Active Sampling for Data Mining

Emanuele Olivetti and Paolo Avesani

ITC-iRST
Via Sommarive 14 - I-38050 Povo (TN) - Italy
{olivetti,avesani}@itc.it

Abstract. Data mining is a complex process that aims to derive an accurate predictive model starting from a collection of data. Traditional approaches assume that data are given in advance and their quality, size and structure are independent parameters. In this paper we argue that an extended vision of data mining should include the step of data acquisition as part of the overall process. Moreover the static view should be replaced by an evolving perspective that conceives the data mining as an iterative process where data acquisition and data analysis repeatedly follow each other.

A decision support tool based on data mining will have to be extended accordingly. Decision making will be concerned not only with a predictive purpose but also with a policy for a next data acquisition step. A successful data acquisition strategy will have to take into account both future model accuracy and the cost associated to the acquisition of each feature. To find a trade off between these two components is an open issue. A framework to focus this new challenging problem is proposed.

1 Introduction

Very often there are initiatives to provide inductive evidence as explanation of a complex phenomena although a collection of data is not available in advance. It is straightforward that in this context a data acquisition plan becomes a strategic preliminary or intermediate goal.

To arrange a data acquisition plan could be not trivial if the collection and the recording of information can not take advantage of electronic devices to automate such a process. Moreover the assumption that the effort spent to collect a vector of data is feature independent could be no more sustainable. For example in the agriculture domain a biological test to fill a feature that describes the presence of a particular pest could be really expensive.

The objective of a data acquisition plan is twofold: to increase the opportunity of a much more accurate model for the next step of data analysis and at the same time to lower the costs associated to a data acquisition plan.

It is to be remarked that in this work we assume that the space of features to be collected can change step by step.

This work aims to define a framework for this kind of challenge as a preliminary step towards the development of working solutions. Therefore this paper doesn't provide yet a solution to arrange successful data acquisition policies.

Let's start with the next section focusing our attention to the agriculture domain. It will explain the motivations of this work illustrating a research project in the area of

pest control management. Starting from this working scenario section 3 will introduce an intuitive definition of the objectives that arise from the previous motivations. A more formal statement of the framework will be provided in section 4 with an introduction of the basic definitions. Attention will be devoted to the specification of the evaluation process.

2 Working Scenario

The motivation of this work arises from a joint project with biologists in the agricultural domain: SMAP[1] (Scopazzi del Melo - Apple Proliferation). Apple Proliferation (AP) plant disease has been a remarkable spread in most of the fruit growing area in Trentino (northern Italy) and south-west Germany in the last 5 years. Apple growers are concerned about the spreading of the infection (due to a phytoplasma) because it leads to complete economic loss of production. At present no curative treatments are known for AP disease.

The main goal of the SMAP project is developing a model of disease, based on the evidence of observable data, to achieve a good comprehension of AP dissemination, evolution and regression. Of course the ultimate objective is to arrange an effective treatment to fight the disease.

The core problem of biologist is not the analysis of collected data, that is performed using well assessed and satisfactory techniques, but the main concern is the planning of new data collection campaign.

The scenario is the following. Biologists have collected in the past seasons a collection of data of potential interest in explaining the AP disease, for example: meteorological parameters, AP carriers presence, closeness to other specific cultivation or wood, geographical characterization etc. Of course such a collection of data allowed the biologists, through a data mining process, to obtain a model of AP disease. Although the model is prone to error the open issue is how to extend the set of feature to be collected to achieve a much more accurate AP comprehension.

Each summer the biologist have to arrange a data acquisition plan that provides the specification of data that will have to be collected in the current season. It is quite intuitive that a simple strategy that tends to over estimate the data that have to be acquired is not sustainable because the cost of data acquisition process, both in terms of money and human effort, may increase very quickly.

Let consider an example of the last season. The biologist advanced the hypothesis on two new features of potential usefulness: the acquisition of the damaged leaves color or the response of a biological test. The former has lower cost but is highly subjective (both at the level of damage recognition and color detection) while the latter is more safe but has high cost (both in term of money and time). Therefore the open issue was: do we have to plan the acquisition of damaged leaves color or do we have to perform a biological test? It is clearly a decision making process that up to now is not supported in the traditional data mining framework.

