ILP Experiments in Detecting Traffic Problems

Sašo Džeroski[1], Nico Jacobs[2], Martin Molina[3], Carlos Moure[3]

[1] J. Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia
[2] K.U.Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium
[3] Universidad Politecnica de Madrid, E-28660 Boadilla del Monte, Madrid, Spain

Abstract. Expert systems for decision support have recently been successfully introduced in road transport management. These systems include knowledge on traffic problem detection and alleviation. The paper describes experiments in automated acquisition of knowledge on traffic problem detection. The task is to detect road sections where a problem has occured (critical sections) from sensor data. It is necessary to use inductive logic programming (ILP) for this purpose as relational background knowledge on the road network is essential. Preliminary results show that ILP can be used to successfully learn to detect traffic problems.

1 Introduction

Expert systems for decision support have recently been successfully introduced in road transport management. Some of the proposals in this direction are TRYS [4], KITS [3] and ARTIST [6]. From a general perspective, the goal of a real time traffic expert system for decision support is to advise traffic management center operators by proposing control actions to eliminate or reduce problems according to the global state of traffic. To asses the global state of traffic, the system periodically receives readings from sensors on the road, which measure magnitudes such as speed (Km/h), flow (veh/h) and occupancy (percentage of time that the sensor is occupied by vehicles), as well as information about the current state of control devices, such as traffic signals at intersections, traffic signals at sideway on-ramps, CMS (Changeable Message Signs), etc. The system interprets sensor data, detects the presence of a problem, gives the possible cause and proposes recommendations about how to solve or reduce it.

The usual approach to building traffic expert systems is to use knowledge based architectures that support the strategies of reasoning followed by operators. This approach requires to develop knowledge bases using symbolic representations (such as rules, frames, or constraints) that include specific domain knowledge of transport management corresponding to the city for which the system is developed. Among other things, knowledge on detecting specific traffic problems is necessary.

On the other hand, traffic management centers have databases that include basic information about different traffic scenarios, such as congestions at certain locations caused by lack of capacity due to accidents or excess of demand (rush hours). This data, collected from sensors on the road, can be used to either generate or improve the knowledge base for problem (incident) detection of

the expert system. The paper explores the possibility to use inductive learning techniques (such as ILP-inductive logic programming) to generate knowledge on traffic problem detection from from historical data that contains parameters recorded by sensors.

The learning experiments described in this paper take place within the context of the traffic management expert system TRYS [4], developed for the cities of Madrid and Barcelona. The system uses knowledge distributed in a collection of knowledge bases that use different representations and address specific tasks (such as data abstraction, incident detection, problem diagnosis, prediction of behaviour, and recommendation of control actions). The knowledge for incident (traffic problem) detection has been formulated by domain experts in a first-order frame-based representation. Therefore, ILP is a suitable tool for learning to detect traffic problems in this context.

Overall, two kinds of input are available to the learning process. The first type is background knowledge on the road network, which is present in and used by the TRYS system. An object oriented representation is used to capture the different types of road sections, the relations among them, and the placement of sensors on individual road sections. The second type is sensor readings on three basic quantities describing traffic behaviour: speed, flow and occupancy. Both types of input will be described in more detail in Section 2. The goal of the learning process is to identify critical sections (where problems have occured) by using sensor readings and road geometry. Technically speaking, a critical section is a section of the road which constrains the road capacity the most, e.g., because an accident has occured just after this section in the immediate past. In the paper, the term accident critical section refers to such a section and not to a section where accidents occur frequently.

Let us note at this point that in practice real sensor data are available. However, we have used simulated data in our experiments for three reasons. The first is that real sensor data were not immediately available because of management reasons. The second is missing sensor data from broken sensors (which amounts to approximately 20% of the sensors). Finally, using a simulator makes it possible to easily generate a wide range of different traffic problems (including accidents that should not be artificially produced in the real world).

We used AIMSUN (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks) [1], a software tool able to reproduce the real traffic conditions of any urban network on a computer. AIMSUN follows a microscopic simulation approach. It means that the behaviour of each individual vehicle in the network is continuously modelled throughout the simulation time period it remains inside the system (i.e. the traffic network), according to several vehicle behaviour models. A model of the urban-ring of the city of Barcelona was developed using this simulator. This model includes exactly the same variables that the real information system records using sensors and was calibrated using information from the real system. Using this model, a collection of examples (including accidents and congestions due to rush hours) were produced for the learning experiments presented in the paper.

2 Road network and sensor data

In TRYS [4], the road network is represented in an object oriented fashion. The basic object in the road network representation is the section. A section refers to a cross-section of the road and typically has an array of sensors associated to it. There exist several types of sections, such as off-ramp, on-ramp or highway. Relations between sections, such as previous and next, are included in the TRYS knowledge base. The complexity of road structures makes it possible for a section to have more than two previous or next sections.

