Journal of Data Analysis and Information Processing, 2017, 5, 131-139
http://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Predicting Precipitation Events Using Gaussian Mixture Model

Haitian Ling[1,2], Kunping Zhu[1*]

[1]School of Science, East China University of Science and Technology, Shanghai, China
[2]Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, USA
Email: haitian95@outlook.com, *kpzhu@ecust.edu.cn

How to cite this paper: Ling H.T. and Zhu, K.P. (2017) Predicting Precipitation Events Using Gaussian Mixture Model. *Journal of Data Analysis and Information Processing*, **5**, 131-139.
https://doi.org/10.4236/jdaip.2017.54010

Received: August 10, 2017
Accepted: October 7, 2017
Published: October 10, 2017

Abstract

In this paper, a Gaussian mixture model (GMM) based classifier is described to tell whether precipitation events will happen on a certain day at a certain time from historical meteorological data. The classifier deals with a two-class classification problem where one class represents precipitation events and the other represents non-precipitation events. The concept of ambiguity is introduced to represent cases where weather conditions between the two classes like drizzles, intermittent or overcast are more likely to happen. Six groups of experiments are carried out to evaluate the performance of the classifier using different configurations based on the observation data released by Shanghai Baoshan weather station. Specifically, a typical classification performance of about 75% accuracy, 30% precision and 80% recall is achieved for prediction tasks with a time span of 12 hours.

Keywords

Gaussian Mixture Model, Classification, EM Algorithm, Precipitation Event

1. Introduction

Predicting precipitation events, as a part of weather prediction, is often done by numerical weather prediction. Numerical weather prediction predicts future weather conditions with the help of partial differential equations. Various attempts to apply machine learning methods to weather prediction have been made but often with other methods than Gaussian mixture model. The earliest attempts to apply machine learning to precipitation prediction were made using perceptrons. More recent researches are often based on artificial neural network [1] and support vector machine [2]. A detailed review is available in [2]. Gaussian mixture model is a simple but effective model for classification and cluster-

ing compared with other classification models. It has many successful applications in various areas such as computer vision, digital signal processing, etc. For example, some recent researches use Gaussian mixture model for object tracking and segmentation [3] [4]. In this paper, we attempt to predict precipitation events using Gaussian mixture model (GMM). Six groups of experiments are carried out to evaluate the performance of the classifier based on the observation data. Furthermore, instead of predicting accurate precipitation, we only considered a single class of precipitation events regardless of how much precipitation is observed. These are the main points how this paper differs from other researches. The rest of this paper is organized as follows. In Section 2, we briefly describe Gaussian mixture model, expectation-maximization (EM) algorithm and our classifier. In Section 3, details of implementing the model are discussed. In the last section, the experimental results are given and an analysis of the results is also presented.

2. Model Description

2.1. Gaussian Mixture Model

Given an n-dimensional vector $\boldsymbol{x}$, a Gaussian mixture probability density function can be written as follows,

$$p(\boldsymbol{x}) = \sum_{i=1}^{m} w_i p_i(\boldsymbol{x}) \tag{1}$$

where m represents the number of mixture components, and mixture weights w_i satisfies $\sum_{i=1}^{m} w_i = 1$ and $w_i \geq 0$. Each component density $p_i(x), i = 1, 2, \cdots, m$ is the probability density function of a Gaussian distribution parameterized by a $n \times 1$ mean vector μ_i and a $n \times n$ covariance matrix Σ_i. Component densities can be written as follows

$$p_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \det(\boldsymbol{\Sigma}_i)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}} (\boldsymbol{\Sigma}_i)^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_i)\right\}. \tag{2}$$

Given the value of m, the value of $3m$ parameters, w_i, μ_i and Σ_i, $i = 1, 2, \cdots, m$, can then be determined. EM algorithm is used to estimate these parameters. For a classifier with K classes, a GMM is trained for each class. These models are denoted by $\lambda_k, k = 1, 2, \cdots, K$. λ_k can also be used to denote its parameters, that is $\lambda_k = \left\{w_i^k, \mu_i^k, \Sigma_i^k\right\}, i = 1, 2, \cdots, m$.

2.2. EM Algorithm

In this paper, the parameters of GMMs are estimated using Expectation Maximization algorithm (EM), an algorithm to find the maximum likelihood estimate of unknown parameters. For a data set with g feature vectors $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_g\}$, the likelihood function of GMM can be written as follows

$$L = \prod_{j=1}^{g} \sum_{i=1}^{m} w_i p_i(\boldsymbol{x}_j) \tag{3}$$

A detailed description of EM algorithm can be found in [5].

