Chapter 2

Modeling Unreliable Data and Sensors

Using Event Log Performance and F-Measure Attribute Selection

Vasanth Iyer, S. Sitharama Iyengar, and Niki Pissinou

Contents

2.1	Introduction	32
2.2	Background	33
2.3	State of the Art	33
2.4	Sensor Measurement and Fire Activity	35
2.4.1	Sorting by BA (ha)	35
2.4.2	Sorting by BA with Temporal Attributes	36
2.4.3	Estimating Training Values with Sample Data	36
2.4.4	Algorithm Complexity	37
2.5	Alarm Ranking Function for Accidental Small Fires	37
2.5.1	Ranking Function	37
2.5.2	F-Measure	38
2.5.3	Performance of Fire Topics Classification Using Temporal Ranking	38
2.5.4	Ranking Accidental Small Fires	39
2.6	Alarm Ranking Function for Large Fires	39
2.6.1	Performance of Fire Topics Classification Using FWI Ranking	40
2.6.2	Misclassification and Cost of False Alarms	40
2.7	Machine Learning Algorithms	41
2.7.1	Naive Bayes	42
2.7.2	User Query	42
2.7.3	Tree Classifier	44
2.7.4	Attribute Selection	45

2.8 Simulation 45

 2.8.1 Simulation Analysis..... 46

 2.8.2 Error Analysis..... 46

2.9 Correlation of Attributes 49

2.10 Better Model for Weather Data 50

2.11 Future Work..... 50

2.12 Summary 51

Acknowledgments 52

References 52

2.1 Introduction

During preprocessing event log, training samples are sorted by their area of fire damage according to high, medium, small, and accidental small fire classes. The preprocessing step allows studying the probability distribution and in our case the samples are highly skewed, giving an estimate that accidental small fires are more likely compared to large fires. Statistically we are interested in the many factors that influence accidental small fires; from the training samples, it follows a normal distribution. Graphically describing normal distribution, it takes the form of a bell-shape curve, which is also known as the Gaussian function. This is a first approximation for a real-valued random variable, which tends to cluster around a single mean value. The sensor model needs to learn the expected ranges for the baseline attributes being measured, giving better density estimation with increasing samples count. The baseline discrete parameters capture only the sensor ranges, making event prediction function hard to train with a Gaussian density function, without specific temporal understanding of the datasets. The dynamic features present in a sequence of patterns are localized and used to predict events, which otherwise may not be an attributing feature to the static data mining algorithm.

We have studied the spatial features and baseline discrete sensor measurements and all the attributes available that have a high classification error [1], which can sometimes amount to 50% of the error in the case of accidental small fire category. To further investigate factors that can cause such fires, we include data pertaining to human specific temporal attributes such as number of visitors and traffic patterns coming into the forest area, thus filtering events with local significance. Temporal attributes is a better estimator, given the type of training samples, which are difficult to calibrate and any approximation may induce false alarms. Relevance-based ranking function is highly suitable to order higher bound sample chosen by domain experts as ideal estimates and still maintaining the desired low false alarm rates. The method of ranking uses the function of sensor precision and event relevance weights, which are then linearly added to represent data from fire activity logs. The rest of the chapter is organized as follows. Sections 2.2 and 2.3 provide related work and state of the art. Section 2.4 defines sensor measurements and fire activity to model the data and algorithm computational complexity. Section 2.5 defines the additive ranking weight to classify accidental small fires using precision and relevance of the event collection. Section 2.6 discusses the performance of the large fire ranking function in terms of Fire Weather Index (FWI) [2] attributes and uses precision and discusses error rate in terms of false alarms. Sections 2.7 and 2.8 discuss preliminary information retrieval (IR) [3] choices with extensive simulation using Machine Learning (ML) probabilistic algorithms. Section 2.9 uses WEKA [4,5] tool to automatically do attribute selection ranking, this allows to a good statistical knowledge for weather data, and in Section 2.10, the constraints related to machine learning model with respect to weather data and

how to use Bayes net are discussed. Future work in Section 2.11 discusses massive datasets and how to computationally use a new framework and port ML algorithms. The chapter concludes in Section 2.12 with summary of results performance.

2.2 Background

The machine learning repository [6] provides collection of supervised databases that are used for the empirical analysis of event prediction algorithms with unsupervised datasets from distributed wireless sensor networks. Sensor network generate huge amounts of data that need to be validated for its relevance; keeping only necessary data helps avoid high computational overloads due to data redundancies. Calibration of sensors may not be always possible and the data aggregating algorithms need to have domain rules to detect any outliers from the datastream, given other parameters are kept constant. Given a dataset of forest fire events for a region, the training algorithm will be able to transform correlated attributes from sensor networks datastream to validate and classify the events and reject the outliers reliably. The preliminary work models the empirical data with a ranking function without spatial information to predict the likelihood of different events. The concept of IR, such as precision and relevance, is used in context to sensor networks, which not only allow to understand the domain topics but also add high reliability to the large dataset. In this case study, we use forest fire and environmental conservation as the theme and study which environmental factors attribute to such events, for example, temporal attributes such as humans, peak weather conditions, surface fuel buildups, and wind spread factors. The Burnt Area (BA), which is the ground truth is broadly studied with respect to small and large fires. The framework needs to extend queries in topics that are spatially aware, making sensing an essential source of discovering information.

2.3 State of the Art

The sensitivity of a sensor network not only depends on the accuracy of individual sensors but also on the spatial distribute collaborative sensing of neighboring sensors. Now we can define the data mining criteria, which is a standard measure to evaluate queries in a sensor networks.

