IoT-Based Garbage Management Information System

Support Vector Regression for Time Series Prediction

Yuanhui Yang*, **Xu Teng**†, **Baiyu Yang**‡ **and Weiyi Zeng**§

Department of Electrical Engineering and Computer Science, Northwestern University

{YuanhuiYang2015*, XuTeng2015†, BaiyuYang2017‡ and WeiyiZeng2015§}@u.northwestern.edu

Abstract

It is the 3rd part of Garbage Collection Management System, focusing on data analysis and time series prediction for information management. It aims to provide a website-based data visualization platform and data mining implementation based on support vector regression technology. As for data visualization, a website (smartgarbagerecycle.com) is built to monitor and record the up-to-date garbage status in the City of Evanston in Illinois. As for data mining, a time series prediction using support vector regression is implemented. As for forthcoming research, the Support Vector Regression Application Programming Interface (API) and the optimized route plan based on the above works will be offered.

Introduction

The Internet of Things (IoT) is a concept in which surrounding objects are connected through wired and wireless networks without user intervention. Garbage management is a primary issue in modern cities. The absence of efficient garbage management information method has caused serious environmental problems and cost issues. As a major application field of IoT, it is providing such an economically efficient garbage management solution.

Support Vector Regression for Time Series Prediction

Overview

Implement Time Series Prediction using Support Vector Regression. Support Vector Regression is one well-known and relatively excellent regression method. Not only performing linear regression, Support Vector Regression can efficiently perform a non-linear regression using kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. Kernel trick includes Linear Function, Radial Basis Function (RBF) and Sigmoid Function. In this research, these 3 kernels will be implemented and compared.

Problem

Given 0^{th} day to $(n-1)^{th}$ day time series array $a[n]$ and then return its next r days data $\hat{a}[r]$

- INPUT: $a[n] = [a_0, a_1, , a_{n-1}]$
- OUTPUT: $\hat{a}[r] = [a_n, a_{n+1}, \cdots, a_{n+r-1}]$

Support Vector Regression

Given training vectors $\vec{x_i} \in \mathbb{R}^p$, where $i = 0, \cdots, n-1$ and a vector $y \in \mathbb{R}^n$ ε-SVR solves the following primal problem:

$$\min_{\omega,b,\zeta,\zeta^*} \frac{1}{2}\omega^T\omega + C\sum_{0}^{n-1}(\zeta_i + \zeta_i^*) \tag{1}$$

subject to:

$$\begin{cases} y_i - \omega^T\phi(x_i) - b \le \varepsilon + \zeta_i \\ \omega^T\phi(x_i) + b - y_i \le \varepsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* \ge 0 \end{cases} \tag{2}$$

Its dual is:

$$\min_{\alpha,\alpha^*} \frac{1}{2}(\alpha-\alpha^*)^T Q(\alpha-\alpha^*) + \varepsilon e^T(\alpha+\alpha^*) - y^T(\alpha-\alpha^*) \tag{3}$$

subject to:

$$\begin{cases} \varepsilon e^T(\alpha-\alpha^*) = 0 \\ 0 \le \alpha, \alpha^* \le C \end{cases} \tag{4}$$

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an n by n positive semidefinite matrix, $Q_{ij} \equiv K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ is the kernel. Here training vectors are implicitly mapped into a higher (maybe infinite) dimensional space by the function ϕ. The decision function is:

$$\sum_{i=0}^{n-1}(\alpha_i - \alpha_i^*)K(x_i, x) + \rho \tag{5}$$

Algorithm Design

Standardization Standardization of a dataset is a common requirement for many machine learning estimators. Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using the transform method.