Since EM typically converges to a local optimum and involves random initialization, the estimated parameters may sometimes result in poor model performance. To solve this problem, a workaround is proposed as described later.

2.3. Precipitation Events Classifier

For a classifier with K classes $\lambda_k, k=1,2,\cdots,K$, a feature vector $\boldsymbol{x}$ is assigned to the class with the greatest posteriori probability. That is, assign $\boldsymbol{x}$ to class λ_j if

$$p\left(\lambda_j|\boldsymbol{x}\right) \geq p\left(\lambda_k|\boldsymbol{x}\right),\ k=1,2,\cdots,K \tag{4}$$

Using Bayes' theorem, this can also be written as

$$\frac{p\left(\boldsymbol{x}|\lambda_j\right)}{p\left(\boldsymbol{x}|\lambda_k\right)} \geq \frac{p\left(\lambda_k\right)}{p\left(\lambda_j\right)},\ k=1,2,\cdots,K \tag{3}$$

where $p\left(\lambda_k\right)$ stands for the priori probability of class λ_k.

λ_1 is used to denote the class of precipitation events, and λ_2 is used to denote the class of non-precipitation events. The precipitation events classifier deals with a two-class classification problem. In this paper, we let $p\left(\lambda_1\right)=p\left(\lambda_2\right)$, thus a vector $\boldsymbol{x}$ is assigned to the class with the greatest Gaussian mixture density value. That is, the classifier reports precipitation events if

$$p\left(\boldsymbol{x}|\lambda_1\right) > p\left(\boldsymbol{x}|\lambda_2\right) \tag{6}$$

and reports non-precipitation events otherwise. In practice, these values are computed and compared in their log form, thus the above inequality is evaluated as follows

$$\log p\left(\boldsymbol{x}|\lambda_1\right) > \log p\left(\boldsymbol{x}|\lambda_2\right). \tag{7}$$

For the precipitation events prediction problem, feature vectors of different classes can appear very close to each other in terms of distance. In such cases, the prediction results are often inaccurate. For this reason, the prediction results are flagged as ambiguous if

$$abs\left\{\log p\left(\boldsymbol{x}|\lambda_1\right) - \log p\left(\boldsymbol{x}|\lambda_2\right)\right\} < \log 2 \tag{8}$$

which is the same as

$$\frac{\max\left\{p\left(\boldsymbol{x}|\lambda_1\right), p\left(\boldsymbol{x}|\lambda_2\right)\right\}}{\min\left\{p\left(\boldsymbol{x}|\lambda_1\right), p\left(\boldsymbol{x}|\lambda_2\right)\right\}} < 2. \tag{9}$$

When classified as ambiguous, the prediction results are considered close to the cases where weather conditions are between the two classes like drizzles, intermittent or overcast. However, the authenticity of the above claim is not tested since doing so will make a multiclass classification problem. In the case of evaluating the classification performance, data points flagged as ambiguous are not involved in the evaluation process. From our experimental results, we found that

in most cases, about 10% of all data are flagged as ambiguous.

3. Implementation

3.1. Data Acquisition and Feature Extraction

In this paper, the meteorological data of Shanghai, China is used for experiments. The data are obtained from Shanghai Baoshan weather station, station id 58,362 (Historical data obtained from http://www.meteomanz.com/). The station issues observation data 8 times each day, with a fixed interval of 3 hours.

We have chosen temperature, relative humidity, sea level pressure, wind direction, wind speed, total cloud cover and precipitation as features. Thus, a set of 7×1 feature vectors can be obtained after feature extraction. Some fields of the observation data are omitted, this is done to avoid the need to cope with too many missing data. Specifically, when wind speed is equal to 0, we let wind direction be 0. When converting original data to feature vectors, a normalization process is applied to ensure all components of the feature vectors have a lower bound of 0 and an upper bound of 100. This is simply done by linear transformations. All the features used by our model are listed in Table 1.

3.2. Preprocessing

Since observation data are given in the SYNOP format (FM-12), all possible weather conditions in observation data are known (see http://weather.unisys.com/wxp/Appendices/Formats/SYNOP.html for detail). These weather conditions are divided into the two classes and the corresponding feature vectors are accordingly classified for training. Specifically, fog, mist, haze and overcast are considered non-precipitation events, intermittent, drizzle and snow are considered precipitation events.