Precision How close a features measurement match at every independent value, when measured in time.

Accuracy How redundant a basic feature sampling measurement assumption are, due to the fact the independent and identically distributed (i.i.d) statistical for measuring correlated spatially arranged sensors do not hold good for medium to large weather data sets (e.g. Confusion matrix shown Table 2.1 and the formula for ranking precision and accuracy are given below). The more standard form of precision and recall is given in Ranking section in 2.1.1.

Table 2.1 Confusion Matrix of Classifier

	<i>Positive</i>	<i>Negative</i>
Positive	tp	fn
Negative	fp	tn

Figure 2.1 Bayes Net representation.

$$\text{accuracy} = \frac{t_p + t_n}{n}$$

$$\text{precision} = \frac{t_p}{t_p + f_p}$$

$$F_\beta = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The first criteria, which is precision, cannot be generalized in the case of weather data, as weather data are not independent and identically distributed (i.i.d.), but happens to be highly correlated feature. As most of the generative models like Naive Bayes assume i.i.d. strictly, one needs to find alternative models, which better approximate the observed samples. A better model using conditional probabilities compared to the basic Nave Bayes is its network dependent version called Bayes Net. This model allows to specify conditional relationships that are dependent and independent of the observed phenomena respectively. A typical Bayes Net is illustrated in Figure 2.1, which identifies entities such as small, large forest fires and its dependencies. The multi-feature allows to analyze the general pattern in weather data, and local events are further studied by including temporal dependencies in the form of human traffic patters. The enhanced model not only improves the statistical assumptions [7], it also performs much better in error handling, which is one of the major disadvantage of large low-resource energy constrained sensor networks. In this work, one of the goals is to better estimate the performance of the underlying algorithms used. To illustrate the need for such a metric we use three algorithms and measure its precision and accuracy as shown in Table 2.2. The average is calculated by combining precision and accuracy. Algorithm III has the highest average even though it has very low precision; to overcome this limitation we adapt a weighted average, which is also called the F-score. The calculated value of F-score is shown in the final column for all the three algorithms, and now using F-score the weights do properly rank the algorithm III as the lowest due to its low precision value.

Adding sensor data to the manual forest fire event logs allows the study of automated correlated real-time information, which properly trained allows estimation and classification of future sensor outputs from the same geographical region. Event logs contain spatial information such as GPS

Table 2.2 Mean Comparison of Precision and Recall

	<i>Precision</i>	<i>Recall</i>	<i>Average</i>	<i>F-score</i>
Algorithm I	0.5	0.4	0.45	0.444
Algorithm II	0.7	0.1	0.4	0.175
Algorithm III	0.02	1.0	0.51	0.0392

and the area of the fire-damaged region but lack correlated information which could lead to better estimation of the fire event under study. Moreover, the spatial information is collected manually and takes considerable amount of time to classify it, while the sensor measurements are measured in real-time and can be approximated by machine learning algorithms to classify events and notify if a fire alarm condition has reached.

Ranking functions allows filtering unrelated data and present only relevant information to the user's query [8–10]. In the IR domain, there are many efficient ways to rank the relevance of a document in a collection given a user query. Similarly, we like to rank the order of an event occurring given the precision and relevance of the prior fire probabilities and a hypothesis. The terms defined as precision and relevance are inversely proportional. In problems where the recorded evidence is small and rare, one can use a precision scale instead to rank the evidence and making other correlated events as relevant balancing out the summed weights used in events ranking. We rank the precision weights higher whenever fire events occur along with higher alarm conditions. The high alarm condition is always true for precision ranking and holds across accidental small, medium, and large fires equally, unlike relevance, which accounts for majority of fires including only the accidental small fire ones. The precision ranking uses ground truth such as actual fire evidence, further eliminating any possibility of errors due to outliers and weak evidences. These ground truths are well-established naturally occurring phenomenon, which occur rarely and leave significant evidence of the BAs in meter square of the forest. A relevance ranking weight does an exhaustive search of all the prior fire events, which makes it unique to the natural habitat of the particular geographical area, where the concept is machine learnt.

2.4 Sensor Measurement and Fire Activity

The samples of the fire events are sorted in terms of BA in hectares(ha), the frequency distribution versus BA is shown in Figure 2.2. We can model the BA in terms of a function as shown in Equation 2.1, which allows us to study the behavior of fire activities over time and predict newer events reliably.

$$BA = f(x) \quad (2.1)$$

2.4.1 Sorting by BA (ha)

The histogram shows that the BA is skewed with large number of small fires and very few large fires, making likelihood of small fires more predictable. We further classify fires into four categories that allows to select the performance of the system in terms of precision and relevance. Figure 2.2b shows the new classes without any other weather attribute that is likely to cause the events.

Figure 2.2 Histograms of empirical samples. (a) Empirical log collection and (b) four-class classification.

