COLLEX - Collecting Labels from Experts

Final Presentation - July 28, 2017
Ahmet Gündüz
Gunnar CS Koenig
Julia M Moosbauer
Franz MJ Pfister
BD Lab

Outline

- Introduction & Background
- Declaration of Transparency
- Technology
- Sprint Overview
- Results
  - Software Engineering
  - Pre-Processing / Cleaning
  - Model Building & Statistical Evaluation
- Key Learnings & Outlook
- Appendix: Sprint Details
Introduction & Background
Parkinson’s Disease

- Second most common neurodegenerative Disease worldwide
- Typical Symptoms:
  - Bradykinesia
  - Rigor
  - **Rest Tremor**
  - Gait Disorders
  - Non-motor symptoms: Depression, urinary dysfunction, autonomous disorders
Purpose & Goal - Specific Use Case

- Application in biomedical research: Parkinson’s Disease (PD)
- (Objective) analysis of PD Symptoms, e.g. rest tremor, is a key success factor in management of the disease
- Many scientific players to solve that problem
- Giant amounts of motion data
- Problem: No / little amount of labels

→ Solution: Fitting various machine learning models enabled by collection of large amounts of expert labels and their meta-information (e.g. confidence)

1 Ching et al 2017, Opportunities and obstacles for deep learning in biology and medicine, published online
Workflow

Patient(s) → Video → Segments/ Snippets → Labels by Raters → Aggregated Label → Machine Learning

Motion Data

Source: © Freepik
Technology used

**LANGUAGES**
Python, R

**FRAMEWORKS**
Django rest framework

**LIBRARIES/PACKAGES**
SciPy, NumPy, Keras, MLR, XGBoost

**DATABASE**
mySQL
Sprint Overview of the Big Data Lab
Sprint Overview

**Sprint 1:** Research and Concept Phase, Data Acquisition

**Sprint 2:** Database Setup / Inner Circle / Model Building 1

**Sprint 3:** Data Cleaning & Model Building 2 (Core)

**Sprint 4:** Model Building 3 / Refine Inner Circle

**Sprint 5:** Statistical Evaluation
Sprint 1: Research and Concept Phase

Goals:

- Definition of Core Technologies
- Acquisition & Preparation of Clinical Research Data
- Component Structure of Project
- Assignment of team roles and responsibilities
- Discussion with supervisors and implementation of feedback

Achievements:
Sprint 2: Database Setup / Inner Circle / Model Building 1

Goals:

- Prototype Inner Circle of the System
  - Process Input Data
  - Setup prototype database
  - Decision on API
- Research on Machine Learning Technology
- Evaluate Backend-Pipeline
- Implementation of first API version
- **Authentication Service**

![Diagram](image-url)
Sprint 3: Data Cleaning & Model Building 2 (Core)

Goals:

- **Connecting the components**
- **Cleaning Service**: Preprocessing of IMU data for Data Science analysis (e.g. read-in, Spectrogram, FFT, Noise reduction, DWT)
- **Machine Learning Component**: Implement Several Machine Learning Models using Test Data
  - Random Forests
  - KNN
  - Trees
Sprint 3: Data Cleaning & Model Building 2 (Core)

Goals:

- **Connecting the components**
- **Cleaning Service**: Preprocessing of IMU data for Data Science analysis (e.g. read-in, Spectrogram, FFT, Noise reduction, DWT)
- **Machine Learning Component**: Implement Several Machine Learning Models using Test Data
  - Random Forests
  - KNN
  - Trees
Sprint 4: Model Building 3 / Refine Inner Circle

Goals:

- Fine-tune current machine learning models
- Add machine learning models, e.g.
  - Logistic / Linear / Spline Regression
  - Boosting (AdaBoost, XGBoost)
  - Support Vector Machine (SVM)
  - Multi Layer Perceptron (MLP)
  - Convolutional Neural Networks (CNN)
  - Long Short Term Memory (LSTM)
- Gathering more labels from more experts
- Refine Inner Circle & Iterations
Sprint 5: Wrap-Up / Statistical Evaluation

Goals:

- Refinement of ML techniques
  - parameter tuning
  - development of cost-sensitive approaches
- Statistical Evaluation of
  - Machine Learning Models
  - Label Statistics
  - Rater Statistics
- Write final report and presentation

Confusion Matrix and Statistics

<table>
<thead>
<tr>
<th>Reference</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>27</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Overall Statistics

- Accuracy : 0.7422
- 95% CI : (0.6574, 0.8154)
- No Information Rate : 0.5547
- P-Value [Acc > NIR] : 9.876e-06
- Kappa : 0.595

McNemar’s Test P-Value : NA

Statistics by Class:

<table>
<thead>
<tr>
<th></th>
<th>Class: 0</th>
<th>Class: 1</th>
<th>Class: 2</th>
<th>Class: 3</th>
<th>Class: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.7324</td>
<td>0.7500</td>
<td>0.8333</td>
<td>0.57143</td>
<td>1.00000</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.8947</td>
<td>0.8152</td>
<td>0.9166</td>
<td>0.98147</td>
<td>0.99206</td>
</tr>
<tr>
<td>Pos Pred Value</td>
<td>0.8566</td>
<td>0.6136</td>
<td>0.58824</td>
<td>0.66667</td>
<td>0.66667</td>
</tr>
<tr>
<td>Neg Pred Value</td>
<td>0.7286</td>
<td>0.8929</td>
<td>0.98198</td>
<td>0.97541</td>
<td>1.00000</td>
</tr>
<tr>
<td>Precision</td>
<td>0.8966</td>
<td>0.6136</td>
<td>0.58824</td>
<td>0.66667</td>
<td>0.66667</td>
</tr>
<tr>
<td>Recall</td>
<td>0.7324</td>
<td>0.7500</td>
<td>0.8333</td>
<td>0.57143</td>
<td>1.00000</td>
</tr>
<tr>
<td>F1</td>
<td>0.8062</td>
<td>0.6750</td>
<td>0.6866</td>
<td>0.61538</td>
<td>0.80000</td>
</tr>
<tr>
<td>Prevalence</td>
<td>0.5547</td>
<td>0.2812</td>
<td>0.89375</td>
<td>0.05469</td>
<td>0.01562</td>
</tr>
<tr>
<td>Detection Rate</td>
<td>0.4062</td>
<td>0.2189</td>
<td>0.87812</td>
<td>0.83125</td>
<td>0.01562</td>
</tr>
<tr>
<td>Detection Prevalence</td>
<td>0.4531</td>
<td>0.3638</td>
<td>0.13261</td>
<td>0.46888</td>
<td>0.02344</td>
</tr>
<tr>
<td>Balanced Accuracy</td>
<td>0.8136</td>
<td>0.7826</td>
<td>0.88649</td>
<td>0.77745</td>
<td>0.99603</td>
</tr>
</tbody>
</table>
Results
of the Big Data Lab
Software Engineering

- Spark/Flink vs Pandas
- SQL vs NoSQL
- Django vs Flask
Known: System Overview

- **Input**
  - Backend Pipeline
  - aggregation service
  - label statistics
  - sampling service
  - frontend
  - experts

- **Active Learning**
- **ML Model Building**
  - ML Model
  - model building service
  - supervised learning dataset
Relational Database Model
Overview of Decisions made

- Backend of Collex System:
  - Flask vs Django
- Framework for Data Preprocessing
  - Spark/Flink vs numpy/pandas
- Data Model
- API specs
- Storage of IMU data
- Storage of Video data
Preprocessing / Cleaning

Challenges:
- Missing Data
- Timestamp correction
- IMU data from different watches
- Matching IMU data with Video segment
Label Collection

- Raw Data from 19 Patients with Parkinson’s Disease (about 10min each), in total n segments of motion data, equals to about **11,400s** of raw motion data
- accounts for **1,275 windows**
- Labels from 4 Expert Raters for 254 snippets each, accounts to **1,016** collected labels
Cleaning & Preprocessing

- Cleaning of Raw Data: Duplicate Timestamps, Transmission Errors, Matching
- Preprocessing of Raw Data (Discrete Wavelet Transform, PSD, Periodogram, Welch Method, Kalman Filtering)
Feature Engineering

- Energy of each segment
- Energy between 3-7 Hz (characteristic frequency band of rest tremor)
- Maximum energy
- Maximum energy between 3-7 Hz
- Power Spectral Density in the frequency bands between 0 and 31 Hz at a 0.5Hz step size
Label Engineering

- 1 Random Rater
- Mean
- Mode
- Aggregation of Agreement
- Cost-Sensitivity
Aggregation Strategies

**Different Aggregation Strategies**

- Mean
- Mode
- Rounded Mean

**Rounded Mean vs Raters**

- Expert Rater
- Rounded Mean
Cost-sensitive learning

- Different ways to do cost-sensitive learning
  - Thresholding
  - Direct weighting
  - Rebalancing the data

- Rebalancing is done by rejection resampling
  - with a probability of $1 - \text{weight}$ reject the observation
  - otherwise add observation to data set

- How to choose weights?
Weighting strategies

\[ \text{weight} = \exp(-\text{std}^4) \]

\[ \text{weight} = (1 - \text{std}/2)_+ \]

\[ \text{weight} = 1_{\text{std}=0} \]
Model Building & Statistical Evaluation

Challenges:
- class imbalance
- label noise
- few data
Interrater Agreement
Comparison of different machine learning approaches

