AN IMPROVED NAIVE BAYES MODEL FOR AIR TEMPERATURE PREDICTION

Binghong Jiang
Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China
E-mail: jbh20030925@stu.xjtu.edu.cn Tel: +86-15389268107

Abstract— **In this paper, a Temperature Probability Prediction with Multi Factor (TPPMF) based on Naive Bayesian algorithm (NB) is designed for weather data including multiple types and forms. The defect of continuous variable analysis is solved through the variable - probability transformation. A temperature increment algorithm is proposed to improve the accuracy of NB. And a compressed encoding method for feature vector of incompatible incidents achieve the dimensionality reduction of data expression, which reduced the training time for about 20%. The experiment results show that the root-mean-square of the mean-absolute-error minimum reached 1.698℃, in the 24-hour air temperature prediction of 12 typical cities.**

Keywords—**Air Temperature Prediction; Naive Bayesian; variable - probability transformation; feature vector compression**

I. INTRODUCTION

The impact of agricultural, ecological, environmental and industrial development on climate changing has always been attention. Among various meteorological characteristics, air temperature is the most concerned one due to its close relationship with energy industry and agriculture. Therefore, accurate temperature prediction models have been the focus of research for years [1].

In academia, there has been searching for data-driven temperature prediction models with high spatial and temporal resolution and accurate results for a long time. Two directions were mainly focused on, time series analysis based on ARMA and machine learning algorithms based on neural network. ARMA is a linear statistical technique for time series analysis and prediction. Reference [2] has made qualitative and quantitative prediction of air temperature in Harbin with ARMA. Based on the unique time correlation and bias of temperature and precipitation in time series, reference [3] proposed a forecasting method for statistical time series based on ARIMA model. Which has achieved remarkable results in regional prediction. Reference [4] achieved monthly temperature and precipitation prediction with Autoregressive Fractional Integral Moving Average (a variant of ARIMA). Meanwhile, other statistical models have also been attempted for temperature prediction. Relying on the monthly periodicity of air temperature, reference [5] achieved temperature prediction in 2000's Guangzhou, using a grey Markov model based on seasonal index. To make full use of prior information, reference [6] selected several different Bayesian models to predict the temperature in Nanning.

In recent years, the ANN (Artificial Neural Networks), SVM (Support Vector Machines) and other machine learning algorithms had been tried to deal with comprehensive analysis of a variety of weather indicators. MLPNN (Multilayer Perceptron Neural Networks) and RBFNN (Radial Basis Function Neural Networks) are the conventional strategies, among which Levenberg-Marquardt and gradient descent are the most commonly used optimization algorithms [7]. However, in real-world applications, once the type of meteorological indicators provided is changed, an extremely long period of time will be needed to retrain the neural network parameters. Meanwhile, the NN algorithms are also too complicated to adjust the edge computing applications with constrained resource.

In this paper, NB (Naïve Bayes) model is introduced to process the non-quantitative meteorological information and construct the Temperature Probability Prediction with Multi Factor (TPPMF). NB model cannot accurately analyze the continuously-changing parameters directly, a variable-probability-transformation function is constructed to transform the continuous time variable into probability, which helps realize the NB analysis of the continuous variable. Probability density of predicting temperature is constructed based on fitting and numerical differentiation, and proposes an accurate algorithm that finding temperature increment to improve accuracy, which aims to supply the deficiency of TPPMF that it only outputs the probability distribution function of temperature change. In addition, in order to avoid the cost of computational efficiency caused by non-quantized parameters, a coding method is presented of feature vector compression based on mutually exclusive events to achieve dimensionality reduction processing of data sets. Finally, we crawled the daily weather data of 3529 days in several cities, and uses the proposed method to forecast the daily average temperature. The results show that the mean absolute error (MAE) of the proposed method is 1.698℃, and the error can be further reduced if data samples of the same season are selected for training or the meteorological conditions of the forecasting city is relatively stable.

II. NB-based Multi-factor Air Temperature Predicting Model

NB is a typical probabilistic analysis method. The model can be used to analyze the inducements of an event, and the analysis results can be used to accurately predict whether the event will occur. NB takes the non-cooccurrence of features as the premise, and learns the joint distribution law from input to output through a given training set.

