Machine Learning Nanodegree

Capstone Proposal

Yuanqi Wang

December 22, 2017

I Domain Background

This project is based on the "WSDM - KKBox's Music Recommendation Challenge" from Kaggle.[1] The aim of this project is to build a better music recommendation system for KKBox, a music streaming service in Asia. KKBox currently employs "a collaborative filtering based algorithm with matrix factorization and word embedding in their recommendation system."[2]

In this particular project, we aim to improve KKbox's recommendation system by improving the ability to predict how likely a user is going to listen a particular song *again* during certain time window. Since the task is to classify whether a user is going to listen a song again or not, this is in essence, a binary classification problem.

Classification problems fall into the umbrella of supervised learning — perhaps one of most common types of Machine Learning. Classification algorithm has wide applications, such as spam filtering, fraud detection, character detection, just to name a few. Some of the algorithms for tackling a classification problem include:

- Logistic Regression: a model that estimates the probability of dependent variable belonging to a class on its input features.[4]

- Decision Trees: a model that splits the datasets and classify instances based on

[1] <www.kaggle.com/c/kkbox-music-recommendation-challenge>

[2] <www.kaggle.com/c/kkbox-music-recommendation-challenge> A recommender system is "a technology that is deployed in the environment where items (products, movies, events, articles) are to be recommended to users (customers, visitors, app users, readers) or the opposite."[3] Loosely speaking, there are two types of recommendation system: content based and collaborative-based. See <www.towardsdatascience.com/how-to-build-a-simple-song-recommender-296fcbc8c85> Content-based system recommends item based on what the history of a particular user, while collaborative-based system recommends item to a user based on what other similar users liked.See <www.analyticsvidhya.com/blog/2015/10/recommendation-engines/>

[4] Hosmer Jr, David W., Stanley Lemeshow, and Rodney X. Sturdivant. Applied logistic regression. Vol. 398. John Wiley & Sons, 2013.

certain criterion.[5] A few of the most well-known decision trees algorithms are C4. 5 (developed by Ross Quinlan) and CART (Classification and Regression Trees).[6]

- Support Vector Machine ("SVM"): a model that find a separating boundaries that maximizes the margin.[7]

- Neural Network: a model "made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."[8]

- Ensemble methods: an algorithm that classifies new instancing by combining the decisions made by a set of models.[9]

Even though classification algorithms have been studied extensively and used widely in industry, new research that improve on existing methods and explore new algorithms keep this active and thriving domain.[10] The decision of choosing a particular algorithm is often based on the specific requirement and constraints of the problem at hand (e.g. speed of classification, tolerance to noise, the number of features and instances, etc.)

Empirical studies using Machine Learning techniques have shed lights on this project. For example, Vandenberghe, et al. (2013) used SVM to classify scans

[5]Murthy, Sreerama K. "Automatic construction of decision trees from data: A multi-disciplinary survey." Data mining and knowledge discovery 2, no. 4 (1998): 345-389.

[6]I will be using CART algorithm provided by scikit-learn. See <www.scikit-learn.org>

[7]Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20, no. 3 (1995): 273-297; Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.

[8]"Dr. Robert Hecht-Nielson as quoted in Neural Network Primer: Part I by Maureen Caudill, AI Expert, Feb. 1989." See Bell, Jason. Machine learning: hands-on for developers and technical professionals. John Wiley & Sons, 2014.

[9]Dietterich, Thomas G. "Ensemble methods in machine learning." Multiple classifier systems 1857 (2000): 1-15. Examples of ensemble methods include Bagging, AdaBoosting, and GradientBoosting.

[10]Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.

of the brain as “normal” or ”Alzheimer-like”.[11] Lemmens and Croux (2006) used bagging and boosting to predict customer churn.[12] Pang, Lee and Vaithyanathan (2002) employed Naive Bayes, maximum entropy classification and SVM to classify sentiment of movie reviews.[13]

II Problem Statement

As stated earlier, this is a binary classification problem. The problem is to predict how likely a particular user listens to a particular song for more than one time.

III Datasets and Inputs

The datasets are obtained from Kaggle. It includes five separate files that contain information about the users, songs, and the system environment when the event is triggered:[14]

1. `train`:
    - `msno`: user id
    - `song_id`: song id
    - `source_system_tab`: the name of the tab where the event was triggered. System tabs are used to categorize KKBOX mobile apps functions.