<table>
<thead>
<tr>
<th>Model</th>
<th>1RR</th>
<th>1ER</th>
<th>SSD</th>
<th>Mod</th>
<th>Avg</th>
<th>Agr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression*</td>
<td>-</td>
<td>-</td>
<td>.61</td>
<td>-</td>
<td>.58</td>
<td>1.06</td>
</tr>
<tr>
<td>Log. Regr. L1/L2**</td>
<td>.31</td>
<td>.31</td>
<td>.48</td>
<td>.31</td>
<td>-</td>
<td>.71</td>
</tr>
<tr>
<td>Random Forest**</td>
<td>.68</td>
<td>.64</td>
<td>.61</td>
<td>.69</td>
<td>.41*</td>
<td>.84</td>
</tr>
<tr>
<td>AdaBoost**</td>
<td>.64</td>
<td>.72</td>
<td>.73</td>
<td>.66</td>
<td>-</td>
<td>.81</td>
</tr>
<tr>
<td>XGBoost**</td>
<td>.63</td>
<td>.69</td>
<td>-</td>
<td>.67</td>
<td>-</td>
<td>.83</td>
</tr>
<tr>
<td>SVM**</td>
<td>.51</td>
<td>.61</td>
<td>.58</td>
<td>.51</td>
<td>-</td>
<td>.71</td>
</tr>
<tr>
<td>FF Neural Network**</td>
<td>.42</td>
<td>.51</td>
<td>-</td>
<td>.40</td>
<td>.42</td>
<td>.53</td>
</tr>
<tr>
<td>CNN**</td>
<td>.52</td>
<td>.52</td>
<td>-</td>
<td>.40</td>
<td>.41</td>
<td>.53</td>
</tr>
<tr>
<td>LSTM (raw)**</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.34</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3: Overview of different Model Performances. Abbreviations: SLA = Best Performance Small Label Amount, LLA = Best Performance Large Label Amount, 1RR = 1 Random Rater Baseline, 1ER = 1 Expert Rater, SSD = Small Standard Deviation, Mod = Mode, Avg = Average / Mean, Agr = Agreement, WCS = Weighted / Cost-Sensitive Approach, *MSE, **Accuracy, in **bold**: best performance
How do raters influence results of the ML model?

- in reality: train and evaluate on labels of one rater
- → high variance in rater’s results
- results of one model (same hyperparameters, same evaluation method)

<table>
<thead>
<tr>
<th></th>
<th>Student</th>
<th>Doctor 1</th>
<th>Expert</th>
<th>Doctor 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF classification*</td>
<td>0.688</td>
<td>0.742</td>
<td>0.652</td>
<td>0.640</td>
</tr>
</tbody>
</table>

- Which setting to choose?
- How to find something near to the “ground truth”?* evaluated by 10-fold CV
Linear / Logistic / Spline Regression

Linear Regression (MSE 0.558)

Spline Regression (MSE 0.468)
SVM - without Tuning

- Kernel SVMs: Gaussian Kernel
- Without Tuning: ACC = 0.477 (Mode)
SVM - Tuning

- With Tuning: ACC = 0.529 (Cost-Parameter C and Kernel Width by Grid Search)
SVM - Challenges for Dataset

- Number of Features
  - no accuracy gain
- Class Imbalance
  - SMOTE: ACC = 0.577
AdaBoost

Resample Result
Task: mode
Learner: classif.ada.multiclass
Aggr perf: acc.test.mean=0.654, mmce.test.mean=0.346
Runtime: 79.4886
XGBoost

- Classification with tree boosters (acc 0.673)
- Regression with linear boosters (mse 0.39)
- Tuning by grid search
Random Forest for mode

Confusion Matrix and Statistics

Prediction 0 1 2 3
Reference
0 322 85 27 0
1 182 380 33 1
2 37 42 180 10
3 7 7 27 67

Overall Statistics
- Accuracy: 0.6941
- 95% CI: (0.68, 0.7198)
- No Information Rate: 0.3749
- P-Value [Acc > NIR]: < 2e-16
- Kappa: 0.5612
- McNemar’s Test P-Value: 0.05879

Statistics by Class:

Class: 0 Class: 1 Class: 2 Class: 3
Sensitivity 0.6946 0.6955 0.6792 0.72826
Specificity 0.894 0.8395 0.9859 0.96534
Pos Pred Value 0.7345 0.6955 0.6545 0.62037
Neg Pred Value 0.9226 0.8395 0.9150 0.9758
Precision 0.7345 0.6955 0.6545 0.62037
Recall 0.6946 0.6955 0.6792 0.72826
F1 0.7140 0.6965 0.6667 0.6780
Prevalence 0.3769 0.3451 0.2078 0.07216
Detection Rate 0.2684 0.2400 0.1412 0.05255
Detection Prevalence 0.3545 0.3451 0.2157 0.08471
Balanced Accuracy 0.7720 0.7675 0.7926 0.84580

RF Feature Importance

mode (19 features), filter = information.gain
Random Forest cost-sensitive

Accuracy : 0.7969
95% CI : (0.7167, 0.8628)
No Information Rate : 0.375
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.7074
Mcnemar’s Test P-Value : NA

<table>
<thead>
<tr>
<th>Class: 0</th>
<th>Class: 1</th>
<th>Class: 2</th>
<th>Class: 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>0.9048</td>
<td>0.7708</td>
<td>0.7000</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.8837</td>
<td>0.9125</td>
<td>0.9388</td>
</tr>
<tr>
<td>Pos Pred Value</td>
<td>0.7917</td>
<td>0.8409</td>
<td>0.7778</td>
</tr>
<tr>
<td>Neg Pred Value</td>
<td>0.9500</td>
<td>0.8690</td>
<td>0.9109</td>
</tr>
<tr>
<td>Precision</td>
<td>0.7917</td>
<td>0.8409</td>
<td>0.7778</td>
</tr>
<tr>
<td>Recall</td>
<td>0.9048</td>
<td>0.7788</td>
<td>0.7000</td>
</tr>
<tr>
<td>F1</td>
<td>0.8444</td>
<td>0.8043</td>
<td>0.7368</td>
</tr>
<tr>
<td>Prevalence</td>
<td>0.3281</td>
<td>0.3750</td>
<td>0.2344</td>
</tr>
<tr>
<td>Detection Rate</td>
<td>0.2969</td>
<td>0.2891</td>
<td>0.1641</td>
</tr>
<tr>
<td>Detection Prevalence</td>
<td>0.3750</td>
<td>0.3438</td>
<td>0.2109</td>
</tr>
<tr>
<td>Balanced Accuracy</td>
<td>0.8942</td>
<td>0.8417</td>
<td>0.8194</td>
</tr>
</tbody>
</table>
Cost-Sensitive Approach & Agreement Aggregation

- tested with Random Forest
- Full agreement portion: 19.4%

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Small Dataset</td>
<td>.84</td>
</tr>
<tr>
<td>(b) Large Dataset</td>
<td>.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF 1 randomly picked rater</td>
<td>.65</td>
</tr>
<tr>
<td>RF mode of all raters</td>
<td>.69</td>
</tr>
<tr>
<td>RF /w cost-sensitivity</td>
<td>.80</td>
</tr>
</tbody>
</table>
Random Forest regression

- MSE of 0.417
- even there weighting according to raters confidence achieved good results
Feed Forward NN

- lower performance than classical machine learning methods
- aggregation approach slightly improved performance
Deep Learning: CNN & LSTM

- too little data
- difficult to choose hyperparameters
- method not as robust, sensitive to variance in labels
Class Imbalance

→ SMOTE 5nn
→ class weights
Conceptual Considerations (1)

**Feature Engineering**
- Manual/Automatic feature engineering
- dimensionality reduction
- filtering
- data augmentation
- outlier removal
- balancing of classes
- ...

**Label Engineering**

Source: Leo Breiman (2001) Statistical Modeling: The Two Cultures
Conceptual Considerations (2)
Summary & Key Learnings

- **Aggregation of multiple labels** is a very helpful method to infer ground truth
- **Aggressive Sample Weighting** is gaining accuracy
- **Cost-Sensitive Approach**, which takes the costs for each label within one segment into account is gaining accuracy
- Large **Inter-Rater Disagreement** / Variability
- Inter-Rater Variability is very valuable in inferring ground truth
- **Quality of Labels** over **Quantity of labels** for medical data?
Outlook

- Add Active Learning Component
- Refining the current use case
  - Collecting more labels from more experts
  - Collecting more data (more severe cases)
- Other use cases
is a formalized framework system to collect large-scale, high-quality, correctly labeled training data from experts for machine learning applications in the biomedical domain.
Team Members & Roles

Ahmet Gündüz: Software Development, Preprocessing, Deep Learning

Gunnar CS König: Software Development, Preprocessing, Deep Learning

Julia M Moosbauer: Software Development, Statistics, Model Building

Franz MJ Pfister: Software Development, Model Building, Domain Knowledge
Thank you for your attention.

Ahmet Gündüz
Gunnar CS Koenig
Julia M Moosbauer
Franz MJ Pfister