2.4.2 Sorting by BA with Temporal Attributes

$$V_{\text{train}}(\text{FireEvent}_{\text{Day of week}}) \leftarrow \quad (2.2)$$

$$\hat{V}(\text{FireEvent}_{\text{Day of week}})_{\text{Temporal Variable}} +$$

$$\hat{V}(\text{FireEvent}_{\text{Day of week}})_{\text{Correlated measurements}}$$

$$\text{Temporal Variables} = \text{Month of the year} + \text{Day of the week} \quad (2.3)$$

$$\text{Correlated measurement} = \text{temperature} + \text{humidity} + \text{wind} + \text{rain} \quad (2.4)$$

$$\text{Classifiers} = \{\text{accidental}; \text{small, medium, large}\} \quad (2.5)$$

2.4.3 Estimating Training Values with Sample Data

Sample datasets are based on 517 Fire Location rules from UCI forest fire repository to classify fire activity for a geographical area. The equation representing the target function of BA from the empirical data is given in Equation 2.1. In our case, the hypothesis to be maximized in terms of the temporal attributes are given in Equation 2.2 for a four-class classification, as given in Equation 2.5 and according to Equation 2.1. The assumption here is that the training set D is an unbiased representation to learn the concept c and can estimate the inputs x_i . The previously defined dependent variable **Fire Location**, which is used to estimate given the independent correlated measurements and its relation to the temporal attributed are given in Equations 2.4 through 2.6. The target concepts are present in the training samples and we like to see the influence of adding unlabelled sensor measurements to further accurately learn the concepts of the human-induced accidental small fires versus the more natural accruing types of the medium and large fires. For the sake of clarity of machine learning domain, we convert the correlated sensor data to ordinal [11,12] types, as illustrated in the following:

$$\text{temperature} = \{\text{cool}; \text{mild}; \text{hot}\} \quad (2.6)$$

$$\text{humidity} = \{\text{normal}; \text{high}\} \quad (2.7)$$

$$\text{wind} = \{\text{true}; \text{false}\} \quad (2.8)$$

The model estimation of the the target function with weights w_1, w_2 as shown allows to minimize the training error, where x_1, x_2 are temporal and correlated measurements.

$$\hat{V} = w_1 x_1 + w_2 x_2 \quad (2.9)$$

The learning algorithm needs to define the best fit for the given hypothesis and adjust the weights to minimize the error-and-misclassifications.

$$E \equiv \sum (V_{\text{train}}(\text{FireEvent}) - \hat{V}(\text{FireEvent}))^2 \quad (2.10)$$

2.4.4 Algorithm Complexity

Search space consists of all the possible patterns of the features, given our data model, $4 \times 3 \times 3 = 36$ possibilities for each rule when using attributes temperature, humidity, and wind. As there are 517 rules from the collected dataset instances, each rule can have 36 possibilities and the complete search space will have 36^{517} different possibilities. To minimize the complexity of search space, we can further cut down on the sample instances by using spatial clustering and removing any redundancies in similar features. Given the (X, Y) positions, we can cluster into groups the possible fire types into accidental small fires and others that have medium and larger BA as large fires. As measuring ambient phenomena are correlated we expect clustering would be best suited. Let us take five clusters to contain all the samples, then the search space reduces to $36^5 = 60 \times 10^6$ possible rule sets. These methods are used with preprocessing to reduce redundancies in the model and are very practical optimizations of machine learning algorithms. To judge the effectiveness of the model and the classification effectiveness, we initially rely on real-valued numeric model such as [13] to estimate the errors. In contrast to the previous approach, we use ordinal values as defined in Equations 2.8 through 2.11 to build a tree classifier and further reduce errors.

2.5 Alarm Ranking Function for Accidental Small Fires

In our example event data log such as forest fires, which may be incomplete and how do we infer knowledge from the missing datasets. Relevance factor of fire event learning concept can be defined as all fire events that are tagged in the log, as we are interested in reporting fire occurrences. The inverse concept precision is seen from the histogram plot in Figure 2.2a where there are very few large fires. When reporting on major fire events, the highest ranked samples are retrieved that has a higher corresponding rank leading to a precision learning concept close to 1.

2.5.1 Ranking Function

The design of a good ranking function needs to balance the relevance and precision of the events in a way to express a summable numerical quantity, which signifies the importance of the new sample and how reliable the prior probabilities were, as shown in Table 2.1 with false positives. As the ranking functions are evaluated for a given query, we define the query criteria for retrieving accidental small fires and large wild fires for our collection.

$$\begin{aligned} \text{Precision} &= \frac{\text{number of relevant forest fire events retrieved}}{\text{number of forest fires retrieved in query}} \\ \text{Relevance} &= \frac{\text{number of relevant forest fire events retrieved}}{\text{number of relevant forest fires classified}} \end{aligned}$$

2.5.2 F-Measure

A measure which combines precision and recall for a small dataset is F-measure and is the weighted harmonic mean of precision and relevance. The traditional F-measure or balanced F-score = $\frac{\text{Precision} \times \text{Recall}}{2 \times (\text{Precision} + \text{Recall})}$, is because recall and precision are evenly weighted. The general formula for nonnegative real β is that it is based on van Rijsbergen's effectiveness E given by

$$\text{Alarm}_{\text{rank}} = 1 - \frac{1}{\alpha^{\frac{1}{p}} + (1 - \alpha)^{\frac{1}{r}}}$$

We further look into accidental small fires, as they are very probabilistic and any conceptual link to the attributes may lead to rank the idea of precision and relevance of the collection. α — the alarm weight are calculated based on the reliability of the ground truth, higher precision weightage is given to large and medium fires compared to accidental small fires. The precision weight factor for small fires can then be evaluated using $\frac{\alpha}{1 - \alpha}$.

2.5.3 Performance of Fire Topics Classification Using Temporal Ranking

$$BA_{\text{Days}\%} < 50 \text{ ha} = \alpha + \beta \ln(NF) \quad (2.11)$$

where $\alpha = 2.895$ and $\beta = 1.265$, which is the variance of the BA data versus fire activity showing logarithmic $O(\lg(BA))$ complexity as shown in Figure 2.3a.

