Combining Observations and Expectations: Application to the Refinement of an Image Sequence Classification

Christine Largouët *
ENSAR, 35042 Rennes, FRANCE
clargoue@irisa.fr

Marie-Odile Cordier †
USARQ-INRA, 35042 Rennes, FRANCE
cordier@irisa.fr

Abstract

The main idea of this paper is to take profit of the available knowledge on the dynamics of a system to improve the quality of a sequence of observations issued by an image classifier. It has been applied to refine the landcover classification of an image sequence. The agricultural system is modeled by a probabilistic timed automaton. One of the original points is the use of both predicted and postdicted expectations which are combined to observations in a bayesian way.

1 Introduction

Remote sensing when used on an image sequence is very useful for analyzing environmental changes. The first step of remote sensing analysis is the classification step which provides a thematic map, where each item is assigned to a label corresponding to a class. In this paper, the application of interest focuses on the landcover classification of images, taken at different time, and describing an agricultural area. The result, a sequence of thematic maps, constitutes the basis knowledge for monitoring the water quality of a watershed. The aim of this paper is to propose a new approach for refining the classification of a sequence of images. The plots of land are considered as dynamic systems and their evolution (knowledge about crop and rotations) is described in a model. This model is used to get the expected landcovers at the image date. Combining the expectations and the observations extracted from the images results in improving the classification of images.
After having situated our approach with respect to theories as Kalman filtering and belief change, the classification refinement problem is defined in section 2. Section 3 focuses on the modeling of the dynamics of the plots and presents probabilistic timed automata. Section 4 explains how the expected states and their respective probabilities are computed. The combination rule of expected states and observations is then defined. Section 5 introduces the decision rules we are using. Experimental results carried out on a sequence of five images are analyzed in section 6. Finally, conclusions and perspectives are given in section 7.

* IRISA, AÏDA Team, F-35042 Rennes
† on leave from IRISA, F-35042 Rennes

2 A classification refinement problem

The basic idea underlying our work is to make use of an explicit model of the dynamics of the system to improve the observations issued by an image classifier. It is then strongly connected with research work dealing with matching expectations and observations as the theory of belief change and as Kalman filtering in the control theory domain. Our intention is not to go into details on any of these two domains [1] but only to sketch these two processes in order to situate our own work by highlighting the similarities but also the specificities we have to deal with[2]. In this section, we define the classification refinement problem we are facing, situate it with respect to these related approaches and sketch the algorithm we propose to solve it.

2.1 Kalman filtering

Kalman filtering is a technique which is widely used in the control theory field for monitoring a physical system. The aim is to track, i.e. to estimate, the current state of the system. It makes use of a numerical state evolution model and of possibly erroneous and noisy observations issued from sensors. The two-step process (see Figure 1) consists first in predicting the state of the system at time t, E_t^{t-1}, [3] from the state estimate at time $t-1$, E_{t-1}^{t-1}, and then to match E_t^{t-1} with the observations[4], O_t, in order to get the current state estimate at time t, E_t^t.

[1] The reader is referred to [GA93] for more details on the Kalman filtering and to [FH99] on belief change.

[2] For a discussion on the links existing between Kalman filtering and belief change, the reader is referred to [CT99].

[3] $E_k^{k'}$ denotes the state estimate at time k when taking into account observations at time k'.

[4] To simplify, we do not make explicit the observation model which allows to extract the expected observation from the expected state. In our case, as will be seen later, the state is directly, even if imperfectly, observable and then no observation model is needed.

