ATLANTIS PRESS

Advances in Computer Science Research, (ACSR), volume 52

2016 International Conference on Computer Engineering and Information Systems (CEIS-16)

Short Term Load Forecasting Using Core Vector Regression Trained with Particle Swarm Optimization

Xin Sun, Xin Zhang

Beijing Electric Power Economic Research Institute

Beijing, China

E-mail: 18614000110@163.com, 18613838637@163.com

***Abstract*-Short term load forecasting is very essential to the operation of electricity companies. In this paper, we propose a new method for short term load forecasting trained by PSO and Core Vector Regression (CVR). The CVR algorithm extend Core Vector Machine algorithm to the regression setting by generalizing the underlying minimum enclosing ball problem. In this paper, we use particle swarm optimization (PSO) to optimize the parameters of the CVR. Experiments show that the PSO optimized method has comparable performance with SVR (Support Vector Regression), but is much faster and produces much fewer support vectors on very large data sets.**

Keywords-short term load forecasting; core vector regression; PSO; kernel parameter

I. INTRODUCTION

In recent years, Short term load forecasting has been an important topic in power system research. Such as expert systems [1], fuzzy inference [2], artificial neural networks (ANN) [3] which does not need the expression of human experience. Because it based on Back propagation algorithm, ANN does not converge optimally and requires much longer time for training, so it difficult for real-time application [4]~[6]. Among them, support vector machines (SVM) and support vector regression (SVR) are especially successful[7]~[8]. We suppose m is the capacity of training datasets, then the training time complexity is $O(m^3)$ and it has at least quadratic space complexity. Most of the earlier works aimed at predicting one-day loads ahead prediction, but it is computationally infeasible if we need to predict the loads for a long period time on very large data sets. Hence another major stumbling block is, the results of the long-term prediction may degenerate due to the error propagation.

Tsang et al. proposed the Core Vector Machine (CVM) [10]~[12] by observing that practical SVM implementations only approximate the optimal solution by an iterative strategy. Typically, the stopping criterion utilizes either the precision of the Lagrange multipliers or the duality gap. For example, in SMO, *SVMLight* and *SimpleSVM*, training stops when the Karush-Kuhn-Tucker (KKT) conditions are fulfilled within a ε -insensitive parameter.

By utilizing an approximation algorithm for the minimum enclosing ball (MEB) problem in computational geometry, the *CVM's* time complexity is linear in m and space complexity is independent of m. Experiments on large data sets also demonstrated that the CVM is as accurate as existing SVM implementations, but is much faster and can handle much larger data sets than existing scale-up methods. And then they propose an enhancement of the CVM that allows a more general quadratic programming (QP) formulation (CVR) [13]. It turns out that this allows the condition on the kernel to be lifted.

In this paper, we propose a PSO-CVR algorithm that allows continuous forecasting a period of loads with a large scale dataset and the CVR parameters do not need to specify manually.

II. CORE VECTOR MACHINE

A. Core Vector Machine

Given a set of points $S=\{x_1, \ldots, x_m\}$, where $x_i \in \mathbf{R}^D$, the minimum enclosing ball of S (denoted MEB(S)) is the smallest ball that contains all the points in S. Let $B(c, R)$ be the ball with center c and radius R. Given an $\varrho > 0$, a ball $B(c, (1+\varrho)R)$ is an $(1+\varrho)$-approximation of MEB(S) if $R \le r_{MEB(S)}$ and $S \subset B(c, (1+\varrho)R)$. In many shape fitting problems, it is found that solving the problem on a subset, called the core-set, Q of points from S can often give an accurate and efficient approximation. More formally, a subset $Q \subseteq S$ is a core-set of S if an expansion by a factor $(1+\varrho)$ of its MEB contains S, i.e., $S \subset B(c, (1+\varrho)R)$, where $B(c, R) = \text{MEB}(Q)$.

We denote the core-set, the ball's center and radius at the t^{th} iteration by S_t, c_t and R_t respectively. Also, the center and radius of a ball B are denoted by c_B and r_B. Given a $\varrho > 0$, the CVM proceeds as follows:

- Initialize S_0, c_0 and R_0.
- Terminate if there is no training point z such that $\tilde{\varphi}(z)$ falls outside the $(1+\varrho)$-ball $B(c_t, (1+\varrho)R_t)$. Where $\tilde{\varphi}$ is a nonlinear feature mapping function associated with a given kernel k, mapping z to a higher dimensional

space.

