

University of Bristol
Computer Science Department
Merchant Venturers Building,
Woodland Road
BS8 1UB, Bristol, UK



Review

“Weather Talk”, extracting weather information by text mining

by

Student: Vasileios Lampos

Supervisor: Professor Nello Cristianini

Marker1: Dr James A. R. Marshall

Marker2: Dr Walterio W. Mayol-Cuevas

COMSM2100

“Project Specification and Design, Advanced”

Faculty of Engineering
Department of Computer Science

May 2008

Contents

1	Introduction	1
1.1	“Weather talk”, the idea	1
1.2	“Weather talk”, the description	2
1.3	Report outline	3
2	Collecting web data	4
2.1	Data and Web Mining	4
2.2	Crawling the web	4
2.3	Alternatives on Crawling	5
3	Classifying web data	7
3.1	Weather states	7
3.2	Using a weather ontology	8
3.3	Deciding a weather state by applying Naïve Bayesian classification	9
3.3.1	Bayes’ theorem	9
3.3.2	Applying Naïve Bayesian classification to “Weather Talk”	9
3.3.3	Naïve Bayesian Classification, an example	12
3.4	Data Fusion	13
4	Hidden Markov Model application	14
4.1	Markov chains and Hidden Markov Model	14
4.1.1	Applying Hidden Markov Model, an example	15
4.1.2	Disadvantages of the scheme	17
4.2	Changing scheme to a pair Hidden Markov Model	17
4.2.1	Pair HMM, a quick example	18
5	Evaluation and further extensions	20
5.1	Evaluation procedure in “Weather talk”	20
5.2	Additional features	21
5.3	Further challenges	21
	Bibliography	23

Chapter 1

Introduction

1.1 “Weather talk”, the idea

It is always about the right combination, everywhere. In life, in mathematics, in thoughts the right combination of “elements” will make the difference. “Weather talk”, in general, is a project that tries to create the right combinations between technically unrelated information in order to extract useful conclusions. It is an idea of Professor Nello Cristianini; an idea with an unconventional title but conventional applications.

“Weather talk” takes advantage of the vast amount of information that lies on the World Wide Web (hereinafter web). The goal of this project is to use statistical analysis on text documents available online in order to extract information about the weather on a specific location. Documents can include blogs, newsgroups and news articles but not officially weather related web sites or pages. These documents form “Weather talk”, the input of our project.

Why has weather been chosen instead of something more challenging? Weather may not seem an interesting subject to investigate as the process of deriving the weather state of a location, one day after, without watching the weather observations of official weather repositories does not offer any extra information. The answer is that the choice of weather is motivated by the easy availability of ground truth.

As an extension to the whole process, we may construct a graphical representation of our predictions on a map. More importantly, if the “Weather talk” model behaves correctly we will experiment by applying it to other types of information, *e.g.* marketing, products, opinions.

1.2 “Weather talk”, the description

What is the general plan of this project? The project will be separated in three serialized steps. In every step, we will get a final result and every next step will use the achievements of the previous one.

- **Step 1 - Infer weather state from observable web data.**

In this step we will infer the major weather state of a day using observable¹ web data (Figure 1.1). Data collection methods as well as classification of the collected data should be applied.

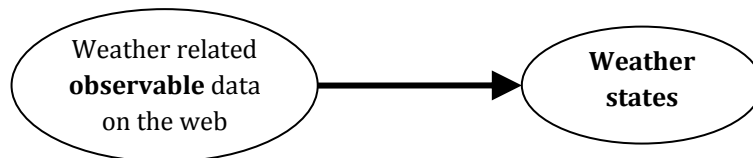


FIGURE 1.1: “Weather talk” Step 1. Infer weather states from observable web data.

- **Step 2 - Infer weather state using observable web data fusion.**

In order to improve our prediction, we will use one or more contexts that include information which is related with or maps to weather states or ‘consequences’ such as topics referring to traffic, sport events, and agriculture (Figure 1.2).

- **Step 3 - Improve result using yesterday’s weather.**

This step concludes our system and introduces the need of memory. It combines all the previous steps with the addition of using previous day’s weather state as a coefficient in the model (Figure 1.3). A Hidden Markov Model will be applied for carrying out this step.

- **Evaluation Step.**

In this step we compare our results to official weather data. Constructing the weather states of our system will be based on the format of ground truth data. Therefore, it is important to decide which format of official weather data is going to be used before starting building the weather states of the system. There are more complicated solutions on that, such as mapping both inferred weather states with official ones to a common weather state system. This step will be carried out after each one of the previous steps, for evaluation purposes.

¹ throughout this report “observable” term will be used to indicate data that derive from unofficial resources and can be observed by a user or an application.

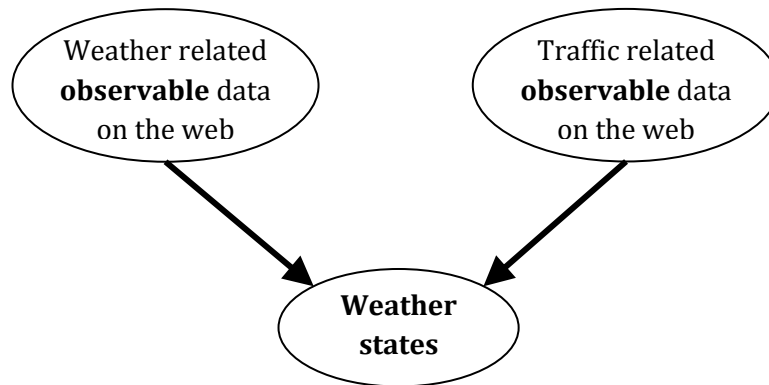


FIGURE 1.2: “Weather talk” Step 2. Infer weather states from observable web data together with traffic data (data fusion).

