Data Driven Legal Reforms

Project Code: P2

Team: Harsh Vardhan Rai, Utkarsh Singh

Advisor: Prof. Nomesh Bolia

Introduction

- Large amounts of Indian district courts legal data obtained from http://services.ecourts.gov.in
- Every court case has following details
  - Case Act & Case Section
  - Registration Date, Hearing Dates, Purpose of Hearing, Decision Date
  - Names of petitioner, respondent, judge
- Such detailed information offers lots of possibilities
  - Developing predictive models to predict disposal time of any given case
- How to learn as much as possible from the data?
  - Probabilistic Modelling - Given ‘x’, probability that case is disposed in ‘y’ days?
  - Sequence Modelling - Make use of the sequence of hearing dates

Chief Judicial Magistrate

Case Details

Case Type	: C R Case	
Filing Number	: 1000201/2013	Filing Date: 30-10-2013
Registration Number	: 201/2013	Registration Date: 30-10-2013
CNR Number	**: ASTN04-000533-2013**	

Case Status

First Hearing Date	: 29th November 2013
Next Hearing Date	: 24th September 2018
Stage of Case	: Evidence before charge
Court Number and Judge	: 6-Additional Chief Judicial Magistrate, Tinsukia

Petitioner and Advocate

1) Smti Rishmita Sarkar

Respondent and Advocate

1) Sri Dulal Sarkar

Acts

Under Act(s)	Under Section(s)
Indian Penal Code	406

History of Case Hearing

Registration Number	Judge	Business On Date	Hearing Date	Purpose of hearing
201/2013	Judicial Magistrate 1st Class, Tinsukia	29-11-2013	30-12-2013	Hearing on Petition
201/2013		30-12-2013	27-01-2014	Hearing on Petition
201/2013	Sub-Divisional Judicial Magistrate(Sadar),Tinsukia	27-01-2014	21-03-2014	Hearing on Petition
201/2013	Sub-Divisional Judicial Magistrate(Sadar),Tinsukia	21-03-2014	31-03-2014	Hearing on Petition
201/2013	Sub-Divisional Judicial Magistrate(Sadar),Tinsukia	31-03-2014	12-05-2014	Hearing on Petition
201/2013	Sub-Divisional Judicial Magistrate(Sadar),Tinsukia	12-05-2014	20-06-2014	Hearing on Petition
201/2013	Sub-Divisional Judicial Magistrate(Sadar),Tinsukia	20-06-2014	01-08-2014	Hearing on Petition

Literature Survey

- Previously tried models:
  - **Random Forests:** Ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
  - **Gradient Boosting:** Another method that is used to create an ensemble of weak predictive models, (typically decision trees) the purpose of which, is to mitigate the tendency of individual decision trees to overfit the training data.
  - **Multilayer Perceptrons:** It is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP.

Literature Survey

- Earlier models used features - State, District, Court, Case Type and the Year of Filing
- Why did these models not perform well?
    - Data has sequence in form of multiple hearings. Problem can be thought of predicting sequence of future hearing dates for predicting disposal time. These models don't learn sequence.
    - Features used for earlier models may not be enough to predict the disposal time. Case acts, dates of hearing, purpose of hearing weren't used.
    - For the above set of features, multiple predictions possible (one-to-many map). Learning probability distribution over disposal times can be useful.

Theory

- **Mixture Density Networks (MDNs):** Discriminative models used to learn the distribution $p(\mathbf{y} \mid \mathbf{x})$. They use the outputs of a neural network to parameterise a mixture distribution. A subset of the outputs are used to define the mixture weights, while the remaining outputs are used to parameterise the individual mixture components.
- **Hidden Markov Fields (HMFs)**: Generative, graphical model of a joint probability distribution between the observations and the states. It consists of an undirected graph G = (N, E) in which the nodes N represent random variables and E represents the edges. Each node/state in the graph satisfies the markov property(memoryless). Mathematically, the joint probability distribution is given by

$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^{T} p(y_t|y_{t-1})p(x_t|y_t)$$

