Fuzzy Velocity-Based Temporal Dependency for SVM-Driven Realistic Facial Animation

Pith Xie[1,2], Yiqiang Chen[1], Junfa Liu[1], and Dongrong Xiao[2]

[1] Institute of Computing Technology, Chinese Academy of Sciences,
Postfach 10 00 80,
Beijing, China
pithxie@yahoo.com, yqchen@ict.ac.cn, jfliu@shict.cn
[2] Department of Information Science and Communication, Nanjing University of Information Science and Technology,
Postfach 21 00 44,
Nanjing, China
xiaodongrong44@163.com

Abstract. Driving a realistic facial animation with Support Vector Machine(SVM) requires determining the shape-to-wrinkle correspondence, which includes not only spatial dependency, but also temporal dependency. A few available frameworks(e.g., *Recurrent Neural Network* and *Long Short-Term Memory*), represent temporal dependency as the dependency of output on position input series, which however may bring about spatial redundancy in some cases. We argue that temporal dependency should be represented as the dependency of output on velocity input series. Besides, due to the weak temporal dependency between shape change and wrinkle change, we put forward *Fuzzy Embedding* to convert velocity into fuzzy velocity. The shape-wrinkle synthesis demonstrates that, in determining the temporal dependency between wrinkle change and shape change, fuzzy velocity provides more valuable information than velocity and thus enhances the degree of the realism effectively.

1 Introduction

Driving a realistic facial animation requires uncovering the facial dynamics, which determines the shape-to-wrinkle correspondence. The ways to represent facial dynamics can be divided into three groups: image-based methods [1,2,3], physically-based methods [4,5] and machine learning method [6].

Avoiding tedious and manual process, machine learning method provide a straightforward way to uncover facial dynamics, which will subsequently be used to synthesize rich expressions. Zhu et al. [6], representing shape change as Facial Animation Parameters(FAP) and wrinkle change as *Eigen Expression Ratio Image*(EigenERI), train a SVM to learn the spatial dependency between FAP and EigenERI. Due to the fact that both shape and wrinkle change temporally, there exists temporal dependency between FAP and EigenERI in addition.

Many domains in machine learning involve discovering dependencies and structure over time. To capture the temporal dependency present in time series,

Q. Yang and G. Webb (Eds.): PRICAI 2006, LNAI 4099, pp. 563–572, 2006.

a variety of frameworks have been constructed by extending standard learning models,e.g., neural network. Jordan M.I. [7] proposed *Recurrent Network* to learn short-term temporal dependency. Hochreiter, S. and Schmidhuber, J. [8] proposed *Long Short-Term Memory*(LSTM) to learn long-term temporal dependency. Based on the fact that Hidden Markov Model(HMM) can encode long-term temporal dependency, Bengio, Y. and Frasconi, P. [9] proposed Input Output HMM (IOHMM) to learn long-term temporal dependency.

All those frameworks encode temporal dependency as various recurrent networks and succeed in solving the specific problem they aim at. As a whole, those frameworks are stimulated only by position input, that is to say, the position output depends on the position input series. However, position input series brings about spatial redundancy while changing slowly over time. In that case, those frameworks will no longer be sensitive to temporal dependency.

According to observation, both shape and wrinkle change slowly over time. Therefore, it is necessary to reduce spatial redundancy brought about by shape change. In this paper, temporal dependency is considered as the dependency of output on velocity input series while spatial dependency as the dependency of output on current position input. Velocity, representing temporal change hidden in the position input series, can effectively reduce the spatial redundancy. Furthermore, to capture the weak temporal dependency between wrinkle change and shape change, we propose *Fuzzy Embedding* to transform velocity into fuzzy velocity.

The paper is organized as follows. In section 2, we introduce temporal dependency and present our approach to capture weak temporal dependency. In section 3, we provide result of Mackey-Glass experiment to support our argument. In section 4, we provide demonstration of shape-wrinkle synthesis to support our argument. Section 5 gives the conclusion.

2 Temporal Dependency

To capture the correspondence between two time series, learning model should take into account not only spatial dependency, but also temporal dependency. Given an input time series $X(t)$ and an output time series $Y(t)$, for example, spatial dependency means that current position output $Y(t)$ depends on current position input $X(t)$ while temporal dependency means that current position output $y(t)$ depends on not only current position input $x(t)$, but also previous position input $x(t - \Delta t), x(t - 2\Delta t) \cdots$.

