DOI 10.1007/s11207-009-9318-9

Short-Term Solar Flare Prediction Using a Sequential Supervised Learning Method

Daren Yu · Xin Huang · Huaning Wang · Yanmei Cui

Received: 15 June 2008 / Accepted: 8 January 2009 / Published online: 6 February 2009

Abstract Solar flares are powered by the energy stored in magnetic fields, so evolutionary information of the magnetic field is important for short-term prediction of solar flares. However, the existing solar flare prediction models only use the current information of the active region. A sequential supervised learning method is introduced to add the evolutionary information of the active region into a prediction model. The maximum horizontal gradient, the length of the neutral line, and the number of singular points extracted from SOHO/MDI longitudinal magnetograms are used in the model to describe the nonpotentiality and complexity of the photospheric magnetic field. The evolutionary characteristics of the predictors are analyzed by using autocorrelation functions and mutual information functions. The analysis results indicate that a flare is influenced by the 3-day photospheric magnetic field information before flare eruption. A sliding-window method is used to add evolutionary information of the predictors into machine learning algorithms, then C4.5 decision tree and learning vector quantization are employed to predict the flare level within 48 hours. Experimental results indicate that the performance of the short-term solar flare prediction model within the sequential supervised learning framework is significantly improved.

Keywords Flare prediction · Photospheric magnetic field · Sequential supervised learning method

D. Yu · X. Huang (✉)
Harbin Institute of Technology, Power Engineering, Harbin, Heilongjiang Province, China
e-mail: huangxinhit@yahoo.com.cn

H. Wang
National Astronomical Observatories, Chinese Academy of Sciences, Beijing, China
e-mail: hnwang@bao.ac.cn

Y. Cui
Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing, China

1. Introduction

High-energy particles and radiation originating from giant solar flares are hazardous to both spacecraft and astronauts (Koskinen *et al.*, 1999), so precise prediction of solar flare helps in the protection of spacecraft and astronauts. However, the initiation mechanism of solar flares has not been well understood (Kusano *et al.*, 2004), and therefore many statistical and artificial intelligent approaches are used to establish a solar flare prediction model based on observational data.

Much effort has been devoted to the establishment of short-term solar flare prediction models based on morphological characteristics of active regions. McIntosh (1990) presented definitions of the McIntosh classification system and their rationale. Based on the McIntosh classification, an expert system (Theo) involving more than 500 decision rules was used for predicting X-ray solar flares. It is not necessary to use all the McIntosh classification parameters to predict solar flares. Multiple liner regression analysis was used to derive the effective solar flare contributions of the McIntosh classification parameters in Bornmann and Shaw (1994). They concluded that the 17 original McIntosh classification parameters could be reduced to 10 and still adequately reproduce the observed flare rates. Historical flare rates from the McIntosh classification could provide initial flaring probability of an active region. The probability of new forecasts was modified using Poisson statistics in Gallagher, Moon, and Wang (2002). Some machine-learning-based systems were employed to provide automated short-term solar flare prediction using the McIntosh classification. Fozzard, Bradshaw, and Ceci (1989) built a three-layer back-propagation neural network (named the TheoNet) to forecast flares. Qahwaji and Colak (2007) optimized cascade-correlation neural networks, support vector machines, and radial basis function networks for short-term solar flare prediction. Li *et al.* (2007) proposed a method combining support vector machine with *k* nearest neighbors to construct a solar flare forecasting model. Wheatland (2005) proposed an objective solar flare prediction method using the observed historical flare data. This method did not rely on the McIntosh classification.

To directly reflect the relationship between the changes in photospheric magnetic field and the flare level, solar flare prediction models based on the characteristics of the photospheric magnetic field were built. Parameters derived from photospheric vector magnetic field were used to identify a preflare signatures. Leka and Barnes (2003) used discriminant analysis to identify those properties that are important for the production of solar flares. Twenty-four flaring and flare-quiet epochs taken from seven active regions were analyzed. Furthermore, Leka and Barnes (2007) analyzed a much larger sample of photospheric vector magnetic field data from 2001 to 2004. The discriminant analysis method produced a “flaring” or “flare-quiet” prediction. Barnes *et al.* (2007) extended this method to a probability method. Cui *et al.* (2006) extracted three physical measures – the maximum horizontal gradient, the length of the neutral line, and the number of singular points – from SOHO/MDI longitudinal magnetograms. Based on these measures, Wang *et al.* (2007) set up a flare forecast model supported with artificial neural network techniques. This model is equivalent to a person with a long period of solar flare forecasting experience.

In all these methods, the prediction of a solar flare is based on the current information of morphological or magnetic properties of an active region. The influence of previous predictors on flare level is not considered. However, the previous predictors are also very important for short-term solar flare prediction. In machine learning methods, sequential supervised learning (Dietterich, 2002) is introduced to add the evolutionary information of predictors into the machine learning algorithms. Short-term solar flare prediction is treated within the sequential supervised learning framework.

The rest of the paper is organized as follows: The data and the influence of evolutionary predictors on flare are introduced in Section 2. The sliding-window technique in flare prediction is discussed in Section 3. Two machine learning methods – the C4.5 decision tree classifier and learning vector quantization – are explained in Section 4. The experimental results and their analyses are reported in Section 5. Finally, the conclusions are given in Section 6.

2. Data and the Influence of Evolutionary Predictors

2.1. Data

Three predictors – the maximum horizontal gradient ($|\nabla_{\mathrm{h}} B_z|_{\mathrm{m}}$), the length of the neutral line (L), and the number of singular points (η) – are extracted from SOHO/MDI full-disk longitudinal magnetograms with a pixel size of $2''$ and a noise level of 20 G from 15 April 1996 to 10 January 2004 (Cui *et al.*, 2006). The time interval between the successive magnetograms is 96 minutes.

Active region location data associated with the solar flare events are obtained from SGD solar event reports (http://www.solarmonitor.org/index.php).

Active regions are selected by using the following two criteria:

1. At least one X-ray flare with magnitude $\geq$ C1.0 is produced.
2. The location of active regions is within 30° of the solar disk center.