In practical application, when there are n mutually exclusive events $A_1, A_2, A_3, \cdots, A_n$ that may induce event B, NB can be defined with (1), in which a series of known cases are used to form analysis data to determine whether event B will occur in any given situation.

$$P(A_1, A_2, A_3 \cdots A_n|B) = \frac{P(B|A_1, A_2, A_3 \cdots A_n)P(A_1, A_2, A_3 \cdots A_n)}{P(B)} \quad (1)$$

NB is usually used to analyze whether a series of features have an impact on a target event and how much the impact is, and to predict whether a specific combination of features will lead to the occurrence of a target event. Weather prediction is a typical nonlinear decision problem which can predict results according to the characteristics of discrete events, so it can be solved by using NB. In this case, the target event is: the 24-hour increase of temperature exceeded a preset specific value.

In the analysis below, all meteorological characteristics that may lead temperature change, such as weather type, wind force, and wind direction, represented by features $F_1, F_2, F_3, \cdots F_m$, form up the set of weather features $\mathbb{L}$, as shown in (2).

$$\mathbb{L} = \{F_1, F_2, F_3, \cdots, F_n\} \quad (2)$$

Set $\mathbb{T}$ is constructed based on $\mathbb{L}$, which indicates the weather conditions corresponding to different observed temperatures. Each element of $\mathbb{T}$, $\boldsymbol{p_i} = (p_{i,1}, p_{i,2} \cdots p_{i,n})^{\mathrm{T}}$ is used to mark the combination of weather features at a particular sampling time, which is a vector with the same length as the size $\mathbb{L}$, as shown in (3).

Since NB is a probabilistic model, each dimension of $\boldsymbol{p_i}$ can only represent the " occurrence" or " non-occurrence" state of weather feature F_j from $\mathbb{L}$ at the sampling time point. Specifically, if the jth feature F_j from L appears, then the jth element $p_{i,j}$ in $\boldsymbol{p_i}$ values 1, otherwise it values 0.

$$\mathbb{T} \coloneqq \left\{ \boldsymbol{p}_{i,j} = \begin{cases} 1, F_j \in \{feature\ combination\ for\ \boldsymbol{p_i}\} \\ 0, F_j \notin \{feature\ combination\ for\ \boldsymbol{p_i}\} \end{cases} \right\} \quad (3)$$

Prediction result vector $\boldsymbol{q}$ is also defined in a similar way, as shown in (4).

$$\boldsymbol{q_{1\times m}} = \left\{ q_i = \begin{cases} 1, & temperature\ increment\ for\ \boldsymbol{p_i} \geq\ preset\ value \\ 0, & temperature\ increment\ for\ \boldsymbol{p_i} <\ preset\ value \end{cases} \right\} \quad (4)$$

From (1), the probability of each feature causing the temperature increment ΔT to be higher than a specific value (denoted as event R) can be obtained, as shown in (5),

$$P(R|F_1, \cdots, F_n) = \frac{P(F_1, \cdots, F_n|R)P(F_1, \cdots, F_n)}{P(R)} \approx \frac{\sum_{p_i \in \mathbb{T}} q_i p_i \sum_{p_i \in \mathbb{T}} p_i}{\left(\sum_{p_i \in \mathbb{T}} q_i\right)^2} \quad (5)$$

where $P(F_1, \cdots, F_n) \approx \sum_{\boldsymbol{p_i} \in \mathbb{T}} \frac{\boldsymbol{p_i}}{m}$ stands for the probabilities that characteristics in training sets turn up, $P(R) \approx \sum_{\boldsymbol{p_i} \in \mathbb{T}} \frac{q_i}{m}$ stands for the probability that the temperature increment ΔT exceeds a specific value, $P(F_1, \cdots, F_n|R) \approx \frac{\sum_{\boldsymbol{p_i} \in \mathbb{T}} q_i \boldsymbol{p_i}}{\sum_{\boldsymbol{p_i} \in \mathbb{T}} q_i}$ stands for posteriori probabilities that the increment exceeds the value given the characteristics have turned up.

Similarly, the probability that the temperature increment ΔT doesn't exceed a specific value can be calculated with (6).

$$P(R'|F_1, \cdots, F_n) \approx \frac{\sum_{p_i \in \mathbb{T}} (1-q_i) p_i \sum_{p_i \in \mathbb{T}} p_i}{\left[\sum_{p_i \in \mathbb{T}} (1-q_i)\right]^2} \quad (6)$$

Obviously, the characteristic F_j may lead to the temperature increment going higher than the specified value (event R) given that $P(R|F_j)$ is much larger than $P(R'|F_j)$. In contrast, the characteristic F_j will avoid event R given that $P(R|F_j)$ is much less than $P(R|F_j)$. When $P(R|F_j)$ is approximately equivalent to $P(R|F_j)$, the characteristic F_j has little effect on the temperature increment.

