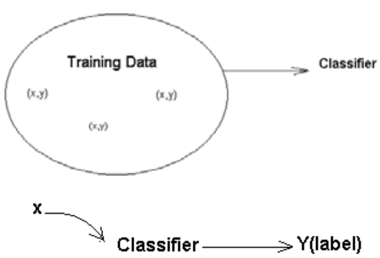
Machine Learning for Sequential Data

Sequential Supervised Learning

Sequential Supervised Learning

- Supervised learning
- Statistical learning problems
- Sequential Supervised Learning(SSL)
- Research issues in SSL
- Methods for addressing SSL problems

Supervised Learning(*backround…*)



Examples

Character recognition
  x : image , y = {A, B,… ,Z}
Cellular telephone fraud detection
  x : telephone call , y ={0,1}
Part of speech tagging
  x : *a word* y : part of speech
Word Recognition Problem
  x: sequence of letters y : word

Sequential Supervised Learning

$\{(x_i,y_i)\}$ for i = 1 to N : Set of N training examples

$\mathbf{x_i} = (\mathbf{x}_{i,1},\mathbf{x}_{i,2},\ldots,\mathbf{x}_{i,Ti})$ $\mathbf{y}i = (y_{i,1},y_{i,2},\ldots,y_{i,Ti})$

Goal : construct classifier h(**x**) ,that predicts new label sequence **y** given an input sequence **x**

Examples

Telephone fraud detection
  distribution of legitimate call
Part of speech tagging
  x = (do you want fries)
  y = (verb pronoun verb noun)

Research Issues *in Sequential Supervised Learning*

three basic research issues:

- Loss functions
- Feature selection
- Computational efficiency

Loss Functions

Classical Supervised Learning

0/1 Loss : Loss = 0 if correctly classified

Loss = 1 incorrectly classified

Non Uniform Loss functions

represented by "***cost matrix***" C(i,j)

C(i,j) : cost when true value = j

predicted value = i

Different types of Loss Functions

Basic Idea :*different errors = different costs*

- Depends on goal

  different goals = different loss functions

  e.g. **goalA** : *predict entire error sequence correctly*

  goalB : *predict as many correct individual labels as possible*

Examples

Cell Phone fraud detection

goal : predict t* = *time when cell phone was stolen*

If predicted time = t, then

if t < t*

penalty , Ca(t* - t) : cost of lost business

else

penalty , Cb(t – t*) : cost of fraudulent calls

Feature Selection

A "**must**" for any method of sequential supervised learning

- *Break overall problem of predicting* $\boldsymbol{y}_i$ *given* $\boldsymbol{x}_i$ *into subproblems of predicting individual output labels* $y_{i,t}$ *, given a subset of* $\boldsymbol{x}_i$ *and* $\boldsymbol{y}_i$

Feature Selection Problem

identify relevant subset for making accurate predictions

Feature selection problem :***Solutions***

Four Strategies :

Strategy One: (Wrappers Approach)

- Make various subsets of features
- Run learning algorithm and find *hypothesis*
- Measure the accuracy of the *hypothesis*

Feature subsets are selected by

- ***Forward Selection***
- ***Backward Elimination***

Feature selection problem , cont..

Strategy Two:

- Include all possible features in the model
- Place Penalty on the values of parameters
  - *Causes parameters with useless features to become small*

e.g. Neural Network weight elimination , ridge regression, support vector machines

Strategy three :

- Compute some measure of feature relevance
- Remove low scoring features

  e.g. through mutual information between a feature and the class

Strategy four :

- First fit a simple model
- Analyze the simple model to identify feature importance

A general approach :

Assumption → *a fixed sized neighborhood is relevant*

eg $x_{i,t-1}$, $x_{i,t}$, $x_{i,t+1}$ *predict* $y_{i,t}$

Two Drawbacks

- not all features relevant
- Longer range interactions "missed"

  e.g. thought and though

** Any successful feature selection methodology needs human expertise and statistical methodology "both"

Computational Efficiency

- Most of the Sequential Learning algorithm are computationally expensive
- Applying learned classifier is expensive too…

One approach:

- *apply cheapest methods first*
- *generate set of possible candidates*
- *apply expensive methods progressively*