Flare data are downloaded from http://www.ngdc.noaa.gov/stp/SOLAR/ftpsolarflares.html#xray.

The importance of a solar flare is conventionally described by its GOES class, for example, C, M, or X. Within a certain time interval, more than one flare may happen. The importance of these flares is summed up with weights. The total importance of flares is defined as follows:

$$I_{\mathrm{tot}} = \sum \mathrm{C} + 10 \times \sum \mathrm{M} + 100 \times \sum \mathrm{X}. \tag{1}$$

Equation (1) considers influence of all the flares within the forward-looking period. For example, for an active region producing C1.2, C2.3, M4.1, and X1.2 flares within 48 hours, we have $I_{\mathrm{tot}} = (1.2 + 2.3) + 10 \times 4.1 + 100 \times 1.2 = 164.5$ (Wang *et al.*, 2007).

A forecasting model usually pays attention to the production of flares with significance above a threshold. Here, the threshold of I_{tot} is set to be 10. This means that the definition of "flaring" versus "nonflaring" is a total importance above 10. The instances with total importance larger than 10 ($I_{\mathrm{tot}} > 10$) are marked as "Class 1"; otherwise they are marked as "Class 0."

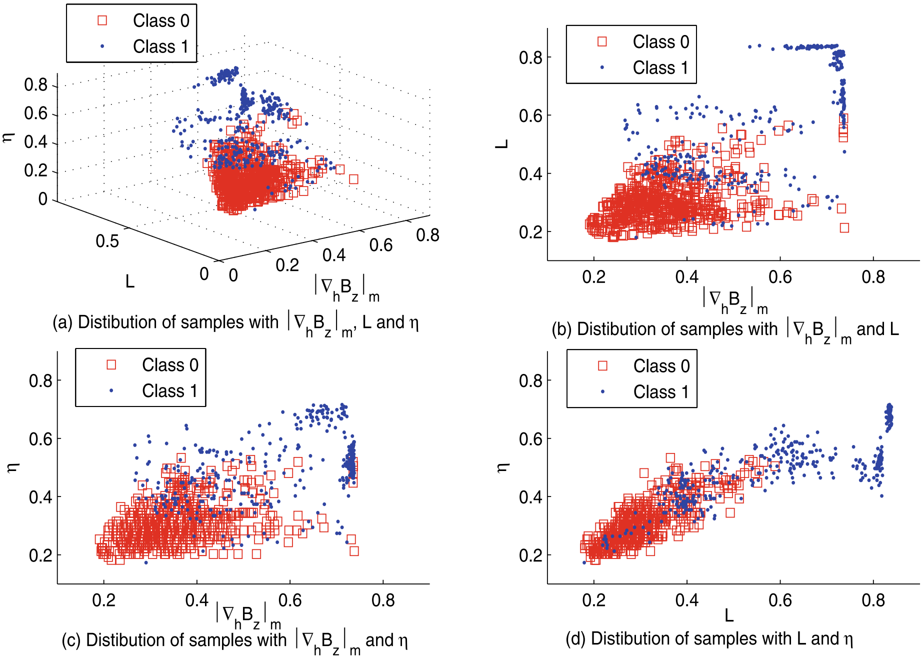
The predictors are preprocessed to incorporate prior information in machine learning algorithms. The relationship between the value of predictors and flare productivity seems to be sigmoidal; therefore the data can be fitted by using the sigmoid function in Boltzmann style (Cui *et al.*, 2006):

$$Y = A_2 + \frac{A_1 - A_2}{1 + \exp[(X - X_0)/W]}, \tag{2}$$

where Y is the flare productivity defined by the ratio of the number of flare-productive samples to the number of total samples, X is the value of the predictor, and A_1, A_2, X_0,

Table 1 Values of parameters in Boltzmann functions.

Threshold	Forward-looking period	Predictor	A_1	A_2	X_0	W
$I_{\mathrm{tot}} = 10$	48 hours	$\lvert\nabla_{\mathrm{h}} B_z\rvert_{\mathrm{m}}$	0.164	0.738	0.360	0.066
		L	0.062	0.848	763.08	382.97
		η	−0.196	0.730	9.343	22.663



(a) Distibution of samples with $|\nabla_h B_z|_m$, L and η

(b) Distibution of samples with $|\nabla_h B_z|_m$ and L

(c) Distibution of samples with $|\nabla_h B_z|_m$ and η

(d) Distibution of samples with L and η

Figure 1 Distribution of samples in space of predictors.

and W are estimated from the curve-fitting process. In this process, parameters A_1, A_2, X_0, and W are optimized to minimize the sum of the squares of the deviations between the observed data and the expected data (Marko, 2003). The values of these parameters are given in Table 1. Then the predictors are preprocessed with the Boltzmann functions.

Figure 1 shows the distribution of samples in the space of the predictors from March 2001 to May 2001. It is easy to see that there are regions where Class 0 and Class 1 overlap. The samples in these regions are difficult to forecast.

2.2. Influence of Evolutionary Predictors

The evolutionary process of an active region has a great influence on flare activity (Rust *et al.*, 1994). To explain the evolutionary characteristics of predictors and describe the influence of the evolutionary process on the flare level, two time series analysis tools – the autocorrelation function and the mutual information function (Bernhard and Darbellay, 1999) – are used.

The autocorrelation function of sequence $\mathbf{x}$ is

$$r_{\mathbf{x}}(k) = \left(\frac{1}{N}\right)\sum_{n=0}^{N} x(t)x\big[t-(k+n)\Delta t\big], \tag{3}$$

where $\mathbf{x} = \{x(t), x(t-\Delta t), \ldots, x(t-N\Delta t)\}$, Δt is the time interval between two observations, and the average value of $\mathbf{x}$ has been subtracted from the data. The autocorrelation function is used to measure the correlation of a time series with its own past values.