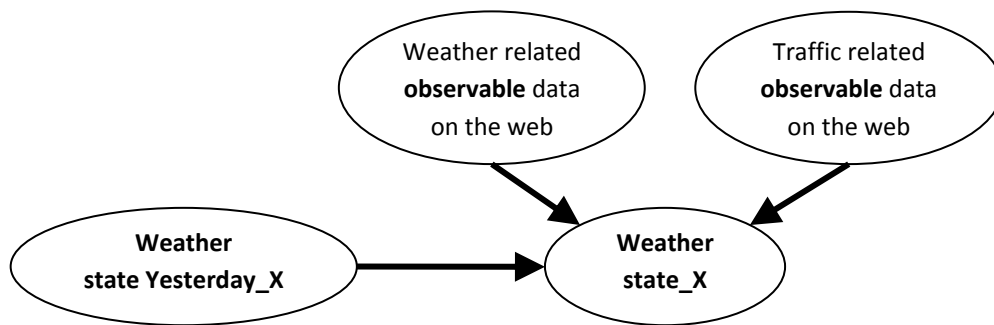


FIGURE 1.3: “Weather talk” Step 3. Improve results using yesterday’s weather.

- **Designing the weather map.**

Instead of having a static text list with city names and weather states, it is in our goals to project our results using a map interface, such as Google Maps API.² However, our main goal is getting the right results, not projecting them.

1.3 Report outline

The content of this report is distributed as follows; in **Chapter 2**, we refer to methods for data retrieval as well as in their possible adaptations to this project. In **Chapter 3**, we provide an analysis of the technique for classifying the collected web information and we introduce data fusion. In **Chapter 4**, we present a Hidden Markov Model which is used

² Google Maps API, <http://code.google.com/apis/maps/>.

to decide the weather state of a day by combining all the previous methods together with the inferred weather state of the previous day. In **Chapter 5**, we describe the methods for evaluating and displaying our results and we discuss about possible extensions, and different applications of this project.

Chapter 2

Collecting web data

In this chapter we make a brief reference to web mining techniques applicable to “Weather talk”. Furthermore, we describe alternatives that can be used nowadays and offer the same level of results.

2.1 Data and Web Mining

Data mining, also known as “Knowledge Discovery in Databases”, is the process of extracting unknown, non obvious or hidden, but conditionally useful information from data stored in a database format using nontrivial techniques [FPSM92]. Web mining is based on the same concept and in most cases uses the same algorithms for retrieving information but it is applied on web data. These data may have been derived from a server’s database but the web mining system cannot have full access to that server. Web mining applications can access information located on the web either it is visible or just unprotected, meaning that the data collection process is a substantial part of the whole mining procedure.

2.2 Crawling the web

The data collection process in Web mining is performed, most of the times, by programs, known as crawlers or spiders or robots. As an example of a very simple and limited crawler, we could have a program that after is given a set of Uniform Resource Locators (hereinafter URLs) as an input, it starts to follow links within the web pages while storing the pages it passes by into a repository.

For this project we have investigated two different crawler types. A “**focused**” crawler follows the proposal of Chakrabarti *et al.* [CvdBD99] and applies goal directed crawling. A document that withholds all the topics of interest is given as an input to the crawler.

As a result, the crawler does not search only for specific keywords but for more general contexts that might describe them. In our project the input could have the form of a weather ontology¹ as it was proposed in [EM03]. A classifier is used in order to fetch pages that are more close to the predefined interest. It may use two possible strategies:

- *Conventional strategy.* None of the fetched pages or paths are withdrawn during crawling. All the pages are sorted based on their probabilities to contain information of interest. The crawling continues by fetching the pages with higher probabilities first.
- *Strict strategy.* For each page the classifier finds the most probable context. If none of the pages which belong to the search path of the current page (ancestors of the current page) is of the requested context, then this path is pruned.

An extension of this crawler is described in [CPS02]. It uses a second classifier, named as “apprentice”, which assigns priorities to pages that have not been crawled yet by analyzing information which is related to them, such as inspecting items of interest inside the Document Object Model (hereinafter DOM) tree of the pages that link to them.

An “intelligent crawling” technique is presented in [AAGY01]. The crawler is given as input a set of “predicates”, which they may be words of interest in the linked URLs or the DOM tree (content) of the pages. Before visiting a page, it calculates the probability that this page is interesting by using the “predicates”. Suppose that we are looking for pages about the weather. We have set only one predicate, the word *weather* and we are looking only whether this word is part of the URL of an unseen page. From our previous results we know $\Pr(\text{subject} = \textit{weather} \cap \text{token in URL} = \textit{weather})$, the probability that a page talks about the weather and includes the word *weather* in its URL. We also know, $\Pr(\text{subject} = \textit{weather})$ and $\Pr(\text{token in the URL} = \textit{weather})$. Then, we may calculate the fraction d ,

$$d = \frac{\Pr(\text{subject} = \textit{weather} \cap \text{token in URL} = \textit{weather})}{\Pr(\text{subject} = \textit{weather}) \Pr(\text{token in the URL} = \textit{weather})}. \quad (2.1)$$

The crawler decides that the unseen page is not interesting if $d \leq 1$, and interesting otherwise. An advantage of this method is that the crawler does not need a seed of URLs in order to start. On the other hand, one should be very careful when deciding for the “predicates”.

2.3 Alternatives on Crawling

Including a crawler in “Weather talk” will result to search engines independence and will give a more flexible and customizable character to the project. However, implementing a smart,

¹ Ontology application is discussed in Chapter 3.

large-scale crawler for a specific purpose (*e.g.* effective searching of weather information) may be a project of its own. For the reason that during this project, we would like to focus on investigating and implementing the core theory of our system, instead of customizing or implementing from scratch a crawler, we use more conventional, alternative methods.²

Alternative methods will include the use of modern search engines' services, such as retrieving dynamically an RSS feed³ which includes all the articles of interest based on the tokens of a search.

² We are more interested in the theoretical investigation and implementation of the concepts presented in Chapters 3 and 4. Whether a crawler will be implemented or adapted to the project's needs, will be decided by the whole progress of the project.