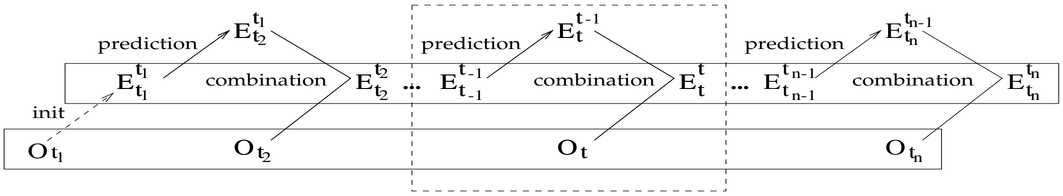

Figure 1: Iterated process based on prediction

Let us point out some characteristics. The Kalman filtering relies on a *numerical* evolution model, generally given as a set of differential equations. The observations are dated and the prediction step takes this date into account. The observations and the expectations are given *symmetrical* roles, both being subject to uncertainty and imprecision. It is implicitly supposed that the merge of these two uncertain information will lead to a better (more realistic) estimate than what can be obtained by taking one of them separately. The model is used in a *predictive* way to infer expectations at time t' from a state at time t with $t' > t$. That is justified by the fact that Kalman filtering is used in an on-line control context. Lastly, the Kalman filtering is clearly an *iterative* process.

2.2 Belief change

Belief change theory has received a lot of attention in the AI community and classically distinguishes belief revision from belief update. Being interested in the following in tracking an evolving world, we will focus on update. Update can be defined as follows: having a set of beliefs on the state of the world at time $t-1$ and getting an observation at time t, the update operation computes a new set of beliefs which is supposed to be better (more realistic) than the previous one. This new set of beliefs corresponds informally to the most plausible evolution of the world from time $t-1$ to time t satisfying the observation. As proposed by [CL95, Bou98], the update operation can be considered as a two-step process. The first one relies on an evolution model of the world. It can be an explicit model (the transition model in [CS95] , the event model in [Bou98]) or an implicit model as an inertia model. This model allows to compute the expected state of the world at time t, E^t_{t-1}, from the belief state at time $t-1$, E^{t-1}_{t-1}. The second step consists in matching the observation, O_t, with these expectations in order to get the belief state at time t, E^t_t. This second step is close to a revision operation where preferences take into account the respective plausibility of the expectations. It can be illustrated as one of the basic step in Figure 1 where the prediction step makes use of an explicit model and where the combination corresponds to revision.

Let us point out some characteristics. The update operation relies, explicitly or implicitly [5], on a *qualitative* transition model expressing the way the world evolves from one instant to a posterior one. The model does not take into account any quantitative temporal information and the observations are not dated. According to the classical Katsuno-Mendelzon postulates, an absolute *priority of the observations* on the expectations (success postulate) is generally assumed, even if [Bou98] generalizes the classical framework and considers noisy observations. The model is used in a *predictive* way even if it is argued by [Bou98] that observation can question previous beliefs. It is related to the fact that belief change does not aim to produce the "best" sequence of belief states but focus on producing the "best" current belief state. It is also related to the lack of emphasis put by belief change theory on iterated revision/update.

[5] The implicit inertia model is generally encoded directly in the update operator, as preferences for instance.

2.3 Classification refinement problem

Let us now describe more precisely our landcover classification problem and situate it w.r.t these two related theories.

A sequence of images, $I_1, \cdots, I_n$ taken at time $t_1, \cdots, t_n$, acquired by different sensors (SPOT, LANDSAT, aerial) represent the same set of landscape plots. Let $\mathcal{C}$ be the set of landcover classes. For a given plot, let O_{t_i} be the probability distribution over the landcover classes resulting from a preliminary per-plot classification of the image I_i.

The problem we are faced with is to improve the results of the classification on the n images sequence. Considering each plot independently along time, we get the following general algorithm.

Preclassification:
for each image I_i
  for each plot P
    classification $\longrightarrow O_{t_i}$;
Classification refinement:
for each plot P
  Plot refinement task
    $(\mathcal{A}, [O_{t_1}, \cdots, O_{t_n}]) \longrightarrow [E^{t_1}_{t_1}, \cdots, E^{t_n}_{t_n}]$
  Decision task
    $[E^{t_1}_{t_1}, \cdots, E^{t_n}_{t_n}] \longrightarrow [P_{t_1}, \cdots, P_{t_n}]$

The *plot refinement task* takes as input the pair $(\mathcal{A}, [O_{t_1}, \cdots, O_{t_n}])$ where $\mathcal{A}$ is the plot evolution model and provides, for each plot, a sequence $[E^{t_1}_{t_1}, \cdots, E^{t_n}_{t_n}]$ where $E^{t_1}_{t_1}, \cdots, E^{t_n}_{t_n}$ are probability distributions over the classes of $\mathcal{C}$. The idea is to combine the two sources of information, the observation sequence and the mo-