Machine Learning methods for *SSL*

1. *Sliding window methods*
2. *Recurrent sliding window methods*
3. *Hidden Markov models*
4. *Maximum Entropy Markov Models*
5. *Input-Output Markov Models*
6. *Conditional random fields*
7. *Graph transformer networks*

Sliding window method

Basic Idea : *convert sequential supervised learning problem into classical supervised learning*

- Construct a window classifier h_w

  h_w maps window of width w → y

classification

Add d ((w-1)/2) **"null"** values on each end of x_i

Convert them into N separate examples

Predict y_t for each example

Concatenate all y_t 's to form **y**

Sliding Window method, cont…

Advantage:

Any supervised learning algorithm can be applied

Drawback :

Correlation between nearby y values not taken into account.

Example : Sejnowski and Rosenberg used..

7 letter sliding window for the task of pronouncing English words.

Recurrent sliding window

Only difference *:* predicted $y_{i,t}$ is fed as input to predict $y_{i,t+1}$

Most recent d predictions

$y_{i,t-d}$ $y_{i,t-d+1}$ ……… $y_{i,t-1}$

and

$x_{i,t-d}$ $x_{i,t-d+1}$ …… $x_{i,t}$ …….. $x_{i,t+d-1}$

to predict $y_{i,t}$

Hidden Markov Models(HMM)

A probabilistic model

represents $P(\mathbf{x},\mathbf{y})$

Defined by:

Transition probability

$P(y_t|y_{t-1})$: *how adjacent y are related*

Observation probability

$P(\mathbf{x}|y)$: *how observed x are related to hidden y*

(both stationary distributions)

HMM , cont…

How x_i and y_i are generated

- *if K possible labels (1 …. K) ,then*

  augment start label = 0 and

  end label = K+1

- *Generate y values ,using* $P(y_{i,t} \mid y_{i,t-1})$ *until* $y_{i,t}= K+1$

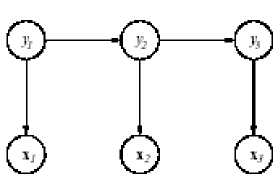
  Set T = t , at this time

- *For t = 1 .. T*
  - *Generate x , according to* $P(x_{i,t} \mid y_{i,t})$

HMM , cont…

Training HMM → learning $P(y_{i,t} \mid y_{i,t-1})$

$P(x_{i,t} \mid y_{i,t})$



HMM in sequential learning problems

$P(y_{i,t} \mid y_{i,t-1})$ *: by looking at all pairs of adjacent y labels*

$P(x_{i,t} \mid y_{i,t})$*: by looking at al pairs of* x_i *y*

$$\overline{\mathbf{y}} = \underset{\mathbf{z}}{\operatorname{argmin}} \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) L(\mathbf{z},\mathbf{y}).$$

HMM , limitations

- *Relationship between separated y values not communicated(e.g y_1 and y_5)*
  solution : sliding window of x_t values

- *x_t is only generated from y_t → more difficult to use an input window*

Maximum Entropy Markov Models(**MEMM**)

- Conditional probabilistic models
  *represents : P(**y** |**x**)*

Learns $\mathbf{P(y_t \mid y_{t-1} , x_t)}$

Trained using Maximum entropy method

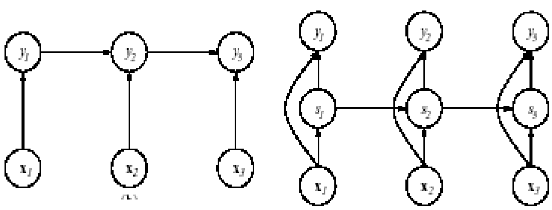
$$P(y_t|y_{t-1}, \mathrm{x}) = \frac{1}{Z(y_{t-1}, \mathrm{x})} \exp\left(\sum_{\alpha} \lambda_\alpha f_\alpha(\mathrm{x}, y_t)\right)$$

$Z(y_{t-1}, \mathbf{x})$: normalizing factor

f_α : boolean feature , depends on y_t and "any" properties of x (input sequence)

"supports long distance interactions"

MEMM & IOHMM



Input Output HMM(IOHMM)

- Similar to MEMM
- with additional "***hidden state variables***" s_t
- s_t : hidden states permit "memory" of long distance effects

