Interactive Machine Learning Interface

Lilong Jiang(jiang.573)

1 Introduction & Motivation

Interactive visualization analysis is an important research area in information visualization, databases and human-computer interaction(HCI). Previous works and commercial products (e.g. Tableau [1]) focus on data analysis area, e.g. how to dynamically filtering datasets [1, 6], zooming and panning [13], brushing and linking [3, 8]. Recently, more works focus on the machine learning and data mining algorithms [2, 15, 11, 14]. There several motivations for this trend. Firstly in many cases, there exists parameter tuning stage for machine learning algorithms and whether the point belongs to something (e.g. cluster, outlier) really depends on what metric to use, what threshold is set, etc. Secondly, interactive visualization provides much better usuability than traditional programming method. And non-experts can easily use this interfae. Thirdly, it is often common for users to verify results because of multiple reasons, e.g. overfitting and underfitting. In summary, it is necessary to provide an usable interactive interface allowing users to adjust machine learning algorithms on-line.

2 Work

The overall user interface (Figure 1, Figure 2, Figure 3) is divided into two parts: the left part shows what kind of machine learning method to run and parameters associated with the algorithm. Also users can choose different dataset and it will show the attributes of current manipulated dataset. The right part shows the visualization of dataset and result. We implement interactive interfaces for three machine learning algorithms in the frontend using javascript, d3 [2], and Tornado [3]: K-means [4] (Figure 1), linear regression [5](Figure 2) and support vector machine (SVM) [6] (Figure 3). Table 1 summarizes operations that users can perform with our interface.

[1] http://www.tableau.com/

[2] https://d3js.org/

[3] http://www.tornadoweb.org/

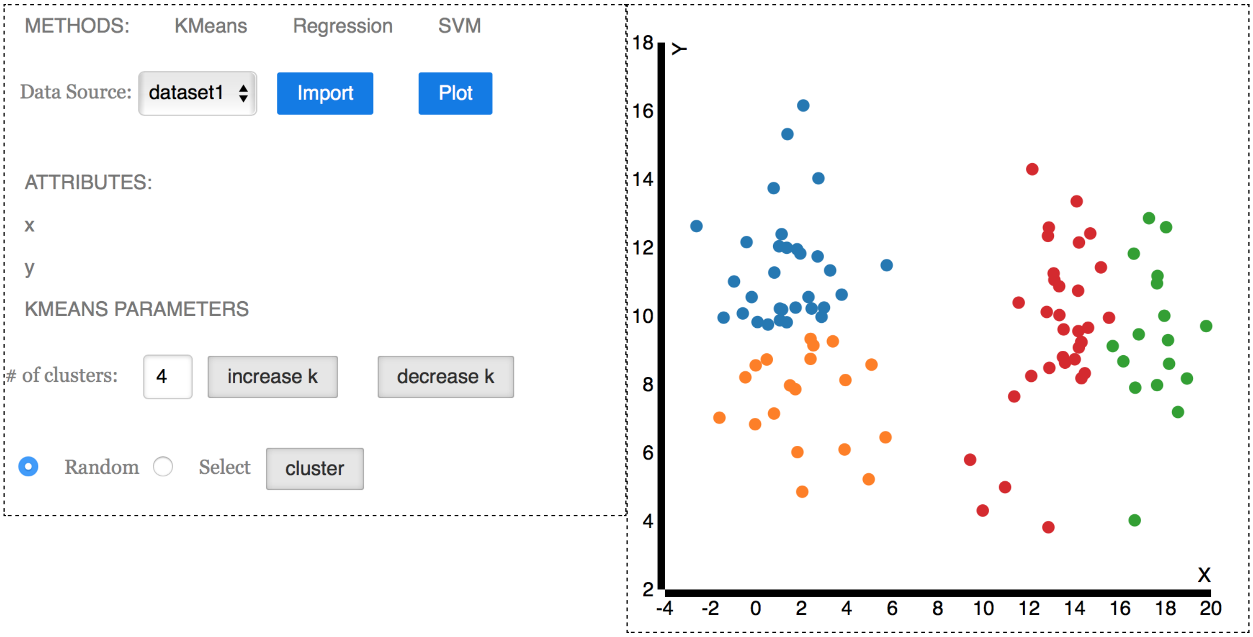
[4] https://github.com/emilbayes/kMeans.js