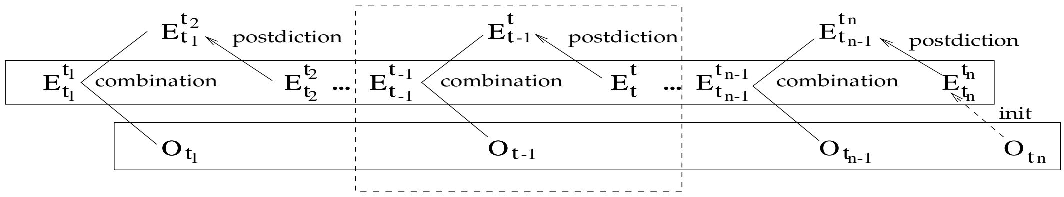

Figure 2: Iterated process based on postdiction

del, in order to get a "better" (more realistic) result than the observation sequence alone.
The *decision task* is in charge of selecting one (P_{t_i}) of the remaining possible landcovers for a plot ($E_{t_i}^{t_i}$) and is commented in section 5.

The plot refinement task can then be viewed as illustrated in Figure 1. The evolution model is used to *predict* the state (the landcover) of the plot, E_t^{t-1}, at the date t from the state of the plot at the date $t-1$. The observation at the date t, O_t, is combined with the predictions to get the resulting state at the date t. Another, less expected, way is illustrated in Figure 2. The evolution model is there used to *postdict* the state of the plot, E_{t-1}^t, at the date $t-1$ from the state of the plot at the date t. The observation at the date $t-1$ is then combined with the postdictions to get the resulting state at the date $t-1$.

Let us then point out some characteristics of our refinement problem and compare it with the Kalman filtering and the belief update. First, the observations are dated which leads us, in order to take a real profit of the expectations, to rely on a quantitative model of the agricultural plots. As seen below, the evolution model includes *quantitative* temporal constraints. Observations and expectations are considered in a *symmetrical* way as both are imperfect: the observations are imperfect due to the intrinsic quality of the image and to the performance of the classifier; the expectations are imperfect due to the uncertainty and imprecision of the model. Contrary to the two previous cases, *prediction and postdiction* are both worth considering. The main reason is that it is an off-line process. The whole sequence of observations being known from the beginning, each observation can be chosen as the initial state of the simulation which can be either forward (in a predictive way) or backward (in a postdiction way). The problem is clearly *iterative* as we are interested in the sequence of resulting states and the quality of the result has to be estimated globally. All these points make our refinement problem as an original one, even if some strong links exist with these related domains. In the following, we explain the main choices we made: i) the way the prediction and postdiction tasks are interleaved is explained in 2.4; ii) the evolution model is described by a probabilistic timed automata as explained in section 3; iii) the uncertainty on observations and expectations is expressed in a probabilistic framework which explains the choice of a bayesian combination operator (see section 5).

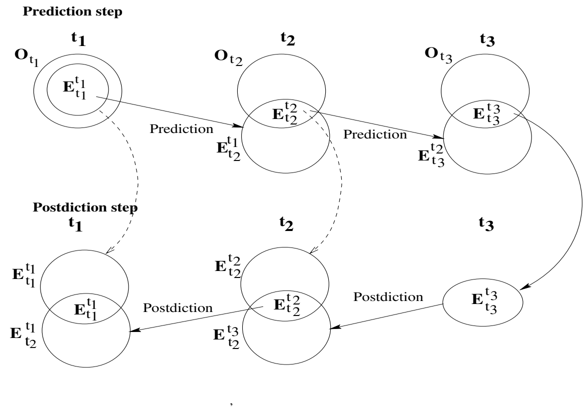

Figure 3: Prediction and postdiction on 3 dates

2.4 Plot refinement task

As said before, the plot refinement task interleaves a prediction step and a postdiction step in order to improve the initial classification as much as possible. The algorithm we implemented starts from the first image I_1 and executes the prediction step until the last image I_n; the postdiction step then begins from the last image until the first one. The plot refinement algorithm is then as follows:

Plot refinement task:
$(\mathcal{A}, [O_{t_1}, \cdots, O_{t_n}]) \longrightarrow [E_{t_1}^{t_1}, \cdots, E_{t_n}^{t_n}]$
1- Prediction
Initialization: $E_{t_1}^{t_1} := O_{t_1}$
For i from 2 to n
Prediction: $E_{t_i}^{t_{i-1}} :=$ prediction $< \mathcal{A}, E_{t_{i-1}}^{t_{i-1}}, t_i >$
Combination: $E_{t_i}^{t_i} := O_{t_i} \otimes E_{t_i}^{t_{i-1}}$
2- Postdiction
For i from $n-1$ to 1
Postdiction: $E_{t_i}^{t_{i+1}} :=$ postdiction $< \mathcal{A}, E_{t_{i+1}}^{t_{i+1}}, t_i >$
Combination: $E_{t_i}^{t_i} := E_{t_i}^{t_i} \otimes E_{t_i}^{t_{i+1}}$

Figure 3 shows the refinement of a classification on a sequence of 3 images.

After explaining how we model the dynamics of the system, we present in section 4 the way the expected states are computed and combined to observations.

3 Modeling the dynamics of the plot

The choice of timed automata for our application is motivated by the fact that it enables us to represent a nondeterministic process, to deal with uncertainty on the temporal transitions and to represent cycles thanks to the clock reset operation. It is then quite adapted to model the dynamics of the agricultural plots we are considering ([LC00]). We propose to extend timed automata to *probabilistic timed automata*. Contrary to [KNSS99], our motivation is not to extend model-checking methods to a probabilistic formalism but only to be able to get the probability of a state proposed by a reachability analysis, this later being performed using a traditional model-checking method.

3.1 Probabilistic timed automata

Timed Automata [AD94] is a formalism used to model and analyze real-time systems. A timed automaton is a finite state machine extended with a set of clocks that allow to express timing constraints. Probabilistic timed automata are classical timed automata in which the locations and the transitions are associated to discrete probability distributions.

As in timed automata, timing constraints are associated to *locations* (vertices of the graph) and to transitions. They are expressed with one or several synchronized clocks. Transitions are instantaneous and allow clocks to be reset whereas time passes in the locations. Timing constraints associated to a transition and to a location are resp. called a *guard* and an *invariant*.

Def. 1 (Timing constraint) Let $\mathcal{X}$, be the set of clock variables, and $\Phi(\mathcal{X})$ the set of timing constraints φ defined by the following grammar:

$$\varphi ::= x \leq c \mid c \leq x \mid x < c \mid c < x \mid \varphi \wedge \varphi$$

where $x \in \mathcal{X}$ and c is a constant in Q. A *clock valuation* is a function that assigns a non-negative real number to every clock $x \in \mathcal{X}$.

The next-location probability distribution, derived from probabilistic systems, is defined as follows.

Def. 2 (Next-location probability distribution) If $\mathcal{S}$ is the location space of the system, the next location probability distribution is a function $p : \mathcal{S} \times \mathcal{S} \times \Phi(\mathcal{X}) \rightarrow [0,1]$. We denote $p(s, s', \lambda)$ the probability that the system moves from the location s to the location s' at a time t with t satisfying the timing constraint λ. We require that $\sum_{s' \in \mathcal{S}} p(s, s', \lambda) = 1$.