The basic intuition is that the biologist can be supported in a process of active sampling to minimize the acquisition effort and to estimate in advance the predictive power

[1] This work is funded by Fondo Progetti PAT. SMAP (Scopazzi del Melo - Apple Proliferation), art. 9, Legge Provinciale 3/2000, DGP n. 1060 dd. 04/05/01.

of candidate sample. Although it seems quite simple, the process of active sampling may increase in complexity whether we take into account all the possible alternatives.

Our contribution in the SMAP project is to design new techniques that may support the biologist in taking such a kind of decision. More in detail the research contribution was twofold: a framework to model the decision making process in the knowledge discovery' and the development of an innovative technique to support one of the issue related to planning of data acquisition planning.

This paper is devoted to the first of the two contribution while the second is illustrated in [14]. Therefore in the following we briefly sketch an overview of the decision making process in active sampling for data mining; after that we introduce a more formal framework as a premise for the design of effective supporting tools.

3 Data Mining and Decision Making

Data mining and decision making are two tasks closely related. Usually from the application point of view their relationship is clear: one is consequential of the other: first will data mining task, that allows to build a model, then the decision support task, that ve exploit such a model. The purpose of the inductive process is to support a deliberati process that should take advantage of a better knowledge of the pest.

Let summarize a sketch of the data mining process carried on very often.

1. collecting and preprocessing of the data;
2. data analysis and synthesis of a data model;
3. validation and deployment of the learned model.

to

The process above is usually accomplished under many tacit assumptions. Let's try get them explicit step by step. into

First of all the data collection is assumed to be not cost-sensitive: it is not taken as- account whether it is expensive or time consuming to acquire more data. From this ver- sumption immediately follows an other: the default data acquisition policy is to do o sampling. If any doubt applies, concerning with the potential usefulness of a given feature, its acquisition is promoted. The favorite policy is to over estimate the collection of the tures that have to be acquired because the filtering of the useless ones is postponed to following data analysis step . The assumption that the set of features to be acquired could be arbitrarily large is coupled with a similar assumption concerning with the size of is data collection. The size of the acquired data that have to support the inductive step supposed to be meaningful and representative even after the preprocessing and cleaning phases. to

A related assumption underlies the data analysis step. A quite common approach is manage the inductive process with a strategy that aims to detect and to discard the useless and features. It is the reason why the data mining is accomplished like a strictly forw or sequence of steps where the data analysis is not in charge to address the previous this further step of data acquisition. Although the active learning approach aims to cover the opportunity, the resulting acquisition policies don't affect the data dimensionality, i.e. the features to be acquired. A deeper discussion related to this aspect is postponed to section 5.

To consider the set of the features fixed it is equivalent to do a close world assumption. Every kind of inductive process can not assume any further features that don't belong to the acquired data. It is consistent with the hypothesis that the data analysis step doesn't have the opportunity to influence the data acquisition. The possibility to collect data respect with new features on demand is not allowed.

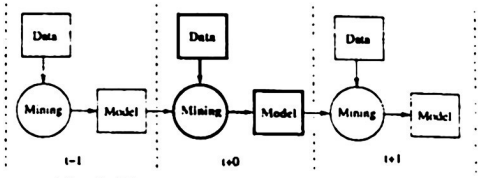

Fig. 1. The current model of data mining

Let's try to explain how these tacit assumptions affect the design of an architecture that combines data mining and decision support in an application environment as depicted in the SMAP projects. A new step takes place when decision making is combined with data mining: the design of an incoming data acquisition campaign. Data are no more a precondition but becomes an intermediate target of the whole discovery process. Not only the discovery of an accurate model of the environment behavior but also the discovery of the data design that better enables this achievement.

The revised architecture, as depicted in the figure 2, extends the previous schema introducing a "sampling policy". Active sampling in data mining becomes an open issue as important as to obtain an accurate model: what kind of input structure provided to the data analysis could allow to enhance the quality of the output?