³ RSS stands for RDF Site Summary, or Rich Site Summary, or Really Simple Syndication depending on its author and version, [http://en.wikipedia.org/wiki/RSS_\(file_format\)](http://en.wikipedia.org/wiki/RSS_(file_format)).

Chapter 3

Classifying web data

In this chapter, we describe methods used for classifying the collected information. The representation of the term “weather state” in our project and its connection with a weather state ontology are explained. In the end, we describe how data fusion is applied in order to conduct more accurate results.

3.1 Weather states

Our initial purpose is to decide and assign a weather state to each target city for each day. A weather state is a general description of the weather of a day indicating the major weather condition and it may consist of one or multiple words that form a phrase, *e.g. sunny, rain, partly cloudy, light rain shower*. Different weather states may be hundreds with small variations with each other. We have to limit this for two significant reasons:

- the weather states derived from the official resources are limited. We will need a mapping between similar, derived weather states to a more general one, used by the official sources.
- making conclusions with that level of accuracy would not be beneficial for the total outcome or will lead to incorrect assumptions. Suppose that observable web information maps to the weather state of *fair weather* with a percentage of 20%, to *bad weather* with 15%, to *cold weather* with 20%, to *wet weather* with 20%, and to *sunny weather* with 25%. Based on these probabilities, we decide the weather state with the highest one, *i.e. sunny weather*. It is obvious, that the other derived states map to the more general *fair weather* state, as bad, cold, and wet are adjectives which describe this condition. The fact that *sunny weather* has been mentioned in the minority of resources, points to the conclusion that it was not the major weather state of the day.

A solution to this problem might come with the use of a weather ontology in a customized manner. The main weather states will be defined from the official weather observations which form the ground truth mechanism. From a detailed weather ontology, mappings between multiple weather states and their contextual descriptions will be extracted. We will end up with a basic set of weather states and their mappings to weather related words or phrases derived from the weather ontology.

3.2 Using a weather ontology

In [Gru93] ontology is defined as a formal but restricted specification of a concept which describes predicted relationships between its elements. Applications of ontologies vary; in this project we will use a weather ontology in the process of transforming unordered and heterogeneous data to usable information. Scenarios describing this type of use for ontologies (“common access to information”) are presented in [JU99].

Building an ontology about a general subject is not a trivial task. At the same time, a very general ontology about the weather will introduce a higher level of complexity in a part of our problem that we want to be as simple and accurate as possible. Our final decision is to research existing ontologies about the weather¹ and to adapt them to a weather ontology suitable for “Weather talk”. The general idea is that the ontology will map several elements (words or phrases) to a weather state (Figure 3.1). This is a way of dealing with the restrictions that ground truth weather states introduce. Editing or building a new ontology may be carried out with Protégé [NSD⁺01], an ontology editor and knowledge acquisition tool.²

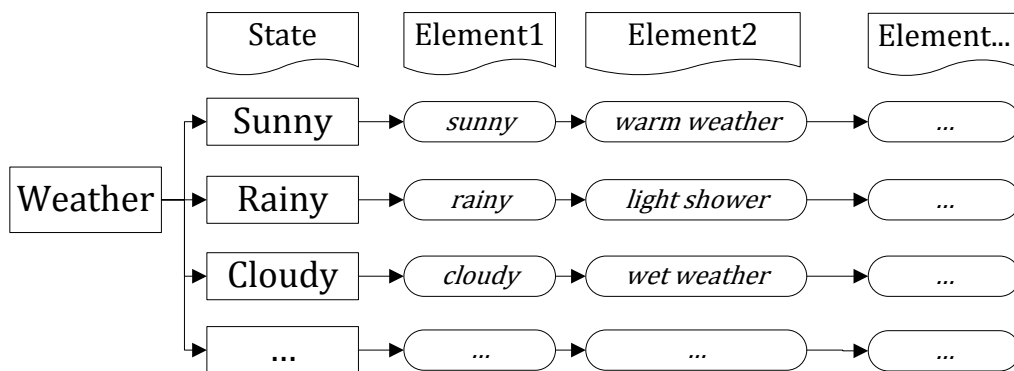


FIGURE 3.1: A partial graphical representation of a weather ontology.

¹ A weather ontology was constructed for a project at the Computer Science Department of Aberdeen, Scotland, UK, <http://www.csd.abdn.ac.uk/research/AgentCities/WeatherAgent/index.php>.

² Protégé, <http://protege.stanford.edu/>.

Using WordNet³ might help in building a more complete and consistent weather ontology. According to [MBF⁺90], WordNet is a different, much more interesting approach of an English language dictionary. It represents general concepts as a collection of several synonym sets (called “synsets”). Words are connected to each other, building a networked scheme, based on similarities in meaning or in the general concept in which they are attached. This networked structure of WordNet could also be useful in improving the search queries mentioned in the previous Chapter.

3.3 Deciding a weather state by applying Naïve Bayesian classification

3.3.1 Bayes’ theorem

In general, Thomas Bayes’ theorem provides a way to calculate the *a posteriori* probability $\Pr(X|Y)$,⁴ from the *a priori* probability $\Pr(X)$, using $\Pr(Y|X)$ and $\Pr(Y)$.

$$\Pr(X|Y) = \frac{\Pr(Y|X) \Pr(X)}{\Pr(Y)}. \quad (3.1)$$

Using (3.1) we are able to calculate the exact probability of an event X , given a set of data Y . In case we want to decide the most probable event based on the same data set Y , then the normalizing denominator $\Pr(Y)$ is not needed as it is the same for all the events X . Our decision is the *maximum a posteriori* hypothesis s and is defined as

$$s = \operatorname{argmax}_{x_i \in X} \Pr(x_i|Y) = \operatorname{argmax}_{x_i \in X} \Pr(Y|x_i) \Pr(x_i). \quad (3.2)$$

3.3.2 Applying Naïve Bayesian classification to “Weather Talk”

This section is based on the relevant chapters in Mitchell’s [Mit97] and Liu’s [Liu07a] books. Naïve Bayesian classification, which will be used for the purposes of this project, is based on equations (3.1) and (3.2). It forms a popular classification algorithm used mainly for supervised learning.⁵

As we have mentioned before, we want to transform observable web data to weather states. Let WS be the class of weather states with values $\langle ws_1, \dots, ws_{|WS|} \rangle$, where $|WS|$ is the

³ WordNet, <http://wordnet.princeton.edu/>.