Even though features in observation data that contain too many missing data are omitted, there are still cases where data can be absent due to difficulty of observation etc. In such cases, these data rows are simply removed since removing these data have no effect on training or testing the classifier. This step can cause a data loss of about 60%.

When training GMMs, diagonal covariance matrices are used instead of full

Table 1. All features used in feature vectors.

Feature	Unit	Value range
Temperature	˚C	[−30, 50]
Relative humidity	%	[0, 100]
Sea level pressure	Hpa	[950, 1050]
Wind direction	˚	[0, 360]
Wind speed	Km/h	[0, 50]
Total cloud cover	N/A	[0, 1]
Precipitation (averaged by hour)	mm	[0, 10]

covariance matrices. This is done because it has been found that doing so will not only make GMMs perform better in practice but will also significantly reduce the computation needed since inversion of matrices is computationally intensive [6].

3.3. Performance Evaluation

For our classification model, we refer to the class of precipitation events as positive class and non-precipitation events as negative class. Subsequently, we denote the number of actual positive data points being classified as positive by true positives (TP) or by false negatives (FN) if being classified as negative. Similar definitions can be given for true negatives (TN) and false positives (FP). To evaluate the performance of the classifier, the definition of classification accuracy is introduced. Instead of defining classification accuracy as the ratio of correctly classified samples to all samples in the data set, we define classification accuracy as follows

$$\text{accuracy} = \left[1 - \frac{1}{2}\left(\frac{\text{FN}}{\text{TP}+\text{FN}} + \frac{\text{FP}}{\text{TN}+\text{FP}}\right)\right] \tag{10}$$

Classification accuracy is defined this way because precipitation events happen less often than non-precipitation events, precipitation events data typically take up only 10% of all data, which will cause FN and TP have little effect on classification accuracy. Precision and recall are also used as key factors to evaluate classification performance, defined as follows

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{11}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{12}$$

Since EM typically converges to a local optimum and involves random initialization, a single test is not enough to assess model performance. Thus, the classification accuracy, precision and recall are averaged over 10 trials and the averages are used as metrics for performance evaluation.

4. Experimental Results

In this section, the experimental results obtained from six groups of experiments carried out to evaluate the performance of the model with different configurations are described. To illustrate the effect of the amount of data, two sets of data are used, one of which contains 3 years of historical data and the other set contains 11 years of historical data. 2015 and 2016 are chosen as the source of the 3-year data set and year 2006 to 2016 as the source of the 11-year data set. Specifically, a subset of the whole data set is chosen as training set and the remainder as test set, the number of data points is about 2:1 for training set and test set respectively. The 11-year data set is used for most of our experiments apart from experiment 2, where the model performances with different data sets are com-

pared.

Similarly, GMMs of different number of mixtures are used, namely 16, 32, 64, 128. 64 mixtures are used except for experiment 1, where the effect of number of mixtures is assessed. A time span of 12 hours is used for predictions except for experiment 5.

In the first experiment, we compared GMMs of different number of mixtures and found that these models have similar performance regardless of their number of mixtures from the results shown in Figure 1. This could mean that a number of mixtures as small as 16 may already be enough to represent the distribution of the feature vectors when there is enough training data.

We can tell that GMM generalize to the observation data well from the fact that there is little performance loss for test data compared with training data.

In the second experiment, the 3-year data set is used to train the GMMs and test their performance. A significant decrease in both accuracy and recall is observed in the experimental results shown in Table 2.