In this essay, the calculations above that analyzes the possibility of temperature increment being higher than the specific value (event R) is summarized as TPPMF, as shown in (7).

$$TPPMF - Bayes(R) = \frac{P(R|\mathbb{P})}{P(R|\mathbb{P})+P(R'|\mathbb{P})} = \frac{\prod P(R|F_{i_j})}{\prod P(R|F_{i_j})+\prod P(R'|F_{i_j})} \quad (7)$$

The larger $TPPMF - Bayes(R)$ is, the greater the probability of event R will be. Therefore, $TPPMF - Bayes(R)$ values at different values of ΔT can be calculated respectively when temperature prediction is carried out, and the maximum value corresponding to ΔT is where the most probable temperature increment lies.

Limited by the characteristics of NB model, (7) can only make inaccurate temperature predictions, but cannot meet the actual needs in practical applications. First of all, the weather features in the algorithm $F_1, F_2, F_3, \cdots, F_n$ can only be in two absolute states of "existence" and "non-existence", but cannot be in some intermediate state, and cannot be used in analysis of time, temperature and other continuously changing quantities. Secondly, the feature vector definition method in the algorithm has a lot of invalid redundant data, which may affect the efficiency of the algorithm. What's more, the results of the algorithm are discrete, and TPPMF can only calculate the high or low probability of temperature increment higher than a specific value, rather than making more accurate temperature predictions. Therefore, improves the TPPMF model will be improved to avoid the defects above in Section 2.

III. Improvements to Multi-Factor Air Temperature Prediction

A. *variable - probability transformation*

In view of the problem that describing features with probability is too absolute, real numbers between 0 and 1 are used to describe intermediate states between "occurrence" and "non-occurrence". By constructing a transformation function $f_j: \mathbb{D} \to [0,1], x \to p_{ij}$, the continuous variable is converted into a decimal that can be used as a probability element in $\boldsymbol{p_i}$s to deal with the problem of describing the continuous variables in NB model. Elements that are changed to decimals are called "probability terms" (or "probability dimensions"), while the functions are called variable-probability-transformation functions. There no fixed expression for transformation functions. Instead, they should be constructed according to the practical application and the characteristics to be expressed. The constructed function should meet the following limitations:

(1) Should be a bijection from the interval that covers all the possible values of the continuous variable to $[0,1]$.

(2) The conversion function should be easy to understand; Several elements in $\boldsymbol{p_i}$, rather than only one, should to be involved to represent one continuous variable if necessary.

For example, for the air humidity feature (which values the percentage of water vapor in the air), the conversion function could be $f_{\text{air\_humidity}}(x) = x$, since air humidity itself is a percentage value between 0 and 1. Another good example is the date feature, the transformation function can be $f_{\text{time}}(x) = \frac{1+\sin\frac{(x+122)}{365/2\pi}}{2}$, because the purpose of date features is to reflect the change of climate (especially temperature) over seasons. In this paper, we construct a year-period sine function (as shown above), whose value reaches the maximum in summer solstice and the minimum in winter solstice.

Therefore, the training set $\mathbb{T}$ in Section 1 can be expanded to include the set of continuous changes such as temperature, humidity and time, as shown in (8).

$$\mathbb{T} \coloneqq \left\{ \boldsymbol{p_i} \middle| p_{i,j} = \begin{cases} \begin{cases} 1, F_j \in \{feature\ combination\ for\ \boldsymbol{p_i}\} \\ 0, F_j \notin \{feature\ combination\ for\ \boldsymbol{p_i}\} \end{cases}, & unquantifiable\ F_j \\ f_j(x), & quantifiable\ F_j : x \end{cases} \right\} \quad (8)$$

B. *Compression of Feature Vector Based on Mutually Exclusive Events*

In weather description, certain features are obviously non-coexisting. For example, "sunny," "cloudy," and "cloudy to clear" are not likely to appear on the weather record at the same time. As a result, once one dimension of the weather feature vector $\boldsymbol{p_i}$ defined by (3) is valued as "1", the remaining dimensions must be valued as "0". Therefore, when the TPPMF model is used for weather feature data processing, there will be a large number of redundant "zeros" in the data set. In other words, if there's a matrix foamed up by all $\boldsymbol{p_i}$s, the matrix must be extremely sparse, which may waste a lot of memories and reduce the execution efficiency of the algorithm.