Let's try to detail better the role of this new step through a sketch of an active sampling policy that we will have to implement:

1. Select a set of candidate new features to be acquired.
2. Preliminary active sampling of the new candidate features.
3. Filling of the most promising predictive features.

The first step is to filter among the huge amount of every possible feature a subset of candidates that will be evaluated through a preliminary assessment. The second point is to rank the subset of candidates doing a kind of subsampling: the goal is to achieve a preliminary estimate of a feature without to accomplish a complete acquisition of the data sample. The third and ultimate point is to complete the acquisition of the most promising features.

Before to discuss the open issues related to this new challenge we have to remark a further revision of the proposed architecture. Up to now another tacit assumption still holds: the data mining is conceived as a "one shot" process. It is not supposed to iterate the basic steps simply because the final result doesn't affect the data acquisition. At the same time a static view of the world brings to neglect the side effect that a deliberative actor has on the environment behavior using a predictive model.

improve model and to guide data collection at the next step. Let $X^* \subseteq X \setminus \overline{X}$ be the set of new instances and $X^{sub} \subseteq X^*$ a subsampling set over X^* that we use to decide whether a new feature $f_j \in F^*$ is good to improve the model or not. Let $\tilde{X} \subseteq X^*$ be the set of new instances we add after determining $\tilde{F}$ and c_f, the cost payed for the introduction of a new feature $f_j \in \tilde{F}$.

Given this framework we introduce the concept of hypothesis evolution of a model and in particular we define $H_k^{\overline{M}}$ as a partial hypothesis at step k of the model evolution over some given dataset $\overline{M}$, $\mathcal{H}$ as the final hypothesis, combination, in a specified way, of the all the $H_k^{\overline{M}}$. The space of all possible hypothesis $H_k^{\overline{M}}$ is said $\mathbb{H}$.

We introduce also the concept of error $\varepsilon : \mathbb{H} \times \mathbb{M} \rightarrow \mathcal{R}$ as an error function of an hypothesis over a given set (a test set) of features and instances.

Finally we introduce π_f, as the policy to promote a feature from the set of new features F^* (suggested by the domain expert) to the set of selected features $\tilde{F}$ to be added to the current features under control $\overline{F}$, and π_i, as the policy to promote an instance from the set X to the set $\tilde{X}$ of the instances to add at the next step.

At every time step of the process k, $\overline{X}_k$, $\overline{F}_k$, $\overline{M}_k$, $\tilde{X}_k$, $\tilde{F}_k$ and F_k^* are equivalent definitions as above.

4.2 Incremental Mining Process

The process to improve the model of the system is iterative and based on successive change of focus on domain, to progressively specialize the model on regions of the instance space where it performs badly. We start from an initial set of instances $\overline{X}_0$, features $\overline{F}_0$ and their values $\overline{M}_0$, to work out an hypothesis $H_0^{\overline{M}_0}$ and its error rate $\varepsilon(H_0^{\overline{M}_0}, T)$, namely the usual accuracy measure. Note that the accuracy error is estimated over a test set, different from the training set which is used to build H_0; in this sense $\overline{M}_0$ should be considered divided in two parts: $\overline{M}_0 = \overline{M}_0^{training} \cup \overline{M}_0^{test}$ and obviously $\overline{M}_0^{training} \cap \overline{M}_0^{test} = \emptyset$, giving $\varepsilon(H_0^{\overline{M}_0^{training}}, \overline{M}_0^{test})$.

Now we restrict to the region of the current instance space where the model performs badly (see later for a more precise definition), let's call it ϕ_0. To collect new information in order to increase accuracy of the model we can choose between 3 independent ways to do it and eventually mix them together; in figure 3 it is showed the matrix of values taken from features extracted on instances $\overline{M}_0$ and the possible ways to acquire new data.