⁴ $\Pr(X|Y)$ denotes the probability of the event X based on the event Y , or closer to our subject the probability of a weather state X based on a data set of observations Y .

⁵ this type of learning is called *supervised* because the input data (or more formally *training data*) are labeled with predefined classes.

number of predefined different states.

$$WS = \langle ws_1, \dots, ws_{|WS|} \rangle. \quad (3.3)$$

Let OD be the data set of weather information collected from observable web data with discrete attributes $\langle O_1, \dots, O_{|O|} \rangle$, where $|O|$ is the number of attributes we use.

$$OD = \langle O_1, \dots, O_{|O|} \rangle. \quad (3.4)$$

For clarity, we mention that this data set does not contain any official weather information. Let o be an instance of OD , $o = \langle O_1 = o_1, \dots, O_{|O|} = o_{|O|} \rangle$. In other words, o is an observation made of collected web unofficial weather data. The number of attributes $|O|$ is the number of parsed words or phrases of interest (*elements*).⁶ Each attribute holds an element of interest that is part of the weather ontology. With high probability there would be attributes which will contain the same element (we will refer to them as *duplicate* attributes). Our model ensures that duplicate attributes will not be ignored as they should strengthen the probability of one weather state.

The purpose of our model is to calculate the probability of a weather state $ws_i \in WS$ based on the observation o . This probability is $\Pr(WS = ws_i|o)$ and using Bayes' theorem can be expressed as

$$\begin{aligned} \Pr(WS = ws_i|o) &= \frac{\Pr(o|WS = ws_i) \Pr(WS = ws_i)}{\Pr(o)} = \\ &= \frac{\Pr(o|WS = ws_i) \Pr(WS = ws_i)}{\sum_{k=1}^{|WS|} \Pr(o|WS = ws_k) \Pr(WS = ws_k)}. \end{aligned} \quad (3.5)$$

We will make a decision about the weather state based on the value of (3.5), picking the state with the greatest probability (*maximum a posteriori probability*). Therefore, as in (3.2) only the numerator will be calculated as the denominator of (3.5) is the same for all the values of the weather state class.

We are able to simplify the formulas above by assuming that for a known weather state, $WS = ws_i$, the weather data set attributes are conditionally independent one another. That means that each element, which maps to a weather state, is not related with the other words included in our data set. Of course, this is not true, but from [Zha04] we assume that even when there are strong dependencies between the words in the text, naive Bayes classification is optimal. The conditional independency can be expressed as

$$\begin{aligned} \Pr(O_j = o_j|WS = ws_i) = \\ \Pr(O_j = o_j|O_1 = o_1, \dots, O_{j-1} = o_{j-1}, O_{j+1} = o_{j+1}, \dots, O_{|O|} = o_{|O|}, WS = ws_i). \end{aligned} \quad (3.6)$$

⁶ in order to refer to words or phrases of interest, the term *element* is going to be used from this moment onwards.

As a result

$$\Pr(o|WS = ws_i) = \prod_{j=1}^{|O|} \Pr(O_j = o_j|WS = ws_i). \quad (3.7)$$

From (3.5) and (3.7) we have that

$$\Pr(WS = ws_i|o) = \frac{\Pr(WS = ws_i) \prod_{j=1}^{|O|} \Pr(O_j = o_j|WS = ws_i)}{\sum_{k=1}^{|WS|} \Pr(WS = ws_k) \prod_{j=1}^{|O|} \Pr(O_j = o_j|WS = ws_k)}. \quad (3.8)$$

Since we only need the most probable weather state for a city, we will not take into consideration the denominator of (3.8), as its value is the same for every instance of the weather state class. Consequently, our decision d is

$$d = \operatorname{argmax}_{ws_i \in WS} \Pr(WS = ws_i) \prod_{j=1}^{|O|} \Pr(O_j = o_j|WS = ws_i). \quad (3.9)$$

the weather state whose probability's enumerator is the greatest.

We will make the following assumptions:

- All the weather states have the same probability to occur,

$$\Pr(WS = ws_1) = \dots = \Pr(WS = ws_{|WS|}) = \frac{1}{|WS|}. \quad (3.10)$$

For a particular location on a particular day, it is obvious that is assumption is not correct as there are areas with “restricted” weather behaviour. On the other hand, this is another type of input, the biggest part of which is based on weather forecasting statistical techniques. It is clear that this is not the scientific field that this project targets.

- All the elements that map to a weather state, have the same probability to map to this weather state with any other element that maps to another weather state. Formally,

$$\Pr(O_j = o_j|WS = ws_k) = \Pr(O_i = o_i|WS = ws_u) = \frac{1}{|O|}, \quad (3.11)$$

for $\forall \{i, j\} \in [1, |O|]$ and for $\forall \{k, u\} \in [1, |WS|]$,
 where $O \rightarrow WS$.

Similarly to the previous assumption, there are elements that should have a stronger mapping to a weather state. Taking this into consideration will require the creation of a dynamic weather ontology, in which the elements that map to a weather state will have weights.

TABLE 3.1: A small weather ontology mapped on a table.