The mutual information function between sequence $\mathbf{y}$ and $\mathbf{z}$ is

$$I_{(\mathbf{y},\mathbf{z})}(k) = I\big[\mathbf{y}(t-k\Delta t), \mathbf{z}(t)\big], \tag{4}$$

where $\mathbf{y}(t-k\Delta t) = \{y(t-k\Delta t), y(t-(k+1)\Delta t), \ldots, y(t-(k+N)\Delta t)\}$, k is the lag order, and $\mathbf{z}(t) = \{z(t), z(t-\Delta t), \ldots, z(t-N\Delta t)\}$.

The right-hand side of Equation (4) is the mutual information between $\mathbf{y}$ and $\mathbf{z}$:

$$I(\mathbf{y},\mathbf{z}) = H(\mathbf{y}) + H(\mathbf{z}) - H(\mathbf{y},\mathbf{z}), \tag{5}$$

where $H(\mathbf{y})$ and $H(\mathbf{z})$ are the entropy of $\mathbf{y}$ and $\mathbf{z}$, respectively. $H(\mathbf{y},\mathbf{z})$ is the joint entropy of $\mathbf{y}$ and $\mathbf{z}$.

The entropy of sequence $\mathbf{y}$ is defined as

$$H(\mathbf{y}) = -\sum_{i=1}^{n} p(y_i)\ln p(y_i), \tag{6}$$

where $p(y_i)$ is probability of y_i, $i = 1, \ldots, n$.

In practical applications, the probability is estimated with samples. For the discrete variables, a histogram is introduced to estimate the probability distribution and entropy. For continuous variables, discretization or fuzzification is used to treat this problem. However, some information is lost because of the hard boundaries of discretization. Compared with the hard boundaries, fuzzy information entropy (Hu *et al.*, 2006) generates soft boundaries to obtain more information. So it is used to estimate entropy.

The mutual information function between set $\widehat{\mathbf{y}}(t-k\Delta t)$ and sequence $\mathbf{z}(t)$ is

$$I_{(\widehat{\mathbf{y}},\mathbf{z})}(k) = I\big[\widehat{\mathbf{y}}(t-k\Delta t), \mathbf{z}(t)\big], \tag{7}$$

where set $\widehat{\mathbf{y}}(t-k\Delta t) = \{\mathbf{y}(t), \mathbf{y}(t-\Delta t), \ldots, \mathbf{y}(t-k\Delta t)\}$ consists of all the sequences from lag 0 to lag k and $\mathbf{z}(t) = \{z(t), z(t-\Delta t), \ldots, z(t-N\Delta t)\}$.

For short-term solar flare prediction, the predictor sequences consisting of the current predictor and its lagged predictors are defined by

$$|\nabla_{\mathrm{h}}\mathbf{B_Z}|_{\mathrm{m}}(t) = \big\{|\nabla_{\mathrm{h}}B_Z|_{\mathrm{m}}(t), |\nabla_{\mathrm{h}}B_Z|_{\mathrm{m}}(t-\Delta t), \ldots, |\nabla_{\mathrm{h}}B_Z|_{\mathrm{m}}(t-N\Delta t)\big\}, \tag{8}$$

$$\mathbf{L}(t) = \big\{L(t), L(t-\Delta t), \ldots, L(t-N\Delta t)\big\}, \tag{9}$$

$$\boldsymbol{\eta}(t) = \big\{\eta(t), \eta(t-\Delta t), \ldots, \eta(t-N\Delta t)\big\}, \tag{10}$$

where Δt is 96 minutes, which is the sampling interval of MDI magnetograms.

As shown in Figure 2, the autocorrelation functions of the predictors, $r_{|\nabla_{\mathrm{h}}\mathbf{B_Z}|_{\mathrm{m}}}(k)$, $r_{\mathbf{L}}(k)$, and $r_{\boldsymbol{\eta}}(k)$, show smooth decrease. So some non-random patterns reflecting the evolutionary characteristics of the active region are present.

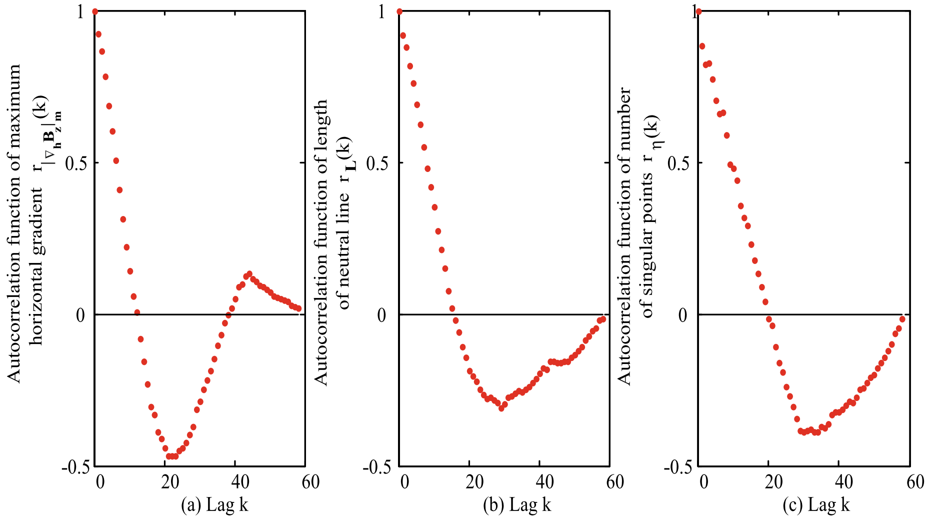


Figure 2 Autocorrelation functions of three predictors in active region 10158.