Theory

- **Conditional Random Fields (CRFs):** Unlike Hidden Markov Fields, which assume that the current observation is independent of the previous observations, Conditional Random Fields does away with the joint probability distribution between the observations and the states, and instead models the conditional probability of the state sequence **y**, over the observations **x**. It can be modelled as

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x})}\exp\left(\sum_j \lambda_j F_j(\boldsymbol{y},\boldsymbol{x})\right).$$

where

$$F_j(\boldsymbol{y},\boldsymbol{x}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, \boldsymbol{x}, i),$$

- **LSTMs:** They are a special kind of Recurrent Neural Network, that are able to retain past information and are able to use it as a context for making predictions along with the recent information. This makes this model more powerful than the RNNs and also give better accuracy, especially on time series data.

Why These Models?

- Simple time series prediction models try to deal with regression problems of single variable
- Here, result depends on several other features such as State, District, Case Type, Case Act, etc
- Therefore, we require models which can work with categorical features
- Hidden Markov Fields (HMFs), Conditional Random Fields (CRFs) and LSTMs are best suited for this purpose in the Machine Learning community right now. MDNs looks a little promising in our context

Problem Objectives

- To build new set of models incorporating new methodologies and features from the available case summaries for more accurate predictions of:
  - Disposal time, given a new case and its features
  - Next hearing date, given an ongoing case and its history
  - Disposal time, given an ongoing case and its history
- Literature survey suggests that models like MDNs, CRFs and LSTMs are expected to learn better from the data to produce better results.

Data Procurement

- Raw data is scraped from http://services.ecourts.gov.in, useful features are extracted from this data and entered to a database, and finally structured data is exported as CSVs for our use
- Several challenges that keeps inhibiting the process of data procurement:
  - Website often goes offline for several days, or its source code gets changed
  - Unreliable server connection, tight exception handling required
  - Huge data table size, sometimes takes days to scrape the data
- Scraping process overall slow. Efficient methods devised to ensure download of more volume at a time.
- Current statistics:
  - Raw data for 431 districts out of 610 has been scraped
  - All of the useful data from 420 districts have been entered into the database

Cleaning of Case Acts

- Some case acts information was found to be bad
  - Separators meant to distinguish between section numbers missing or noisy
  - Section information of the form ‘443A,212,121,33’ present in database as ‘443adr212121wqw33’
- This information can’t be used the way it is, and needs to be cleaned
- Heuristic approach devised to clean them, which may not be very reliable
  - Maximize the number of 3 digit case acts due to greater likelihood
  - Approach uses dynamic programming
- A better heuristic that we have been working on:
  - Define P(x) as probability that a given valid segmentation of section numbers ‘x’ occurs
  - Length of string mostly less than 15, so number of valid segments small
  - Can find ‘x’ for which P(x) is maximum by checking over all possible values of ‘x’
  - Domain knowledge for establishing relationship between case act and section number required

