CS6301: Machine Learning for Engineers and Scientists
Final Project Report

Tianning Gao — Electrical Engineering — txg150930@utdallas.edu — *50%*

Siva Saket Sripada — BioMedical Engineering — sxs180054@utdallas.edu — *50%*

 # contribution

I. Problem description and motivation

Bike-sharing systems are in place in several cities in the world and are an increasingly important support for multimodal transport system. One of the major problems is that demand and supply at bike stations are not balanced for most of the time. Most bike share systems employ active rebalancing to ease the pressure at peak times. This means transporting a certain number of bikes from inactive stations to more active stations, or between stations and storage, in order to maximize the usage of each bike and ease supply and demand unbalance problems across bike stations at different times.

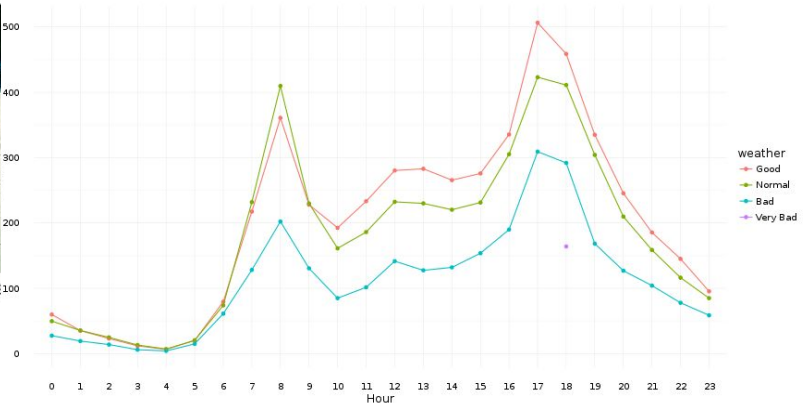
The objective of this project is to find a predictive method for bike demands at each station within one-hour time interval based on features such as date, time slot, workday or not and weather. We will use SVM and Logistic Regression for prediction of number of bikes rented. A third algorithm may be taken into consideration depending on how well these two perform. The models will be built using "sklearn" tool which is written in python.

A data set from UCI Machine Learning Repository will be used. This data set contains records of weather condition during each one-hour time slot, holiday and workdays and number of bikes rented in time slots for every day from 2011 to 2012. All the data will be pre-processed before training.

Tianning will build models based on SVM and Saket will build models using Logistic Regression.

(a) Station-free bike sharing

(b) Bike sharing system with docking st



Bike sharing models had existed in a fixed docking-station paradigm where stations could be spatially distributed in a way to simultaneously minimise user inconvenience and load-imbalance. However,

the more popular paradigm is the station free one that allows users to drop off the bike wherever desired. This however, causes severe imbalance in the demand and availability of bikes not just at a given station but also across stations within a city - especially during the peak/rush hours as can be seen from the time-plot of cycles used.

II. Challenges

Time series forecasting involves that demand of bikes at a given time is dependent not only on current weather conditions but also past weather conditions and, more importantly, on demand at previous time points. To handle this we incorporate auto-regression into our ensemble models. Issues with the data itself include null entries in wind-speed which can wreck the algorithm.

III. The dataset

The original data was obtained from Capital Bikeshare and maintained by UCI Machine Learning repository. The important features are categorical *weather, season and vacation/working day, and temp, wind-speed, humidity, count* and an interesting composite feature *'atemp'* that encodes "feels like temp". Feature "*count*", which denotes bike demand of each one-hour time slots, is what we want the models to predict.

IV. Pre-processing

Data scaling was observed to strip features of relative significance and perhaps thus the algorithmic predictions for few time-points were negative values. Scaling also resulted in slower convergence and poorer performance.

Features Registered(r) and Casual(c) were dropped since they severely skewed weights of learners, effectively reducing the problem to a linear c + r = d.

It is assumed that current bike demand is dependent on data of one or multiple previous time slots. The original dataset was shifted by different time intervals for model training to verify this assumption.

V. Machine Learning Methods

There are many research and discussion on bike sharing demand problem. From these materials, several methods are proved to be effective and accurate. We also choose some classical methods for comparison. Our models are all built with auto-regression with different time delays, which use data from previous time slots for training. Predicted results without auto-regression are recorded for comparison. In addition, we choose two neural networks since they performs well on time series dataset. The following are methods we applied to this problem:

Classical regressors: Support Vector Regressor, Ridge Regressor

Ensemble regressors: Bagging Tree Regressor, Random Forest Regressor, Adaboost Tree Regressor, Extra Tree Regressor

Neural networks for time series regression: RNN and LSTM

The network architecture used for the Neural networks is a dense (512,512) layer in sequence to a RNN/LSTM layer, that has a single output which is the forecasted bike demand. Different optimizers like RMSprop(), SGD(), Nadam() and Adam() were used.