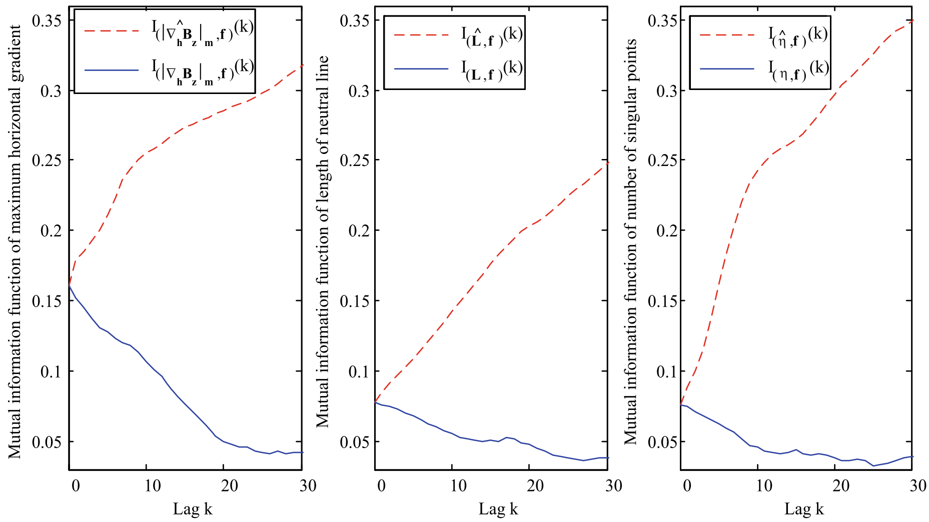


Figure 3 Mutual information functions between each lagged predictor and flare level (solid line) and mutual information functions between the set of all lagged predictors and flare level (dashed line).

The mutual information functions between each lagged predictor and the flare level, $I_{(|\nabla_{\mathbf{h}}\mathbf{B_Z}|_{\mathrm{m}},\mathrm{f})}(k)$, $I_{(\mathbf{L},\mathrm{f})}(k)$, and $I_{(\eta,\mathrm{f})}(k)$, measure the information amount provided by the lagged predictors, and the mutual information functions between the set and the flare level, $I_{(|\widehat{\nabla_{\mathbf{h}}\mathbf{B_Z t}}|_{\mathrm{m}},\mathrm{f})}(k)$, $I_{(\widehat{\mathbf{L}},\mathrm{f})}(k)$, and $I_{(\widehat{\eta},\mathrm{f})}(k)$, measure the information amount provided by the set of k lagged predictors.

As shown in Figure 3, the set of lagged predictors can provide more information if the number of lagged predictors increases. However, a single lagged predictor provides less information when k increases. The results show that the evolutionary information of the predictors is important for short-term solar flare prediction.

3. A Sliding-Window Technique for Flare Prediction

Short-term solar flare prediction is treated as a sequential supervised learning problem. Many methods have been used to solve this problem (Cohen and Carvalho, 2005; Klema *et al.*, 2008). The sliding-window method is used to convert the problem to a standard supervised learning problem (Kontaki, Papadopoulos, and Manolopoulos, 2007).

The principle of the sliding-window method is shown in Figure 4. The properties of the photospheric magnetic field are observed at t. They are used to forecast whether or not flares will happen within $t + F$. The span between $t - W\Delta t$ and t is called the sliding window, where W is the window size. A predictor is extended to $W + 1$ dimensions by using the sliding window. Data are regularly sampled in each active region, but the durations of active regions are not equal, so the sliding-window method is used in each active region. There are not enough data for the sliding window at the beginning of the evolution of each active region. By considering the operability of real-time flare prediction and the characteristics of observed data, the first value of samples is repeated W times.

The key problem with the sidling-window method is how to specify a proper window size. If the window size is too small, the evolutionary characteristics cannot be introduced adequately. Otherwise, too much irrelevant information will be introduced. According to the data characteristics, the following two conditions for determination of window size should be satisfied:

1. The window size should be smaller than the number of observed data points in the majority of active regions with flares.
2. The mutual information functions between the set of lagged predictors and the flare level should no longer increase rapidly.

For the first condition, we compute the cumulative number of active regions with flares versus the number of observed data points in the active regions. As shown in Figure 5, there are more than 45 observed data points in the majority of active regions, which suggests a window size not larger than 45.

For the second condition, the mutual information functions between the set of lagged predictors and the flare level is shown in Figure 6. We see that if the lags of predictors are greater than 45, the mutual information functions no longer increase rapidly. Based on this analysis, the window size is set to 45.

It is physically reasonable for the window size to be 45. Much effort to explain the influence of the evolution of the active region on flare occurrence has been made (Rust *et al.*, 1994). It is widely accepted that emerging flux regions play an important role in flaring (Wang and Shi, 1993). The magnetic flux emergence is strongly related to the predictors

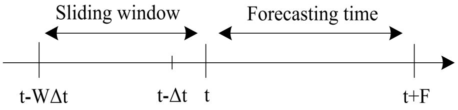


Figure 4 Principle of the sliding-window method. Here Δt is the time interval between two observations, W is the window size, and F is the forecasting time.

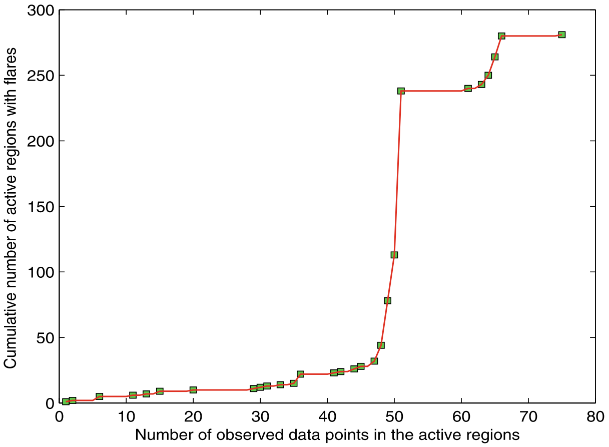


Figure 5 Cumulative distribution for number of active regions in which at least one flare happened *versus* number of observed data points in the the active regions.