Def. 3 (Probabilistic Timed Automata (PTA)) A probabilistic timed automaton $\mathcal{A}$ is a tuple $< \mathcal{S}, \mathcal{X}, \mathcal{L}, \mathcal{E}, \mathcal{I}, \mathcal{P} >$ where:

- $\mathcal{S}$ is a finite set of locations and $s_o \in \mathcal{S}$ is the initial location.
- $\mathcal{X}$ is a finite set of clocks.
- $\mathcal{L}$ is a finite set of labels.
- $\mathcal{E}$ is a finite set of edges, each edge e is a tuple $(s, l, \varphi, \delta, s')$ such that e connects the location $s \in \mathcal{S}$ to the location $s' \in \mathcal{S}$ on symbol $l \in \mathcal{L}$. The enabling condition is captured in φ and $\delta \subseteq \mathcal{X}$ gives the set of clocks to be reset when the edge is triggered.
- $\mathcal{I} : S \rightarrow \Phi(\mathcal{X})$ maps each location s with a timing constraint called an invariant.
- $\mathcal{P} : S \rightarrow \mu(s)$ associates to each $s \in \mathcal{S}$ a set of next-location probability distributions from s.

At the initial location s_o, all the clocks are initialized with 0. At any point of time, when the system is in a current location s, it can remain in this location, with a *timed transition*, or move to a location s' through a *discrete transition*. The probability assigned to a timed transition is $p(s, s, \lambda)$ with λ satisfying $I(s)$ and the probability assigned to a discrete transition is $p(s, s', \lambda)$ with λ satisfying φ.

3.2 Plot evolution model

We now make the PTA framework more concrete in modeling the plot evolution. The sets of locations of the automaton is $\mathcal{S} = \mathcal{C} \cup \{init\} \cup \{end\}$ where $\{init\}$ and $\{end\}$ are respectively referred to initial and final locations of the crops. Dates of crop calendar are expressed in a number of days between 0 and 365 and September 1 is the origin of the crop cycle. Two clocks are defined: x referring to the days and y referring to the years. Clocks x and y are initialized at September 1 of the first cycle of study. Since at the beginning of each cycle the clock x is reset, the value of the clock x represents the value of the clock y modulo 365. The timing constraints used in the next-location probability distributions correspond to time intervals and are inferred from the automaton invariants and guards. A next-location probability having as a timing constraint *true* is independent of time. The data set used to build the automaton was acquired from interviews with agronomists of the area of the study. The PTA we have used for our experiments contains 41 locations, 62 transitions and 100 next-location probabilities.

4 Combining expected states and observations

This section explains how expected states and their respective probabilities are computed by making use of reachability analysis on a timed automaton. Lastly, we present the rule used to combine expected states and observations.

4.1 Computing expected states

Reachability analysis is one of the symbolic verification methods usually dedicated to the validation of timed systems [HNSY94]. It is defined as follows: *given two locations of a timed automaton, the problem consists in verifying if there is a path leading from one location to the other location.* It is then quite adequate to implement the prediction and postdiction mechanisms. Reachability analysis delivers all the sequences of transitions $(e_1, e_2, \cdots, e_n)$ leading from a location s at time t to a location s' at time t'[6].

[6] For sake of simplicity, we consider here that time t and t' capture the value in days and years.

They are denoted $K_{[s,t,s',t']}$ and referred as *scenarios* in the following. The computation of scenarios is implemented with the tool Kronos [Yov97] where reachability properties are expressed in the TCTL temporal logic [AD94]. It is detailled in [LC00].

4.2 Probabilities of expected states

The objective of this step is to define the probabilistic distribution $E^t_{t'}$ of the expected classes, given the model and the set of possible scenarios.
The probability of a scenario $K_{[s,t,s',t']}$ occurring from a location s at time t and leading to a location s' at time t' such that: $s \overset{e1}{\rightarrow} s_1 \overset{e2}{\rightarrow} \cdots \overset{e_{n-1}}{\rightarrow} s_n \overset{e_n}{\rightarrow} s'$, is defined recursively as follows:

$$p(K_{[s,t,s',t']}) = p(s', s', \lambda') \prod p(s_i, s_{i+1}, \lambda')$$

with t' satisfying λ' and $i \in [1, n]$.
The probability to reach the location s' at time t' from the location s at time t is the sum of the probabilities of the v scenarios leading from the location s at time t to the location s' at time t'.