This could be a clue that 2 years of training data may not be enough as opposed

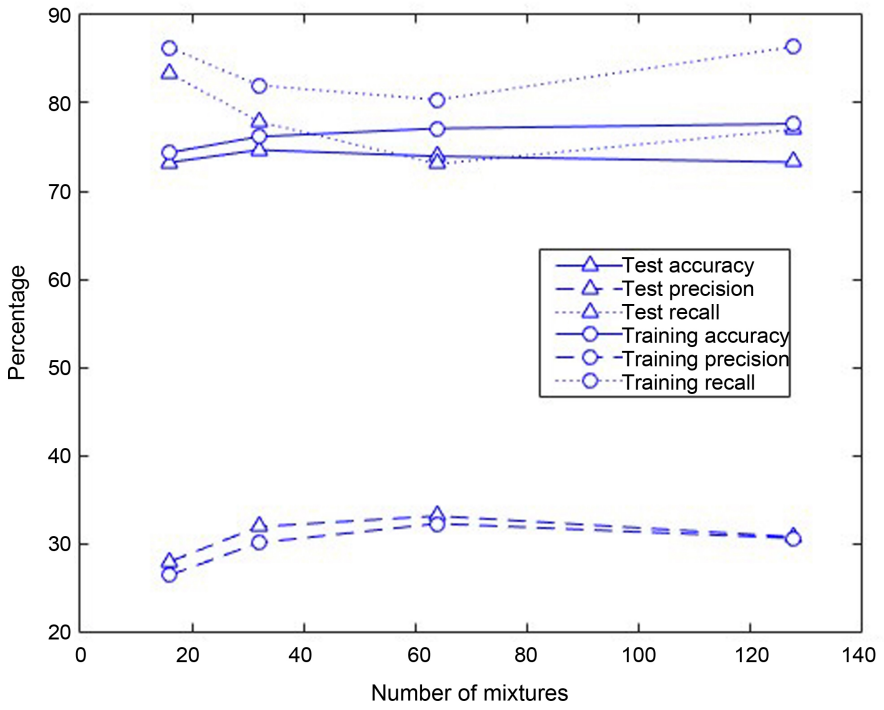

Figure 1. Comparison of classification performance for different number of mixture components.

Table 2. Experimental results for experiment 2, comparison of data sets.

	Training accuracy	Training precision	Training recall	Test accuracy	Test precision	Test recall
11-year	77.11%	32.22%	80.36%	73.95%	33.10%	73.08%
3-year	80.40%	47.77%	85.39%	65.86%	32.85%	56.16%

to 7 years of training data and that a greater number of training data points can lead to better classification performance. Additionally, a slightly higher training performance but lower test performance is observed for the 3-year data set, this means that GMM is slightly overfitting the training data. Though not strictly tested 10 times, a test run using the 3-year data set and 128 mixture components have shown a training accuracy of over 90% and a test accuracy of about 65%, which is apparently a sign of overfitting.

We tried training GMMs with separated daytime and nighttime data and separated season data in experiment 3 and experiment 4 respectively as shown in Table 3. Concretely, a set of GMMs is trained solely for daytime observation data and another set of GMMs for all nighttime observation data in experiment 3. In experiment 4, a set of GMMs is trained solely for spring observation data and another set of GMMs for summer observation data. In either case, the model performance is tested against their corresponding test data. The test results suggest that separating training data has no obvious effect on model performance.

In experiment 5, we measured how fast the predicting power of the model decrease with increasing prediction time span. The results are illustrated in Figure 2. Little prediction power is observed particularly for the 48-hour time span model, with an extremely low precision and a classification accuracy of about 50%. This suggests that the GMM based classifier may have little application value beyond the point of 24 hours.

In the last experiment, we tried adding more information to the original feature vector by appending a feature vector of observation data 12 hours before the prediction is being made. Doing so forms a new 14×1 feature vector that is virtually two combined 7×1 feature vector. The test results indicate a slightly negative effect on classification performance as compared with the first experiment, which is shown in Table 4. The increase in the complexity of feature vectors may have made it harder for GMM to model the relationship between features and classes.

In this paper, the same priori probabilities are chosen for both classes and log2 as the threshold for determining ambiguity. This is done because, for one thing, we want to ensure the availability of enough training data since model generalize poorly with insufficient training data. For another, the purpose of this paper is

Table 3. Experimental results for experiment 3 and 4, separating day/night spring/summer data.

	Training accuracy	Training precision	Training recall	Test accuracy	Test precision	Test recall
Full	77.11%	32.22%	80.36%	73.95%	33.10%	73.08%
Day	79.35%	35.86%	83.40%	73.72%	35.18%	72.33%
Night	79.34%	32.47%	83.35%	72.28%	29.97%	67.13%
Spring	79.14%	39.40%	87.74%	72.78%	35.68%	65.29%
Summer	78.92%	33.50%	84.97%	68.80%	36.73%	65.78%