As we can see we can work on 3 different portions of the matrix:

1. extracting new features on current instances
2. adding new instances and extract on some of the current features
3. adding new instances extracting on only new features

We can consider the 3 parts independently at a first glance (mixing will be addressed later) and refer generically to *adding information* not specifying the actual schema. Giving a set of features (old or new) $\tilde{F}_0$ we apply an active feature policy π_f to get a feature ranking specific for ϕ_0; here's the formal definition:

Definition 1. *¿From a subset of the feature space F (actually a subset of $F^* \cup \overline{F}$), a dataset $M \in \mathbb{M}$ and and an hypothesis H^M the* active feature policy π_f *determines the*

improve model and to guide data collection at the next step. Let $X^* \subseteq X \setminus \overline{X}$ be the set of new instances and $X^{sub} \subseteq X^*$ a subsampling set over X^* that we use to decide whether a new feature $f_j \in F^*$ is good to improve the model or not. Let $\tilde{X} \subseteq X^*$ be the set of new instances we add after determining $\tilde{F}$ and c_f, the cost payed for the introduction of a new feature $f_j \in \tilde{F}$.

Given this framework we introduce the concept of hypothesis evolution of a model and in particular we define $H_k^{\overline{M}}$ as a partial hypothesis at step k of the model evolution over some given dataset $\overline{M}$, $\mathcal{H}$ as the final hypothesis, combination, in a specified way, of the all the $H_k^{\overline{M}}$. The space of all possible hypothesis $H_k^{\overline{M}}$ is said $\mathbb{H}$.

We introduce also the concept of error $\varepsilon : \mathbb{H} \times \mathbb{M} \rightarrow \mathcal{R}$ as an error function of an hypothesis over a given set (a test set) of features and instances.

Finally we introduce π_f, as the policy to promote a feature from the set of new features F^* (suggested by the domain expert) to the set of selected features $\tilde{F}$ to be added to the current features under control $\overline{F}$, and π_i, as the policy to promote an instance from the set X to the set $\tilde{X}$ of the instances to add at the next step.

At every time step of the process k, $\overline{X}_k$, $\overline{F}_k$, $\overline{M}_k$, $\tilde{X}_k$, $\tilde{F}_k$ and F_k^* are equivalent definitions as above.

4.2 Incremental Mining Process

The process to improve the model of the system is iterative and based on successive change of focus on domain, to progressively specialize the model on regions of the instance space where it performs badly. We start from an initial set of instances $\overline{X}_0$, features $\overline{F}_0$ and their values $\overline{M}_0$, to work out an hypothesis $H_0^{\overline{M}_0}$ and its error rate $\varepsilon(H_0^{\overline{M}_0}, T)$, namely the usual accuracy measure. Note that the accuracy error is estimated over a test set, different from the training set which is used to build H_0; in this sense $\overline{M}_0$ should be considered divided in two parts: $\overline{M}_0 = \overline{M}_0^{training} \cup \overline{M}_0^{test}$ and obviously $\overline{M}_0^{training} \cap \overline{M}_0^{test} = \emptyset$, giving $\varepsilon(H_0^{\overline{M}_0^{training}}, \overline{M}_0^{test})$.

Now we restrict to the region of the current instance space where the model performs badly (see later for a more precise definition), let's call it ϕ_0. To collect new information in order to increase accuracy of the model we can choose between 3 independent ways to do it and eventually mix them together; in figure 3 it is showed the matrix of values taken from features extracted on instances $\overline{M}_0$ and the possible ways to acquire new data.

As we can see we can work on 3 different portions of the matrix:

1. extracting new features on current instances
2. adding new instances and extract on some of the current features
3. adding new instances extracting on only new features

We can consider the 3 parts independently at a first glance (mixing will be addressed later) and refer generically to *adding information* not specifying the actual schema. Giving a set of features (old or new) $\tilde{F}_0$ we apply an active feature policy π_f to get a feature ranking specific for ϕ_0; here's the formal definition:

Definition 1. *¿From a subset of the feature space F (actually a subset of $F^* \cup \overline{F}$), a dataset $M \in \mathbb{M}$ and and an hypothesis H^M the* active feature policy π_f *determines the*

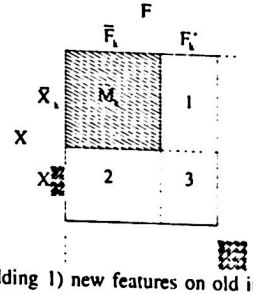

Fig. 3. Incremental acquisition adding 1) new features on old instances, 2) new instances on old features, 3) new instances on new features

subset of features that could be useful for the improvement of the model at the next mining step:

$$\pi_f : 2^F \times \mathbb{M} \times \mathbb{H} \rightarrow 2^F$$

Where there isn't previous information (i.e. section 3 of fig.3) for some feature, evaluation will be done over a subsampling X^{sub} of the instances where the feature is not extracted.