- Find (core vector) *z* such that $\tilde{\varphi}(z)$ is furthest away from c_t. Set $S_{t+1} = S_t \cup \{z\}$. This can be made more efficient by using the probabilistic speedup method in (*Smola & Sch¨olkopf*, 2000) that finds a z which is only approximately the furthest.
- Find the new MEB (S_{t+1}) and set $c_{t+1} = c_{MEB(S_{t+1})}$ and $R_{t+1} = r_{MEB(S_{t+1})}$.
- Increment *t* by 1 and go back to Step 2.

B. Core Vector Regression

Applicability of the CVM procedure depends on the following two conditions being satisfied:

1) k(x, x) is a constant.

2) The QP problem is of a special form: In particular, there is no linear term in the QP's objective. However, the dual objective of SVR contains a linear term and so is not of the required form.

The MEB finds the smallest ball containing all $\tilde{\varphi}(z_i)$'s in *S*. Tsang et al. augment each $\tilde{\varphi}(z_i)$ by an extra $\Delta_i \in R$ to form $[\tilde{\varphi}(z_i)', \Delta_i]'$, and then find the MEB for these augmented points, while constraining the last coordinate of the ball's center to be zero.

The algorithm in Section 2.1 can now be modified accordingly as:

- Initialize S_0, c_0 and R_0.
- Terminate if there is no training point zi such that $\tilde{\varphi}(z_i)$ falls outside the (1 + ϱ)-ball B (ct, (1 + ϱ) Rt). i.e., $\sqrt{\|c_t - \varphi(z_i)\|^2 + \Delta_i^2} > (1 + \varrho)$ Rt. where φ is a feature mapping function associated with a given kernel k, mapping zi to a higher dimensional space.
- Find z_i such that $\tilde{\varphi}(z_i)$ is furthest away from c_t. Set $S_{t+1} = S_t \cup \{z_i\}$.
- Find the new MEB (S_{t+1}) and set $c_{t+1} = c_{\mathrm{MEB(St+1)}}$ and $R_{t+1} = r_{MEB(S_{t+1})}$.
- Increment *t* by 1 and go back to Step 2.
- Thus, the only modification is to change the distance computation between c_t and any point z_i (in Steps 2 and 3) to $\sqrt{\|c_t - \varphi(z_i)\|^2 + \Delta_i^2} > (1 + \varrho) R_t$. Moreover, this preserves all the properties of the original CVM algorithm as expected.

III. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization is an idea based on human social behavior. Kennedy and Eberhert in 1994 presented the concept. It models problem as a set of n particles each representing a dimension of solution space. These particles move in solution space in search of optimal solution. The particles follow three principles as follow:

- Evaluating: learning through self-experience;
- Comparing: learning through comparative study;
- Imitating: learning through adapting the best trend.

In PSO, the key is how to choice the weighting function *w*. A larger *w* has better global search capacity, while smaller are in a stronger local search Capacity. Thus, with the increase in the number of iteration, the *w* should continue to reduced.

We denote a position parameter and a velocity parameter to be updated for all the dimensions of all the particles. The updating logic lies with the global best parameter of all the particles and the local best parameter of a single particle. Our PSO is based on continuous state variables with real values. Detailed procedure for PSO is explained as follows:

```
//S: initial swarm, V: velocity, X: position
   Initialize S with random V and X
// w: weighting function, c1 = c2: weighting factors
   Initialize w, c1, c2
//k: number of iteration
  k →
    Begin
       X_i^k :position of a particle
       pbest_i :personal best position of i^th particle
       V_i^k :velocity of i^th particle
       gbest :global best of all the particles of S

//batch: total number of inputs
      batch →
         Begin
            Calculate the output and thus error
         End
      Calculate MSE for each particle in swarm

      For each particle i in swarm
         If( X_i^k < pbest_i )
            Then pbest_i = X_i^k
         If( pbest_i < gbest )
            Then gbest = pbest_i
Position and velocity are then updated as per the following equations:
  V_i^{k+1} = w*V_i^k + c1*r1*(pbset_i - X_i^k) + c2*r2*(gbest - X_i^k)
  X_i^{k+1} = X_i^k + V_i^{k+1}
end
```

IV. MODELING AND FORECASTING PROCESSES

C. Data Analysis

In this paper, the data we used include electricity load and temperature. We also used a list of holidays. The load data set contains the load per half hour of each day from 1997 to 1999 Jan while the temperature data

set provides the average daily temperature from 1997 to 1999 Jan. like most data mining tasks, we have to analyze the data first before applying any techniques on them. Some properties observed are as follow:

1) Load periodicity: In our analysis, the load is changing with the season: high demand for electricity in winter while low demand in summer. Furthermore, in the weekend the load is usually lower. But the load on Saturday is a little higher than that on Sunday. The details can be showed in figure 1:

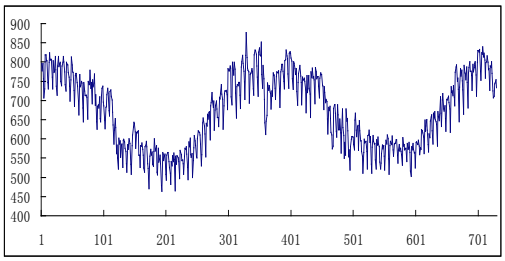

Figure 1. Max load from 1997 to 1998

2) Holiday Effect: Earlier works showed that holiday factor might influence the load. From the datasets we used, we found that the load usually lowers down on holidays. In addition, the load is also depends on what holiday it is. On some major holidays such as Christmas or New Year, the load may be more different compared with other days.

3) Weather Influence: As we explained above, the load data have some seasonal variation, this means the temperature have a great influence on the demand for electricity. We can easily to see that because of the heating device use. A winter day with higher temperature may causes lower demands. The correlation between the average load and the temperature in 1998 can be showed in figure 2:

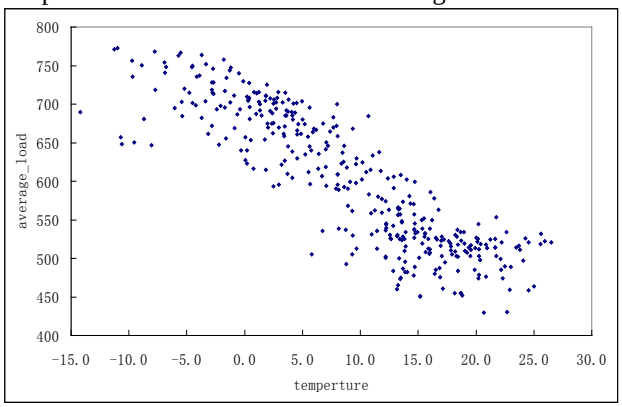

Figure 2. Correlation between the load and temperature

D. Data Sets Constructing

If only we considering the load information itself, the load data can be seen as a simple time series, we can predict future load by training past load data. In addition, we also know the average temperature, the calendar dates and all holidays. So it is necessary to encode the all the information above if possible.

For the jth time point of the ith day, the training data has an output yij as the target value and other attributes below:

- Seven inputs L1 (lj-7…lj-1) indicate seven loads before j;
- Seven inputs L2 (li-7, j…li-1, j) indicate seven loads in the same time point of the past seven days;
- Seven inputs Wi (w1…w7) indicate which day in a week;
- One input H_i indicate whether this is a holiday or not;
- One input T_{iavg} for the average temperature of the i^{th} day;

E. PSO-CVR Modeling

The detailed procedure of the PSO-CVR we proposed is below:

- Initializing PSO parameters. Denote the velocity and position by $V:(v_1,v_2,...,v_m)$ and $X:(x_1,x_2,...,x_m)\in R^n$, which x_i is a random particle in n-dimensional space R^n. Then initialize the weighting factors c1 and c2, the weighting function w, the max number of iteration k, and the initial swarm S with random V and X.
- Estimating swarm S. denotes MSE function by $F=\sum_i^m (y_{ij}-\hat{y}_{ij})^2$, $\hat{y}_{ij}:the\ actual\ load$ .the smaller of the MSE, the better of the position of S.
- Updating the position and velocity as per the following equations:
- $V_i^{k+1}=w*V_i^k+c1*r1*(pbset_i-X_i^k)+c2*r2*(gbest-X_i^k);$
- $X_i^{k+1}=X_i^k+V_i^{k+1}$;$r1,r2:random\ between[0,1]$
- Checking the termination condition. If current iteration number is k or the MSE is less than we expected, then terminate this procedure, else increase the iteration number and go to step 2.
- Transferring the optimized position vector (C,ε,δ) to CVR.
- Using CVR to train a model with the training data set.
- Using the model to predict the load with the predicting data set.