The broad classification of topics in the training samples collection [14] are reflected in two categories, accidental small fires and large wild fires. To evaluate the performance of the two we use weighted precision versus relevance to estimate the ranking information $F@0.5$ for the previous equation. The F-scores are calculated and weighted for high reliability by using $F@0.5$, which is twice the precision compared to its equivalent relevance scale. Reliability and precision are proportionally weighted while relevance is inversely proportional. The performance scores show

Figure 2.3 Fire activity plot and its F-score transform. (a) Fire activity vs. burnt area yields a logarithmic relation and (b) twice the precision vs. relevance.

Table 2.3 Performance of Ranking with False Alarm Rates

<i>Other Test</i>	<i>Yellow Region</i>	<i>Blue Region</i>	<i>Temporal</i>	<i>FWI</i>
Alarm	H = Hit	H = Hit	H = 311	H = 69
No alarm	M = Miss	M = Miss	M = 111	M = 3
False alarm	F = False alarm	Z = Null hypothesis	F = 8	F = 307

that query evaluation for accidental small fires is 1.7 times higher when compared to queries for large wild fires or from the same collection. The plot in Figure 2.3b shows queries for accidental and small fires for the temporal attribute days of the week, where the accidental small fire F-scores have much higher values.

2.5.4 Ranking Accidental Small Fires

Accidental small wild fires are possible all through the year, making them a viable application for automated sensor measurements. The measurements such as temperature, humidity, and wind gust are automated, while temporal attributes such as human traffic and day of the week are used to study the small fire events (Table 2.3). The peaks of the plot in Figure 2.4 suggest high alarm during weekends followed by only one high alarm day during the normal week. The weightage of the ranking suggest that small fires events are caused due to temporal attributes such as human traffic and vehicular routes more than any observed correlated sensor measurements.

2.6 Alarm Ranking Function for Large Fires

In the previous section we used BA and data from sensor networks to classify forest fires into four classes. In this section we will instead use domain knowledge to precisely predict fires by using

Figure 2.4 Fire alarm days ranked using $F@(0.5)$.

Table 2.4 FWI Classes

<i>FWI Types</i>	<i>Measure (Normalized)</i>
Low (AF)	0–8
Medium (SF)	8–13
High (MF)	13–32
Very high (LF)	$32 > FWI < 80$

FWIs {low; moderate; high; very high} as shown in Table 2.4. FWI is calculated using ISI and BUI, where ISI represents the Initial Spread Index and BUI represents the Build Up Index, which indicate fire behavior and respectively represent rate of fire spread, fuel consumption, and fire intensity. All FWI indexes are significantly correlated with the number of fires and the burned area, especially when $BA > 100$ is the area burnt by large fires. The average FWI index variation during the year is shown in Figure 2.5a and b. It increases during the month of May and peaks in August and September and starts reducing in the month of October. The following equation shows a numerical representation of BA when using FWI classes. Which is mean daily burned area per month and the mean daily number of fire events per month (NF).

$$BA_{FWI} > 50 \text{ ha} = (BUI) + (ISI)^x \quad (2.12)$$

Where ISI and BUI are calculated from the environment for a given fuel type, x is the estimated geometrical fire spread factor. The FWI index is highly correlated with the number of fires and the BA. The plot in Figure 2.5 shows the correlated region when $FWI > 33$ (very high) in the case of large fires.

2.6.1 Performance of Fire Topics Classification Using FWI Ranking

Large fire occurrences damage more than 50 ha in total amounts for majority of the BA (ha). It is a high priority to avoid large fire incidents and help forest conservation. As they are hard to detect and have a varying threshold, it is also a cause of false alarms [15] in an automated system. Plotting all the correlated FWI components that relate to fire activity, plot from Figure 2.5a shows that peak months have a gradual increase of FWI index and are also correlated with large fire incidents. The area of high correlation is shown in yellow with lowest false alarm for a given FWI threshold. The lower bound conditions for fire activity are shown in blue, which have higher false alarm due to valid area above the yellow region.

The temporal correlation for large fires using the FWI F-score is plotted in Figure 2.5b. It shows that large fires are invariant to temporal changes and perform better than small fires for a given precision and recall measure.

2.6.2 Misclassification and Cost of False Alarms

$$\begin{pmatrix} & \text{Decisionmade} \\ & \text{LargeFire} & \text{SmallFire} \\ \text{LargeFire} & 0 & 1000 \\ \text{SmallFire} & 1 & 0 \end{pmatrix}$$

Cost function matrix for misclassification

Figure 2.5 Lower bound range of FWI ≤ 33 (blue) and higher bound range FWI > 33 (yellow) show all large fires (dotted line) fall into the yellow region. (a) FWI-based classification and (b) FWI for large fires shows invariance

While testing the performance of the algorithm error rates are simulated, which allows to find the sensitiveness of the system to false positives. False positives have more significance with higher bound values such as large fires, which are very rare and hard to classify. We define a hit in terms of precision to avoid false alarms [15] and Table 2.1. From the given cost matrix we show that using FWI the alarms are very precise, while using temporal the alarms are more accurate, which includes large amounts of false positives.

2.7 Machine Learning Algorithms

Probabilistic algorithms when used with density estimation and class classification yields lowest error. This allows to provide a baseline analysis of the system attributes being used.

2.7.1 Naive Bayes

One can use Naive Bayes [16] that by design presumes the class densities a priori, which have been determined and accurate. The model calculates the class conditional probabilities of the input feature vectors. To understand the underlying skewed structure of the dataset, we further create thresholds for accidental small fires compared to medium and large fires as shown in Table 2.6. So we have the four possible values for the target variable as given in Equation 2.7.