2.1 Spatial Redundancy

Position input describes a stationary spatial state. While varying rapidly over time, position input series represents temporal dependency effectively. While varying slowly over time, however, position input series involves much repeated spatial content and thus brings about spatial redundancy.

For example, given an 1-dimension position input series: [5.2], [5.6], [5.4], [5.2], the repeated spatial content can be approximately quantized into value 5 due to the big proportion it occupies in all position inputs. When position input series varies slowly over time, repeated spatial content inside it will result in spatial redundancy which finally acts as a noise. Though normalization can reduce global spatial redundancy, it still reserves local spatial redundancy.

2.2 Velocity

Reflecting the temporal change of position input, velocity reduces repeated spatial content, i.e. spatial redundancy. So we can represent temporal dependency as the dependency of output on velocity input series.

Given a time series with a fixed interval Δt, the velocity at time point t can be approximated by the following calculation:

$$R(t) = \frac{X(t) - X(t - \Delta t)}{\Delta t} \tag{1}$$

where $R(t)$ represents the velocity of the position input $X(t)$.

Based on Cantor Set, velocity exhibits as exact thresholds. In the case such as shape-to-wrinkle mapping, however, the temporal dependency of wrinkle change on shape change is weak. Therefore, wrinkle change depends on fuzzy velocity of shape change providing an approximate description more than velocity of shape change.

2.3 Fuzzy Velocity

Fuzzy Mathematics describes vague concept with membership function. Mapping infinite domain to the closed interval $[0, 1]$, membership function describes concept too roughly to lose much descriptive information, which however is valuable and necessary in machine learning.

Therefore, we propose *Fuzzy Embedding*, substituting for membership function, to transform velocity into fuzzy velocity so as to capture detailed fuzzy concept. Due to the operation of clustering, fuzzy velocity is vaguer than velocity. Ranging between $[0, +\infty)$, fuzzy velocity is more detailed than the grade of membership. The algorithm is given as follows:

1. Compute velocity of each position input in time series. Assumed these velocities amount to n;
2. Get all velocities together as a velocity set. The *sum-of-squared-error criterion*[11] is used here as a criterion function. Denote the number of clusters as k and the corresponding *sum-of-squared-error criterion* as $J(k)$. Then set $k = 1$ and cluster velocity set with K-Means;
3. Set $k = k + 1$ and cluster velocity set with K-Means.
   - If $k \geq \frac{n}{100}$ or $\frac{J(k)}{J(k-1)} > \epsilon$,then goto 4;
   - Else goto 3.
4. Denote the centers of clusters as $m_1, m_2, \cdots, m_k$, then map each velocity $R(t)$ to k-dimension fuzzy velocity $R'(t)$:

$$R'(t) = [\| R(t) - m_1 \|, \| R(t) - m_2 \|, \ldots, \| R(t) - m_k \|] \tag{2}$$

The condition $k \geq \frac{n}{100}$ ensure that the average volume of clusters be no less than 100 elements, while ϵ is set in advance to ensure that the result of clustering can reflect the distribution of elements well. $\parallel R(t) - m \parallel$ represents the Euclid distance between $R(t)$ and cluster center m. The fuzzy velocity $R'(t)$, resulting from *Fuzzy Embedding*, reflects the similarity between velocity and each cluster.

3 Mackey-Glass Experiment

The intensity of temporal dependency is the degree to which the position output depends on the position input series. Mackey-Glass equation is often used in some research[12, 13] to generate stationary time series, which can simulate the transfer of the intensity of temporal dependency with several parameters being adjusted.

Mackey-Glass equation is expressed as the following formula:

$$\frac{dx(t)}{dt} = \frac{a \bullet x(t - d \bullet \Delta t)}{1 + x(t - d \bullet \Delta t)^{10}} - b \bullet x(t) \tag{3}$$

a, b, d and Δt are parameters of this equation. Among them, we fix the following three parameters: $a = 2$, $b = 1$, $\Delta t = 1ms$. Parameter d represents the time-delay. The smaller d is, the greater role the previous elements play in determining current element, and the stronger the hidden temporal dependency is. On the contrary, the bigger d is, the less role the previous elements play in determining current element, and the weaker the hidden temporal dependency is.

Initially, $\forall t \leq 0, x(t) = 9$. Then the equation is iterated for 4005 times with an interval of Δt to generate a time series consists of 4005 elements. With time window size being fixed to be 5ms, we can acquire an example set consisting of 4000 temporally ordered input-output pairs, where the position input can be expressed as the following 5-dimension vector:

$$X_t = [x(t-5), x(t-4), \ldots, x(t-1)] \tag{4}$$

and the output can be expressed as the following 1-dimension vector:

$$Y_t = [x(t)] \tag{5}$$

In the experiment, we first train three SVMs: Standard SVM, Velocity SVM and Fuzzy Velocity SVM, then test their generalization performance.