Cluster Level Data Analysis

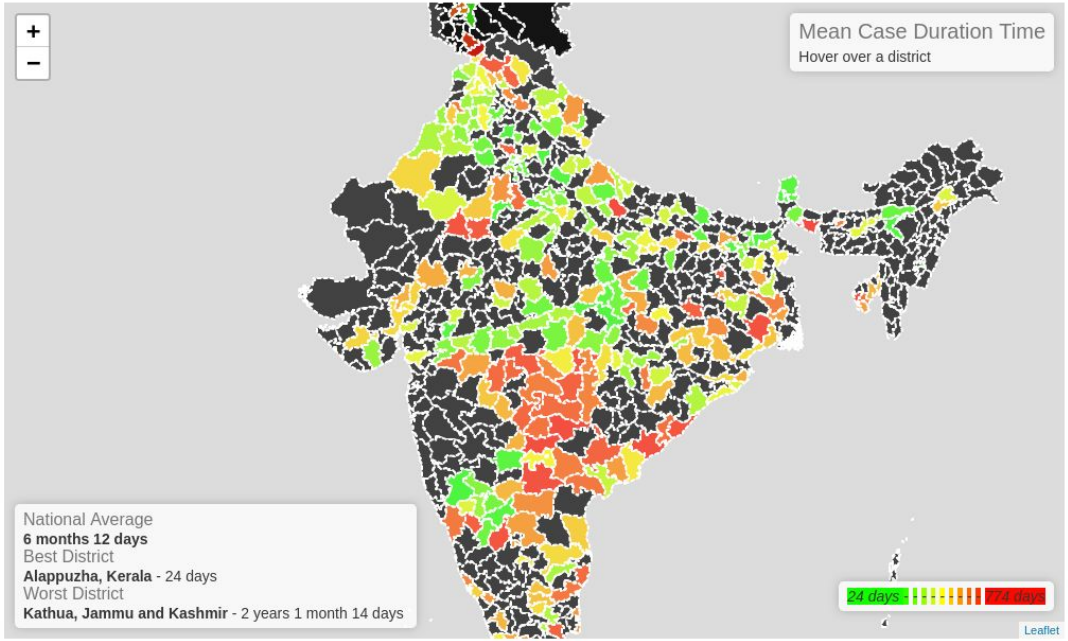
- Case types observed across all the districts were clustered into 16 final case types (called clusters) for the purpose of developing a holistic, standard list
    - Civil Act, Civil Appeal, Civil Arbitration, Civil Application, Civil Case, Civil Petition, Criminal Act, Criminal Appeal, Criminal Arbitration, Criminal Application, Criminal Case, Criminal Petition, Special Case, Small Cause Case, Sessions Case, Other
- Important to understand the behavior of the cases under the bracket of these new case types
    - Mean Disposal Time
    - Zero Day Disposal
    - Tail Bounds
- Results published on project website - https://www.cse.iitd.ac.in/dair/courtanalytics

Indian Legal Dataset Analysis

DAIR, IIT Delhi

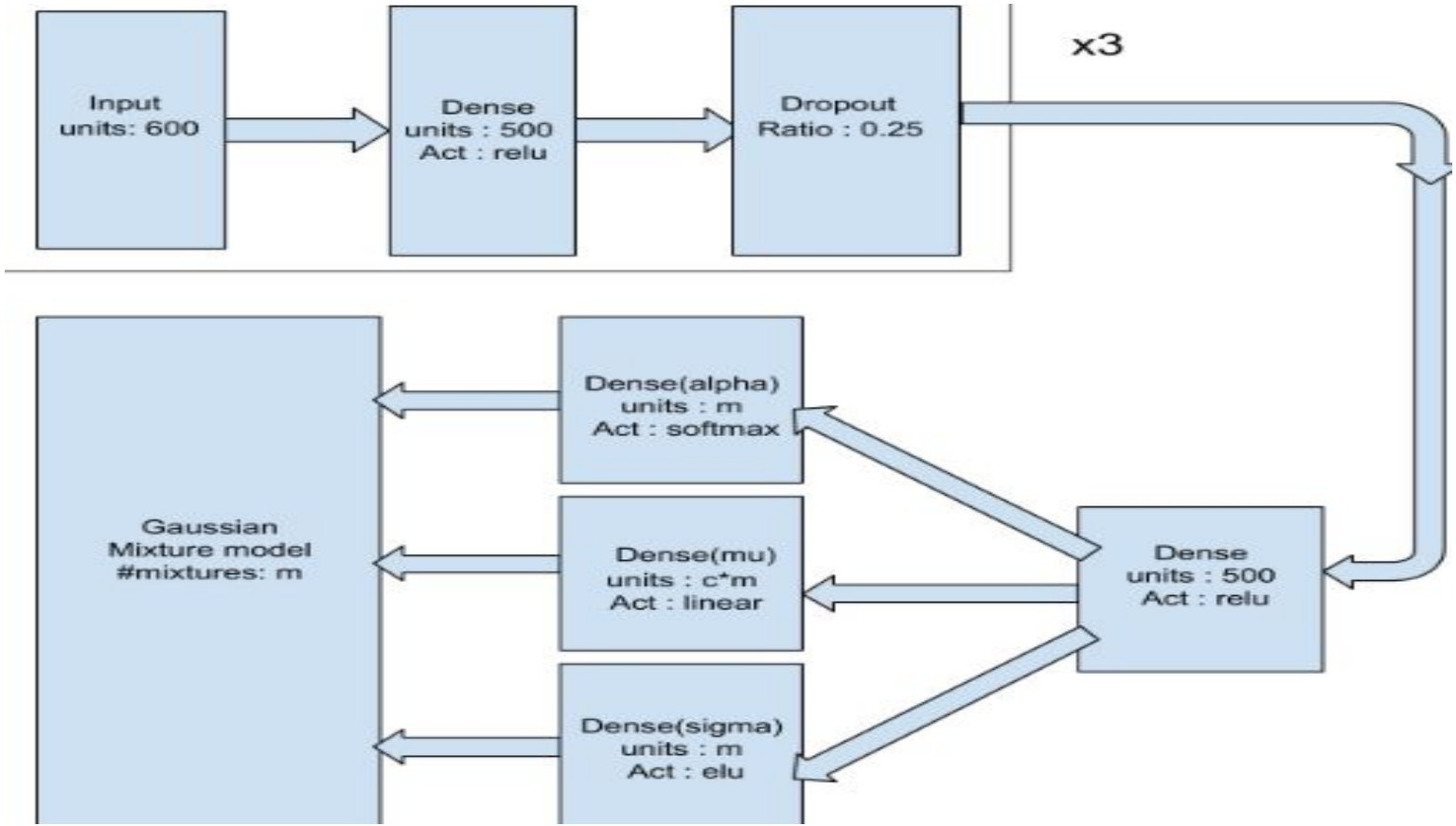
Home | National-Districts | National-States | States | Districts | Compare | Rankings | FAQ | About Us