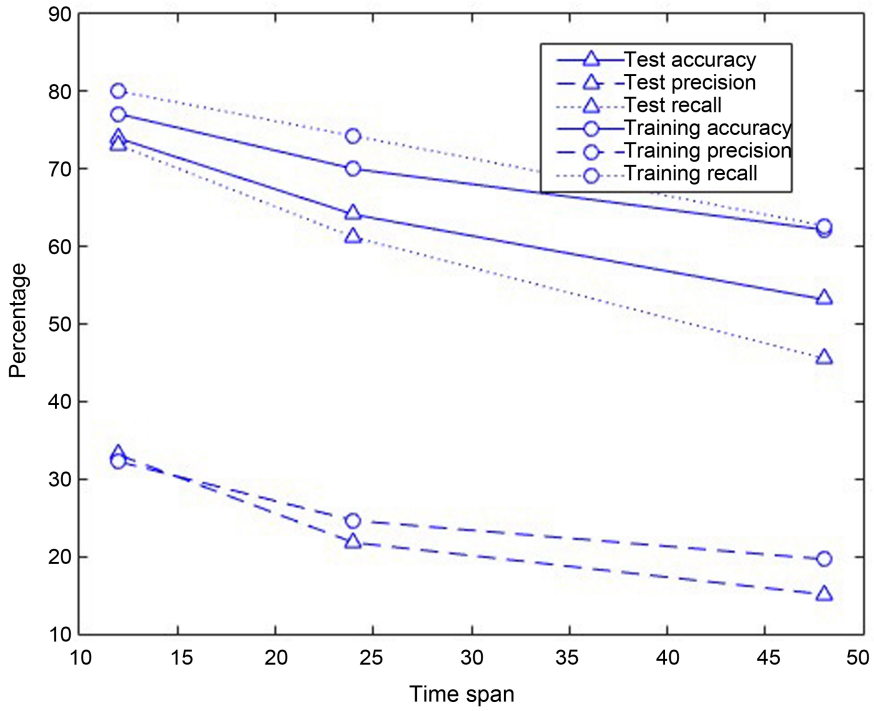

Figure 2. Comparison of classification performance for different time span.

Table 4. Experimental results for experiment 6, comparison of different features.

	Training accuracy	Training precision	Training recall	Test accuracy	Test precision	Test recall
7-dimensional	77.11%	32.22%	80.36%	73.95%	33.10%	73.08%
14-dimensional	74.27%	29.54%	75.94%	68.30%	25.44%	63.67%

not focused on determining the best performance the classifier can achieve but to find out how a GMM based classifier behaves to the precipitation events prediction task. To determine the optimal value for the above parameters, cross validation sets should be introduced which will cause training sets and test sets have access to less data.

5. Conclusion

From the above experimental results, the conclusion can be drawn that GMM is an effective model for predicting precipitation events. In this paper, the classifier built is one with high recall and low precision, such a classifier may be desirable when failure to prepare for precipitation events would bring about serious consequences. By altering the value of priori probabilities, a classifier with higher precision and lower recall can be obtained, since lowering the priori probability of precipitation events will make the classifier report precipitation events only when the classifier is very confident about the result. In this sense, the classifier may also be useful in cases where false alarm should be avoided. In our experiments, it is estimated that one of the most important factors affecting classifica-

tion performance may be the availability of features. Cases where an unambiguous feature vector belonging to λ_1 is classified to λ_2 are observed and vice versa. This may mean that the 7-dimensional feature vector does not contain sufficient information to almost uniquely determine future weather conditions even for a time span of 12 hours. The claim can also be verified from the training accuracy shown in **Figure 1**. Thus, predicting precipitation events using other classification models like SVM (Support vector machine) may bring no improvements. The observation data are restricted to ground observation in this paper, utilizing other types of data such as satellite observation may be an effective way to improve the performance of classification.

References

[1] Liu, L. and Ye, W. (2010) Precipitation Prediction of Time Series Model Based on BP Artificial Neural Network. *Journal of Water Resources and Water Engineering,* **21**, 156-159.

[2] Ortiz-García, E.G., Salcedo-Sanz, S. and Casanova-Mateo, C. (2014) Accurate Precipitation Prediction with Support Vector Classifiers: A Study Including Novel Predictive Variables and Observational Data. *Atmospheric Research*, **139**, 128-136.

[3] Genovese, M. and Napoli, E. (2014) ASIC and FPGA Implementation of the Gaussian Mixture Model Algorithm for Real-Time Segmentation of High Definition Video. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **22**, 537-547. https://doi.org/10.1109/TVLSI.2013.2249295

[4] Chauhan, A.K. and Krishan P. (2013) Moving Object Tracking Using Gaussian Mixture Model and Optical Flow. *International Journal of Advanced Research in Computer Science and Software Engineering*, **3**, 243-246.

[5] Webb, A.R. (2003) Statistical Pattern Recognition. Second Edition, Wiley, Hoboken, NJ.

[6] Reynolds, D.A., Quatieri, T.F. and Dunn R.B. (2000) Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, **10**, 19-41. https://doi.org/10.1006/dspr.1999.0361