2.7.2 User Query

To validate the model let us predict the fire activity outcome of a peak summer month from the dataset [17]. August has significant number of reported fires compared to other months. Estimating the unknown probabilities (?) using temporal features of fire events given the attribute values for the class.

$$\begin{aligned} ? &= \{Month = August; Day = Monday\} \\ &\{Temperature = Cool; Humidity = High; Wind = True\} \end{aligned}$$

The estimated class conditional densities for the independent variables temperature, humidity, and wind conditions are calculated using temporal attributes **month** for the dataset shown in Table 2.5. The datasets further are explored using two temporal variables, **month** and the **day of the week**, as shown in Tables 2.7 and 2.8. The temporal variables introduced into the dataset help gain the

Table 2.5 Posterior Probabilities for Background Weather Data for the Peak Month August

<i>Burnt Area (ha)</i>	<i>August</i>	<i>Monday</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Prior Probability</i>	<i>Predictor Variance (%)</i>
>1 ha	► 0.34	► 0.14	► 0.46	► 0.17	► 0.42	► 0.47	► 57
>1 ha <=10 ha	0.39	0.15	0.35	0.13	0.38	0.33	25.0
>10 ha <=50 ha	0.30	0.14	0.43	0.19	0.50	0.13	17.0
>50 ha	0.33	0.08	0.16	0.08	0.37	0.04	0.02

Table 2.6 Target Variable Occurrences

<i>Fire Types</i>	<i>Recorded</i>
Accidental (AF)	247
Small (SF)	175
Medium (MF)	71
Large (LF)	24

Table 2.7 Likelihood of Fires for the Month of August

<i>Fire Type</i>	<i>Month=August</i>
Accidental	0.004
Small	0.002
Medium	0.001
Large	0.00004

Table 2.8 Posteriors Probabilities for Temporal Feature Day of the Week

<i>Days</i>	<i>Accidental</i>	<i>Small</i>	<i>Medium</i>	<i>Large</i>
Mon	► 35	► 27	► 10	► 2
Tue	28	21	11	4
Wed	22	24	5	3
Thu	30	21	9	1
Fri	42	31	12	0
Sat	42	24	11	7
Sun	48	27	13	7
Total	► 247	► 175	► 71	► 24

insight of users' dependencies with fire prediction model.

$$g_i(x) = P(\omega_i | \mathbf{x}) = \frac{p(\mathbf{x} | \omega_i) P(\omega_i)}{\sum_{i=0}^{i=4} p(x | \omega_j) P(\omega_j)} \quad (2.13)$$

Substituting the corresponding highlighted values from Tables 2.5 through 2.8 in the Equation 2.13, we get the posterior probability of accidental small fire.

$$\hat{\text{fire}}_{\text{accidental}} = \frac{0.0007547}{0.003565} = 57\% \quad (2.14)$$

$$\hat{\text{fire}}_{\text{Small}} = \frac{0.000333}{0.003565} = 25\% \quad (2.15)$$

$$\hat{\text{fire}}_{\text{medium}} = \frac{0.000223}{0.003565} = 0.17\% \quad (2.16)$$

$$\hat{\text{fire}}_{\text{large}} = \frac{0.000000287}{0.003565} = 0.02\% \quad (2.17)$$

From the posterior probabilities for the month of August for the data collected in Portugal [17], the likelihood of accidental small fires are very high. From cross-validating from the known fact

that in summer the likelihood of wild fires is higher, the Bayes rule is able to classify the dataset for accidental and small fires with high accuracy. We use a simulation framework in the following sections to further prove our initial conclusion from the datasets. It is shown that the training time for Naive Bayes scales linearly in both the number of instances and number of attributes.

2.7.3 Tree Classifier

In this section, we focus on the domain rules, which are applicable to the learning system. Tree classifiers lend themselves to use ML rules [11] when searching the hypothesis by further branching on specific attributes. The design of such a classifier needs to sort the weights or entropies [16] of the attributes, which is the basis of its classification effectiveness.

ID3 is a popular tree-classifier algorithm, to implement ID3 as illustrated in Figure 2.6a and Table 2.9 with our attributes. Let (*S*) be a collection of samples, then using the tree algorithm that uses entropy to split its levels entropy is given by

$$\text{Entropy}(S) = \sum_{i=0}^{i=c} p(i) \log_2 p(i)$$

(2.18)

Let us assume a collection (*S*) has 517 samples [17] with 248, 246, 11, and 12 of *accidental*, *small*, *medium*, *large* fires, respectively. The total entropy calculated from Equation 2.18 is given by

$$\begin{aligned} \text{Entropy}(S) &= \frac{248}{517} \log_2 \frac{248}{517} + \frac{246}{517} \log_2 \frac{246}{517} + \frac{11}{517} \log_2 \frac{11}{517} \\ &\quad + \frac{12}{517} \log_2 \frac{12}{517} = 1.23 \end{aligned}$$

Figure 2.6 Weka algorithm toolkit. (a) Tree classifier and (b) Bayes network.

Table 2.9 Gain Ratio Calculation for Tree Using Entropy

Month	Temperature	Wind
Not shown	Info: 1.08	Info: 1.20
Not shown	Gain: 1.23 – 1.08 = 0.192	Gain: 1.23 – 1.08 = 0.025

2.7.4 Attribute Selection

ID3 uses a statistical property called information gain to select the best attribute. The gain measures how well the attribute separates training targeted examples, when classifying them into fire events. The measure of purity that we will use is called information and is measured in units called bits. It represents the expected amount of information that would be needed to specify whether a new instance should be classified as accidental, small, medium, or large fires, given that the example reached that node. The gain of an attribute is defined by and illustrated in Table 2.9. Using the calculated attribute for information gain, we show that *temperature* attribute is used before the *wind* attribute to split the tree after the tree root.