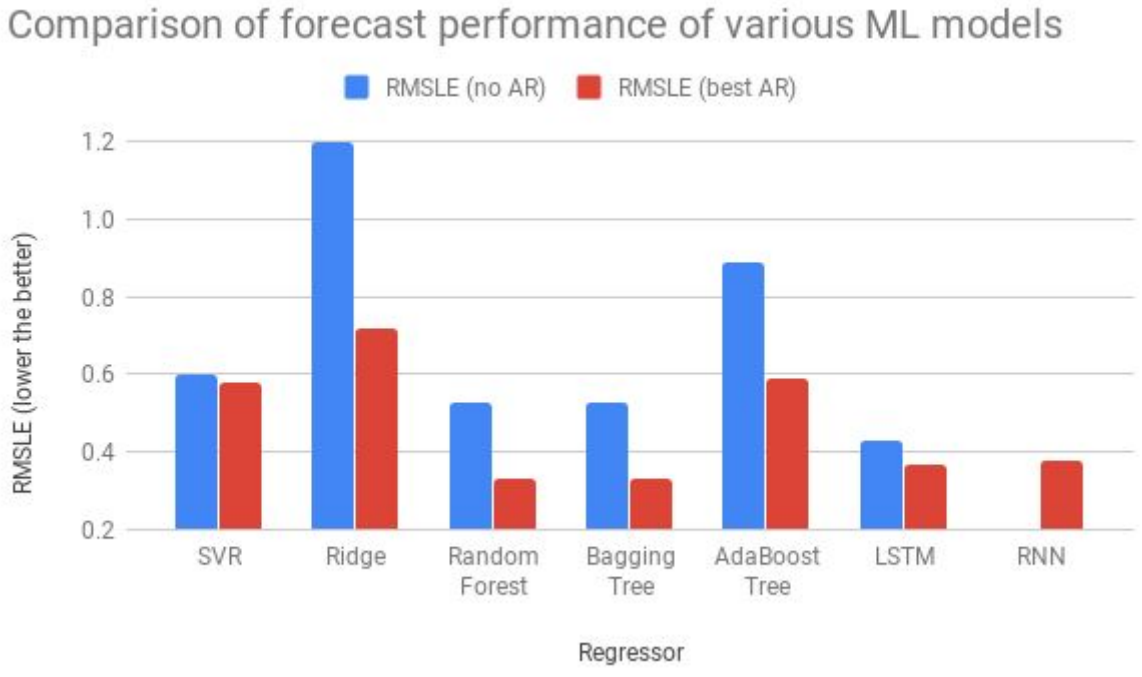
Grid-search cross validation was used for model selection and hyper-parameter tuning for the classical and ensemble learning models. RMSLE reported is on the test test, which is 40% of the total data, obtained using train_test_split function of sklearn.

VI. Results & Analysis

We choose Root-Mean-Square-Log-Error as our evaluation metrics as shown below. The output of a log function increases slower when its input becomes bigger. Therefore, under estimation of models will result in higher error rate than over estimation.

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y\_pred_i + 1) - \log(y\_true_i + 1))^2}$$

Regressor	SVR	Ridge	Random Forest	Bagging Tree	AdaBoost Tree	LSTM	RNN
RMSLE (no AR)	0.60	1.20	0.53	0.53	0.89	0.43	0.45
RMSLE (best AR)	0.58	0.72	0.33	0.33	0.59	0.37	0.37



Our dataset is time series and depends on data in previous time slots. Applying auto-regression takes this dependency into consideration during training and thus the results are improved comparing to models without auto-regression.

The best results are given by Random Forest Regressor and Bagging Tree Regressor which is the best result that other people got so far. Contrary to expectations, SVR and Ridge Regressor do not perform very well. The reason for this might be that the true demand function is discontinuous. Since SVR and Ridge Regressor derives continuous prediction function from training, they possibly won't get any better results than ensemble models with Decision Tree Regressor as base estimator.

Though LSTM and RNN do not give the best result, their validation errors are low enough because they are suitable for time series problems.