$$p(s', t'|s, t) = \sum_{K_j} p(K_{[s,t,s',t']}) \quad j \in [1, v]$$

A landcover type C is represented by several locations of the automata denoted $Loc(C) \subseteq \mathcal{S}$. The probability for the plot to belong to the class C' at time t' given the fact that it is in the class C at time t is denoted by $p(C', t'|C, t)$. It is the sum of the probabilities of all possible successions between locations of $Loc(C)$ and locations of $Loc(C')$, each of which being weighted by the *a priori* probability of the locations of $Loc(C)$.

$$p(C', t'|C, t) = \sum_{s' \in Loc(C'), s \in Loc(C)} p(s', t'|s, t).p(s, t)$$

This probability is normalized taking into account all the possible successions between landcover types of E^t_t and those of $E^t_{t'}$. The probability that the plot belongs to the class C_i at time t' is then:

$$p^t_{t'}(C_i) = \frac{\sum_{C_j \in E^t_t} p(C_i, t'|C_j, t)}{\sum_{C_k \in E^t_{t'}} \sum_{C_j \in E^t_t} p(C_k, t'|C_j, t)}$$

4.3 Combination rule

The method we used to combine the observations and the expectations is the Bayes rule. We denote $p(C_i|O_{t'})$ the probability that the plot P belongs to the class C_i at time t', given the observation. $p^t_{t'}(C_i)$ is the probability of C_i obtained as described above. The probability of the combined hypothesis is:

$$p^{t'}_{t'}(C_i) = \frac{p(C_i|O_{t'}) p^t_{t'}(C_i)}{\sum_{C_k \in \mathcal{C}} p(C_k|O_{t'}) p^t_{t'}(C_k)}$$

5 Decision task

This section presents the criterion, called a "decision rule", used to choose the more realistic landcover type on ambiguous plots remaining after the refinement task. Three decision rules are considered:

maximum of probability The class C chosen for the plot P at time t_i is the one having the greatest probability on each image.

maximum of probability in prediction The class C chosen for the plot P at time t_i is, among the possible landcover successors of the class C' chosen for the plot P at time t_{i-1}, the one having the greatest probability.

maximum of probability in postdiction The class C chosen for the plot P at time t_{i-1} is, among the possible landcover predecessors of the class C' chosen for the plot P at time t_{i+1}, the one having the greatest probability.

6 Experimental results

In the project "Bretagne Eau Pure", the landcover evolution is an important information to assess the risk of pollution. The area of study is a watershed supplying the city of Rennes in Brittany (northwest of France). A sequence of five images, each of which contains 2124 plots, is available. The characteristics of the images are the following: 1) aerial (18/04/1997), 2) Landsat TM (28/07/1997), 3) SPOT (05/12/1997), 4) SPOT (25/05/1998) and 5) SPOT (07/08/1998). A set of sample plots, about 110 for each image, provided from human observations and called the "ground truth", characterized *a priori* the classes of the images.

6.1 Preliminary classification

The preliminary classification is performed using the Arkemie software [Ark96]. It provides, as a result for each plot, a distribution of probabilities over the classes. A threshold fixed to 0.1 is used to discard the less representative classes.

6.2 Quality evaluation

In order to evaluate the method, it is necessary to judge the quality of the classified sequence before and after the refinement. First of all, we define two criteria to assess the accuracy of the classification:

- ambiguity rate: it describes the percentage of clear plots (i.e. the plots identified with only one "sure" class) in an image before having applied the decision rule.
- identification rate: it describes the correctness of the classification in relation with the ground truth. It is calculated by dividing the number of correct plots by the total number of sample plots.

In our experiment we have a dual objective for each classified image: i) to increase the number of clear plots before the decision rule, ii) to obtain a reasonable identification rate.

6.3 Analysis of the results

The results obtained after the preliminary classification are described in Table 1 which shows, for each image I_i, the identification rate τ_i, the number and the percentage of clear, ambiguous and non labeled plots. Non-labeled plots result from difficulties encountered by Arkemie to classify too small plots. The class chosen as labeling the plot is the one having the maximum of probability.