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{i=0}^{i=c} \frac{S_v}{|S|} \text{Entropy}(S_v) \quad (2.19)$$

$$\text{Entropy}(S_{\text{Hot}}) = \frac{9}{36} \log_2 \frac{9}{36} + \frac{23}{36} \log_2 \frac{23}{36} + \frac{3}{36} \log_2 \frac{3}{36} + \frac{1}{36} \log_2 \frac{1}{36} = 1.282$$

$$\text{Entropy}(S_{\text{Medium}}) = \frac{23}{96} \log_2 \frac{23}{96} + \frac{65}{96} \log_2 \frac{65}{96} + \frac{3}{96} \log_2 \frac{3}{96} + \frac{5}{96} \log_2 \frac{5}{96} = 1.175$$

$$\text{Entropy}(S_{\text{Cool}}) = \frac{117}{269} \log_2 \frac{117}{269} + \frac{146}{269} \log_2 \frac{146}{269} + \frac{2}{269} \log_2 \frac{2}{269} + \frac{4}{269} \log_2 \frac{4}{269} = 1.05$$

$$\text{Entropy}(\text{temp}) = \frac{43}{517} \times 1.282 + \frac{139}{517} \times 1.175 + \frac{335}{517} \times 1.05 = 1.08$$

$$\text{Gain}(S, \text{temp}) = 1.23 - 1.08 = 0.192$$

$$\text{Entropy}(S_{\text{High}}) = \frac{162}{249} \log_2 \frac{162}{249} + \frac{72}{249} \log_2 \frac{72}{249} + \frac{8}{249} \log_2 \frac{8}{249} + \frac{7}{249} \log_2 \frac{7}{249} = 1.1952$$

$$\text{Entropy}(S_{\text{Low}}) = \frac{68}{133} \log_2 \frac{68}{133} + \frac{59}{133} \log_2 \frac{59}{133} + \frac{2}{133} \log_2 \frac{2}{133} + \frac{4}{133} \log_2 \frac{4}{133} = 1.24$$

$$\text{Entropy}(\text{wind}) = \frac{361}{517} \times 1.1952 + \frac{156}{517} \times 1.24 = 1.20$$

$$\text{Gain}(S, \text{wind}) = 1.23 - 1.20 = 0.025$$

The internal tree representation for m attributes from n samples will have a complexity of $O(\lg n)$. With increasing inputs given by parameter n the height of the tree will not grow linearly as in the case of Naive Bayes. On the other hand, complexity of building a tree will be $O(mn \lg n)$.

2.8 Simulation

Open-source workbench called WEKA [4] is a useful tool to quantify and validate results, which can be duplicated. WEKA can handle numeric attributes well, so we use the same values for the weather data from the UCI [6] repository datasets. The class variable has to be a nominal one to

allow WEKA [4]; we convert all fire types to “0” or “1,” where “0” is for accidental small fire and “1” is for large fires making it a two-class classifier; the results are shown as confusion matrix in Tables 2.11 through 2.13. Naive Bayes correctly classifies accidental and small fires (209 out of 247), while the J48 tree classifier does far more, 219 out of 247 and SVM with high precision (235 out of 247).

As WEKA uses kappa [4] statistics internally for evaluating the training sets, a standard score of >60% means training set is correlated. Using J48 simulation we get 53.56% just below and when using SVM we get 0.68 above the correlated index. The comparison on results shows that J48 tree classifier does better than Naive Bayes by 25% and the corresponding SVM does 35% overall showing least bias of the three models. Therefore, using sensor network measurements accidental and small fires can be predicted with high precision using SVM classifier.

2.8.1 Simulation Analysis

WEKA attribute statistics for training set and cross validated testing and its effective correlation kappa score. Tables 2.10 through 2.12 show kappa and other comparison statistics for Naive Bayes, J48 tree, and Support Vector Machine show classifiers for small fires. The experiment is repeated using FWI, which are shown for J48 tree classifier in Table 2.14.

2.8.2 Error Analysis

Equation 2.11 specifies the regression model error and its following confusion matrix from the simulation scores are shown in Tables 2.13 through 2.16, upper bound of small fire (AF+SF) has over 90% precision in SVM, 80% accuracy for J48-Tree, and 61% overall for Naive Bayes. The corresponding baseline performances including all fires categories is 82% for SVM, 72.1% for J48-Tree, and Naive Bayes is 51.64%, which is due to bias toward small fires and only SVM by design is the least biased (Tables 2.17 through 2.19).

When FWI classification [17] is as given in Equation 2.12 for large fires prediction, it is more precise with better precision as shown in the confusion matrix Table 2.20. The percentage of correctly classified is >95%, making it reliable with few false alarms.

