariat is expected to race tomorrow

Secretariat is expected to race tomorrow

Secretariat is expected to race tomorrow

Secretariat is expected to race tomorrow

Secretariat is expected to race tomorrow

MEMM

CRF

MEMM

CRF

MEMM

CRF

MEMM

CRF

MEMM

CRF

MEMM

CRF

MEMM

CRF

MEMM

CRF
MEMM v.s. CRF

Inference (Viterbi)

Objective function for training

Given the training data $D = \{(x^j, y^j)\}_{j=1}^N$
and $p(y \mid x) = \frac{1}{Z(x)} \exp \sum \lambda \bullet F(y, x)$

Objective function:
conditional likelihood $L(\lambda) = \log L(\lambda) = \sum \log p(y^j \mid x^j)$
equiv. to optimize $\delta(\lambda) = \sum \log p(y^j \mid x^j) = \sum \log \frac{1}{Z(x)} \exp \sum \lambda \bullet F(y^j, x^j)$

CRFs Software:
- Mallet (http://mallet.cs.umass.edu/),
- CRF++ (http://crfpp.sourceforge.net/),
- CRF (http://crf.sourceforge.net/)