Standard SVM takes into account only spatial dependency, which is stimulated by one and only position input. Velocity SVM takes into account both spatial dependency and temporal dependency, which is stimulated by position input as well as previous velocity input series. Fuzzy Velocity SVM also takes into account both spatial dependency and temporal dependency, which however is stimulated by position input as well as previous fuzzy velocity input series. In the following, we give detailed procedures to train and test these SVMs.

3.1 Standard SVM

First, normalize the example set to generate a new example set;

Second, a training set consisting of the former 2000 examples in the new example set is used to train a SVM to learn the mapping $F : X_t \longrightarrow Y_t$;

Third, a test set consisting of the latter 2000 examples in the new example set is used to test the generalization performance of above SVM.

3.2 Velocity SVM

First, incorporate three successive velocities V_{t-2}, V_{t-1}, V_t into the input of the example set, and generate new input X'_t as follows:

$$X'_t = [V_{t-2}, V_{t-1}, V_t, X_t] \tag{6}$$

Second, normalize the example set to generate a new example set;

Third, a training set consisting of the former 2000 examples in the new example set is used to train a SVM to learn the mapping $F : X'_t \longrightarrow Y'_t$;

Fourth, a test set consisting of the latter 2000 examples in the new example set is used to test the generalization performance of above SVM.

3.3 Fuzzy Velocity SVM

First, convert V_t to fuzzy velocity V'_t with *Fuzzy Embedding* under the constraint $\epsilon = 0.99$;

Second, incorporate three successive fuzzy velocities V'_{t-2}, V'_{t-1}, V'_t into the input of the training set and test set, and generate new input X''_t as follows:

$$X''_t = [V'_{t-2}, V'_{t-1}, V'_t, X_t] \tag{7}$$

Third, normalize the example set to generate a new example set;

Fourth, a training set consisting of the former 2000 examples in the new example set is used to train a SVM to learn the mapping $F : X''_t \longrightarrow Y''_t$.

Fifth, a test set consisting of the latter 2000 examples in the new example set is used to test the generalization performance of above SVM.

3.4 Experimental Results

The experiment is repeated on 16 example sets generated from Mackey-Glass equations with different d. Thus, we acquire 16 results as Table. 1 to compare the 3 SVMs. From left to right, column 1 stands for the value of d; column 2, 3 and 4 stand for the value of Mean Squared Error(MSE); column 5, 6 and 7 stand for the value of Squared Correlation Coefficient(SCC). MSE reflects the local fitting effect while SCC reflects the global fitting effect. The smaller MSE is, the better the local fitting effect is. The bigger SCC is, the better the global fitting effect is.

The intensity of temporal dependency decreases gradually while d increases step by step(see Table. 1). When d increases from 1 to 5, the temporal dependency keeps strong and Velocity SVM performs the best. When d increases from

Table 1. Experimental results from Mackey-Glass on Standard SVM(S-SVM),Velocity SVM(V-SVM) and Fuzzy Velocity SVM(FV-SVM)

parameter		**MSE**			*SCC*	
d	**S-SVM**	**V-SVM**	**FV-SVM**	*S-SVM*	*V-SVM*	*FV-SVM*
1	0.0786	0.0405	0.0596	0.755	0.865	0.78
2	0.101	0.0404	0.0783	0.756	0.873	0.714
3	0.104	0.0475	0.0851	0.696	0.839	0.691
4	0.229	0.168	0.203	0.0007	0.163	0.0557
5	0.293	0.258	0.257	0.0003	0.0115	0.0105
6	0.28	0.265	0.269	0.0136	0.018	0.0221
7	0.242	0.236	0.241	0.0695	0.072	0.0771
8	0.203	0.198	0.202	0.16	0.165	0.168
9	0.173	0.17	0.171	0.252	0.255	0.264
10	0.145	0.142	0.142	0.348	0.352	0.361
11	0.13	0.129	0.128	0.395	0.398	0.406
12	0.115	0.115	0.114	0.451	0.451	0.458
13	0.103	0.103	0.102	0.502	0.502	0.508
14	0.0906	0.0906	0.0895	0.559	0.559	0.565
15	0.0819	0.0816	0.0806	0.596	0.597	0.601
16	0.073	0.0729	0.0718	0.637	0.637	0.642

6 to 16, the temporal dependency becomes weak and Fuzzy Velocity SVM performs the best on the whole. This demonstrates that fuzzy velocity, as opposed to velocity, provides more valuable information in determining the weak temporal dependency.