Learning

1. $[a_0, a_1, \cdots, a_{k-1}]$ is able to determine a_k;
2. $[a_1, a_2, \cdots, a_k]$ is able to determine a_{k+1};
3. $[a_2, a_3, \cdots, a_{k+1}]$ to determine a_{k+2};
4. $\cdots$

That is,

$$\begin{cases} a_k \Leftarrow [a_0, a_1, \cdots, a_{k-1}] \\ a_{k+1} \Leftarrow [a_1, a_2, \cdots, a_k] \\ a_{k+2} \Leftarrow [a_2, a_3, \cdots, a_{k+1}] \\ \cdots\ \cdots\ \cdots \\ a_{n-1} \Leftarrow \left[a_{n-2-(k-1)}, a_{n-1-(k-1)}, \cdots, a_{n-2}\right] \end{cases} \tag{6}$$

Cross-validation Learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called overfitting. To avoid it, cross-validation is common practice.

Prediction Similar to the above learning work,

1. $\left[a_{n-1-(k-1)}, a_{n-(k-1)}, \cdots, a_{n-1}\right]$ is able to predict a_n;
2. $\left[a_{n-(k-1)}, a_{n+1-(k-1)}, \cdots, a_n\right]$ is able to predict a_{n+1};
3. $\cdots$
4. $\left[a_{n+r-2-(k-1)}, a_{n+r-1-(k-1)}, \cdots, a_{n+r-2}\right]$ is able to predict a_{n+r-1}.

That is,

$$\begin{cases} \left[a_{n-1-(k-1)}, a_{n-(k-1)}, \cdots, a_{n-1}\right] \Rightarrow a_n \\ \left[a_{n-(k-1)}, a_{n+1-(k-1)}, \cdots, a_n\right] \Rightarrow a_{n+1} \\ \cdots\ \cdots\ \cdots \\ \left[a_{n+r-2-(k-1)}, a_{n+r-1-(k-1)}, \cdots, a_{n+r-2}\right] \Rightarrow a_{n+r-1} \end{cases} \tag{7}$$

Experimental Results

This research develops 3 kinds of Support Vector Regression (SVR) Application Program Interface (API), Linear-SVR and RBF-SVR as well as Sigmoid-SVR. Of these SVRs, Linear-SVR and RBF-SVR are popular, and this research compare the two with Sigmoid-SVR in terms of robust and accuracy.