Limitation of MEMM and IOHMM

- **Label bias problem**

probability mass received by y_{t-1} "must be" Transmitted to y_t (at time t) regardless x_t

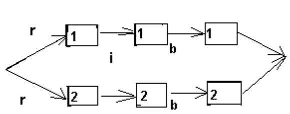
Label Bias problem , example …

y = {1,2}

*for **x** = "rob" **y** = "111" &*

*for **x** = "rib" **y** = "222"*

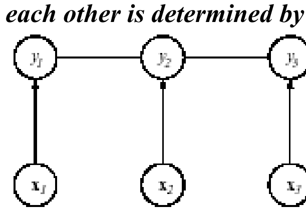
"rib" and "rob" has equal probability



Conditional Random Fields(CRF)

To overcome the problem of *"**Label Bias**"*

- ***The way adjacent pairs y_t and y_{t-1} influence each other is determined by input features***



CRF ***Advantages****:*

- *Overcomes Label Bias Problem*
- *Takes care of long distance interactions*

Drawback :

- *Training is expensive*

Results

Problem *: Part of Speech tagging*

error rates

HMM : ***5.69%*** *, MEMM :* ***6.37%*** *, CRF :* ***5.55%***

Graph Transformer Networks

Neural network methodology for solving sequential supervised learning problems

- **Graph Transformer Network**

is a neural network, (input graph → output graph)

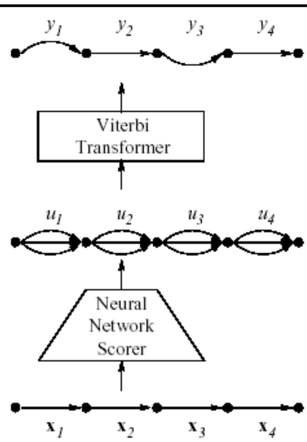
Input graph : linear sequence of x_t(feature Vector)

Output graph : collection of u_t values

u_t = (class label ,score)

Viterbi transformer : finds "*lowest score path"*

Training by *"gradient descent"*



Current Research Issues

How to *:*

- Capture and exploit sequential correlations
- Represent and incorporate complex loss functions
- Identify long distance interactions
- Make learning algorithms fast

summary….

- Supervised learning
  - Some problems don't fit in supervised learning
- Sequential supervised learning
- Fundamental issues in sequential supervised learning .. Like Loss functions , feature selection , computational efficiency

summary….

- Machine learning methods for SSL problems
  - HMM, IOHMM, MEMM ,CRF, GTN , sliding window and recurrent sliding window
  - Advantages and drawbacks for these methods
- Research issues
  - Capture sequential relations , increasing computational efficiency ..etc.

Machine Learning For Sequential Data: A Review

Commentator: Krishna

04-02-2003

- Supervised Learning
- Construct Classifier that can predict the classes.
- Consider scenarios where the correlation between data matters

Example : Text To Speech

Pronunciation depends on characters encountered or some character that is at a distance.

eg. Rich / Rice

Though / Thought

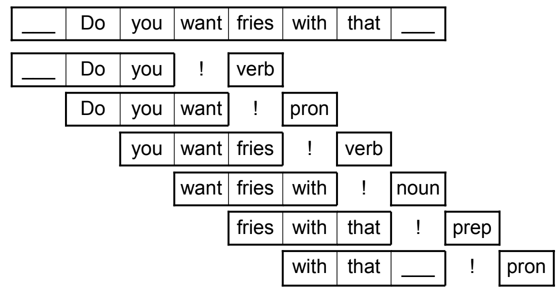
Loss Function

- Predicting the Class
- Scenario based consideration
- Role of Loss Function
- Scenarios
  - Two Class – simple
  - Multi-Class Problems- M x M array of possible classifications.
  - Rock / Diamond Problem
  - Fraud Detection – Not to loose a potential customer
  - Blood Type Match - No room for even a minor error

The Sliding Window Methods For SSL

Comments By: Navdeep Kaur

Sliding Windows



Recurrent Sliding Windows

- Include y_t as input feature when computing y_{t+1}.
- During training:
  - Use the correct value of y_t
  - Or train iteratively (especially recurrent neural networks)
- During evaluation:
  - Use the predicted value of y_t

Recurrent Sliding Window Method

- English pronunciation problem

  e.g. for pairs of words like “photograph” and “photography”.