Dealing with this problem, for the mutually exclusive events group in the weather features, the group can be defined as an "non-coexisting set". $\mathbb{T}_u$ is the set composed of all "non-coexisting sets", as shown in (9).

$$\mathbb{T}_u := \left\{\mathbb{G} = \{F_{k_1}, F_{k_2}, \cdots\}, unquantifiable\ F_{k_i} \middle| \forall i, \sum p_{ik_j} = 1\right\} \quad (9)$$

In addition, it is defined that the "maximum non-coexisting set" is the "non-coexisting set" that is not any subset of the "non-coexisting set". Set $\mathbb{T}_{umax}$ composed of all the "maximum non-coexisting sets" is given by (10).

$$\mathbb{T}_{umax} := \{\mathbb{G}_{max} \in \mathbb{T}_u | \nexists \mathbb{G} \in \mathbb{T}_u, s.t. \mathbb{G}_{max} \subset \mathbb{G}\} \quad (10)$$

For "non-coexisting sets", a more efficient representation method of feature vectors is designed, as shown in Table 1.

Table 1 Compress feature vector by non-coexisting set

	Ranking by frequency in training set			Coding			Calculate vectors
	features	frequency		Frequency ranking	Code		Feature vector
19 features in total	SE wind	2371	⇒	max	1	⇒	(0,0,0,0,1)
	NW wind	244		2^{nd} max	3		(0,0,0,1,1)
	⋮	⋮		⋮	⋮		⋮
	E wind	18		medium	19		(1,0,0,1,1)
	⋮	⋮		⋮	⋮		⋮
	E-SE wind	2		2^{nd} min	4		(0,0,1,0,0)
	N-NW wind	1		min	2		(0,0,0,1,0)

The specific methods are as follows:

Step 1: Count the frequency of occurrence of all events in $\mathbb{T}_u$ and order them from most to least;

Step 2: Code the events from both sides to the middle of frequency ranking. The event with highest frequency codes 1, the one with lowest frequency codes 2, 2^{nd} most frequent one codes 3, 2^{nd} less one codes 4, 3^{rd} most one codes 5, and so on. Finally, the event with the frequency ranks the most middle codes the maximum number.

Step 3: The compressed feature vector of the event F_{k_i} is obtained by converting the encoded value obtained in Step 2 into binary.

With the process above, the length of the feature vector of $\mathbb{T}_u$ for n kinds of events are reduced from n bit to $[\log_2 n]$ bit, while the coding method in Step 2 can make the probability of each dimension to be 1 after compression as close as possible to 0.5 to reduce the error.

C. Accurate Method to Predict Temperature Increment

In the utilization of temperature prediction, only obtaining the trend prediction result like " temperature will drop tomorrow" cannot meet the requirement in practice. Instead, the results are often need to be accurate to degrees, while its reliability is also of concern. Based on the TPPMF model in Section 1, we solve the problem of using NB to calculate probability distribution and expected value by linear fitting, with reference to the polynomial-fitting-and-derivation methods in numerical differentiation. The specific methods are as follows:

Step 1: Set several thresholds $\{a_1, a_2, a_3 \cdots a_n\}$ in advance, so that the value of the target variable X is more likely to lie near the thresholds.

Step 2: Use TPPMF to calculate the probability P_k that the variable value is larger than (or less than) a specified value a_k.

$$\text{TPPMF-Bayes}(S_k) = P_k, S_k: x \leq a_k \quad (11)$$

Step 3: Carry out n-degree polynomial fitting of $\{a_k\}$ and $\{P_k\}$, and get the fitting result of the form $B(y) = \sum_{i=0}^{n} b_i y^i$, where b_i is the fitting coefficient.

Step 4: Carry out m-degree polynomial fitting $\{a_k\}$ and $\left\{\frac{B(a_k)}{P_k}\right\}$, with $m > n$, and get the fitting result of the form $C(y) = \sum_{i=0}^{m} c_i y^i$.