All case types | Mean Disposal Time



Results - Mixture Density Networks

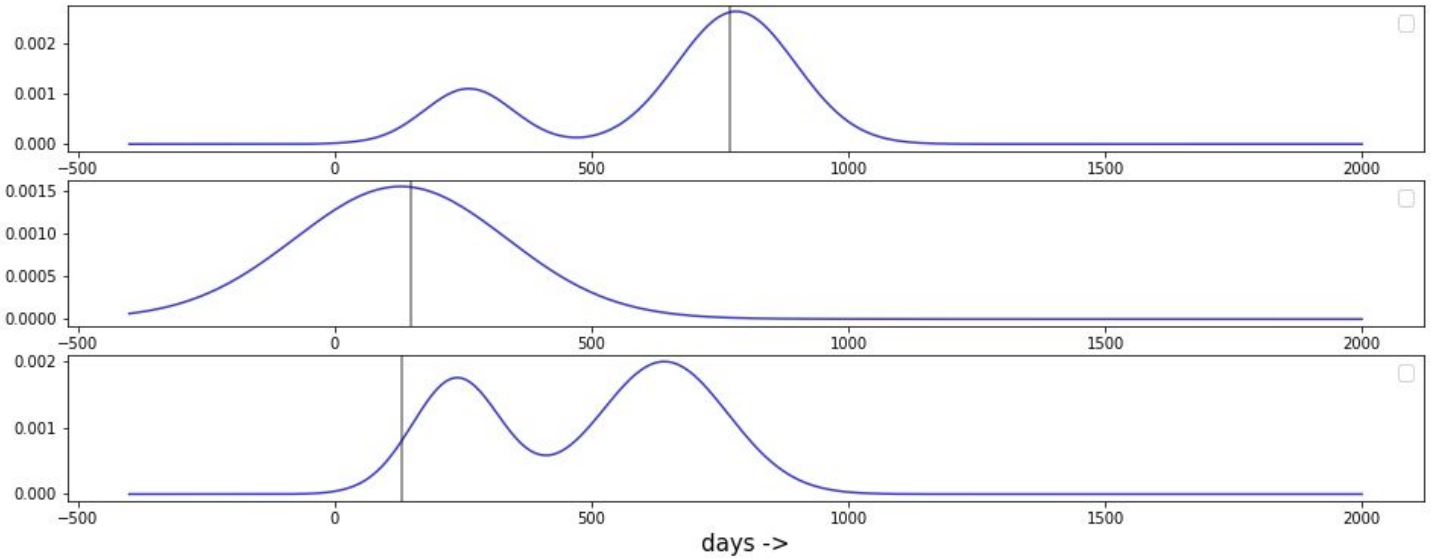
- For our model, the features corresponding to ‘x’ were the same as those used in earlier models:
  - State
  - District
  - Court
  - Case Type
  - Year of Filing
- Aim was to observe how well probabilistic modelling approach works for above set of features compared to the earlier tested models (Random Forest, Gradient Boosting, MLPs)
- We can obtain P(y|x), ‘y’ being the disposal time of a case in days
- Following slide shows the model visualisation