4 Shape-Wrinkle Synthesis

In shape-wrinkle synthesis, we represent shape change as FAP and wrinkle change as EigenERI. According to physiology, there exists causality between them which can be considered as mapping. However, there exists more or less temporal dependency between EigenERI and FAP, which belongs to weak dependency.

We capture a 1840-frame video containing 16 expressions with the resolution of 130×200 pixels. In accordance with MPEG-4, we label 35 feature points and 3 rectificative points on human face. Firstly, we align all images according to rectificative points. Then we capture 43-dimension FAP by optical flow and compute ERI. High dimension make ERI difficult to be used in machine learning, therefore we extract 10-dimension EigenERI from ERI through Principal Component Analysis(PCA).

We derive a example set containing 1840 examples from 1840-frame video, where the input stands for FAP and the output stands for EigenERI. Similar to Mackey-Glass experiment, we compare three SVMs: Standard SVM, Velocity SVM and Fuzzy Velocity SVM. The experimental procedures differs from the Mackey-Glass experiment in two ways. First, ϵ in *Fuzzy Embedding* is set to be

Table 2. Experimental results from shape-wrinkle synthesis on Standard SVM(S-SVM),Velocity SVM(V-SVM) and Fuzzy Velocity SVM(FV-SVM)

EigenERI		**MSE**			*SCC*	
index	**S-SVM**	**V-SVM**	**FV-SVM**	*S-SVM*	*V-SVM*	*FV-SVM*
1	0.073	0.0694	0.0641	0.858	0.867	0.876
2	0.0676	0.0643	0.0593	0.858	0.8666	0.876
3	0.0643	0.0613	0.0565	0.856	0.864	0.875
4	0.0699	0.0666	0.0616	0.856	0.865	0.874
5	0.0775	0.0739	0.0682	0.856	0.864	0.874
6	0.0468	0.0443	0.0407	0.862	0.871	0.881
7	0.0657	0.0624	0.0578	0.862	0.87	0.879
8	0.0727	0.069	0.0641	0.862	0.871	0.879
9	0.065	0.0615	0.0569	0.861	0.87	0.879
10	0.0555	0.052	0.0486	0.867	0.877	0.884

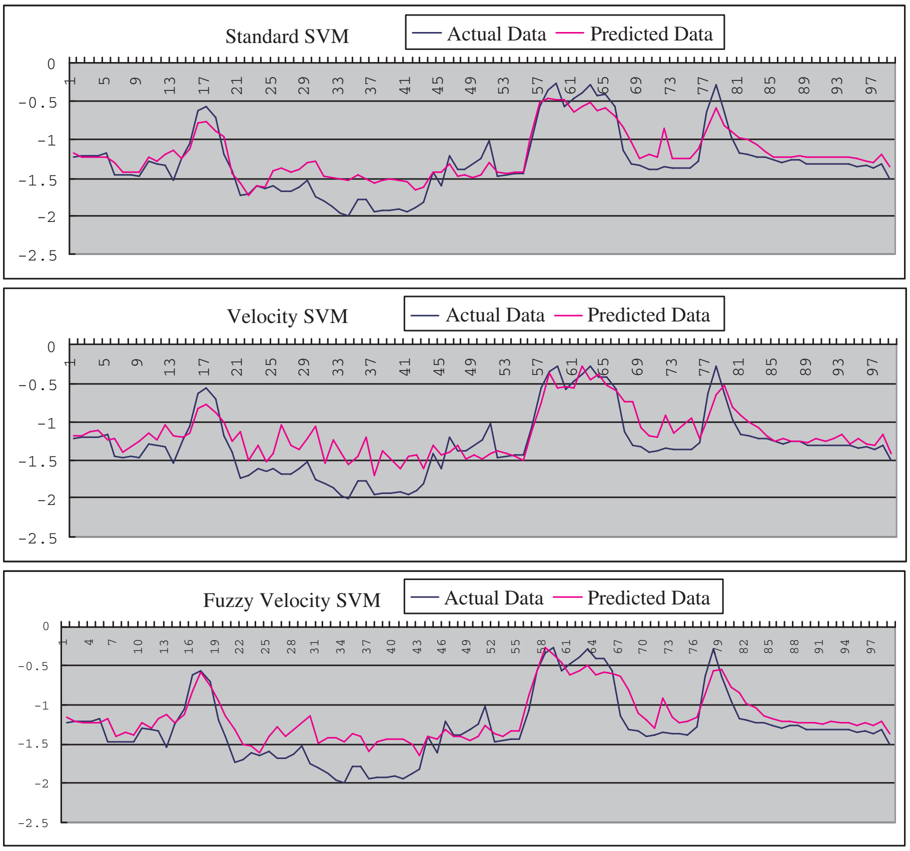

Fig. 1. Fitting curves generated from Standard SVM, Velocity SVM and Fuzzy Velocity SVM