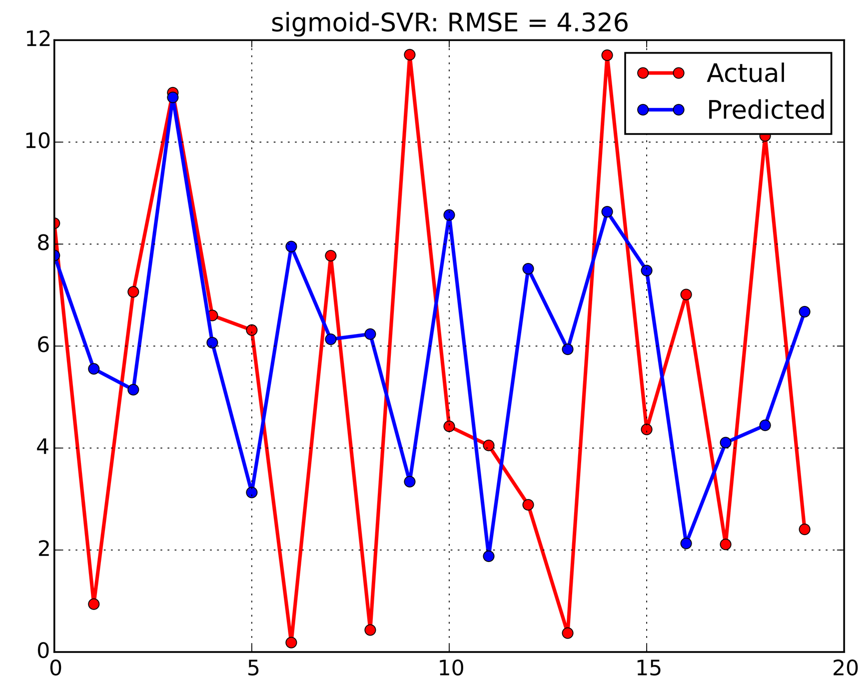


Figure 1: Sigmoid kernel - Support Vector Regression

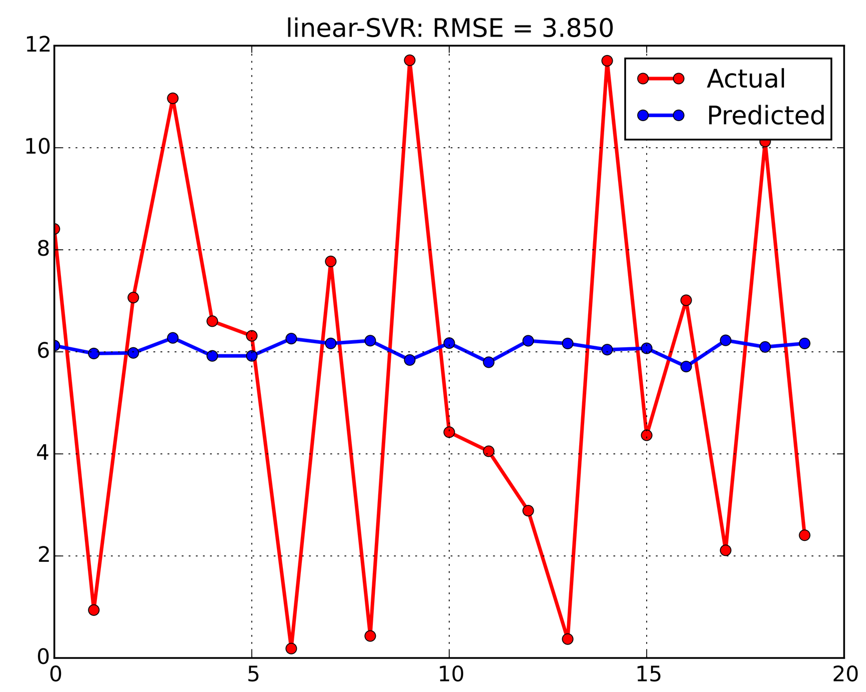


Figure 2: Linear kernel - Support Vector Regression

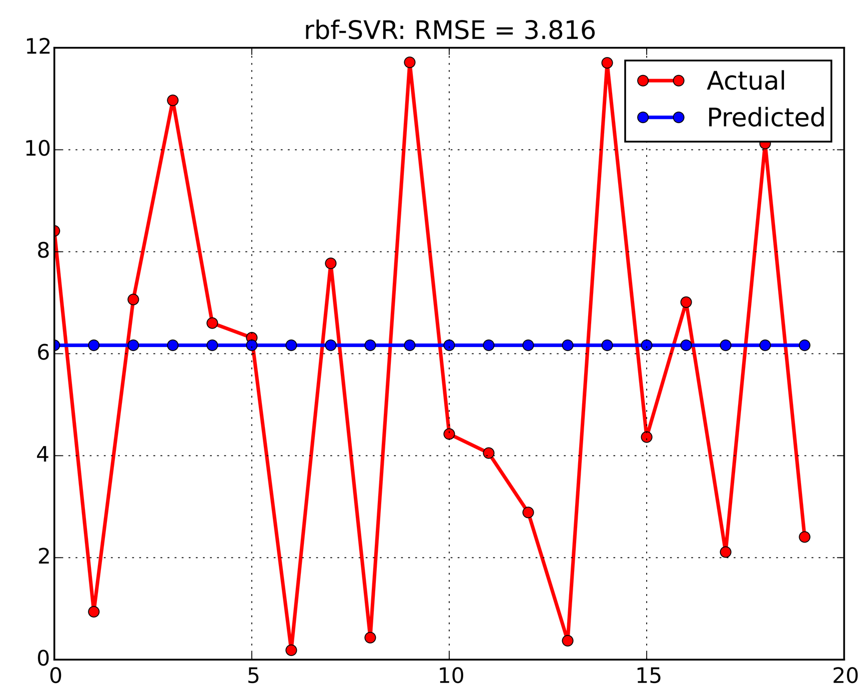


Figure 3: RBF kernel - Support Vector Regression

Conclusions

Comparing with Linear-SVR and RBF-SVR, Sigmoid-SVR owns the strongest robust property and excellent prediction accuracy.