Input
units: 600
Dense
units : 500
Act : relu
Dropout
Ratio : 0.25
x3
Gaussian
Mixture model
#mixtures: m
Dense(alpha)
units : m
Act : softmax
Dense(mu)
units : c*m
Act : linear
Dense(sigma)
units : m
Act : elu
Dense
units : 500
Act : relu

Results - Mixture Density Networks

- Critical observation: Zero day disposals are causing a lot of trouble!
  - For every feature, they accounted for significant portion of disposals - ~5.06% over the dataset
  - They were hurting MDN performance, and removing them improved it by quite a margin
- How would we know whether an unseen example will have a zero day disposal?
  - One of us tried to use Logistic Regression to classify zeros from non-zeros
  - Surprisingly, accuracy: 0.842 with true positive rate: 0.94929 and true negative rate: 0.733095
  - Dense(124)->Relu->Dense(60)->Relu->Dropout(0.25)->Dense(10)->Relu->Dense(1)->sigmoid
  - So before trying to make a prediction, we can filter out the zero day disposals
- MDNs were then trained after removing the zero day disposals

Results - Mixture Density Networks



Results - Mixture Density Networks

- Plots shown represents the probability distribution for following randomly chosen tuples:
  - (Maharashtra; Nandurbar; Civil Court Junior Division, Dhadgaon; Civil Petition; 2014)
  - (Punjab; Gurdaspur; Chief Judicial Magistrate, Taluka Court, Batala; Criminal Application; 2014)
  - (Himachal Pradesh; Kangra; Addl. Chief Judicial Magistrate, TC Nurpur; Civil Petition; 2015)
- Vertical line on the plot represents a ground truth from the test dataset
- Visually, it looks like MDNs do give a decent performance
- But how to assess their performance on a more formal basis?
  - P-value test!

Results - Mixture Density Networks

- For our p-value tests, we first define the following metric for sake of clarity:
  - MAE: It is the mean absolute error from the nearest peak in the distribution, for all the examples with disposal time less than some threshold value
  - VAR: It is the variance of the error in disposal time in test data
- Using the above measures, we make the following observations:
  - For disposal time < 100, MAE = 12.351 days, VAR = 16 days
  - For disposal time < 200, MAE = 20.333 days, VAR = 26 days
  - For disposal time < 400, MAE = 39.74 days, VAR = 54 days
  - For disposal time < 800, MAE = 67.74 days, VAR = 124 days
  - For disposal time < 1000, MAE = 93 days, VAR = 150 days