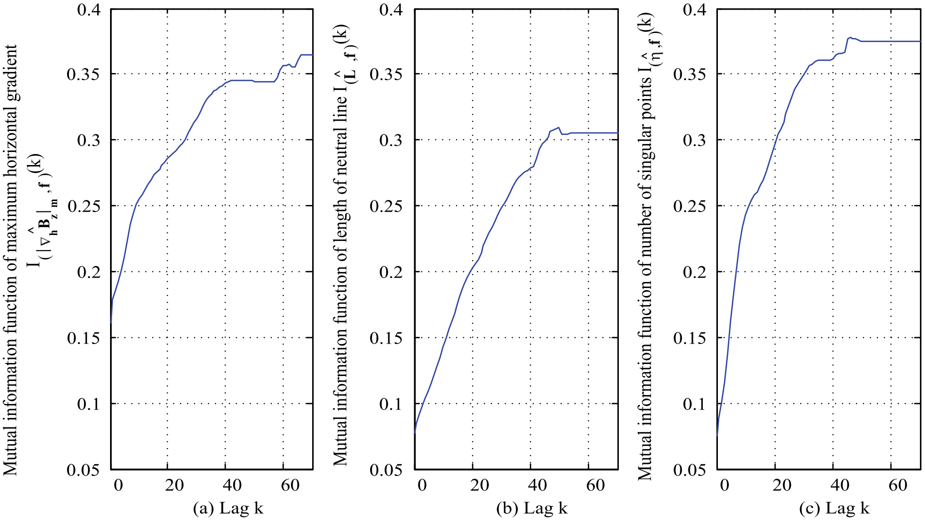


Figure 6 Mutual information functions between the set of lagged predictors and flare level.

used in this prediction model. Newly emerging fluxes colliding with preexisting fluxes produce a strong magnetic gradient, making the magnetic topological structure more complex so that the number of singular points increases. Meanwhile new flux regions often appear in unipolar areas, so the length of the neutral line would grow (Cui *et al.*, 2006). The whole process of flux emergence should be considered. Generally speaking, flux emergence lasts for about 3 – 5 days, so it is reasonable to use the 3-day observed data ($\frac{3\times 24\times 60}{96} = 45$ window size) of predictors to forecast the flare level.

A general schematic view of the flare prediction method with the sliding-window technique is shown in Figure 7. The predictors are extended by using the sliding window. Then the standard machine learning method based on the extended predictors can be used to generate a prediction model reflecting the evolutionary information of the active region.

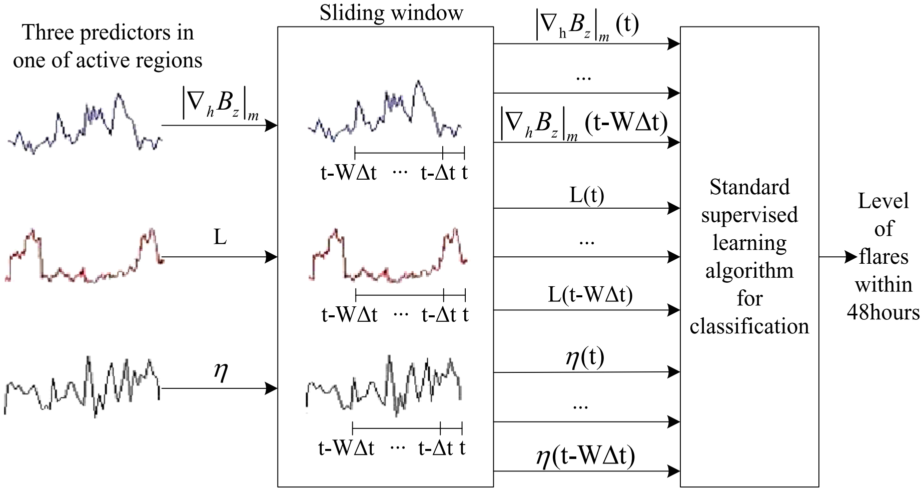

Three predictors in one of active regions
Sliding window
$|\nabla_h B_z|_m$
L
η
t-WΔt ··· t-Δt t
t-WΔt ··· t-Δt t
t-WΔt ··· t-Δt t
$|\nabla_h B_z|_m$ (t)
...
$|\nabla_h B_z|_m$ (t-WΔt)
L(t)
...
L(t-WΔt)
η(t)
...
η(t-WΔt)
Standard supervised learning algorithm for classification
Level of flares within 48hours

Figure 7 General schematic view of short-term solar flare prediction with the sliding-window method.

4. Machine Learning Methods

After the flare prediction problem is converted to a standard classification problem, two machine learning methods – the C4.5 decision tree classifier (C4.5) and leaning vector quantization (LVQ) – are used to learn a prediction model from the data. C4.5 is one of the most popular decision tree algorithms. Its comprehensibility is a useful characteristic (Zheng, 1998). Based on competitive learning criterion, LVQ can become an effective method to forecast the level of flare activity.

4.1. C4.5 Decision Tree

The C4.5 decision tree classifier (Quinlan, 1993) is a type of induction algorithm. An example of a decision tree is shown in Figure 8. A decision tree consists of test nodes, branches, and leaf nodes. A test node represents the selected predictor, which is used to divide the samples into subsets. Each branch descending from that node corresponds to one of the possible values for this predictor. Finally, the leaf node provides the classification of samples in the subsets.

Generally speaking, a decision tree is constructed from a set of samples by using the divide-and-conquer strategy; that is, a best predictor is selected in each test node to split samples into the smaller subsets. The quality of predictors is evaluated by using the information gain ratio (Mitchell, 1997). When the dataset is divided into several subsets by the predictor, the information gain ratio is used to measure the reduction of the uncertainty of samples associated with this process. This means that the samples with the same classification are sorted into the same subset as much as possible. For a continuous predictor P, the threshold t in $P \le t$ should be found to maximize the information gain ratio (Quinlan, 1996). The samples are sorted on their values of predictor P to give ordered distinct values $v_1, v_2, \dots, v_N$. Every pair of adjacent values suggests a potential threshold $t = \frac{(v_i + v_{i+1})}{2}$. When the best predictor and its threshold are determined, the predictor is used to test these samples at the root node of the tree. A descendant of the root node is then created according to the threshold of this predictor, and the training samples are sorted into an appropriate descendant node. C4.5 employs the top-down and recursive splitting technique to produce the

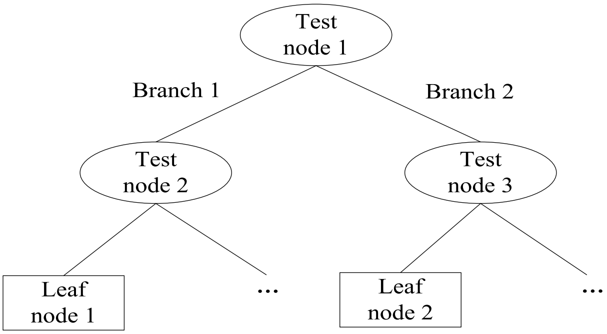


Figure 8 Example of a decision tree. Each test node is used to split samples into subsets by the selected predictor. Each branch corresponds to the value of the predictor used in its test node. Each leaf node assigns the samples in the subset into a certain classification. The three dots (...) represent parts of the decision tree that are omitted for clarity.