A link describes a logical group of sections. For instance, the section just before and just after an off-ramp, together with the off-ramp itself, form an off-ramp-link. There are about ten different types of links. TRYS also uses other concepts like nodes, problem areas and measurement points, but these were not used in our experiments.

The information about sections and links is static. Each section is of a certain type and is associated to a number of sensors (as many sensors as there are lanes at that cross-section of the road) and each link is of a certain type and links a predefined set of sections. These relationships can therefore be considered background knowledge for the learning process.

Sensors provide us with a continuous stream of information, sending five readings each minute that refer to the last minute and each of the four minutes preceding it. Typically, flow (number of cars that passed the sensor in the last minute) and occupancy (the pro mille of time the sensor is occupied) are measured. Some sensors (which are actually double sensors) also measure the average speed of the cars that passed the sensor during the last minute. The measurements of sensors related to a single section are aggregated: flow is summed across lanes, while occupancy and velocity are averaged across lanes. Saturation is a derived quantity defined as the ratio between the flow and the capacity of a section: the latter depends on the number of lanes and is part of the background knowledge.

The TRYS system stores its information in two formats: in CONCEL format, which is a frame-based format, and in Prolog format. The Prolog format is object-oriented and consists mainly of facts about the predicates `instance` and `value`. For simplicity reasons, we transform these facts in the following fashion: facts of the form `instance(Instance,Class)` are translated to facts of the form `Class(Instance)` and facts of the form `value(Instance,Attribute,Value)` are translated to facts of the form `Attribute(Instance, Value)`. For example, the fact `instance(salida_a_rambla_Prim,off_ramp)` is transformed to `off_ramp(salida_a_rambla_Prim)`.

3 An experiment with CLAUDIEN

In a preliminary experiment, nine accidents and two congestions at two different off-ramp links were simulated. In addition to the data transformation described above, the values for speed, saturation and occupancy were discretized according to expert provided thresholds that are in use in TRYS. One of the reasons for discretizing was the small number of examples used.

The static information about sections and links was used as background knowledge. Examples consisted of facts specifying the speed, occupancy and saturation for all sections in the relevant problem area at one moment in time. Each example also contained exactly one fact of the form `accidentat(X)` or `congestionat(X)`, where `X` is the critical section. The task was to find rules that identify critical sections by using sensor values and road geometry.

The ILP system CLAUDIEN [5] was used in this experiment for two reasons. First, the small number of examples dictates the use of a strong declarative bias (which is provided by CLAUDIEN) in order to obtain reasonable rules. Second, CLAUDIEN generates all valid rules, providing some redundancy that might be useful in the light of missing sensor information which will occur in real world data.

Three rules cover all 9 accident examples. The first says there is an accident at critical section X, which is the previous section of off-ramp link Y (enlace de salida) with next section O and ramp section R, if the speed (velocidad) at X is not high (alta), the speed at O is high and the saturation on R is low (baja). The predicate names originating from the TRYS-system are in Spanish.

```
accidentat(X) :-
  seccion(X), seccion_anterior(Y,X), seccion_posterior(Y,O),
  enlace_de_salida(Y), velocidad(X,VX), not VX = alta,
  velocidad(O,VO), VO = alta,
  seccion_en_rampa(Y,R), saturacion(R,SR), SR = baja.
```

There were also two examples of congestion at an off-ramp and two rules rules covered both examples. The first of these says there is a congestion at the ramp section X (seccion en rampa) of the off-ramp link Y (enlace de salida) when the occupancy of X is not low. All five rules describe sensible conditions that were already known to the domain experts. This indicated that ILP might be useful in this domain, and encouraged us to undertake further experiments.

4 Experiments with TILDE

An extended dataset containing 66 examples of congestion and 62 examples of accidents on different locations (off-ramp, on-ramp and highway sections) was generated using the simulator. The aim of the experiments with the extended dataset was to understand which measurements and road geometry predicates are relevant to the learning task at hand. Given this aim and the larger set of simulations, the task was formulated as a classification task.

Each section at a particular moment of time was treated as an example, classified into one of three classes: an accident critical section, a congestion critical section or a non critical section. In this way we obtained a dataset consisting of 5952 examples. Facts on sensor values (which were not discretized) were moved to the background knowledge, which also included facts on road geometry. Predicates that allow access to sections before and after a given section, as well as predicates that calculate the speed-, saturation- and occupancy-gain (also in percentages) between sections were added to the background knowledge.

The TILDE system [2] — based on top down induction of logical decision trees — was used for experiments with this dataset for a number of reasons. First, TILDE addresses classification problems in a first-order setting. Second, it allows for a very weak language bias that easily handles a variety of situations (unlike our preliminary experiment where all critical sections were on an off-ramp). Third, it can deal with real-valued sensor measurements directly, performing discretization itself. Finally, TILDE is very efficient, an important aspect for our problem where we have background knowledge of size approx. 1 MB and 5952 examples.