Definition 2. subsampling *is meant to be a preventive extraction of some possibly interesting features F_k^* over some instances X_k^{sub} to estimate the degree of interest of those features to reach the desired goal of better accuracy of the model. In our case subsampling is performed every time we need to get information from new features in order to make some decision about what to collect in the next round of the data acquisition process. Subsampling can be done with a budget limitation that can consider different costs of each feature.*

Applying budget criteria we can extract some useful features on some useful instances to get new information and build an improved hypothesis $H_1^{\overline{M}_1}$, where $\overline{M}_1 = \overline{M}_0 \cup \widetilde{M}_0$. The number of extractions will depend on the budget and the different cost of the features. We can apply this process iteratively to improve incrementally the model we are building. In the following you will find details and open issues about each part of this process.

At each step of the incremental mining process, when selecting new instances, a key point is to define the region of interest where to concentrate next feature extractions; we focus our attention where the model performs badly as stated in next definition.

Definition 3. *We define* difficult region $\phi_k^s \subset X$ *(at step k for strategy s) a subset of the instance space where the estimated performance of the model is under a certain threshold that is considered critical for the goal of good classification:*

$$\hat{\epsilon}(H_k, F(\phi_k^s)) < threshold$$

A practical way to implement can be a segmentation of the current feature space estimating the local current accuracy to each region. For example, In case of nominal valued/discrete valued features we can assign the local current accuracy to each different combination of feature values.

An analogous concept, to guide the feature extraction, is about the degree of information inside new data. Assume that all features are nominal valued (discrete without order); we define as *most informative region* the subset of S, characterized by specific values for already extracted features ($\alpha = (x_1, ..., x_n)$), in which there is the sample that most alters the current knowledge if added to the current dataset to train the model. This approach is described in [14], where alteration of knowledge is considered in terms of variation of estimated entropy of the system. In [14] the entire training phase is performed each time a new single values is added to the dataset.

Other definitions of these regions (difficult or most informative) can be made if they agree with observations explained above; particular definitions can be more more closely related to the particular problem under investigation so other ideas can be applied. In the following we will address as *difficult regions* all these possible definitions (difficult, most informative or others).

Fixing the definitions of these regions allows to the define active instance policy.

Definition 4. *¿From a particular instance in X, a dataset $M \in \mathbb{M}$ and an hypothesis H^M the* active instance policy π_i *determines if the instance is useful for the improvement of the model at the next mining step:*

$$\pi_i : X \times \mathbb{M} \times \mathbb{H} \rightarrow \{0, 1\}$$

Applying previous definitions we have:

$$x \in \phi_k^s \rightarrow \pi_i(x, M_k, H_k) = 1$$

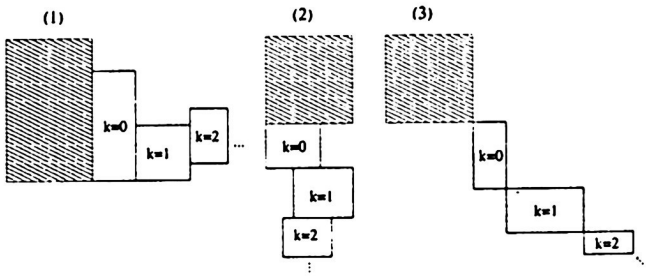

Fig. 4. Incremental acquisition strategies schema: (1) Strategy 1, (2) Strategy 2, (3) Strategy 3