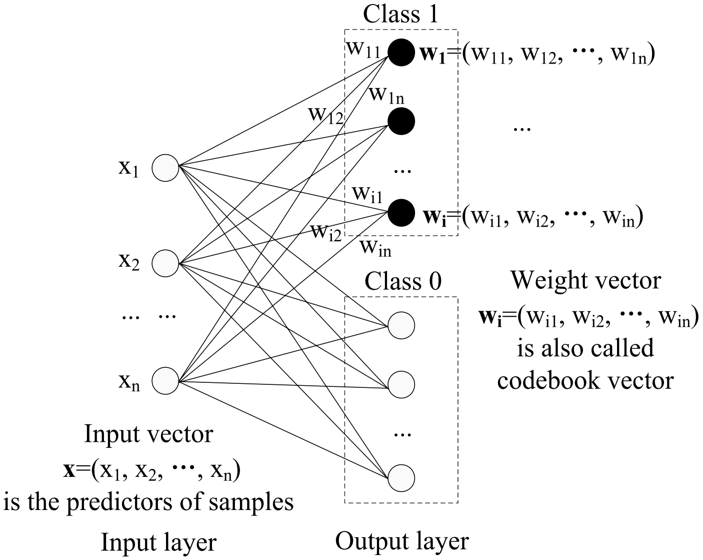

Class 1
w_{11}
$\mathbf{w_1}=(w_{11}, w_{12}, \cdots, w_{1n})$
w_{1n}
w_{12}
...
...
w_{i1}
$\mathbf{w_i}=(w_{i1}, w_{i2}, \cdots, w_{in})$
w_{i2}
w_{in}
x_1
x_2
... ...
x_n
Class 0
Weight vector
$\mathbf{w_i}=(w_{i1}, w_{i2}, \cdots, w_{in})$
is also called
codebook vector
...
Input vector
$\mathbf{x}=(x_1, x_2, \cdots, x_n)$
is the predictors of samples
Input layer
Output layer

Figure 9 Topology of an LVQ network.

subtree. When samples in the subset have the same classification or all possible tests have the same class distribution, the leaf node is generated. The classification of the samples in the leaf node is the same as the most frequent classification in this leaf node.

4.2. Learning Vector Quantization

Learning vector quantization (Kohonen, 2001) based on the competitive learning criterion is one type of neural network. The topology of an LVQ network is shown in Figure 9. This network consists of an input layer and an output layer. The input vector of the network is equal to the values of predictors and the output of the network is the possible class label of samples. Input units fully connect to output units using weight values. The weight vector associated with each output unit is also called a codebook vector in an LVQ algorithm. As in other neural network methods, the most important work of LVQ is to calculate these weight values of the network. The LVQ is a competitive network, so only the weight vector connected with the winner unit is modified. The winner unit is defined as the closest output unit to the input vector.

The basic algorithm for modifying weight vectors is described as follows: The input vector is

$$\mathbf{x} = (x_1, x_2, \dots, x_i, \dots, x_n), \tag{11}$$

where x_i is the ith predictor of samples. The ith weight vector $\mathbf{w}_i$ is

$$\mathbf{w}_i = (w_{i1}, w_{i2}, \dots, w_{ii}, \dots, w_{in}), \tag{12}$$

where $i = 1, 2, \dots, k$ with k the number of weight vectors (codebook vectors). The closest weight vector to $\mathbf{x}$ is defined as $\mathbf{w}_\mathrm{c}$:

$$\|\mathbf{x} - \mathbf{w}_\mathrm{c}\| = \min_i \bigl\{\|\mathbf{x} - \mathbf{w}_i\|\bigr\}. \tag{13}$$

The weight vector is updated by using the following rules:

1. If $\mathbf{x}$ is in the same class as $\mathbf{w}_\mathrm{c}$, this weight vector is renewed by

$$\mathbf{w}_\mathrm{c}(t+1) = \mathbf{w}_\mathrm{c}(t) + \alpha(t)\bigl[\mathbf{x}(t) - \mathbf{w}_\mathrm{c}(t)\bigr], \tag{14}$$

   where $\alpha(t)$, the learning rate, decreases monotonically with time, $\mathbf{w}_\mathrm{c}(t+1)$ is the updated weight vector, and $\mathbf{w}_\mathrm{c}(t)$ is the original weight vector.
2. If $\mathbf{x}$ is not in the same class as $\mathbf{w}_\mathrm{c}$, this weight vector is computed as

$$\mathbf{w}_\mathrm{c}(t+1) = \mathbf{w}_\mathrm{c}(t) - \alpha(t)\bigl[\mathbf{x}(t) - \mathbf{w}_\mathrm{c}(t)\bigr]. \tag{15}$$

3. If the weight vector is not activated, that is, $i \neq c$, the weight vector does not change:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t). \tag{16}$$

Once all the weight values are given, the model is complete.

5. Experimental Results

5.1. Performance Evaluation

There are three predictors in the raw dataset, which is grouped into two classes. In this case, the prediction model has four different possible outcomes, as shown in Table 2.

The assumed two classes of samples are denoted as "positive" and "negative," respectively. Samples correctly classified as positive are defined as True Positive (TP); samples correctly classified as negative are defined as True Negative (TN). Samples wrongly predicted as positive are defined as False Positive (FP) and samples wrongly predicted as negative are defined as False Negative (FN).