Step 5: Obtain the fitting distribution function $F(y)$.

$$F(y) = \frac{B(y)}{C(y)} = \frac{\sum_{i=0}^{n} b_i y^i}{\sum_{i=0}^{m} c_i y^i} \approx \text{TPPMF-Bayes}(S), S: x \leq y \quad (12)$$

Step 6: Derivate and normalize the obtained function $F(y)$ to calculate fitting possibility density $\Psi(y)$.

$$\Psi(y)dy = \lambda dF(y) \quad (13)$$

where λ is the normalized constant, and its value is such that

$$\int_{-\infty}^{+\infty} \Psi(y)dy = 1 \quad (14)$$

Step 7: Calculate the expectation of random variable X with $\Psi(y)$, which is the final prediction.

$$y_e = \int_{-\infty}^{+\infty} \Psi(y) y dy \quad (15)$$

D. Improved Multi-factor Air Temperature Prediction Algorithm

Based on the analysis above, variable-probability transformation and feature vector compression were involved to improve NB algorithm to achieve precise solution of temperature increment. The specific algorithm is as follows.

Algorithm 1 Air Temperature Prediction Model Training

Input: raw data, unquantifiable feature set *F*, quantifiable feature set *G*, feature set size *m*, training set size *n*, variable-probability transformation functions *f*.

Output: matrices *R_Feature* and *R'_Feature* describing correlation between weather features and R or $\bar{R}$

1: Calculate training set vectors $\boldsymbol{p_i}$ with raw data through (8)

and (9) and variable-probability transformation functions *f*.
2: **for** i=1 **to** n **do**
3: **for** j=1 **to** m **do**
4: Calculate *R_Feature[i][j]*s with $\boldsymbol{p_i}$ -related temperature increment in training set through (5).
5: Calculate *R'_Feature[i][j]*s with $\boldsymbol{p_i}$ -related temperature increment in training set through (6).
6: **end**
7: **end**

Algorithm 2 Air Temperature Prediction

Input: sample data for prediction, matrices *R_Feature* and *R'_Feature* describing correlation between weather features and R or $\bar{R}$, variable-probability transformation functions *f*, threshold array *x* for fitting.
Output: prediction *x_result*.
1: Calculate sample data vectors $\boldsymbol{p'}$ with sample data for prediction through (8) and (9) and variable-probability transformation functions f.
2: **for** i=1 **to** m **do**
3: Calculate value *P[i]*s of temperature possibility distribution function on *x[i]*s with $\boldsymbol{p'}$, *R_Feature* and *R'_Feature* through (7).
4: **end**
5: Fit temperature possibility distribution function *F* with *x[i]*, *P[i]* through (12).
6: Calculate possibility distribution Ψ with F derivation through (13).
7: Calculate expectation of temperature *x_result* as result with Ψ through (15).

The specific form of each transformation function f in the above algorithm is determined by its corresponding variable characteristics. In this essay, the transformation function is used to calculate the two dimensionally reflecting seasons in the feature vector, and all the remaining information is directly transformed into the dimensions in the feature vector through (4) and (5). According to season and periodicity of seasonal temperature change, sine function is adopted for variable-probability transformation between time and a_1. a_2, which indicates the season to differentiate time situation when a_1 values the same in spring and fall, equals 1 from winter solstice to summer solstice, or values 0 in other time during the year, reflecting the temperature change characteristics of rising temperature in the first half of a year and falling temperature in the second half of a year, as shown in (16),

$$f_T(a_1,a_2)\begin{cases} a_1=\dfrac{1+\sin\frac{2\pi(T-T_s)}{365}}{2} \\ a_2=\operatorname{sgn}\left[\cos\dfrac{2\pi(T-T_s)}{365}\right]\end{cases} \tag{16}$$

Where $\{a_1,a_2\}$ are the two elements in feature vector that indicate season, while T is for time, T_s stands for the time of summer solstice, and $\operatorname{sgn}()$ is the sign function.

IV. Simulation Experiment and Result Analysis

A. Dataset

In order to verify the correctness and availability of the algorithm above, and to explore the influence of different environmental factors on the accuracy of the model, we programmed a crawler to obtain the historical weather dataset of 13 cities from January 2011 to August 2020 in the Weather Post-report Website, for the detection of the predicting model. The weather dataset includes daily temperature, wind direction, wind strength, weather conditions, etc. In order to verify the model in this essay, the date of each group in the above data was taken as a quantifiable feature, and the weather conditions, wind force and wind direction were taken as quantifiable features, and the increment of the average temperature of the day compared with the one of day before was predicted based on such features through the improved NB model. The unquantifiable features in the data are shown in Table 2.