It can also be seen from the comparison of RMSLE scores with and without autoregression that Time series forecasting indeed benefits from using previous time points (more than 3 hours did not make sense nor improve our results). The 7 fold improvement in predictions is testimony (2 fold improvement in ln score).

Baseline for our prediction is from the kaggle's public leaderboard with an RMSLE of 0.33756, and the best RMSLE posted was 0.319 with feature modifications. Our RMSLE score of 0.333 is thus in a great standing, having accounted for all features and not modified original feature data.

VII. Conclusion, Discussion and Future Work

In the current project, SVR with polynomial and radial basis kernels were used and grid search cross validation was performed for hyperparameter selection. Similar pipeline was used with Ridge regression and ensemble models. However, for an autoregression time series problem it might be expected that boosting regressors perform better due to sequential learning - something that could be explored to its promising potential in the future by combining sequential boosting and bootstrapping of AdaBoost, xgBoost regressors with the powerful RBF kernel by using SVR as base estimator.

Similarly, LSTMs and RNNs with and without embedding were tested as mentioned in the methods section , however a different architecture (interspersing RNN layers with dense and dropout layers) could improve the performance.

VIII. References

[1] Mrazovic P., Larriba-Pey J.L., Matskin M. "A deep learning approach for estimating inventory rebalancing demand in bicycle sharing systems", IEEE Computer Software and Applications Conference (2018), pp. 110-115

[2] Borgnat P., Abry P., Flandrin P., Rouquier J.-B., Fleury E. "Shared bicycles in a city: A signal processing and data analysis perspective", Adv. Complex Syst., 14 (03) (2011), p. 1100295

[3] Yin Zhang, Haoyu Wen, Feier Qiu, ZieWang, Haider Abbas "iBike: Intelligent public bicycle services assisted by data analytics", Future Generation Computer Systems, Volume 95, June 2019, Pages 187-197

Milestones

1. PCA and drop features and autocorrelation for 'cnt' (y_trn) -
    a. **manifold learning (LLE - locally linear embedding)**
    b. **Doesn't conserve inputspace - interpretability of originality**
    c. **Recursive feature elimination (or use an L1)**
2. Implement autoregression using SVR
3. Choose between HMM, RNN
4. Keep 6 months of 2012 for val and 6 for train?? (based on hyperparameter and model selection)
5. Tune feature "windspeed"??
6. For final project we compare results of a classical, ensemble and NN algorithm
    a. compare SVR, logistic and Ridge regression using Cross_val for alpha selection, Bagging Regressor (Base SVR)
    b. Randomforest / Adaboost regressions (Base SVR?)
    c. Read LSTMs (long short-term memory), HMMs (Hidden Markov Models) and DBNs (Dynamic Bayesian Nets)

Tianning

Saket

Findings

Algorithms previously used with great success:

1. ridge regression
2. Adaboost regression
3. multi-input and multi-output deep learning model
    a. Mrazovic P., Larriba-Pey J.L., Matskin M.**A deep learning approach for estimating inventory rebalancing demand in bicycle sharing systems,** IEEE Computer Software and Applications Conference (2018), pp. 110-115
4. XGBOOST regression
5. Bagging Regression (Base: DecisionTreeRegressor)
6. ICA??

Other algorithms used with slightly lesser success (Borgnat P., Abry P., Flandrin P., Rouquier J.-B., Fleury E.**Shared bicycles in a city: A signal processing and data analysis perspective,** Adv. Complex Syst., 14 (03) (2011), p. 1100295):

1. support vector regression
2. stochastic prediction trees
3. Gradient-enhanced regression trees

→ https://www.sciencedirect.com/science/article/pii/S0167739X18322787#b22

https://www.kaggle.com/amelnozieres/bike-sharing-demand-rmsle-0-3194/code

https://github.com/hmmlearn/hmmlearn

https://github.com/MarvinBertin/HiddenMarkovModel_TensorFlow

https://github.com/dwiel/tensorflow_hmm

Problem Description and Motivation:

Demand forecasting

Dynamic load balancing

Time series regression, etc.,)

• **Challenges**: describe the main challenges one would face when dealing with this domain/data; this can be challenges in data gathering, representation, formulation,learning, scalability or domain-specific challenges

• **Machine Learning Methods**: describe what ML techniques you will use and why you chose them; briefly describe the implementation (or provide links to resources you're using)

• **Evaluation Plan**: describe your experimental setup, and especially your baselines; what metrics will you use to compare your methods

• **Results and Discussion**

• **Conclusions**: discuss your results (including negative results), what you've learned and possible future work