The TRSL Project

Internship at IISc, to implement a research paper
Interns: Kshithij Karthick, Ravikumar L

Sponsor: Prof. T V Sreenivas, Speech and Audio Group, IISc
Mentor: Kiran Subbaraman

Project: https://github.com/iisc-sa-open/trsl

Agenda

- Purpose of this work
- Core algorithm – Decision tree building, and predicting the target
  - Tree construction
    - What the algorithm is; How it is build & Why it was built this way
  - Set construction
- Observations

Purpose

- Algorithm Implementation
  - This algorithm is based on the paper: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=32278
    - “A Tree-Based Statistical Language Model for Natural Language Speech Recognition”, (1989) by L R Bahl, P F Brown, P V de Souza, R L Mercer
  - Use Python, and its related libraries to build a usable implementation of the algorithm

“What” is the Algorithm

- **Step 1:** For every leaf node in the tree, initially the root
  - **1.a:** Ask questions, which have not been asked before (in this lineage)
  - **1.b:** Identify the question which results in minimum *average conditional entropy*
- **Step 2:** Select the leaf-nodes, from the tree, which do not result in an '*over-fit*'
  - **2.a:** From these leaf nodes, choose the node, which minimizes the average conditional entropy.
    - **2.a.i:** Is the stopping criteria met
      - **2.a.i.a:** If not, For this leaf-node create two child nodes that represent the 'Y', and 'N' paths. Go to Step 1, for the child nodes
      - **2.a.i.b:** If yes, tree constuction is complete
- **Step 3:** Get a probability distribution for the target at every leaf node

Step 1.a: Ask questions not asked before, in this lineage

1. Training Samples

X1	X2	Target
how	was	the
how	was	it
why	was	the
where	is	a
where	are	the
where	was	I

2. Vocabulary grouped into disjoint Sets

1	{how}
2	{where, why}
3	{was, are, is}
4	{the, a}
5	{it, I}

3. Words replaced with their set identity

X1	X2	Target
1	3	4
1	3	5
2	3	4
2	3	4
2	3	4
2	3	5

4. Questions of this form are asked:

X1 = 1 ?
X2 = 2 ?
X1 = 2?

Step 1.b: Min. Avg. Conditional Entropy

$$\bar{H}_c(Y|X_i = k) =$$

$$- Pr\{X_i = k \mid c\} \sum_{j=1}^{n} Pr\{target = j \mid c\}$$

$$. \log_2 Pr\{target = j \mid c, X_i = k\}$$

$$- Pr\{X_i \neq k \mid c\} \sum_{j=1}^{n} Pr\{target \neq j \mid c\}$$

$$. \log_2 Pr\{target \neq j \mid c, X_i \neq k\}$$

- The average conditional entropy at a node c due to the question Xi = k ?
- n is the number of sets
- Pr {target = j | c} is the probability that the target is j given that the immediate set history leads to c in the decision tree

Step 2: Avoid Overfitting

- Overfitting happens in a decision tree when the tree perfectly mirrors the training samples.
- This can be avoided by not splitting a node when too few training samples reach one or both of the tentative child nodes.

Step 2.a.i: Stopping Criteria

$$\frac{Average\ entropy\ of\ leaf\ distributions}{Entropy\ at\ root} > S$$

- S is a tunable parameter.
- Represents the fraction of entropy we want to reduce on an average before a prediction in our model.

$$\bar{H}(Y) = \sum_{i=1}^{L} H_i(Y) \cdot Pr\{l_i\}$$

- Average entropy of leaf distributions
- L is the number of leaf nodes
- Hi(Y) is the entropy at leaf node li
- Pr{li} is the probability of reaching leaf node li

$$H_i(Y) = -\sum_{j=1}^{n} Pr\{target = j \mid l_i\} \cdot \log_2 Pr\{target = j \mid l_i\}$$

- n is the number of sets
- Pr{target = j | li} is the probability that the target is j given that the immediate history of sets leads to the leaf node li

Step 3: Probability dist. for target at every leaf node

Set Construction

- Word2Vec
- K-means

Observations

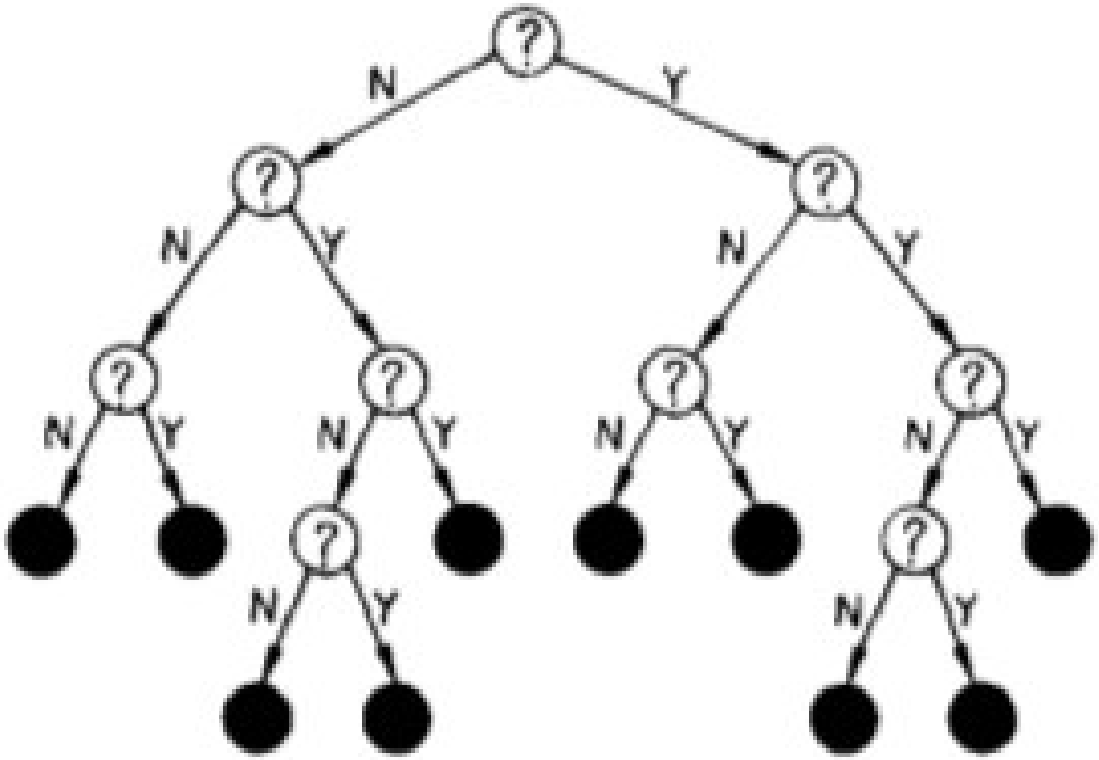
- Metrics
- Class based n-gram model

Backup slides

Terminology

- Memory (Not Computer memory)
- Sliding window
- Word Sequence
- Sets
- Set Sequence
- Predictor Variables (X1, X2, ... Xm)
- Target

Decision Tree



Lineage?

Binary Decision Tree. Source: Tree based statistical language model paper