Two experiments were performed. In the first experiment TILDE had to build a classifier for all three classes, while in the second experiment it was only given critical sections and had to build a classifier that distinguishes between the two types of critical sections. In both experiments a 6-fold cross-validation was performed.

The first experiment gave some encouraging results: 80% of the congestion critical sections were classified correctly and only 39 out of the 5824 non critical sections were classified incorrectly. None of the congestion critical sections were classified as accident or vice versa. The results for accident critical sections were much worse: only 38 out of the 62 examples (61%) were classified correctly. Why accidents are harder to classify than congestions needs to be investigated. A potential problem is also the extremely skewed class distribution (only 128 of almost 6000 examples are critical sections).

When we take a look at the predicates used, we see that the trees very rarely refer to previous sections, but often refer to sections downstream (the use of the gain-predicates is not considered as a reference to the previous section). Regarding the predicates related to sensor measurements, speed (used 60 times), occupancy gain (57) and saturation (54) seem to be important concepts, whereas the gain and percentage gain predicates seem to be less important.

As expected, the second task of predicting the class of a given critical section is much simpler that the first: 96.9% of the congestions and 96.7% of the accidents were classified correctly. Moreover, the decision tree was built very fast (about 3 seconds, compared to the 4 hours it took in the first experiment). Surprisingly, very few predicates were used: saturation, occupancy and the type of section were used in most trees, whereas a reference to the next section appears in only one of the six trees. One of the decision trees states that a section is accident critical if its saturation is below 42.75, otherwise it is congestion critical unless of type highway (when it is again accident critical).

5 Discussion

We have presented a novel application domain for inductive logic programming, namely the domain of detecting traffic problems. The task addressed was to learn rules that identify critical road sections due to accidents or congestions. Background knowledge on road geometry is available, requiring the use of ILP for this task. While simulated data were used for our experiments, it should be noted that the simulator is very realistic and has been calibrated using real-world data.

In a preliminary experiment with CLAUDIEN interesting (but already known) rules were found, encouraging further experiments. A larger set of examples generated using the simulator was supplied to TILDE. The trees generated indicate that sections downstream provide important information on whether the section at hand is a critical one, as well as the predicates providing the values of speed, occupancy gain and saturation.

Much work remains to be done. High on the priority list is the task of learning to distinguish between non critical and any type of critical section. A difficulty that has to be taken into account is the skewed class distribution. Distinguishing among different types of critical sections seems to be an easier task as indicated by our second experiment with TILDE.

Exploring the use of other ILP systems and other biases (background knowledge predicates) will also receive considerable attention. A practical issue of utmost importance is the issue of using real sensor data instead of simulated data. Missing sensor values are a problem that has to be dealt with here and redundant rules will have to be built for this purpose.

Other issues to be addressed include mapping the induced problem detection rules into a frame-based representation with which experts are familiar and using the time series of sensor values instead of the current values only. The domain of traffic control also holds other challenges for machine learning techniques. Detecting traffic problems is only one step of the traffic management process: suggesting actions to alleviate the problems is the natural next step. Since examples of operator actions in response to detected problems exist, there is hope that the problem of suggesting appropriate actions for alleviating traffic problems can also be addressed using machine learning and inductive logic programming.

Acknowledgements Nico Jacobs is financed by a specialisation grant of the Flemish Institute for supporting scientific-technological research in the industry (IWT). This work was supported by the ESPRIT IV Project 20237 ILP2.

References

1. Barcelo, J., Ferrer J.L., and Montero, L. (1989). *AIMSUN: Advanced Interactive Microscopic Simulator for Urban Networks. Vol I: System Description, and Vol II: User's Manual.* Departamento de Estadistica e Investigacion Operativa, Facultad de Informatica, Universidad Politecnica de Cataluna, Barcelona, Spain.
2. Blockeel, H., and De Raedt, L. (1997). Lookahead and discretization in ILP. In *Proc. 7th Intl. Workshop on Inductive Logic Programming*, pages 77–84, Springer, Berlin.
3. Cuena, J., Ambrosino, G., and Boero M. (1992) A general knowledge-based architecture for traffic control: The KITS approach. In *Proc. Intl. Conf. on Artificial Intelligence Applications in Transportation Engineering.* San Buenaventura, CA.
4. Cuena, J., Hernandez, J., and Molina, M. (1995). Knowledge-based models for adaptive traffic management systems. *Transportation Research: Part C*, 3(5): 311-337.
5. De Raedt, L., and Dehaspe, L. (1997). Clausal discovery. *Machine Learning*, 26: 99–146.
6. Deeter, D.L., and Ritchie, S.G. (1993). A prototype real-time expert system for surface street traffic management and control. In *Proc. 3rd Intl. Conf. on Applications of Advanced Technologies in Transportation Engineering*, Seattle, WA.

This article was processed using the LaTeX macro package with LLNCS style