In order to analyze the influence of different environmental factors on the prediction accuracy of the model, we selected the temperature data of 12 cities including Kaohsiung, Haikou, Shanghai, Tianjin, Chongqing, Jinan, Yinchuan, Nanchang, Xi 'an, Harbin, Urumqi and Lhasa for analysis according to different combinations of climate and hydrological characteristics. The selected environmental factors and control variables that have influence on the prediction accuracy of the model are shown in Table 3. The cities corresponding to the combination of climatic and hydrological characteristics are shown in Table 4.

Table 2 Unquantifiable Features and Their Classification

Classification	Features
Weather condition	29 features include Cloudy, Light Snow, etc.
Wind Strength	13 features include Breeze(unmeasurable)、3-4 gale、≤3 gale, etc.
Wind Direction	19 features include Cyclostrophic, E, Unstable, etc.

Table 3 Environmental factors that have influence on the accuracy of model prediction and control variables (Abbreviations)

Regional Climate Characteristics (C.C.)	Regional Hydrological Characteristics	Size of Training Set
Equatorial Monsoon (TM) Sub-tropic Monsoon (STM)	Seaside (S) Riverside (R)	30(1 month) 60(2 months)

Temperate Monsoon (NM) Temperate Continental (NC) Alpine (A)	Inland (I)	90(1 quarter) 182(half year) 365(1 year) 1096(3 years) 1826(5 years)

Table 4 Cities corresponding to the combination of climatic and hydrological characteristics

C.C.	S	R	I
TM	Kaohsiung Haikou (beside strait)	-	-
STM	Shanghai	Chongqing	Nanchang
NM	Tianjin	Jinan	Xi'an Harbin (near frigid zone)
NC	-	Yinchuan	Urumqi
A	-	-	Lhasa

B. Experiment Scheme

Ex. 1. Evaluation of Model availability

Divide the weather data of Xi 'an from 2011 to 2018 into 9 batches according to the year, and feed them into the model as training sets separately. Predict the daily temperature in 2019 with the model whenever a batch was fed and the model was trained. Evaluate the availability of the model according to the error of predicting temperatures.

Ex. 2. Evaluation of sensitivity to size of training set

Set up batches with the sizes shown in Table2, column 3 (more different sizes were tried for higher accuracy of analysis), which include the temperature data in Xi'an of days up to 31th, Dec, 2018. Train the model with the batches above separately, and predict the air temperature of days during Jan, 2019. Evaluate the sensitivity of the model according to the difference of errors.

Ex. 3. Evaluation the impact of geographic environment (of the predicting place)

Set up batches with different sizes and include the temperature data of different cities up to 31th, Dec, 2019. Train the model with the batches above separately, and predict the air temperature of days during Jan, 2020. Evaluate the influences of the environments according to the difference of errors.

The errors of the model in the experiments above are evaluated by Mean Absolute Error (MAE), as shown in (17).

$$S_{MAE} = \frac{\sum_{i=1}^{n}|y_i - y_{i,real}|}{n} \quad (17)$$

C. Experiment Results

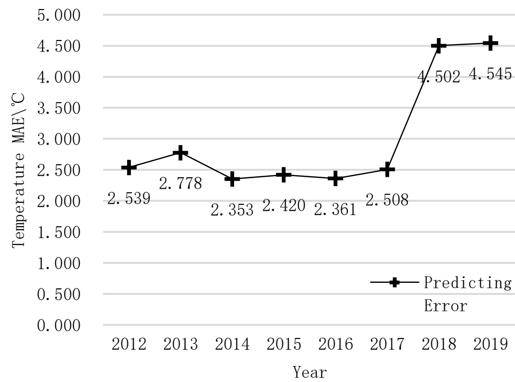

Fig. 1 Curve of predicting error of the model changing by year

The experiment indicates that the average MAE of the prediction results of the model is approximately 3℃, as shown in Fig.1. Before 2017 (the year 2017 is included), the MAE stabilized between 2.361℃ and 2.778℃, which is acceptable for practical applications. However, the value saw a sudden increase to around 4.5℃ in 2019 and 2020, which is presumably caused by the extreme weather in Xi'an from winter 2017 to summer 2019. Therefore, an accuracy prediction may be produced by the model in this paper given that the climate condition is relatively stable.