I_i	τ_i	clear plots		ambiguous plots		non-labeled plots	
I_1	90.91%	1788	84.2%	330	15.5%	6	0.3%
I_2	89.29%	1697	79.9%	386	18.2%	41	1.9%
I_3	75.68%	796	37.5%	1306	61.5%	22	1%
I_4	64.49%	958	45.1%	1161	54.7%	5	0.2%
I_5	63.55%	541	25.5%	1583	74.5%	0	0%

Table 1: Preliminary classification results

Table 2 shows the results obtained after the refinement of the classification. The first column shows the number of clear plots obtained after the prediction/postdiction process before any decision rule has been applied. The last three columns present the identification rate obtained after having applied the following decision rules on the remaining ambiguous plots: 1) maximum of probability, 2) maximum of probability in prediction, 3) maximum of probability in postdiction. Although the low number of samples prevents giving definite conclusions, the decision rule 3 appears to produce the better sequences. The results show that the number of clear plots increases on all images (up to 54.5% on Image I_3) and that the identification rate gets better whatever the decision rule applied.

I_i	clear plots		rule 1 τ_i	rule 2 τ_i	rule 3 τ_i
I_1	2040	96%	90.91%	90.91%	91.74%
I_2	1947	91.7%	90.48%	90.48%	90.48%
I_3	1954	92%	78.38 %	79.73%	81.08%
I_4	1745	82.2%	70.09%	68.22%	70.09%
I_5	1550	73%	74.77%	74.77%	74.77%

Table 2: Refinement of the classification results

7 Conclusion

This paper proposes a new approach to improve the quality of a sequence of dated observations. It takes profit of domain knowledge, here the dynamic evolution of a plot, and relies on combining expectations and observations. The formalism chosen to model the plot evolution, probabilistic timed automata, defines in a explicit way uncertain temporal constraints on states. The evolution model is then undeterministic and imprecise. Because of noisy sensors and of the lack of precision of the preclassification, the observations are not necessary reliable. However, exploiting both allows us to discard some unrealistic hypotheses and to rank the remaining hypotheses more adequatly. The experiment shows the interest of the method which increases the quality of each thematic map and ensures the global consistency of the sequence.

One original point is the cooperative use of prediction and postdiction mechanisms. An analysis of the results (not reported here) have shown that prediction and postdiction mechanisms contribute equally to the global improvement. We are currently experimenting the effects of a change in the initial states on the quality of the result. Another original point is to consider globally the sequence of observations in order to improve it, which is not the case in related domains as the Kalman filtering method in control theory or the belief change theory.

The choice of probabilities to express uncertainty is related to the fact that the preliminary classifier we are using provides us with probability distributions. However, we are currently studying the use of the Dempster-Shafer theory in order to be able to exploit the existing confusions known *a priori* between the classes of observations.

References

[AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(183):235, 1994.

[Ark96] http://ourworld.compuserve.com/homepages/arkemie/.

[Bou98] C. Boutilier. A Unified Model of Qualitative Belief Change: a Dynamical Systems Perspective. *Artificial Intelligence*, 1-2:281–316, 1998.

[CL95] M-O. Cordier and J.. Lang. Linking transition-based update and base revision. In *ECSQARU'95*, pages 133–141, Fribourg, 1995.

[CS95] M-O. Cordier and P. Siegel. Prioritized transitions for updates. In *ECSQARU'95*, pages 142–151, Fribourg, Suisse, 1995.

[CT99] C. Cossart and C. Tessier. Filtering vs Revision and Update: Let us Debate! In *ECSQARU'99*, pages 116–127, 1999.

[FH99] N. Friedman and J.Y. Halpern. Modeling belief in dynamic systems. part ii: revision and update. *Journal of Artificial Intelligence Research*, 10:117–167, 1999.

[GA93] M.S. Grewal and A.P. Andrews. *Kalman Filtering - Theory and Practice.* Prentice Hall, 1993.

[HNSY94] T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.

[KNSS99] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. In LNCS vol. 1601, pages 75–95, March 1999.

[LC00] C. Largouet and M-O. Cordier. Timed Automata Model to Improve the Classification of a Sequence of Images. In *European Conference of Artificial Intelligence (ECAI)*, 2000.

[Yov97] S. Yovine. Kronos : A verification tool for real-time systems. *International Journal of Software Tools for Technology Transfer*, 1, 1997.