Prediction performance is measured by using the TP rate and the TN rate.

The TP rate is defined as the ratio of the number of positive class samples predicted as positive to the number of actual positive class samples:

$$\text{TP rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{17}$$

Table 2 Different outcomes of two-class prediction.

	Predicted positive class	Predicted negative class
Actual positive class	True Positive	False Negative
Actual negative class	False Positive	True Negative

The TN rate is defined as the ratio of the number of negative class samples predicted as negative to the number of actual negative class samples:

$$\text{TN rate} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{18}$$

In solar flare prediction, the samples with flares are labeled as the positive class. Otherwise, they are labeled as the negative class. The TP rate and TN rate are used to evaluate the accuracy of "flaring" and "flare-quiet," respectively.

5.2. Experimental Results and Analyses

Data from 15 April 1996 to 10 January 2004 for Class 1 and the same number of data for Class 0 are selected to form the dataset. The dataset is divided into ten folds and then nine folds are used for training and the remaining fold for testing. The results of the ten tests are given as the mean of all tests.

The algorithms are implemented in WEKA (Waikato Environment for Knowledge Analysis), which is a data mining software written in java (Witten and Frank, 2005). WEKA can be freely downloaded from http://www.cs.waikato.ac.nz/ml/weka/. The default settings of C4.5 called J48 in WEKA are used. A type three learning vector quantization (LVQ3) algorithm is implemented by using the WEKA plug-in downloaded from http://wekaclassalgos.sourceforge.net/. Based on the basic principles of LVQ, the samples located in the crossing surface of the class distribution are considered more elaborately in the LVQ3 algorithm (Kohonen, 1990). The number of codebook vectors is set to 500 and the number of total training iterations is set to 50 000. Other parameters are kept at the default settings.

We change the window size from 0 to 45 and observe the variation of prediction performance with window size. Figures 10 and 11 show that the TP rate and the TN rate both increase with window size. When the window size is 45, the prediction accuracy is greatly improved. For a larger window size, an enormous amount of time should be used to build the prediction model but its performance is not greatly improved. Furthermore, too much data are supplied at the emergence of the active region for a larger window size. This influences the actual evolutionary process. So the experiment is not performed for window size larger than 45.

The TP rate of C4.5 is 74.88% with the original predictors and 81.65% with a window size of 45. It has increased by 6.77%, while the TN rate of C4.5 has increased by 11.18% as well. The performance of the LVQ algorithm has improved more remarkably. The TP rate of LVQ has changed from 70.1% to 82.6% (an increase of 12.5%), while the TN rate of LVQ has changed from 71.5% to 84.1% (an increase of 12.6%). With increasing window size, there is a rising tendency for both the TP rate and the TN rate for the two algorithms. This means that the evolutionary information of the predictors introduced by the sliding window is effective, and the C4.5 and LVQ can extract this information.

Figure 10 TP rate and TN rate of C4.5 with increased window size.

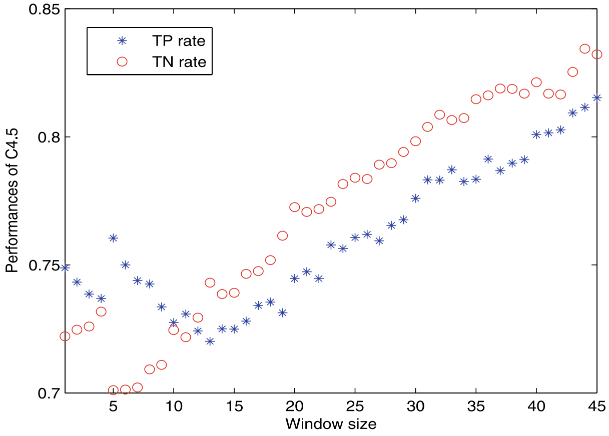
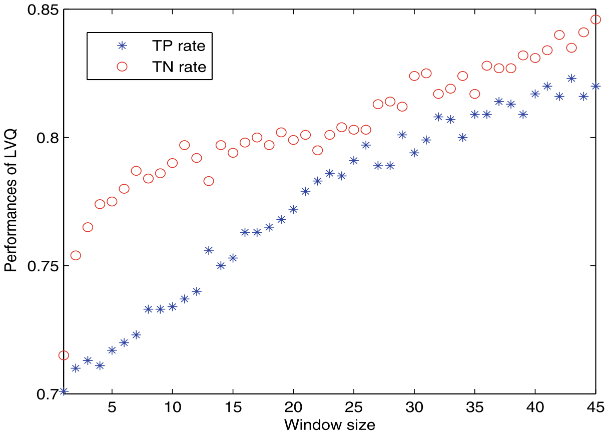


Figure 11 TP rate and TN rate of LVQ with increased window size.



5.3. Comparison with Other Prediction Methods

Many methods to predict solar flare level have been proposed. Some methods (Barnes *et al.*, 2007; Wheatland, 2005; McIntosh, 1990) assign a probability to the occurrence of a flare within a given time interval. Others (Qahwaji and Colak, 2007; Leka and Barnes, 2007; Wang *et al.*, 2007; Li *et al.*, 2007) assign a Class 0 or Class 1 to forecast whether the flare will happen. The forward-looking time is 6 hours (Qahwaji and Colak, 2007), 24 hours (Barnes *et al.*, 2007; Leka and Barnes, 2007; Wheatland, 2005), or 48 hours (Wang *et al.*, 2007; Li *et al.*, 2007). Some methods (McIntosh, 1990; Bornmann and Shaw, 1994; Gallagher, Moon, and Wang, 2002; Li *et al.*, 2008; Qahwaji and Colak, 2007) are based on the McIntosh classification of sunspots. Others (Leka and Barnes, 2007; Barnes *et al.*, 2007; Wang *et al.*, 2007) are based on the properties of the photospheric magnetic field.