Weather State	Element1	Element2	Element3	Element4
sunny	<i>sunny</i>	<i>warm</i>	<i>lots of sun</i>	<i>clear</i>
rainy	<i>rainy</i>	<i>light rain</i>	<i>cold</i>	-
cloudy	<i>cloudy</i>	<i>cold</i>	<i>fair weather</i>	-

It is trivial to notice that there would exist probabilities with zero values, when a mapping between a parsed element and a weather state does not occur. To avoid losing important information, we apply the Lidstone’s law of succession [Lid20] by adding a $\lambda = \frac{1}{|O|}$ to both the numerator and denominator of the probabilities in (3.11). This process is called *smoothing*. As a result, instead of zero probabilities, the minimum probability is now equal to $\frac{1}{|O|^2+1}$.

3.3.3 Naïve Bayesian Classification, an example

In this section, we provide a minimal example of how exactly Bayesian inversion will be applied to “Weather talk”. Suppose that we work with the very small ontology which is mapped on the table 3.1. Every weather state is described with some elements. These elements we are trying to spot in the observable web data.

Suppose again, that the data we observe (for a specific location and day) include the following set of terms:

$$\langle \text{cold, warm, clear, cloudy, cloudy} \rangle .$$

We assume that each weather state is equally probable. As a result,

$$\Pr(W S = \text{sunny}) = \Pr(W S = \text{rainy}) = \Pr(W S = \text{cloudy}) = 1/3.$$

Each element should map equally to a weather state. We want that to happen in order not to assume higher probabilities for elements which belong to a weather state that does not have the biggest number of different elements (in our example *rainy* and *cloudy* have an element less than *sunny*). Consequently, all the probabilities of the type $\Pr(O = o_i | W S = w s_j)$ are equal to $\frac{1}{|O|}$, where $|O|$ is the biggest allowed number of elements in our model. We have assumed, that

$$\Pr(O = \text{sunny} | W S = \text{sunny}) = \dots = \Pr(O = \text{cloudy} | W S = \text{fairweather}) = 0.25.$$

As mentioned in the previous section, we will apply Lidstone’s law of succession with $\lambda = \frac{1}{|O|} = 0.25$. The probabilities will be equal now to $\frac{1+\lambda}{|O|+\lambda}$,

$$\Pr(O = \text{sunny} | W S = \text{sunny}) = \dots = \Pr(O = \text{cloudy} | W S = \text{fairweather}) = \frac{5}{17}.$$

We start to investigate weather states by calculating the enumerators of their probabilities since all the denominators are equal, using (3.9):

Sunny weather state:

$$\begin{aligned} & \Pr(W S = \textit{sunny}) \Pr(O = \textit{cold} | W S = \textit{sunny}) \Pr(O = \textit{warm} | W S = \textit{sunny}) \cdot \dots \\ & \dots \cdot \Pr(O = \textit{clear} | W S = \textit{sunny}) \Pr(O = \textit{cloudy} | W S = \textit{sunny}) \cdot \dots \\ & \dots \cdot \Pr(O = \textit{cloudy} | W S = \textit{sunny}) = 5.8691 \cdot 10^{-6}. \end{aligned}$$

Rainy weather state:

$$\begin{aligned} & \Pr(W S = \textit{rainy}) \Pr(O = \textit{cold} | W S = \textit{rainy}) \Pr(O = \textit{warm} | W S = \textit{rainy}) \cdot \dots \\ & \dots \cdot \Pr(O = \textit{clear} | W S = \textit{rainy}) \Pr(O = \textit{cloudy} | W S = \textit{rainy}) \cdot \dots \\ & \dots \cdot \Pr(O = \textit{cloudy} | W S = \textit{rainy}) = 1.1738 \cdot 10^{-6}. \end{aligned}$$

Cloudy weather state:

$$\begin{aligned} & \Pr(W S = \textit{cloudy}) \Pr(O = \textit{cold} | W S = \textit{cloudy}) \Pr(O = \textit{warm} | W S = \textit{cloudy}) \cdot \dots \\ & \dots \cdot \Pr(O = \textit{clear} | W S = \textit{cloudy}) \Pr(O = \textit{cloudy} | W S = \textit{cloudy}) \cdot \dots \\ & \dots \cdot \Pr(O = \textit{cloudy} | W S = \textit{cloudy}) = 29.3457 \cdot 10^{-6}. \end{aligned}$$

Based on the above calculations, we assign weather state *cloudy* to a location for a specific day.

3.4 Data Fusion

Data fusion can be defined as the combination of heterogeneous kinds of information during the process of decision making in order to enhance the overall accuracy of the decision. Lee in [Lee97] proves that combinations of information will improve the final conclusion. Especially, when integrating (“fusing”) non relevant contexts, the improvement is even greater [VC99]. Before we begin the fusing process, we have to choose a general data concept to fuse. A good choice is the use of observable web data about traffic, as traffic’s behaviour is connected to the weather conditions.

There are two alternatives in applying data fusion on “Weather talk” project. These alternatives refer to the level of the model on which data fusion is applied. Following the definitions in [HL97], “raw data” fusion, combines data from all the resources at the time of retrieval. If we had chosen this approach, then we would have to collect data related with weather and traffic and then, using a more complex multi-ontological scheme, to map them to weather states. A better approach is to use “decision level” fusion, meaning that the fusing procedure is carried out on the already made decisions of two separate models.

The implementation of “decision level” fusion will request to follow the same procedure for traffic data (as described in the previous sections). At the end, using a weight mechanism we will merge the two different based decisions into the final one. The values of each weight will be determined through experiments.

Chapter 4

Hidden Markov Model application

In this chapter, we use a model based on Markov chains, the Hidden Markov Model (hereinafter HMM), in order to use yesterday’s weather state as an additional input to the previously analyzed “Weather talk” design. This input will transform our model from memoryless to memory based, as it will have to “remember” the previous day’s weather state.

4.1 Markov chains and Hidden Markov Model

This section is based on the relevant chapters in Durbin’s *et al.* [Dur98a] and Chakrabarti’s [Cha03] books. We would like to describe a basic Markov chain by using its graphical representation (Figure 4.1).