Website

Overview

A fully functional hosted website, smartphone friendly and resizable to fit any size screen. Clean and neat UI design to introduce smart garbage recycle system from hardware to software, from sensors to backend server all the way to the database and frontend development.

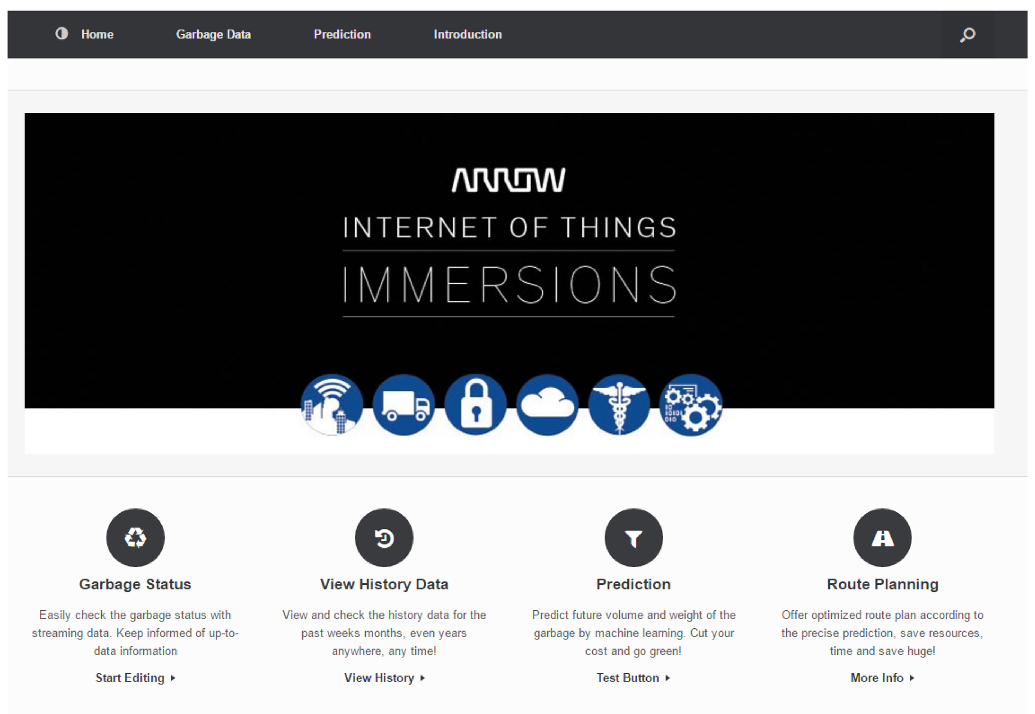


Figure 4: Homepage of smartgarbagerecycle.com

Implement data visualization using WP-Table technology. Allow displaying real time data, applying different filters to narrow down searching range. Note that this website is mobile friendly. The following 2 screenshots are taken on mobile.