Table 2.10 Evaluation on Training Set for Naive Bayes

WEKA Stats	Results	Summary (%)
Correctly classified instances	267	51.64
Incorrectly classified instances	250	48.35
Kappa statistic	0.1371	
Mean absolute error	0.3022	
Root mean squared error	0.3902	
Relative absolute error	94.86%	
Root relative squared error	97.84%	
Total number of instances	517	

Table 2.11 Evaluation on Training Set for J48 Tree Classifier

<i>WEKA Stats</i>	<i>Results</i>	<i>Summary (%)</i>
Correctly classified instances	373	72.14
Incorrectly classified instances	144	27.85
Kappa statistic	0.5356	
Mean absolute error	0.1938	
Root mean squared error	0.3113	
Relative absolute error	60.83%	
Root relative squared error	78.04%	
Total number of instances	517	

Table 2.12 Evaluation on Training Set for SVM Linear Classifier

<i>WEKA Stats</i>	<i>Results</i>	<i>Summary (%)</i>
Correctly classified instances	421	81.43
Incorrectly classified instances	96	18.56
Kappa statistic	0.6893	
Mean absolute error	0.0928	
Root mean squared error	0.3047	
Relative absolute error	29.14%	
Root relative squared error	76.40%	
Total number of instances	517	

Table 2.13 Confusion Matrix for Naive Bayes Using Training Set

	<i>LF</i>	<i>MF</i>	<i>SF</i>	<i>AF</i>
LF	0	1	7	16
MF	0	5	12	54
SF	0	7	53	115
AF	0	0	38	209

Table 2.14 Confusion Matrix on Training Set for J48 Tree Classifier

	<i>LF</i>	<i>MF</i>	<i>SF</i>	<i>AF</i>
LF	7	0	7	10
MF	0	29	15	27
SF	1	7	118	49
AF	0	5	23	219

Table 2.15 Confusion Matrix on Testing Set for J48 Tree Classifier

	<i>LF</i>	<i>MF</i>	<i>SF</i>	<i>AF</i>
LF	0	2	8	14
MF	2	7	2	40
SF	3	12	71	89
AF	4	15	59	168

Table 2.16 Confusion Matrix on Training Set for SVM Linear Classifier

	<i>LF</i>	<i>MF</i>	<i>SF</i>	<i>AF</i>
LF	7	0	5	12
MF	0	31	15	25
SF	0	1	148	26
AF	0	0	12	235

Table 2.17 Confusion Matrix on Testing Set Using SVM

	<i>LF</i>	<i>MF</i>	<i>SF</i>	<i>AF</i>
LF	0	0	7	17
MF	0	1	17	53
SF	0	3	42	130
AF	0	4	51	192

Table 2.18 Confusion Matrix on Training Set Using Bayes Network

	<i>LF</i>	<i>MF</i>	<i>SF</i>	<i>AF</i>
LF	0	0	5	19
MF	0	4	5	62
SF	0	4	30	141
AF	0	0	23	224

Table 2.19 Confusion Matrix on Testing Set Using Bayes Network

	<i>LF</i>	<i>MF</i>	<i>SF</i>	<i>AF</i>
LF	0	0	5	19
MF	0	2	8	61
SF	0	4	23	148
AF	0	2	28	217

Table 2.20 Confusion Matrix on Training Set for *FWI* > 32

	<i>Very High</i>	<i>High</i>
Very high	371	0
High	17	0

2.9 Correlation of Attributes

From statistical point of view, if the attributes have similar values then it creates high bias creating what is called over-fitting error during learning. In our case, **temperature** and **humidly** may have similar values and need to be avoided and substituted with a suitable attribute. To pre-process and analyze, we use all the available sensor measurements in the dataset and WEKA provides the attribute selection as illustrated in Table 2.21.

We use the attribute selection wizard of WEKA to find out the best match. The analysis shows that the Month(100%), Day(10%), and Wind(0%) are highly dependent on the precision. In a two-class classification, the quantitative data are biased toward small fires and SVM does better due to better generalization (Table 2.22). In the qualitative analysis that is based on the frequency of attributes, WEKA picks Month, which is a temporal type. The F-score of small fires is higher, when using temporal attributes as shown in Figure 2.3b, which is also true for WEKA predictions.

**Table 2.21 Attribute Selection
10-Fold Cross-Validation (Stratified)**

<i>Number of Folds (%)</i>	<i>No.</i>	<i>Attribute</i>
10(100%)	1	Month
1(10%)	2	Day
0(0%)	3	Temp
0(0%)	4	RH
0(0%)	5	Wind

**Table 2.22 Evaluation *FWI* > 32 on Training Set for J48
Tree Classifier**

<i>WEKA Stats</i>	<i>Results</i>	<i>Summary (%)</i>
Correctly classified instances	371	95.61
Incorrectly classified instances	17	4.38
Kappa statistic	463.64	
Mean absolute error	0.0838	
Root mean squared error	0.2047	
Relative absolute error	97.5159%	
Root relative squared error	99.9935%	
Total number of instances	388	

2.10 Better Model for Weather Data

In the error analysis, we show that the true error performance depends on the learning algorithms [1] and how the weights are learnt. As all the data mining algorithms assume samples to be i.i.d., it cannot perform well with highly correlated weather data. In the domain of weather data where features tend to be highly correlated, careful feature selection and a better model are needed to address the shortcomings of earlier models. Bayes network allows to define a better model to define classes and events that are dependent and independent of each other. This model allows to distinguish the overlapping statics using correlated and similar features having the same range and values. The basic Bayes net performance is good while testing with 88% precision as shown in Figure 2.7a and b and has similar accuracy as other models proving its discriminative power for the underlying data.

2.11 Future Work

As machine learning is getting popular, its scalability and study of the family of algorithms need to be studied. Development of a large-scale high-speed system is needed to perform *k*-nearest neighbor

Figure 2.7 Plot of F-score for each model and its performance with weather training data. (a) ML algorithm performance and (b) Bayes network.

searches. The availability of such a system is expected to allow more flexible modeling approaches and much more rapid model turnaround for exploratory analysis. We like to explore Hadoop ecosystem to develop machine learning algorithms, data reference set of 20 million profiles to sort the best 1000 records in less than 3 h. Some initial work using the Mahout machine learning framework is proposed to port the described algorithms in a distributed framework running Hadoop clusters.