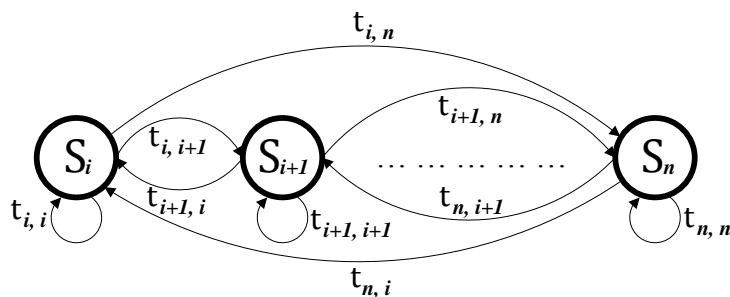


FIGURE 4.1: A simple Markov chain between n states.

A Markov chain consists of a number of states and visualizes the relationships between consequent ones. In Figure 4.1, states S_i are connected with labeled arrows. Each arrow has one direction and holds a value $t_{i,j}$, where i identifies state S_i , and j , state S_j . $t_{i,j}$

is called transition probability, and is equal with the probability of moving from S_i to S_j . Formally,

$$t_{i,j} = \Pr(\text{current state} = S_j | \text{previous state} = S_i). \quad (4.1)$$

The key characteristic of a Markov chain is that each state depends only on the previous one. This is expressed by the equation

$$\Pr(S_i | S_{i-1}, \dots, S_1) = \Pr(S_i | S_{i-1}). \quad (4.2)$$

and it is also known as the *Markov Property*.

A well known extension to classical Markov chains is the HMM. In HMM apart from transition probabilities between the states, emission probabilities of the states are introduced. An emission probability of a state S_i , $e(S_i)$, is defined as the probability of a state based on an observation o ,

$$e(S_i) = \Pr(S = S_i | o). \quad (4.3)$$

For “Weather talk” project, we have calculated these probabilities by first performing Naïve Bayes Classification on the observable data (*i.e.* weather and traffic related data) and then applying data fusion. According to [Rab89], we are searching for the “optimal sequence” (succession in weather state) associated with the observed data. As a result, the probability of a weather state based on the observational data o^1 as well as on the weather state of the previous day is

$$\Pr(W S = ws_j) = \frac{\Pr(W S_t = ws_j | W S_y = ws_k) \Pr(W S = ws_j | o)}{\sum_{i=1}^{|WS|} \Pr(W S_t = ws_i | W S_y = ws_k) \Pr(W S = ws_i | o)},^2 \quad (4.4)$$

where WS_t denotes the weather state today, WS_y the weather state yesterday, ws is a weather state of our scheme, and $|WS|$ is the total number of weather states.

We want to make a decision d of the most probable weather state. As in the previous Chapter, the denominator is the same for each weather state probability, and if we are not interested in its specific value, then we may calculate only the enumerator of (4.4). Consequently,

$$d = \operatorname{argmax}_{ws_i \in WS} \Pr(W S_t = ws_j | W S_y = ws_k) \Pr(W S = ws_j | o). \quad (4.5)$$

4.1.1 Applying Hidden Markov Model, an example

In a minimal version of our system, we only have three weather states.

$$ws = [sunny, rainy, cloudy].$$

¹ This probability combines all the observed data because data fusion has been applied.

² Equation (3.1) was used.

Following the procedure which has been described in Chapter 3, we have assigned probabilities to each weather state WS based on a observation o .

$$\Pr(WS = \textit{sunny}|o) = 0.15,$$

$$\Pr(WS = \textit{rainy}|o) = 0.4,$$

$$\Pr(WS = \textit{cloudy}|o) = 0.45.$$

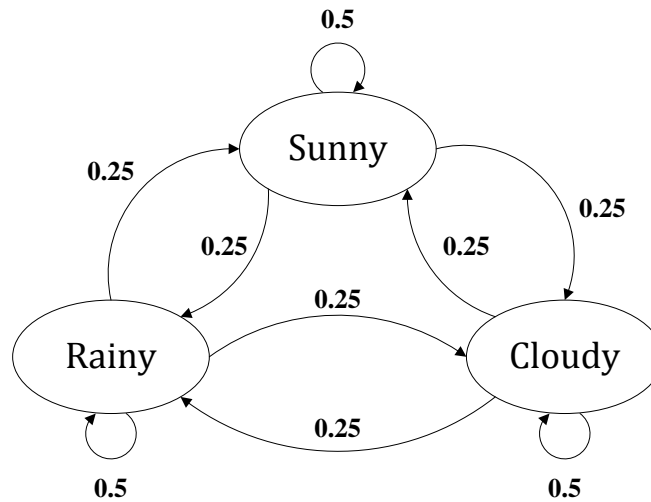


FIGURE 4.2: Applying HMM to “Weather talk”. In this minimal example, we present a chain between three weather states. Arrows have been assigned with the transition probabilities.

We also know that the weather state of yesterday is “rainy”. In Figure 4.2, we have designed a Markov chain that denotes the transition probabilities between weather states. Following the notation of the previous section, these transition probabilities can be formally written as

$$\Pr(WS_{-t} = ws_i | WS_{-y} = ws_j) = 0.25, \text{ where } i \neq j, \text{ and}$$

$$\Pr(WS_{-t} = ws_i | WS_{-y} = ws_i) = 0.5.$$

We calculate the exact probability of each weather state using equation (4.4),

$$\Pr(WS = ws_j) = \frac{\Pr(WS_{-t} = ws_j | WS_{-y} = \textit{rainy}) \Pr(WS = ws_j | o)}{\sum_{i=1}^3 \Pr(WS_{-t} = ws_i | WS_{-y} = \textit{rainy}) \Pr(WS = ws_i | o)}.$$

We have that,

Sunny weather state:

$$\Pr(WS_t = \textit{sunny} | WS_y = \textit{rainy}) \Pr(WS = \textit{sunny} | o) = 0.0375.$$

Rainy weather state:

$$\Pr(WS_t = \textit{rainy} | WS_y = \textit{rainy}) \Pr(WS = \textit{rainy} | o) = 0.2.$$

Cloudy weather state:

$$\Pr(WS_t = \textit{cloudy} | WS_y = \textit{rainy}) \Pr(WS = \textit{cloudy} | o) = 0.1125.$$

As a result, we decide *rainy* as the weather state of this day (for a specific location).