[5] https://github.com/Tom-Alexander/regression-js

[6] https://github.com/karpathy/svmjs

algorithms	operations
k-means	(1) Select initial centers or generate centers randomly (2) Adjust number of clusters on-line (3) Color shows different clusters
linear regression	(1) Adjust function's intercept by moving the line (2) Adjust function's slope and intercept by dragging end points (3) Dynamically show the function and square loss
svm	(1) Select different kernel (2) Select which features to use by dragging into or out space (3) Dynamically show accuracy, precision, recall and confustion matrix

Table 1: Operations

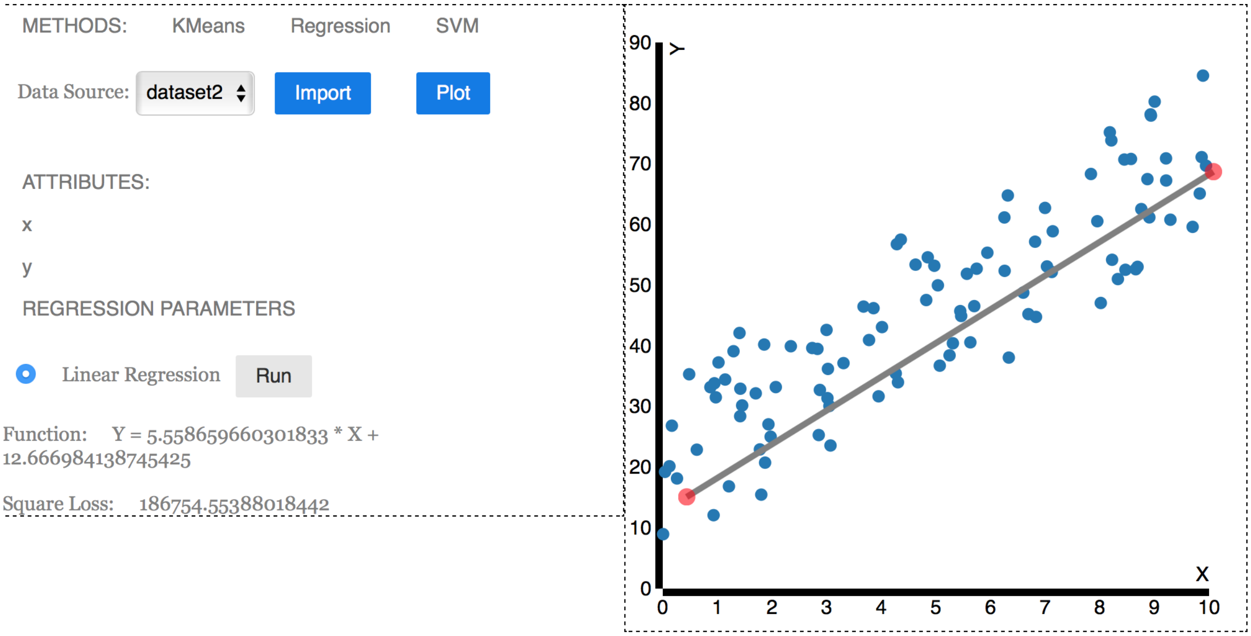


Data Source: dataset1

# of clusters: 4

● Random ○ Select

Figure 1: K-means



Data Source: dataset2

● Linear Regression

Function: Y = 5.558659660301833 * X + 12.666984138745425

Square Loss: 186754.55388018442

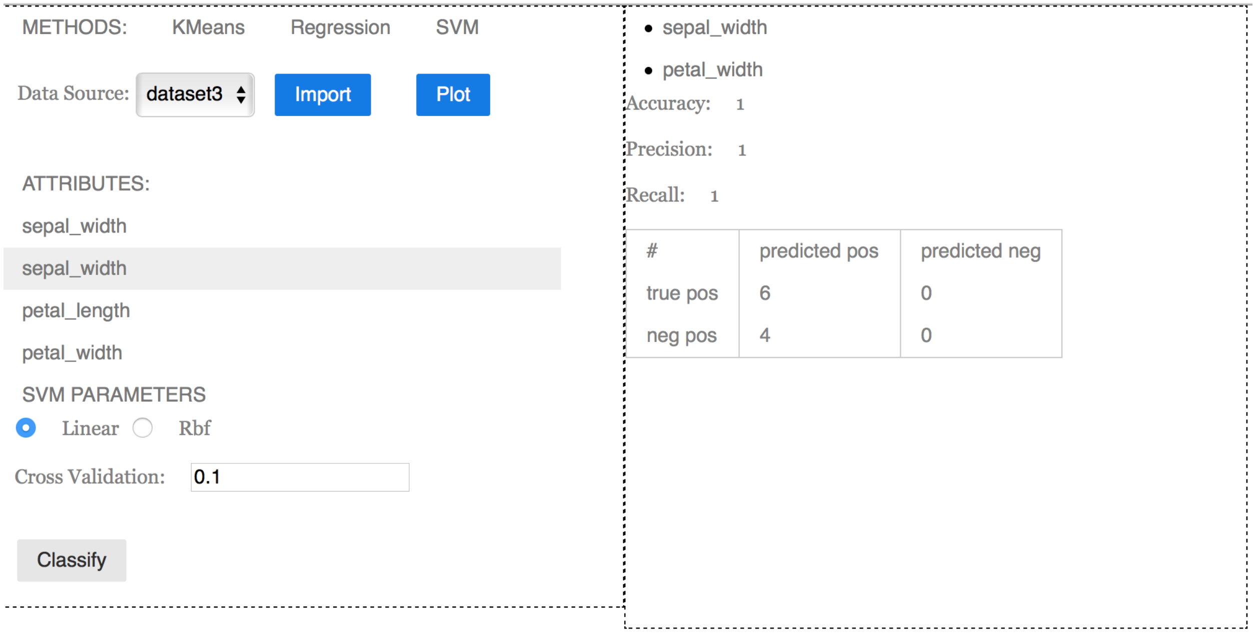
Figure 2: Linear Regression

3 Performance Experiments

