Mortality Prediction from Noisy Time Series Data

Shantanu Thakoor, 13D100003
Karan Vaidya, 130050019
Sai Sandeep, 130050052
Shudhatma Jain, 130050024

Dataset Description

- Time series data of patients: 31 medical readings, age, admittance in ICU
- Time series data has gaps, not all readings taken at each timestamp
- 83% of all data is missing!
- 97% of non-vital data is missing!
- Label for each patient: whether he/she died while in the ICU
- Random noise has been added to each available data

Approaches for training our model

- One training example per patient: find a representative of each feature for each patient
  Does not work well, due to long time series (on day 1, everyone is healthy)
  We cannot predict danger to patients early

- One training example per timestamp: Ideally, we would like to predict a patient being in danger as early as possible; hence we predict on each timestep rather than only the last few, and still give label 1 for timestamps where he/she looked healthy

Tackling Time Series Data

- Exponentially weighted moving average: take average, where weights are exponential in time till end of series scaled down by a constant factor
- Adding features corresponding to variance in the data in certain features based on the recent few values
- Hyperparameter: how much to scale time differences by!

Importance of Oversampling

- The amount of 0’s in the data far exceed amount of 1’s
- Hence, perform oversampling; duplicate data points corresponding to infrequent label
- Also attempted dataset augmentation via adding random noise; however this did not improve significantly
- This may be since original data also had some random noise added to it

From Classification to Regression

- Problem is posed intuitively as a classification problem: based on data at a given timestamp, does the patient live or die?
- Desired: predicting deaths as soon as possible
- Problem: Far from time of death, healthy and dying individuals have similar data
- Solution: Exponentially decay the labels!
  New interpretation: label is probability of patient dying

From Classification to Regression (contd)

- Loss function can still be cross-entropy!
- This is due to probabilistic semantics of the labels
- We can change the decision threshold by binary searching on what gives us highest accuracy. Also experimented with fuzzy thresholds: assigning a random label if score between some bound
- Kind of surrogate loss function used; calculating loss per timestep, even for per person accuracies

Neural Network Model

- For each person, for each timestep, found a score output from neural network
- If greater than X timesteps have score greater than Y, we predict death
- X and Y are hyperparameters to be tuned
- Implemented feed forward neural network with hidden layers of size 100 and 30

Neural Network Model (contd)

- Can be thought of as many weak models, voting together to form a stronger model - ensemble methods

- Accuracy of around 89% achieved
  Model tends to overestimate living

- Future work could include: giving a weight to each of the timestamps, and giving more importance to earlier timestamps

- This would also encourage catching patients in danger, earlier

Adaboost Model

- Continuing on the ensemble methods track, implemented and compared against Adaboost algorithm on decision tree regressors

- Gives much lower accuracy than neural network model (around 11%)

- Possible explanation: since 97% of the data is missing, learning weak trees on a subset of features is not a powerful enough model; less robust to oversampling

Acknowledgements and Future Work

- Implemented and taken cues from paper at http://www.cinc.org/archives/2012/pdf/0261.pdf
- Anonymized data based on http://physionet.org/challenge/2012/set-a/
- Future work: implementing an LSTM to take advantage of inherent sequential nature of data; using more sophisticated surrogate loss
- Use of more sophisticated data wrangling; removing some features based on how many gaps there are, how significant the data is, etc.

Emotion Classification

- Earlier, had worked on emotion classification
- However, certain datasets were not available to us
- Worked using only a very small publicly available dataset
- Achieved 86% accuracy on binary classification (happy vs sad)
- Achieved 42% accuracy on 7-way classification (7 basic emotions)
- However, due to small dataset could not generalize to images our faces
- Hence had to change from our idea of video based emotion recognition