The MAE-sample number curve in Fig.2 reflects the influence of the number and type (seasons) of training samples on the accuracy of the model in this paper. It can be seen that the MAE value decreases with the increase of data quantity at the beginning, which reaches the very low point (2.372℃) when the number of samples reaches 60 days (2 months); then it rises, reaches the highest point (3.127℃) when the number of samples reaches about 365 days (1 year); finally it declines and gradually approaches a certain limit value (about 1℃). The reason for the occurrence of minimum point is that the time of the sample data adopted is basically in the same season with the time of temperature prediction, so the climate pattern is similar, and the learning effect of the model is relatively good. At the maximum point, although the training samples at this time can basically cover the weather type data of the whole year, the climate pattern learned at this time are not complete enough, so the prediction accuracy is unacceptable. With the increase of training set size, the climate pattern obtained from the model is gradually completed, so the prediction accuracy is gradually increased and converges to the limit value of algorithm accuracy.

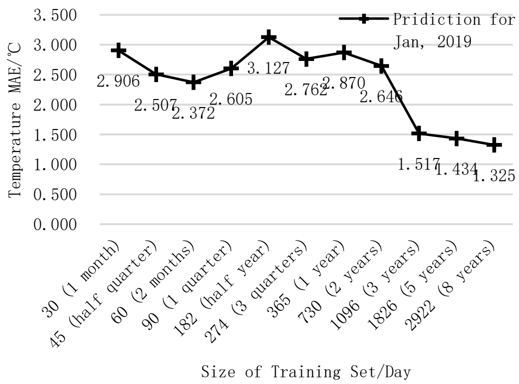

Fig. 2 Curve of predicting error of the model changing by size of training sets

In conclusion, we believe that training sets from the same season as the predicting time may ensure the accuracy of the model when the size of training set or the computing source of equipment is limited. Meanwhile, obtaining training sets with size over thousand may also help to lessen the error to a stable value efficiently.

Table 5 shows the influence of geographical factors of predicted locations and the number of training samples on the accuracy of the model. From the perspective of model accuracy, the MAE value of the model prediction results related to the urban climate characteristics presents a pattern of " Alpine (A) < Equatorial Monsoon (TM) < Sub-tropic Monsoon (STM) < Temperate Monsoon (NM) < Temperate Continental (NC)", and the related hydrological characteristics presents a pattern of "Riverside (R) < Seaside (S) < Inland (I)". Generally, the patterns above fit the law that the more stable the environment is, the more accuracy the prediction will be. From the perspective of the optimal size of training sets, it is advisable to use short-term samples, especially the samples gained nearly one month, to forecast the temperature of cities with relatively stable meteorological conditions; for cities with unstable conditions, a large training set with long-term sample data should be considered for prediction. Anyway, the accuracy of this model for temperature prediction in different cities has reached a high level in general. According to calculation, the square mean root of the lowest MAE in Table 3 is 1.698℃. Therefore, it shows that the model in this paper has good adaptability to utilizations in different geographical environments.

Table 5 Model error varies in size of training data and predicting cities

Cities	30	60	90	182	365	1096	1826
Haikou (TM-S)	1.320	1.430	1.876	**1.308**	1.776	1.275	**1.252**
Kaohsiung (TM-S)	2.611	2.379	**2.083**	2.226	2.156	1.908	**1.699**
Shanghai (STM-S)	2.817	3.296	3.607	**2.147**	3.055	2.453	**2.234**
Chongqing (STM-R)	**1.471**	1.732	1.893	2.074	1.896	**1.481**	1.491
Nanchang (STM-I)	**2.485**	4.643	4.093	3.371	2.866	**4.058**	4.453
Tianjin (NM-S)	**1.917**	2.423	2.227	2.166	3.088	1.955	**1.949**
Jinan (NM-R)	**1.414**	1.448	2.701	1.858	2.317	1.642	**1.935**
Xi'an (NM-I)	2.613	2.980	2.552	2.173	**2.064**	2.446	**1.687**
Harbin (NM-I)	3.909	4.772	3.809	**3.274**	3.771	2.451	**1.833**
Yinchuan (NC-R)	3.198	**2.337**	2.790	2.339	2.389	1.886	**1.651**
Urumqi (NC-I)	2.207	2.418	2.661	1.941	**1.506**	**1.373**	1.465
Lhasa (A)	**0.851**	1.415	1.992	1.792	**1.524**	2.194	2.242

In order to verify the effect of feature vector compression based on mutually exclusive events, statistics on the model training time before and after optimization in the case of input of different amounts of data was conducted, as shown in Fig. 3. With the increase of the amount of data, the advantages of feature vector compression become more and more obvious.