4.3 Strategy 1: New Features on Old Instances

With this strategy we aim at increasing the information (see figure 4), to improve our model, looking for new interesting features to describe better current instances of the dataset where the current model accuracy is not good enough. We start looking for difficult regions (or most informative regions) ϕ_0^1 and concentrate next steps over it. We perform

a subsampling on these regions extracting some new interesting features $\widetilde{F}_0$ taken from the pool F_0^* suggested by domain experts during this iteration[2]. We can rank this candidate features $F_{i0}^* \in F_0^*$ according to relevance for classification of the subjects given the feature $\overline{F}_0$ and optimizing the cost for feature extraction. Relevance can be estimated building a specific hypothesis for this region with old extractions and new ones for each F_{i0} and estimating accuracy as ranking parameter. Estimation has to be evaluated over a proper test set, extracted separately during this phase. Given the ranking and according to a budget limitation (see later for economic details) we can start extracting selected features and build an improved local model better than $H_0^{\overline{M}_0}$ in these regions. The new hypothesis $H_1^{\overline{M}_1}$ can be built in this way: $H_1^{\overline{M}_1} = H_0^{\overline{M}_0} \oplus H_0^{\widetilde{M}_0}$, where $\oplus$ means for example that

$$H_1^{\overline{M}_1} = \begin{cases} H_0^{\widetilde{M}_0} \text{ in } \phi_0^1 \\ H_0^{\overline{M}_0} \text{ elsewhere} \end{cases}$$

Further steps following this strategy will lead to this recursive description:

$$H_{k+1}^{\overline{M}_{k+1}} = \begin{cases} H_k^{\widetilde{M}_k} \text{ in } \phi_k^1 \\ H_k^{\overline{M}_k} \text{ elsewhere} \end{cases}$$

and

$$\overline{M}_{k+1} = \overline{M}_k \cup \widetilde{M}_k$$
$$\overline{F}_{k+1} = \overline{F}_k \cup \widetilde{F}_k$$
$$\overline{X}_{k+1} = \overline{X}_k \cup \widetilde{X}_k$$

Other implementations are possible; an example of a possible implementation with details is [14].

4.4 Strategy 2: New Instances on Old Features

With this strategy we look for new interesting instances extracting them on some old features (see figure 4) considered more informative than others. At a given step of the iterative process we define the difficult/informative region to work on and perform feature ranking on known features; ranking is done according to relevance for classification as described before and with some limitation of the budget (see later). Now we can try to add instances in two ways: randomly or exploiting some control we have over some features.

A feature is *under control* if we can decide its value, on a certain instance or set of instances, a priori of the extraction. For example, in the case of apple trees, we can choose a priori to collect only trees growing at altitude between 500m and 600m, selecting which fields satisfies that constraint; so the altitude feature extracted on these trees will

[2] During different steps different features can be suggested by domain experts due to improvements and evolution of knowledge of the domain related or not to this process.As an example, in our application on apple diseases, new knowledge comes from biology research community each year.

be decided and known (in a certain range) in advance. An example of not *under control* feature, in the same context, is a laboratory test like Elisa: you have to collect leaves from trees and perform the chemical test to know each outcome, so it is not possible to have it in advance and collect only trees with certain Elisa characteristic.

If we have some features under control we can exploit the constraints given by the regions found above over these features; given the sub-ranking restricted to them we obtain a subset of constraints (with a ranking induced by feature ranking) that can be applied in instance selection. Given the instances, feature extraction can be performed over high ranking features as far as budget allows. If selected instances are less than budget can handle, less important constraints can be relaxed to increase instances number and avoid to exceed the budget.

A new hypothesis can now be built on the regions of interest, specific of that local area and the current model can be updated as it was for strategy 1:

$$H_{k+1}^{\overline{M}_{k+1}} = \begin{cases} H_k^{\widetilde{M}_k} \text{ in } \phi_k^2 \\ H_k^{\overline{M}_k} \text{ elsewhere} \end{cases}$$

4.5 Strategy 3: New Features on New Instances

The 3rd part of the matrix (see figure 3 and 4) can be filled in two ways, leading to two different situations:

1. if strategy 1 is performed in advance, at a certain step of the incremental process, we have new features to work for strategy 2; if we assume that strategy 2 is performed before 3 and after 1, we can define strategy 3 as part of strategy 2 if we accept to restrict only to the new features that were selected during strategy 1. In this case strategy 3 ends here.
2. if we assume that strategy 3 is independent from 1 and 2 (and that can be performed, for example, as first acquisition strategy), we can face it mixing what we said for strategy 1 and 2: with a subsampling phase we can estimate which new features can be useful for our goal (as in 1) and try to exploit controllable features among them to select new instances. In this case strategy 3 reduces to application of 1 and 2 (in this order) with the restriction to extract only new features for instances selected in phase 2.