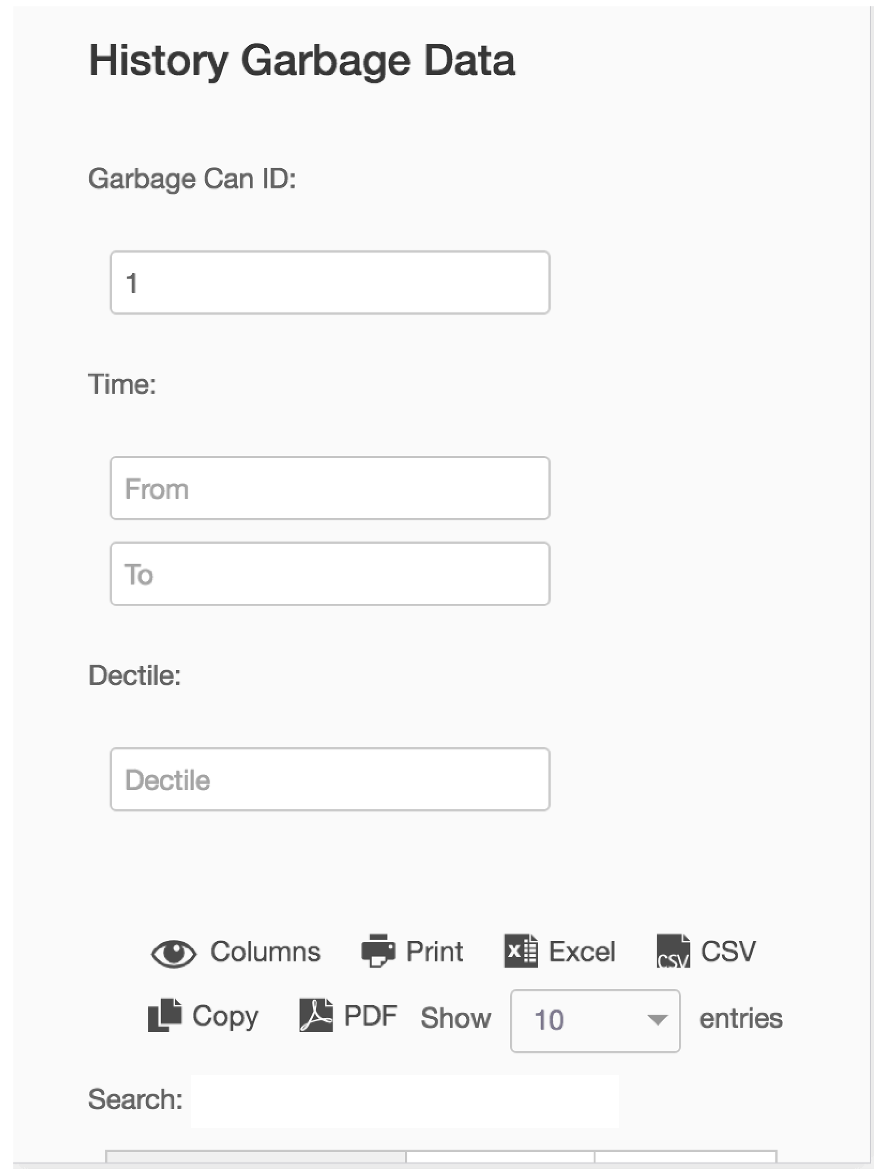
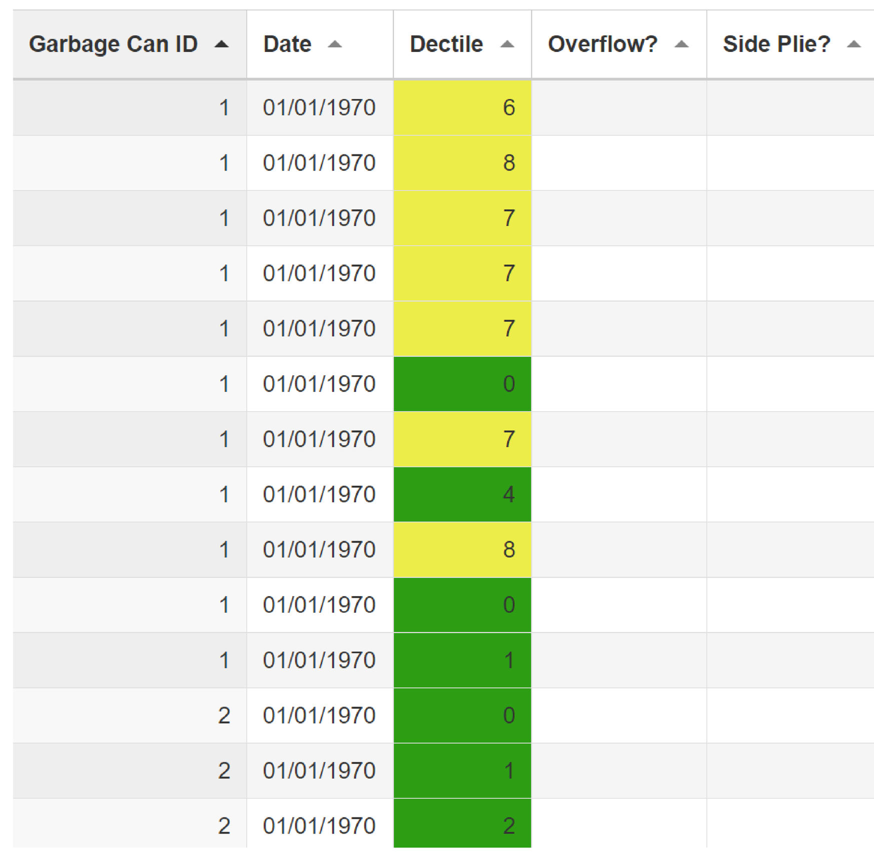