The performance of the present method is compared with the performance of the method proposed by Wang *et al.* (2007). The back-propagation neural network is used to forecast the flare level within 48 hours in Wang *et al.* (2007), but the evolutionary information of predictors is not considered. As shown in Table 3, our method with window size 0 has a higher TP rate but lower TN rate than Wang's method. The performance of our method improves considerably with increasing window size. When the window size is 45, the TN

Table 3 Comparison of present method with method proposed by Wang *et al.* (2007).

	Wang's method	Our method			
		C4.5 $W=0$	LVQ $W=0$	C4.5 $W=45$	LVQ $W=45$
TP rate	69.8%	74.9%	70.1%	81.7%	82.6%
TN rate	84.7%	72.2%	71.5%	83.4%	84.1%

rate is comparable with Wang's method, and the TP rate is 11.9% for C4.5 and 12.8% for LVQ higher than that of Wang's method.

6. Conclusion

The influence of the evolutionary information of predictors on flare level is analyzed by using autocorrelation functions and mutual information functions. We found that not only does flare occurrence depend on the current photospheric magnetic field characteristics but it also is related to previous magnetic properties. A short-term solar flare prediction model is established within a sequential supervised learning framework. A sliding-window method is used to convert a sequential supervised learning problem into a standard supervised learning problem. Two guidelines are proposed for the determination of the window size, and window size is set to 45. The photospheric magnetic field information in three days or 45 samples is introduced into machine learning algorithms by using a sliding-window method. C4.5 and LVQ are employed to learn prediction models from this information. The experimental results show that short-term prediction of solar flares within the sequential supervised learning framework is effective. Compared with the method proposed by Wang *et al.* (2007), the accuracy of the "no flare" forecast is equal to that in Wang's method and the accuracy of the "flare" forecast is 10% higher than in Wang's method.

Some improvements can be made in future work. Both C4.5 and LVQ should be improved to give the probability outputs, and the influence of the sequence of previous flares on the current flare level should be considered.

Acknowledgements This work is supported by the National Basic Research Program of China (973 Program) through Grant No. 2006CB806307 and the National Natural Science Foundation of China (NSFC) through Grant Nos. 10673017 and 10733020. Dr. Han He and Dr. Qinghua Hu carefully revised this paper. Xin Huang especially thanks Dr. Kecheng Hao of Portland State University for carefully revising the manuscript. This paper has benefited from the comments of an anonymous reviewer. We thank the SOHO/MDI consortium for the data. SOHO is a project of international cooperation between ESA and NASA.

References

Barnes, G., Leka, K.D., Schumer, E.A., Della-Rose, D.J.: 2007, *Space Weather* **5**, S09002.
Bernhard, H.P., Darbellay, G.A.: 1999, *Proc. Acoust. Speech Signal Process.* **3**, 1297.
Bornmann, P.L., Shaw, D.: 1994, *Solar Phys.* **150**, 127.
Cohen, W.W., Carvalho, V.R.: 2005, In: *Proc. IJCAI* **1**, 1.
Cui, Y.M., Li, R., Zhang, L.Y., He, Y.L., Wang, H.N.: 2006, *Solar Phys.* **237**, 45.
Dietterich, T.G.: 2002, In: *Proc. of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* **2396**, 15.
Fozzard, R., Bradshaw, G., Ceci, L.: 1989, *Adv. Neural Inf. Process. Syst.* **1**, 248.
Gallagher, P.T., Moon, Y.J., Wang, H.: 2002, *Solar Phys.* **209**, 171.
Hu, Q.H., Yu, D.R., Xie, Z.X., Liu, J.F.: 2006, *IEEE Trans. Fuzzy Syst.* **14**, 191.

Klema, J., Novakova, L., Karel, F., Stepankova, O., Zelezny, F.: 2008, *IEEE Trans. Syst. Man Cybern. Part C, Appl. Rev.* **38**, 3.

Kohonen, T.: 1990, *Proc. IEEE* **78**, 1464.

Kohonen, T.: 2001, *Self-Organizing Maps*, Springer, New York.

Kontaki, M., Papadopoulos, A.N., Manolopoulos, Y.: 2007, *Data Knowl. Eng.* **63**, 478.

Koskinen, H., Eliasson, L., Holback, B., Andersson, L., Eriksson, A., Mälkki, A., Norberg, O., Pulkkinen, T., Viljanen, A., Wahlund, J.E., Wu, J.G.: 1999, SPEE final report, *FMI Reports* **1999:4**, Finnish Meteorological Institute, Helsinki.

Kusano, K., Maeshiro, T., Yokoyama, T., Sakurai, T.: 2004, *Astrophys. J.* **610**, 537.

Leka, K.D., Barnes, G.: 2003, *Astrophys. J.* **595**, 1296.

Leka, K.D., Barnes, G.: 2007, *Astrophys. J.* **656**, 1173.

Li, R., Wang, H.N., He, H., Cui, Y.M., Du, Z.L.: 2007, *Chin. J. Astron. Astrophys.* **7**, 441.

Li, R., Cui, Y.M., He, H., Wang, H.N.: 2008, *Adv. Space Res.* **42**, 1469.

Marko, L.: 2003, *Ind. Phys.* **9**, 24.

McIntosh, P.S.: 1990, *Solar Phys.* **125**, 251.

Mitchell, T.M.: 1997, *Machine Learning*, McGraw-Hill, New York.

Qahwaji, R., Colak, T.: 2007, *Solar Phys.* **241**, 195.

Quinlan, J.R.: 1993, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco.

Quinlan, J.R.: 1996, *J. Artif. Intell. Res.* **4**, 77.

Rust, D., Sakurai, T., Gaizauskas, V., Hofmann, A., Martin, S., Priest, E., Wang, J.: 1994, *Solar Phys.* **153**, 1.

Wang, H.N., Cui, Y.M., Li, R., Zhang, L.Y., He, H.: 2007, *Adv. Space Res.* **42**, 1464.

Wang, J., Shi, Z.: 1993, *Solar Phys.* **143**, 119.

Wheatland, M.S.: 2005, *Space Weather* **3**, S07003.

Witten, I.H., Frank, E.: 2005, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco.

Zheng, Z.: 1998, In: *Proc. of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining* **1394**, 348.