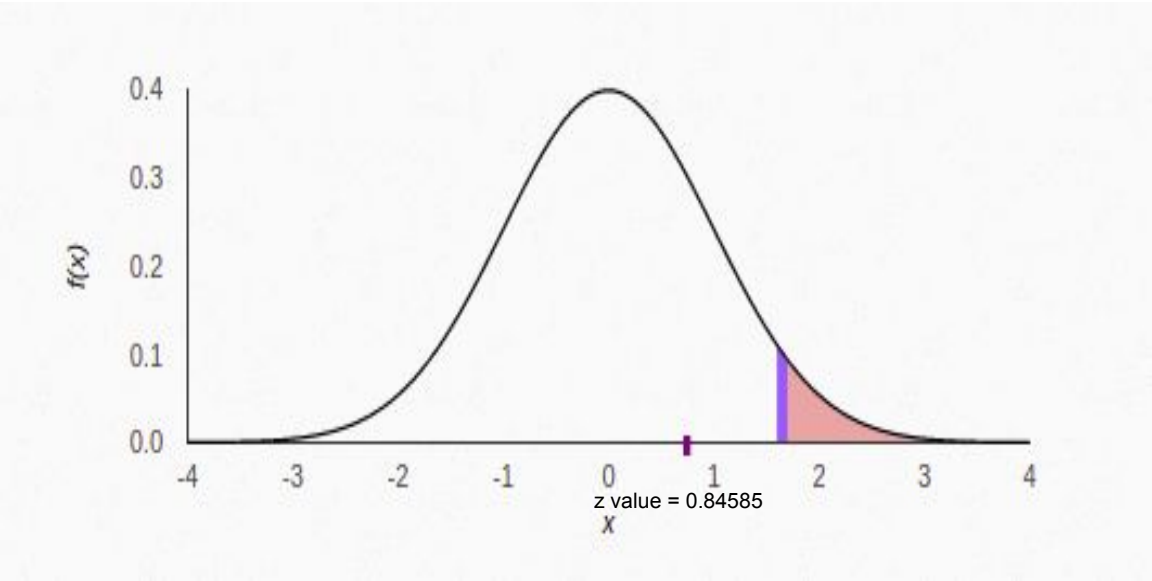
Results - Mixture Density Networks

- We now define our null hypothesis as follows:
  - H_0: MAE <= threshold
  - H_1: MAE > threshold
  - ‘threshold’ that can be adjusted to decide where the null hypothesis fails to reject
  - Test done at 0.05 level of significance
- Some information on the training data:
  - Total training data: 606580 examples
  - Total test data: 151645 examples
- Procedure:
  - Sample some examples randomly from the test data
  - Calculate the z score as z = (MAE - μ) / (σ/sqrt(N)) [σ is the population variance]
  - Use the z score table to apply p-value test and decide whether to reject H_0 or not

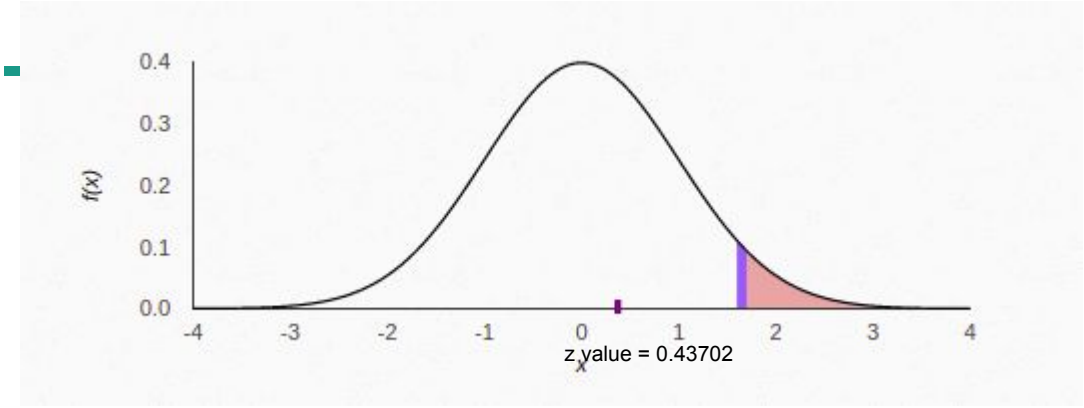
Results - Mixture Density Networks

- Following observations were made:
  - For cases with disposal time < 200, setting threshold = 20 days, sample size = 90356 we get z score = 0.43702 and p-value = 0.33105 > 0.05
  - For cases with disposal time < 200, setting threshold = 19 days, sample size = 10000 we get z score = 0.43702 and p-value = 0.0.004709 > 0.05