4.1.2 Disadvantages of the scheme

The model described in the two previous sections has an obvious disadvantage. In order to work properly, it needs a set of transition probabilities between the weather states. These probabilities should be a result of weather research and will vary from one location to another. The decisions of the model might be correct for some locations, but we cannot accept that they will be universally correct.

Another disadvantage is the static value of yesterday's weather. We retrieve this value from an official resource making our model dependent to an official weather source. We would like a model that is performing well through time, without applying "corrections" via official, and accurate weather states.

4.2 Changing scheme to a pair Hidden Markov Model

The general concept of this section is influenced by the reading of [Dur98b]. In this section we describe how a pair Hidden Markov Model is used in "Weather talk" in order to address the disadvantages mentioned in the previous section. In this model we only have two main states, namely Y and T (Figure 4.3).³

From the Figure 4.3, it is obvious that each state "knows" the probabilities of a set of elements x_1, \dots, x_n . In our case we want state Y to represent the previous day (*yesterday*) and state T the current day (*today*). State Y "knows" the probabilities of each weather state for the previous day, and state T "knows" the probabilities for each weather state for the current day but based only on an observation o (emission probabilities).

³ You may interpret T , and Y as *Today's*, and *Yesterday's* weather state respectively.

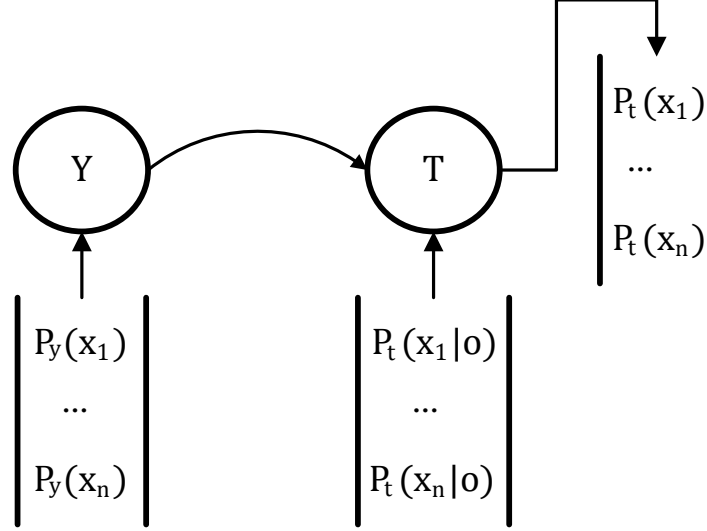


FIGURE 4.3: A simplified pair Hidden Markov Model that addresses “Weather talk” project. In this model we have two states (Y and T). We know the probabilities of the events in Y , the probabilities of the events in T based on an observation o and we want to calculate the probabilities of the events in T based on the provided information.

Rewriting the latter using the notation used for the purposes of this project, we know all the probabilities

$$\Pr_y(W S = w s_i), i \in [1, |W S|], \quad (4.6)$$

where $W S$ is a weather state, $w s_i$ its respective value, and $|W S|$ the number of different weather states. The pointer y is used to denote that we refer to probabilities of the previous day. From the observational data and after emphdata fusion procedure, we have received all the probabilities

$$\Pr_t(W S = w s_i | o), i \in [1, |W S|], \quad (4.7)$$

where pointer t is used to denote reference to the current day. No we can calculate the probability for every possible weather state for the current day using the equation

$$\Pr_t(W S = w s_i) = \frac{\Pr_t(W S = w s_i | o) \Pr_y(W S = w s_i)}{\sum_{j=1}^{|W S|} \Pr_t(W S = w s_j | o) \Pr_y(W S = w s_j)}.^4 \quad (4.8)$$

⁴ Equation (3.1) was used.

As before, the denominator of (4.8) is the same for all the weather state probabilities, and it can be omitted. Our final decision d is given by the equation

$$d = \operatorname{argmax}_{ws_i \in WS} \Pr_t(WS = ws_i | o) \Pr_y(WS = ws_i). \quad (4.9)$$

4.2.1 Pair HMM, a quick example

Suppose that our system supports only three weather states,

$$\langle \textit{sunny}, \textit{rainy}, \textit{cloudy} \rangle .$$

For the previous day we know that

$$\begin{aligned} \Pr_y(WS = \textit{sunny}) &= 0.5, \\ \Pr_y(WS = \textit{rainy}) &= 0.2, \\ \Pr_y(WS = \textit{cloudy}) &= 0.3. \end{aligned}$$

From the observation o we learn that

$$\begin{aligned} \Pr_t(WS = \textit{sunny} | o) &= 0.3, \\ \Pr_t(WS = \textit{rainy} | o) &= 0.2, \\ \Pr_t(WS = \textit{cloudy} | o) &= 0.5. \end{aligned}$$

Using the equation (4.8), the probabilities for the weather states of the current day are

$$\begin{aligned} \Pr_t(WS = \textit{sunny}) &= 0.44117, \\ \Pr_t(WS = \textit{rainy}) &= 0.11766, \\ \Pr_t(WS = \textit{cloudy}) &= 0.44117. \end{aligned}$$

In that case we can decide a combined weather state, such as “*sunny and cloudy*”.

Chapter 5

Evaluation and further extensions

In this chapter, we refer to possible evaluation methods on the results we have produced using the procedures described in the previous chapters. We also mention additional features which might be applied on “Weather talk”. In the end, we make some thoughts about how the general model we follow can be applied to other objects of interest.