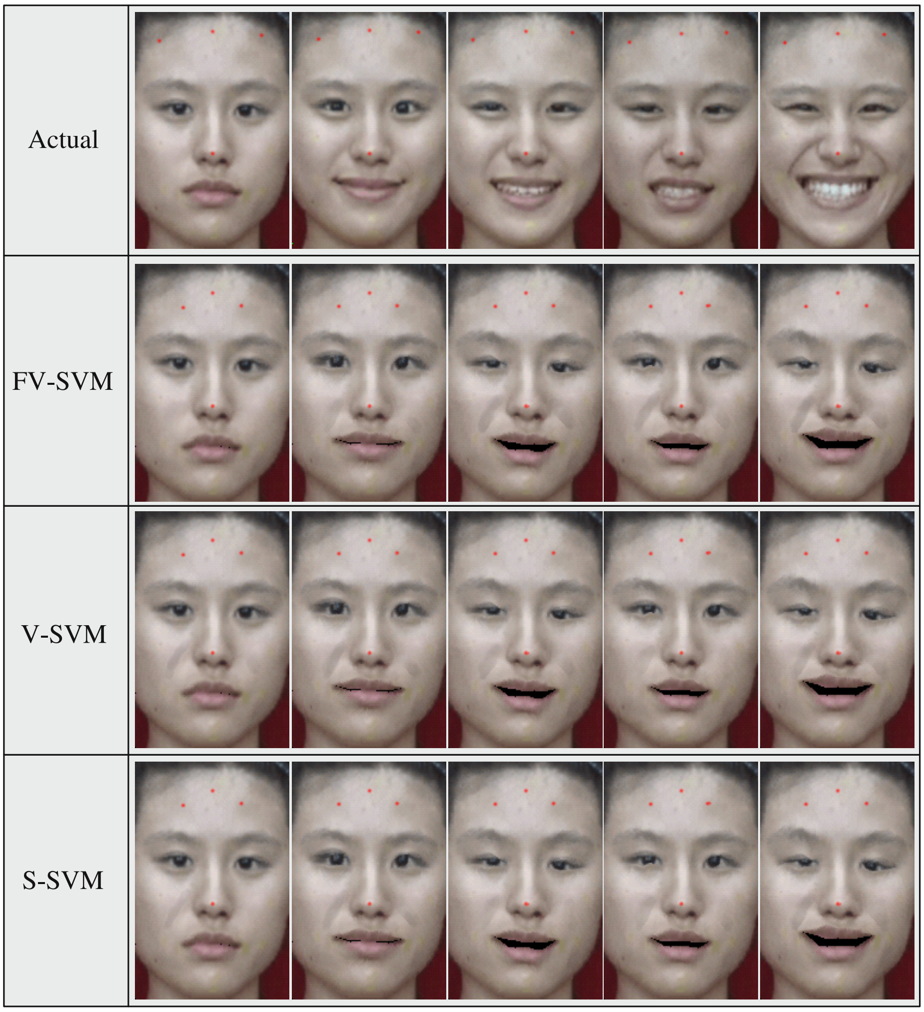

Fig. 2. From top to bottom: true facial animation, facial animation driven by Fuzzy Velocity SVM, facial animation driven by Velocity SVM, facial animation driven by Standard SVM

0.999. Second, the test set is just the whole new example set while the training set is derived from the new example set by sampling one every three examples.

Table. 2 compares the generalization performance of the 3 SVMs. From left to right, column 1 stands for the index of component in EigenERI; column 2, 3 and 4 stand for the value of Mean Squared Error(MSE); column 5, 6 and 7 stand for the value of Squared Correlation Coefficient(SCC). As shown in Table. 2, Fuzzy Velocity SVM performs the best while Standard SVM performs the worst.