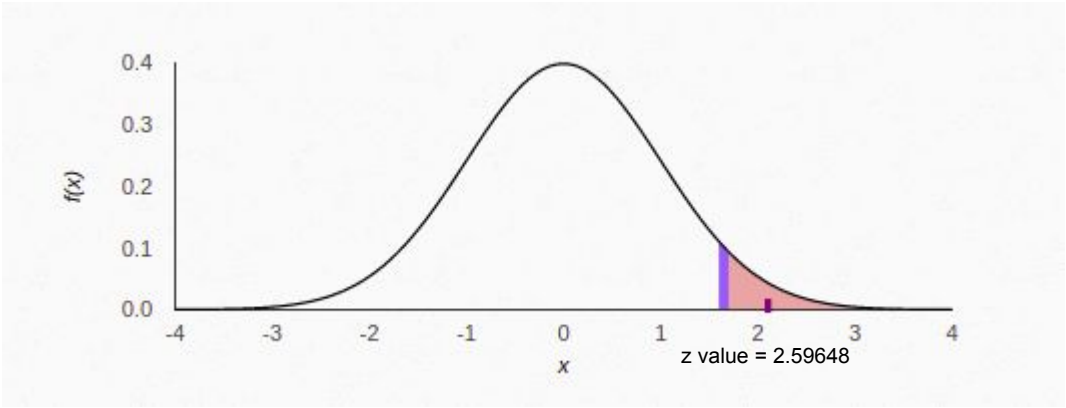
  - For cases with disposal time < 500, setting threshold = 52 days, sample size = 123269 we get z score = 0.33104 and p-value = 0.2899 > 0.05
  - For cases with disposal time < 500, setting threshold = 50 days, sample size = 10000 we get z score = 2.003738 and p-value = 0.02254 > 0.05



Days < 1000, null hypothesis = 93 days



Days < 200,
null hypothesis = 20 day



Days < 500,
null hypothesis = 50 day

Results - Mixture Density Networks

- Some problems that we faced during the entire procedure:
  - Originally, the model was learning roughly same average sample distribution across all features
    - This required lot of hyperparameter tuning to fix
  - As most of the training examples had values in a small range, there was less separation between them
    - Larger values of disposal times were mostly acting as ‘outliers’
    - To increase separation, transformation: $y_{new} = 150(1-\exp(- y_{old}/500))$
    - This gives more importance to the smaller values

Results - How to implement LSTMs?

- Theoretically, the idea of using LSTM for our modelling seems sound
  - But it remains a challenge as to how to feed the features to the model
- Information such as State, District, Court, Case Type can be considered ‘one-time’
  - Don’t need these information at every time step of training
- How should we train the LSTMs then?

Results - How to implement LSTMs?

- One idea - Use LSTMs along with MLPs
  - LSTMs encode the information on hearing dates in a 128 dimensional vector
  - This vector is appended to the one-time information and used to train the MLP
  - Also represents the claim that the more we know about hearing dates, the better model can predict
  - Problem - This model can’t be used for predicting the next hearing date
- Another proposal - Learning LSTM states
  - Train LSTMs separately for the one-time information
  - Remember the internal state of LSTM after training has been done
  - Now when test example comes, LSTM states can be loaded based on the one-time information

Discussion

- Case summaries from from 157 districts have been scraped since the beginning of this BTP.
  - Expected to reach 200, but the challenges discussed earlier inhibited the process to some extent
  - Data being regularly entered to the database to keep things updated
- DP heuristic was tried out for segmenting section no., and a better one was proposed.
- MDNs are able to represent probability distribution over the disposal time reasonably well
  - Can be thought of as a more general model compared to the models tried earlier
- Predictive model incorporating LSTMs for learning from the data has been proposed
  - It is to be seen how it will fare

Conclusion and Further Work

- MDNs exhibited that probabilistic modelling seems to be promising when features are less
  - CRFs model the same distribution as MDNs
  - As CRFs are more complex, we expect a better performance from them
- Cleaning of case acts information will be essential for obtaining contextual information
  - This is because case types are now at a very low level of granularity
- Proposed LSTM models need to be tested for performance