[11] Vandenberghe, Rik, Natalie Nelissen, Eric Salmon, Adrian Ivanoiu, Steen Hasselbalch, Allan Andersen, Alex Korner et al. ”Binary classification of 18 F-flutemetamol PET using machine learning: Comparison with visual reads and structural MRI.” NeuroImage 64 (2013): 517-525.

[12] Lemmens, Aurlie, and Christophe Croux. ”Bagging and boosting classification trees to predict churn.” Journal of Marketing Research 43, no. 2 (2006): 276-286.

[13] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. ”Thumbs up?: sentiment classification using machine learning techniques.” In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp. 79-86. Association for Computational Linguistics, 2002.

[14] See: www.kaggle.com/c/kkbox-music-recommendation-challenge/data

- `source_screen_name`: the name of the layout a user sees.
- `source_type`: the entry point a user first plays music on mobile apps. An entry point could be album, online-playlist, song... etc.
- `target`: target variable. target=1 means there are recurring listening event(s) triggered within a month after the users very first observable listening event, target=0 otherwise.

2. `members`:
   - `msno`: user id (for merging)
   - `city`
   - `age`
   - `gender`
   - `registration method`
   - `registration_init_time`: format %Y%m%d
   - `expiration_date`: format %Y%m%d
3. `songs: song id, song length, genre ids, artist name, composer, lyricist, language`
   - `song_id`
   - `song_length`: in ms
   - `genre_ids`: genre. Some songs have multiple genres and they are separated by |
   - `artist_name`
   - `composer`
   - `lyricist`

- `language`

4. `song_extra_info: song id, song name, international standard recording code`

    - `song_id`
    - `song name:` - the name of the song
    - `isrc:` International Standard Recording Code, theoretically can be used as an identity of a song. However, what worth to note is, ISRCs generated from providers have not been officially verified; therefore the information in ISRC, such as country code and reference year, can be misleading/incorrect. Multiple songs could share one ISRC since a single recording could be re-published several times.

The number of observations in the training set is 7,377,418. After merging the `train, members, songs,` and `song_extra_info`, there are a total of 19 features (excluding `target`). These features could be informative in calculating the likelihood of a song being listened again by a user. Certain variables are categorical variables (e.g. genre_ids, , etc), therefore they need to be transformed through one-hot encoding. Several variables have relatively large amount of missing observations (such as gender, lyricist, etc.), more data exploration will shed light on how these missing values should be treated. Finally, certain variable contains information that needs to be extracted. For example, isrc contains which country the song is published in, and which year it was released. Extracting such information from existing variables will also create a richer dataset. Lastly, approximately 50% of the target variable is 1, while the rest is 0. Therefore, it is a relatively balanced dataset.

IV Solution Statement

I plan to test and compare a variety of algorithm including SVM, Logistic Regression, Random Forests, Boosting, as well as Neural Networks (inspired by the discussion on Kaggle forum).

As a first pass, all available features will be included in the model. Categorical features will be transformed via one-hot encoding. Missing data will either be imputed (mean or median for numerical variables, "unknown" as a new category for categorical variables). New features will be generated — the number of times a song has been played by any users; the number of times an artist were listened by any users, the year the song is released, the language of the song, whether the artist and lyricist are the same entity, interaction terms between user and song, etc.

Depending on the features space — if there are a lot more features than instances, I might drop a few features of less importance, to ameliorate the curse of dimensionality. The importance of features can be obtained from implementing a Random Forest methods.

V Benchmark Model

I will use Logistic Regression without any feature engineering as the Benchmark Model.

VI Evaluation Metrics

The evaluation metrics is "area under the ROC curve between the predicted probability and the observed target." or "AUC".[15] Therefore, it is the rank of the probability that matters.

[15] <www.kaggle.com/c/kkbox-music-recommendation-challenge#evaluation>

VII Project Design

This will be an iterative process:

1. Combine datasets
2. Exploratory data analysis (dealing with missing data, observe distribution, etc.)
3. Feature engineering
4. Set up Cross Validation
5. Train and fine tuning different classifiers; Use GridSearch technique when applicable.
6. Compare results (AUC) from different classifiers.
7. Identify the most promising classifier.