Figure. 1 shows the fitting curves of the first component in EigenERI, which are generated from Standard SVM, Velocity SVM and Fuzzy Velocity SVM during the same interval. As shown in Figure. 1, Fuzzy Velocity SVM performs the best while Standard SVM performs the worst. Figure. 2 compares actual facial animation and the facial animations driven by Standard SVM, Velocity SVM and Fuzzy Velocity SVM. As domenstrated in Figure. 2, Fuzzy Velocity SVM performs the best while Standard SVM performs the worst.

5 Conclusion

To avoid spatial redundancy, we argue that output should depend on previous velocity input series and current position input, which individually provide temporal information and spatial information. When the temporal dependency is weak, however, output depends on fuzzy velocity input more than velocity input. In that case, we propose *Fuzzy Embedding* to convert velocity input into fuzzy velocity input.

As demonstrated in the shape-wrinkle synthesis, fuzzy velocity, providing more valuable information than fuzzy velocity in determining the temporal dependency between wrinkle change and shape change, can enhance the degree of the realism effectively. At this stage, we directly add previous velocity input series into current input to memorize short-term temporal dependency, which only takes into account the fixed-sized temporal dependency. In future, we will attempt to construct recurrent network driven by velocity or fuzzy velocity so as to learn unfixed-sized temporal dependency.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (grant 60303018, 60575032, 60473043),Shanghai Qingmingxing Project (05QMX 1447) and Beijing Multimedia and Intelligent Software Technology Laboratory. Thanks to the reviewer's comments.

References

1. Cyriaque Kouadio, Pierre Poulin, Pierre Lachapelle: Real-time facial animation based upon a bank of 3D facial expression. Proc. Computer Animation'98, pp:128-136, 1998
2. Moiza. G, Tal. A, Shimshoni. I, Barnett. D, Moses. Y: Reconstruction of movies of facial expressions. IEEE ICCV 2001, Page(s):8 - 15, 2001
3. F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. H. Salesin: Synthesizing Realistic Facial Expressions from Photographs. SIGGRAPH 1998, pp. 75-84, 1998
4. Terzopoulos D, Waters K: Physically-Based Facial Modeling and Animation. Journal of Visualization and Computer Animation, Vol. 1, pp. 73-80, 1990
5. Goudeaux, K., Tsuhan Chen, Shyue-Wu Wang, Jen-Duo Liu: Principal component analysis for facial animation. Proc. ICASSP'01. Page(s):1501-1504 vol.3.2001
6. Wenhui Zhu, Yiqiang Chen, Yanfeng Sun, Baocai Yin and Dalong Jiang: SVM-Based Facial Texture Driving for Realistic Expression Synthesis. ICIG 2004. Dec.18-20,2004

7. Jordan, M.I.(1986).Serial order: A parallel distributed processing approach. Institute for Cognitive Science Report 8604.University of California,San Diego
8. Hochreiter, S., Schmidhuber, J. (1997): Long short-term memory. Neural Computation, 9(8), 1735–1780
9. Bengio, Y., Fransconi, P. (1995): An input-output HMM architecture. In J.D. Cowan, G. Tesauro and J. Alspector (Eds.), Advances in Neural Information Processing Systems 7 (pp.427- 434), Cambridge, MA: MIT Press
10. J.L. Elman: Finding structure in time. Cognitive Science, 14:179–211, 1990
11. Richard O. Duda, Peter E. Hart, David G. Stork: Pattern Classification (2nd Edition), John Wiley and Sons, 2001
12. Ming-Wei Chang, Chih-Jen Lin, Weng,R.C.-H: Analysis of switching dynamics with competing support vector machines. IEEE Transactions on Neural Networks, 15(3),720 - 727
13. J. Kohlmorgen, S. Lemm, G. Rätsch, and K.-R. Müller: Analysis of nonstationary time series by mixtures of self-organizing predictors. In Proceedings of IEEE Neural Networks for Signal Processing Workshop, pages 85-94, 2000
14. V. Vapnik: Statistical Learning Theory. John Wiley, 1998
15. E. Hüllermeier: Fuzzy methods in machine learning. and data mining: Status and prospects. Fuzzy Sets and Systems, vol. 156, pp.387-406, 2005
16. A. Joshi et.al: On Neurobiological, Neurofuzzy, Machine Learning and Pattern Recognition Techniques. IEEE Transaction on Neural Networks, vol. 8, no.1, pp. 18-31, Jan.1997
17. Ian Cloete, Jacobus van Zyl: A machine learning framework for fuzzy set covering algorithms. SMC(4):3199-3203,2004