2.12 Summary

We use a query log approach of search engines and standard statistical ranking measure to do a baseline analysis and use data from inexpensive sensors to validate against probabilistic ML algorithms. From Table 2.23, we show that precision- $\oplus$ and accuracy- $\ominus$ as denoted, the F-measures matches

Table 2.23 F-Measure Performance for All Tests Compared with WEKA

<i>Experiments</i>	<i>Model Performance</i>	<i>F@0.5 Measure</i>		<i>Confusion Matrix</i>	
		<i>Small Fire</i>	<i>Large Fire</i>	<i>Small Fire</i>	<i>Large Fire</i>
Ranking—Temporal	Bias	0.22 $\oplus$	0.17	—	—
Ranking—FWI	Bias	0.1	0.9 $\ominus$	—	—
Weka(NB)—Temporal	Generative	0.6 $\oplus$	0.1	0.6	0.3
Weka(Bayes Net)—Temporal	Better model	0.88 $\oplus$	—	0.88	0.0
Weka(J48)—Temporal	Decision tree	0.7 $\oplus$	0.5	0.75	0.3
Weka SVM—Temporal	Generalized	0.87 $\oplus$	0.7	0.9 $\ominus$	0.3
Weka(J48)—FWI	Invariant	—	0.95 $\ominus$	0.1	0.9 $\ominus$

the expected WEKA simulation statistics. FWI is able to boost the weak performance of the raw data from sensors, which are typically hard to calibrate. In the qualitative analysis of small fires, the performance of a generalized classifier such as SVM is preferred. Similarly, the qualitative performance for large fires is done by careful attribute selection and we show that an invariant attribute selection such as ($FWI > 50$) yields high classification precision. The use of a better model like Bayes network in the case of weather data helps to increase the performance.

Acknowledgments

The authors thank Shailesh Kumar of Google, Labs, India for suggesting the machine learning framework WEKA and related topics in statistical methods in data mining. The authors would also like to thank Srinivasa Srivathsan for working on the Big Data Analytics and Hadoop framework in context to Machine Learning. Vasanth Iyer appreciates the support from advisors Dr. S. S. Iyengar, Dr. Rama Murthy, and Dr. Rawat in this research effort, which was funded under the FIU, Miami Research Grant.

References

1. Tom M. Mitchell. *Machine Learning*. McGraw-Hill Publications, 1997.
2. Forest fire research and wildland fire safety: *Proceedings of IV International Conference on Forest Fire Research 2002 Wildland Fire Safety Summit*, Luso, Coimbra, Portugal, 18–23 November 2002, pp. 83.
3. Timothy G. Armstrong, A. Moffat, W. Webber, and J. Zobel. EvaluatIR: An online tool for evaluating and comparing IR systems. Computer Science and Software Engineering, University of Melbourne, 2009.
4. Paulo, C. and A. Morais. *A Data Mining Approach to Predict Forest Fires using Meteorological Data*. Department of Information Systems-R&D Algoritmi Centre, University of Minho, Guimaraes, Portugal.
5. Ian H. Witten and E. Frank. *Data Mining, Practical Machine Learning*. Elsevier, 2005.
6. WEKA Machine learning software. <http://www.cs.waikato.ac.nz/~ml/weka> [Accessed May 15, 2011].
7. Mark D. Smucker, J. Allan, and B. Carterette. *A Comparison of Statistical Significance Tests for Information Retrieval Evaluation*. Department of Computer Science, University of Massachusetts, Amherst, MA.
8. Raymond Wan and A. Moffat. Interactive phrase browsing within compressed text. Computer Science and Software Engineering, University of Melbourne, 2001.
9. H. Harb and L. Chen. A Query by example music retrieval algorithm. Maths-Info Department, Ecole Centrale de Lyon. Ecully, France, 2001.
10. Brian Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search, in *Proceedings of the 12th International Conference on Computer Vision (ICCV)*, 2009.
11. Vasanth, I., S.S. Iyengar, G. Rama Murthy, N. Parameswaran, D. Singh, and M.B. Srinivas. Effects of channel SNR in mobile cognitive radios and coexisting deployment of cognitive wireless sensor networks. *IEEE 29th International, Performance Computing and Communications Conference (IPCCC)*, pp. 294–300, Albuquerque, NM, 2010.
12. Bhaskar Krishnamachari and S.S. Iyengar. Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks, *IEEE Trans. Comput.* 53, 3, 2004, pp. 241–250.
13. Vasanth, I., S.S. Iyengar, G. Rama Murthy, and M.B. Srinivas. Modeling unreliable data and sensors: Analyzing false alarm rates using training weighted ranking. ISBN: 978-1-880843-83-3, in *The Proceedings of International Society for Computers and Their Applications*, Honolulu, HI, 2011.

14. David Hawking and S. Robertson. On collection size and retrieval effectiveness. CSIRO Mathematical and Information Sciences, Canberra, ACT, Australia. [Accessed February 22, 2011].
15. Richard R. Brook and S.S. Iyengar. *Robust Distributed Computing and Sensing Algorithm*, ACM, 1996.
16. Frank, A. and A. Asuncion. UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science, 2010. [Accessed May 20, 2011].
17. Vasanth, I., S.S. Iyengar, G. Rama Murthy, and M.B. Srinivas. Machine learning and dataming algorithms for predicting accidental forest fires. SENSORCOMM 2011, in *The Fifth International Conference on Sensor Technologies and Applications*, August 21–27, 2011, French Riviera, France, 2011.