Figure 5: Filter,save and print out the search result into csv file, excel file or pdf file.

Easy to check current status and history data of any garbage can at certain location including the information of garbage can id, time, dectile.



Garbage Can ID	Date	Dectile	Overflow?	Side Plie?
1	01/01/1970	6		
1	01/01/1970	8		
1	01/01/1970	7		
1	01/01/1970	7		
1	01/01/1970	7		
1	01/01/1970	0		
1	01/01/1970	7		
1	01/01/1970	4		
1	01/01/1970	8		
1	01/01/1970	0		
1	01/01/1970	1		
2	01/01/1970	0		
2	01/01/1970	1		
2	01/01/1970	2		

Figure 6: Review history garbage data according to garbage ID, time, and dectile

Potentially check garbage can status on map.

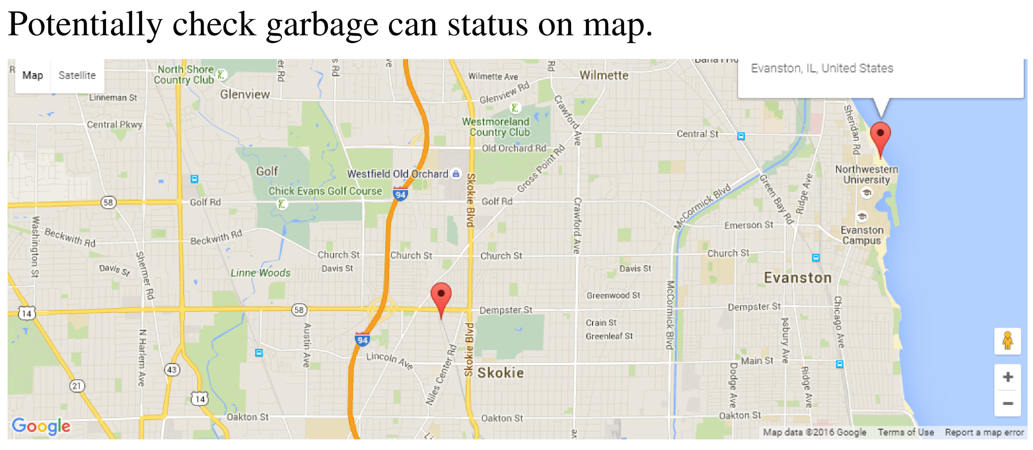


Figure 7: Check garbage can status on map

Forthcoming Research

1. To retrieve real-time data from database and showing different form of data visualization.
2. To generate another layer on the current map and mark the garbage cans such that data could be directly checked on the map.
3. To improve the accuracy of prediction and combine the information from current recycle schedule, more collection data to do route planning.