The first case seems to be more interesting in general situations and lead to simplification of the whole process (3 remains inside 1+2). The second case is interesting when we can't collect some new features on old instances, because of some constraints (for example, time constraints as we have in an hypothetical feature "number of apples when tree was 2 years old": if we didn't collect it in the right moment, we can't have it after): in that case case previous considerations holds.

4.6 Mixing Strategies

As we can see from previous subsections each strategy can be chosen independently from others; but if we want to adopt a mix of them, there is a natural order that comes out, and

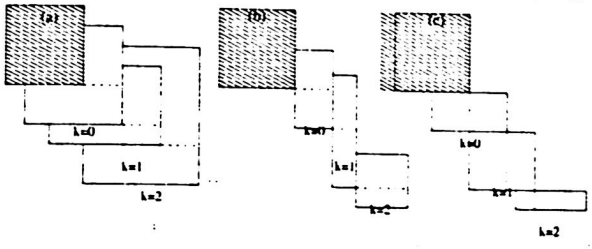

Fig. 5. Mixing strategies, examples of simpler combinations and coupling: (a) 1+2 , (b) 1+3, (c) 2+3. Dashed line divides different strategies applied in the same step

different schemas lead in some way to this ordering. So if we want to follow the mix of the three strategies we should start from 1 (acquire new features), then 2 (get new instances on features we know, old or from step 1) and then strategy 3 (extract new features on new instances).

If we want to restrict to simpler combinations of the strategies the following schema is suggested (see figure 5), remarking possible cooperation between them:

- strategies 1 and 2 : is equal as 1,2 and 3 in case there aren't constraints as described before (see 4.5). At each step strategy 2 extracts at least all new features introduced by 1 (eventually some old ones), that leads to a larger instance basis for its new partial model improvement .
- strategies 1 and 3 : at each acquisition step we have to explore new instances and skip old features. Strategy 3 can exploit results of strategy 1 extracting on new instances on the same features 1 added, gaining a larger basis of values to build its specialized model.
- strategies 2 and 3 : at each acquisition step we can't collect anything on old instances; strategy 2 and 3 can be coupled to work on the same instances to build an unique hypothesis, or can be independent and autonomous following directions mentioned above (see 4.4 and 4.5).

4.7 Handling Budget

We can suppose that each feature has a different cost due to its nature: in the domain of our application problem some features requires visual inspection of an expert, others chemical test on collected samples from trees (i.e. leaves) and sometimes some features are almost inexpensive at all as parameters like altitude of the tree that can be established a priori from maps. Different costs lead to inhomogeneous comparison during feature ranking; from a practical point of view it is necessary to do the same number of extractions from each features during a certain step of the process for a particular strategy, unless we can't build an hypothesis due to missing values (modulation of the number of the samples is not explored here and can be an interesting open issue). So each time we need to determine which subset of features to extract with some budget limitation, we have to find the number of instances to analyze (to determine how much of the budget can be

spent for each instance) and which is the subset of interesting features to get; the two problems seems to be weakly coupled but strongly dependent of the particular problem under consideration. The second one (how to choose the subset of features, given a *budget per instance*) can be approached in many ways, for example with a threshold of relevance to increase till reaching the budget (per instance), or mixing single cost and relevance to find the subset with highest sum of relevances and upper limit with budget (exploring the combinatorial problem).

This economic model is an open issue and an ongoing work of our research.

4.8 Evaluation Process

To evaluate the model produced and to compare different implementations of the decision policies, we can think about two alternative strategies : *on line* versus *off line*. They can be non mutual exclusive and can be referred as *run-time* or *incremental* evaluation and *a posteriori* evaluation.