F. Load Forecasting Based on PSO-CVR

We want to predict for the loads of a certain period of time (half hour for one time point). The problem is this might degenerate due to the error propagation. But the experiments in section V show that our model can solve this problem and the results can be acceptable. For continuous forecasting loads for a period of time, the newly forecast load could be included as an attribute and used for the next prediction if insufficient information provided. For example, when we get the approximate load at 00:30 of January 1, 1999, it is used with loads at 21:30-00:00 in December 31, 1998, and loads at 01:00 of December 25-31, 1998 for forecasting the load at 01:00 in January 1. We continue this way until get all the approximate loads we want.

Another issue is how the size of the training data set influences forecasts. As we described in section IV.B, the training data set we used contains the load per half hour of each day from 1997 to 1998, while loads in Jan 1999 as the testing set. If we use all the data above, the size of the training data set is 365*2*48=35040, which can be seemed as large scale data set.

In step 1, in order to avoid some attributes too large or too small which might have a too big impact on the results, we need scale up all attributes in dataset to [0, 1]. In addition, the predicting data must be treated in the same way in step 2.

V. EXPERIMENTS AND RESULTS

In this section, for comparison, 15%, 30%, 60% and 100% of the training set is used for training while the data set in 1999 Jan is for testing. We also run the SVR implementations of LIBSVM. The root mean squared error (RMSE) and the mean relative error (MRE) which defined as

$$RMSE = \frac{1}{\max y_i}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|$$

are used as evaluation criteria. In addition, we use the Gaussian kernel

$$K(\vec{x},\vec{y}) = \exp(-\frac{\|\vec{x}-\vec{y}\|}{2\sigma^2})$$

to predict and all implementations are in C++. First we trained the full training data set in SVR and PSO-CVR; in addition, we also use PSO for optimizing SVR parameters. Then we use these trained model to continuous predict the loads for one week. The *Figure 3* is the comparison of forecasting results and it illustrate that CVR could do as good as or even better than SVR, CVR training only used about 1480 seconds but 8938 seconds used in SVR. Then we trained 15% (about 5k patterns), 30% (about 10k patterns), and 60% (about 20k patterns) of the training data set for comparing the difference feature in difference data set size. Figure 4a- Figure 4c show the details.

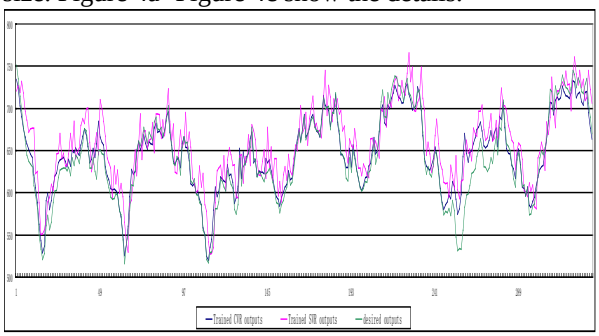

Figure 3. Continuous forecasting for one week

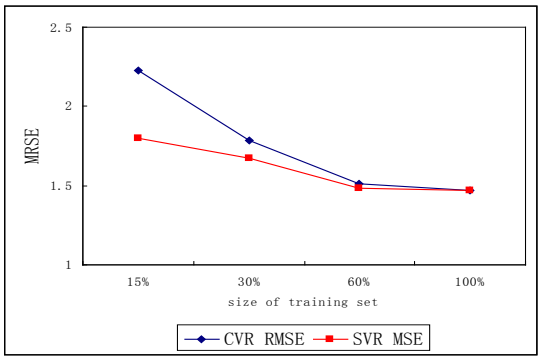

Figure 4. a. MRSE

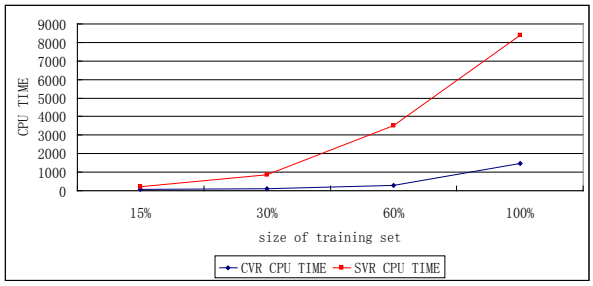

Figure 4. b. CPU TIME

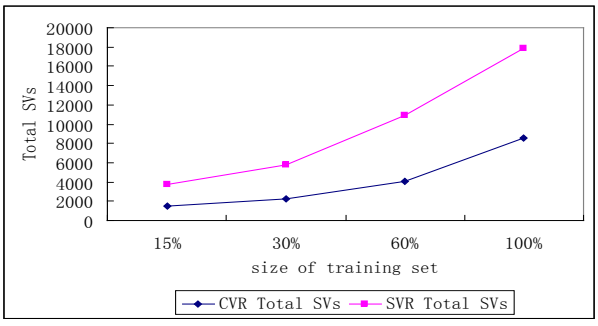

Figure 4. c. Total SVs

To sum up, it can be seen that:

- Training a large scale training set might has a better model to predict (Figure 4a).
- A smaller training set has a faster training speed and worse MRSE. Figure 4a illustrate that CVR is worse than SVR when training a small data set but the CPU time is almost same as SVR. This phenomenon is particularly prominent in 15% training set. So if training set has less than about 20k (60%) patterns, SVR is the better choice (Figure 4a, Figure 4b).
- When the account of training set patterns is larger than 20k (60%), the difference of MRSE between SVR and CVR is very close but the CPU time of CVR is much smaller(Figure 4a, Figure 4b). So CVR has a better performance in this case.

In case of PSO-CVR the best result was found to be, MRSE of 1.47037%, with C= 17520, μ = 8.76, ε =0.682e-5.

We also tried different options but they do not show significant improvements so it is not included. For example, while training the model we tried to reduce the weight for holiday attribute. Originally each attribute has values between [0, 1] after scaling but the holiday attribute can be further reduced to a smaller ranger like [0, 0.3], but the results doesn't change much better.

VI. CONCLUSIONS

Applications of Core Vector Regression (CVR) in electric load forecasting is a greater alternative to improve the load forecasting accuracy significantly, but the most important is the training time of CVR is much faster than SVR on large scale training set. Its advantage in power system is very obvious.

Furthermore, the proposed PSO-CVR appropriately applying PSO algorithm for optimizing CVR parameters so that we no longer need to manually select the CVR input parameters, this may greatly reduce the workload of the prediction process

REFERENCES

[1] K.J. Hwan, G.W. Kim, A short-term load forecasting expert system, in: Proceedings of the Fifth Russian-Korean International Symposium on Science and Technology, 1 (1), 2001, pp. 112–116.

[2] Hiroyuki Mori, Hidenori Kobayashi.Optimal Fuzzy Inference for Short-Term Load Forecasting.IEEE Trans on Power Systems, 1996,11(1):390~396.

[3] D.C.Park, M.A.El-Sharkawi, R.J.Marks II. Electric Load Forecasting Using an Artificial Neural Network, IEEE Trans on Power Systems.1991,6(2):442~449

[4] J. Yang and J. Stenzel, Short-term load forecasting with increment regression tree, Electric Power Systems Research **76** (2006), pp. 880–888.

[5] Wenjin Dai, Ping Wang. Application of Pattern Recognition and Artificial Neural Network to Load Forecasting in Electric Power System. Third International Conference on Natural Computation (ICNC 2007)

[6] Sanjib Mishra , Sarat Kumar Patra.Short Term Load Forecasting using Neural Network trained with Genetic Algorithm & Particle Swarm Optimization. First International Conference on Emerging Trends in Engineering and Technology 2008.94 606-611

[7] Vapnik V. N, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[8] Hu Shuju, Li Jianlin, Xu Honghua. A SVM Method based on Active Area Vector. DRPT2008 6-9 April 2008 Nanjing China.2564-2568

[9] Ivor W. Tsang, James T. Kwok, Pak-Ming Cheung, Core Vector Machines: Fast SVM Training on Very Large Data Sets, Journal of Machine Learning Research 6 (2005) 363–392

[10] Ivor Wai-Hung Tsang, James Tin-Yau Kwok, and Jacek M. Zurada, Fellow, IEEE. Generalized Core Vector Machines. IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 17, NO. 5, SEPTEMBER 2006

[11] Ivor W.Tsang James T.Kwok. Very Large SVM Training using Core Vector Machines. Department of Computer Science The Hong Kong University of Science and Technology.

[12] Ivor Wai-Hung Tsang, András Kocsor,and James Tin-Yau Kwok. Large-Scale Maximum Margin Discriminant Analysis Using Core Vector Machines. IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 19, NO. 4, APRIL 2008

[13] Ivor W. Tsang, James T. Kwok, Kimo T. Lai, Core Vector Regression for Very Large Regression Problems, Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 2005.