When the training set is large enough, the feature vector compression can reduce the model training time by about 20%.

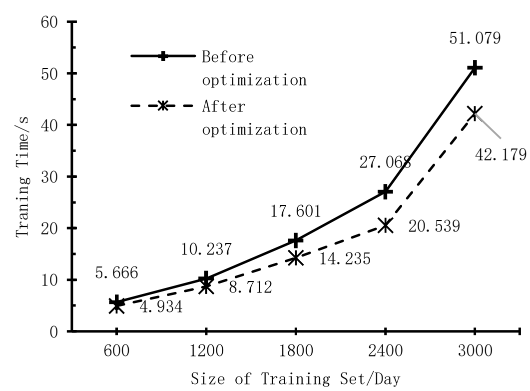

Fig. 3 Training time before and after optimization with different amount of training data

We also compared the training speed and accuracy of our model with some popular neural network model[8][9]. As shown in Table 6, our algorithm indicates a great advantage on the accuracy (still measured by MAE) while consuming such a short training time compared to the NN models. Meanwhile, the neural network models also show a great inflexibility while dealing with unquantifiable features (input in form of binary data): almost all neurons become inactivated (even under regularization) because of the monotonous 0-1 input, which finally leads to a larger error.

Table 6 Training time and accuracy of some neural network models and our model

Method	Training Time (s)	MSE (°C)
MLP[8]	330.1	2.376
LSTM[9]	9.1	2.472
LSTM&MLP[8]	17.5	2.539
NB (ours)	1.7	1.687

V. Conclusion

In this paper, traditional air temperature prediction algorithm, with defects of low universality, high complexity and difficult to carry out multi-variant analysis, is improved in three directions based on NB algorithm to carry out predictions more efficiently:

By constructing a set of result processing methods with fitting and derivation as the core, the expectation of temperature is calculated with the analysis results of NB algorithm, which makes the NB algorithm can be applied to accurate temperature value prediction. "Variable-probability transformation" makes it possible to input quantifiable and continuous variables into NB algorithm for analysis, which enables NB algorithm to deal with more different types of inputs. The mutually-exclusive-events-based feature vector compression coding effectively reduces the data storage space of the feature vector and improves the computational efficiency of the algorithm. In the actual application of multi-city temperature prediction from 2011 to 2020 based on weather conditions and wind conditions, the algorithm produces a series of high-accuracy results, and the lowest MAE reaches 1.698℃. Meanwhile, the prediction of cities with relatively stable meteorological conditions or the using training sets in the same quarter with the samples to be predicted can further reduce the model error.

References

[1] Li Zechun, Bi Baogui, Jin Ronghua, et al. The development and application of the modern weather forecast in China for the recent 10 years[J]. Acta Meteorologica Sinica, 2014, 72(06): 1069-1078.

[2] Lai, Yuchuan, Dzombak, David A. Use of the autoregressive integrated moving average (Arima) model to forecast near-term regional temperature and precipitation[J]. Weather and Forecasting, 2020(6): 959-976

[3] Georgia Papacharalampous, Hristos Tyralis, Demetris Koutsoyiannis. Predictability of monthly temperature and precipitation using automatic time series forecasting methods. Acta Geophys. 2018, 66, 807–831.

[4] Zhou Huaren, Li Haoran, Sun Xuejin, et al. A Temperature Forecasting Model of Grey-Markov Based on Seasonal Index[J]. Mathematics in Practice and Theory, 2016, 4: 167-173.

[5] Zhu Qingjie, Chen Jing, Lu SHilin, et al. The Application of Bayes prediction model in the prediction of temperature variety[J]. Journal of North China University of Science and Technology (Natural Science Edition)，2002，4：91-98.

[6] Jenny Cifuentes, Geovanny Marulanda, Antonio Bello, et al. Air Temperature Forecasting Using Machine Learning Techniques: A Review[J]. Energies,2020, 13(16)

[7] Sungjae Lee, Yung-Seop Lee, Youngdoo Son. Forecasting Daily Temperatures with Different Time Interval Data Using Deep Neural Networks[J]. Applied Sciences, 2020, 10(5)

[8] Kreuzer David, Munz Michael, Schlüter Stephan. Short-term temperature forecasts using a convolutional neural network — An application to different weather stations in Germany[J]. Machine Learning with Applications, 2020, 2 : 100007.