In this section, we focus on performance experiments since in the interactive environment, it is necessary to keep an interactive performance.

Computer Configuration: The experiment is run a MacBook Pro with 2.7 GHz Intel Core i5 and 8G memory.

Dataset: For K-means, we sample from 2 bivariant normal distributions and run K-means for 100 interations. For linear regression, we generate dataset from a linear function and add random noise to each point.



METHODS: KMeans Regression SVM

Data Source: dataset3 Import Plot

ATTRIBUTES:

sepal_width

sepal_width

petal_length

petal_width

SVM PARAMETERS

● Linear ○ Rbf

Cross Validation: 0.1

Classify

- sepal_width
- petal_width

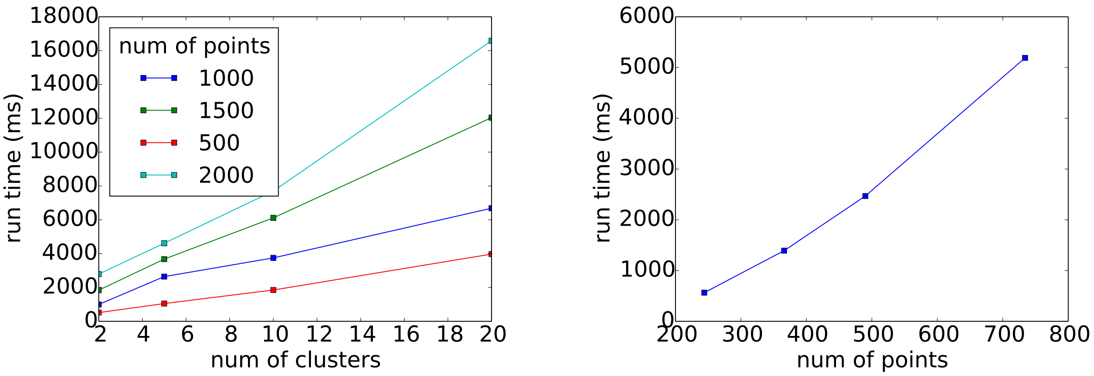
Accuracy: 1

Precision: 1

Recall: 1

#	predicted pos	predicted neg
true pos	6	0
neg pos	4	0

Figure 3: SVM



(a) K-means performance

(b) SVM performance

Figure 4: Performance

For SVM, we use Skin Segmentation Data Set from UCI [12]. There are totally 3 numerical attributes and 2 classes. We randomly sample from both classes.