5.1 Evaluation procedure in “Weather talk”

The whole procedure of this project is based on the evaluation pattern. This was the main reason for choosing weather in the first place; the easy availability of ground truth. The several weather states of “Weather talk” will be extracted from official weather observations. This states will form the basis of the weather ontology. As a result, we should start backwards by assuring these sources and their structure before implementing the core modules of the project.

There are three possible ways in evaluating the final results. One approach would be a manual evaluating method. We compare the experimental results with the official just by observing them. For a small amount of data, this might be a good way to judge whether your model is working. But it is not a proper way for making conclusions and it will not work for bigger amounts of data.

Another approach would have been a strict automatic evaluation. The term strict is used to indicate that an experimental result is correct only if it is exactly the same with the official one. This approach does not take into account small differences that may occur between the experimental and the official results; it just discards everything different.

The third approach, which we will try to follow, expresses a more flexible automatic evaluation mechanism, which has embedded the similarities (*e.g.* in percentages) between weather

states. Consequently, when it comes to a situation of dissimilar results it returns a percentage of similarity. If we assign the value of one in each correct result and the value of the similarity percentage (between the official and experimental result) in incorrect results then the percentage of success S is given by the equation

$$S = \frac{N + \sum_{i=1}^M sim(i)}{N + M}, \quad (5.1)$$

where N and M are the numbers of correct and incorrect results respectively, and $sim(i)$ is the similarity percentage for the i th element that is incorrect.

5.2 Additional features

Instead of displaying “Weather talk” results in text mode, we may try to draw a graphical representation of them on a map. Furthermore, we may try to extend our model in order to include temperature or humidity estimations. This could be carried out by mapping the differences in the probabilities of weather states into temperature or humidity arithmetic equivalents. However, it would be difficult to produce a general solution, as *e.g.* a *warm* day is understood differently in different geographical locations. As a result, for each location, we would have to use a different understanding (or mapping) of the observational data. Parsing temperature’s (or humidity’s) values directly from the text would have been desirable, but people seldom mention them, when talking about the weather.

5.3 Further challenges

“Weather talk” project is not about the weather. It is about investigating, designing, and implementing a model which could be easily evaluated. If the operation of this model is successful, then the next step should be its application to more interesting concepts and contexts. A field that has received a lot of attention recently is “Opinion mining” [Liu07b], the procedure of classifying opinions about products, individuals, places, etc. According to [HL04], one should try first to identify the different features or attributes that a document is mentioning and then decide whether an opinion about an identified feature is positive, negative or neutral.

Another challenge would be the prediction of opinions about “objects” in the future. Having a model which is able to mine opinions from the web enables us to use a large amount of training data on specific opinion targets and study how this information evolves through time. Then, we could make estimations of the transition probabilities of the system, *e.g.* what is the probability that an opinion A about a feature of an object turns to opinion B in the next state of the system. It may sound illogical but having a large amount of information at the current moment, may lead us to predictions of future opinions. By

combining different data concepts on the same purpose (data fusion) we may end up with a very sophisticated system which could try to make targeted assumptions for the future, *e.g.* whether a sports equipment company should use a specific material or not in the future.

Bibliography

- [AAGY01] C.C. Aggarwal, F. Al-Garawi, and P.S. Yu. Intelligent crawling on the World Wide Web with arbitrary predicates. *Proceedings of the 10th international conference on World Wide Web*, pages 96–105, 2001.
- [Cha03] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*, chapter 9, pages 290–297. Morgan Kaufmann, 2003.
- [CPS02] S. Chakrabarti, K. Punera, and M. Subramanyam. Accelerated focused crawling through online relevance feedback. *Proceedings of the 11th international conference on World Wide Web*, pages 148–159, 2002.
- [CvdBD99] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: A new approach to topic-specific Web resource discovery. *COMPUT. NETWORKS*, 31(11):1623–1640, 1999.
- [Dur98a] R. Durbin. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, chapter 3, pages 46–79. Cambridge University Press, 1998.
- [Dur98b] R. Durbin. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, chapter 4, pages 80–99. Cambridge University Press, 1998.
- [EM03] M. Ehrig and A. Maedche. Ontology-focused crawling of Web documents. *Proceedings of the 2003 ACM symposium on Applied computing*, pages 1174–1178, 2003.
- [FPSM92] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. Knowledge Discovery in Databases: An Overview. *AI Magazine*, 13(3):57–70, 1992.
- [Gru93] T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [HL97] DL Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.
- [HL04] M. Hu and B. Liu. Mining and summarizing customer reviews. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

- [JU99] R. Jasper and M. Uschold. A framework for understanding and classifying ontology applications. *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW*, 99:16–21, 1999.
- [Lee97] J.H. Lee. Analyses of multiple evidence combination. *ACM SIGIR Forum*, 31:267–276, 1997.
- [Lid20] G.J. Lidstone. Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192, 1920.
- [Liu07a] B. Liu. *Web data mining: exploring hyperlinks, contents, and usage data*, chapter 3, pages 87–97. Springer, 2007.
- [Liu07b] B. Liu. *Web data mining: exploring hyperlinks, contents, and usage data*, chapter 11, pages 411–447. Springer, 2007.
- [MBF⁺90] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–244, 1990.
- [Mit97] T.M. Mitchell. *Machine Learning*. WCB, chapter 6, pages 154–200. Mac Graw Hill, 1997.
- [NSD⁺01] N.F. Noy, M. Sintek, S. Decker, M. Crubézy, R.W. Fergerson, and M.A. Musen. Creating Semantic Web Contents with Protégé-2000. *IEEE INTELLIGENT SYSTEMS*, pages 60–71, 2001.
- [Rab89] LR Rabiner. A tutorial on hidden Markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [VC99] C.C. Vogt and G.W. Cottrell. Fusion Via a Linear Combination of Scores. *Information Retrieval*, 1(3):151–173, 1999.
- [Zha04] H. Zhang. The optimality of naive Bayes. *Proceedings of the Seventeenth Florida Artificial Intelligence Research Society Conference*, pages 562–567, 2004.