Results: For linear regressions, for 2 million number of points, it just takes less than 200ms to calculate the regression function and for each update. It is able to keep an interactive performance. However, for k-means and SVM, it is kind of hard. Figure 4a and Figure 4b show that the running time can easily be greater than 1 second for both k-means and SVM. Optimization is necessary for both algorithms to maintain an interactive performance.

4 Limitations & Conclusions & Future Work

In this project, we focus on usability point. There are several limitations for this project. The first is interactive performance when the number of points is large. One method is to run machine learning algorithm in the backend and once it is done, then send it to frontend. However, users need to pay the internet cost.

Another method is to use an incremental method, which update results incrementally as more data is processed. Works including [2, 4] adopts this method.

The second is to visualize high-dimensional dataset, and this can be achieved by projection, which is a technique that maps high-dimensional dataset into a small set of dimensions. There are many methods for projection. Some works use dimension reduction, that create latent dimensions that summarize dataset. For example, Principal Component Analysis (PCA) [9], Multidimensional Scaling (MDS) [10]. This method can reveal hidden variables. However, the generated dimensions are usually less intuitive to users. Another method is called feature selection, which selects subset of features to explore. Interactive feature selection is often used to identify relationship between features [7, 16] and aid user remove redundant features and choose appropriate features. All these methods can be easily incorporated into our project.

In conclusion, we build an interactive machine learning interface which allows users to visualize dataset, run different kind of machine leanring algorithms, and compare results. Incorporating high-dimensional visualizations, expanding linear regression into other kind of regression types, and allowing supporting large datasets are ideal future work.

References

[1] Christopher Ahlberg et al. Dynamic queries for information exploration: An implementation and evaluation. In *SIGCHI*, 1992.

[2] Andrew Crotty, Alex Galakatos, Emanuel Zgraggen, Carsten Binnig, and Tim Kraska. Vizdom: interactive analytics through pen and touch. *Proceedings of the VLDB Endowment*, 8(12):2024–2027, 2015.

[3] Helmut Doleisch et al. Smooth brushing for focus+ context visualization of simulkation data in 3d. 2002.

[4] Danyel Fisher, Igor Popov, Steven Drucker, et al. Trust me, i'm partially right: incremental visualization lets analysts explore large datasets faster. In *SIGCHI*, 2012.

[5] Michael Gleicher. Explainers: Expert explorations with crafted projections. *TVCG*, 2013.

[6] Jade Goldstein et al. Using aggregation and dynamic queries for exploring large data sets. In *SIGCHI*, 1994.

[7] Diansheng Guo. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization*, 2003.

[8] Helwig Hauser et al. Angular brushing of extended parallel coordinates. In *INFOVIS*, 2002.

[9] Ian Jolliffe. *Principal component analysis.* 2002.

[10] A Mead. Review of the development of multidimensional scaling methods. *The Statistician*, 1992.

[11] Eli Packer et al. Visual analytics for spatial clustering: Using a heuristic approach for guided exploration. *TVCG*, 2013.

[12] Abhinav Dhall Rajen Bhatt. Skin segmentation dataset. *UCI Machine Learning Repository.*

[13] Ramana Rao et al. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In *SIGCHI*, 1994.

[14] Cagatay Turkay et al. Interactive visual analysis of temporal cluster structures. In *Computer Graphics Forum*, 2011.

[15] Eugene Wu et al. Scorpion: Explaining away outliers in aggregate queries. *VLDB*, 2013.

[16] Jing Yang et al. Value and relation display for interactive exploration of high dimensional datasets. In *INFOVIS*, 2004.