The on line approach assumes that $\widehat{M_i}$ is not available but only $\overline{M_i}$. Due to the fact that the decision process determines at each step which is the best direction (features and instances) to take and there isn't any possibility to step back and change previous decisions; but avoiding to get back and try different alternatives doesn't allow to compare other policies, so other solutions seems to be taken.

The off line approach assumes a full availability of $\widehat{M_i}$. This is the case of pre-existing datasets to use for performance testing purpose. $\widehat{M_i}$ is first partitioned in two datasets (train and test), and the simulation of adaptive iterative process is performed on the train one. An initial set of instances and features is extracted and then a fixed number of sampling and filling steps are performed on available data exploiting the full information accessibility. We propose two possible alternatives about the target to reach in the off line approach to evaluate strategies:

- We fix the number of the mining steps and compare different solutions (policies) giving them homogeneous sampling: a fixed amount of instances is acquired at each step to maintain equally representative samples needed for future comparisons. Another assumption we need to be able to compare is to use the same modeling technique (induction trees, neural networks etc.) The target optimization to be compared between different solutions is the costs/accuracy of the resulting models. The different stress over costs or accuracy can modulate the target to satisfy application needs.
- We fix the amount of budget that every solution can spend for the learning phase, so the number of instances that can be taken at each step is constrained by the costs of features selected by the feature policy. After spending all the budget, maybe in a different number of steps, alternative solutions can be compared respect to the target of accuracy that should be maximized. With this method, different approaches that prefer feature oriented acquisitions (more features, less instances), can be fairly compared with approaches that prefer extensive samples with few features.

5 Related Work

Two main research areas dealing with our issue are *Feature Selection* and *Active Learning*. Feature Selection operates to look for the most relevant subset of features to focus the attention while modeling the system ([4],[6]). Active Learning looks for relevant examples (instances) to be labeled, during the learning process, discriminating them respect to useless examples, which cannot give new information to improve current hypotheses ([3], [11], [10]). Active Learning and Feature Selection reduce the number of instances and features for the learning phase and this is useful for 4 main reasons:

- learning is computational intensive, so reducing data speeds up the process and implies more efficiency.
- labeling cost is high, so asking the expert (teacher or oracle) is not always possible due to limited resources.
- as the learning process requires time, we can reach better accuracy more quickly selecting better examples and features ([2]).
- data collection is expensive even without labeling, as we have said in our working scenario: reducing the cost of data collection can be a main target in real life applications.

In [2] every Feature Selection strategy is classified with respect to the policy of adding/removing features, the organization of search in feature space, the policy to compare alternative features to get the more useful ones and the stopping criterion of the search. Our approach proposes a forward selection, because it starts with few features and tries to add new ones, with the possibility to remove some useless old features at each step; the organization of the search in feature space is guided by the domain expert and it is an open point where to propose new solutions; comparison between possible sets of features is performed by a *wrapper* method ([6], [4]) based on a subsampling of instance space and an induction algorithm to work hypothesis to estimate possible gain in accuracy; no stopping criterion is given because the process is meant to follow new needs of improved accuracy in time or changing of the environment.

As in Active Learning community, we specify which instances are useful to focus the attention on to cover problematic region of the domain, but we haven't full control over the instance space ([3]), so we can give only *feature constraints* to advice data collector. The way to translate feature constraints in specific instances in the domain is still an open problem that remains outside this framework nowadays.

Another work on dynamical aspects of data sampling focus the attention on the fairness of the database sample [5] and assume that the database exists and is accessible; we relax this hypothesis proposing another important aspect of data: the collection cost as the main limitation on sampling.

Because of the presence of costs in inductive learning we face to a broad literature on the subject ([12], [13]), but we restrict to cost of cases to acquire and policy to take care of it (as in [8]), shifting to a different direction of cost-sensitive learning, concerning instances and features, not well stressed until now.

6 Conclusions

In this paper we have presented the motivations that, moving from the real experience on agriculture domain, detect a new demand for extended decision support related to data mining. We focused the problem of data acquisition plan as a relevant problem where the environmental setting doesn't provide the opportunity to automate the collection of data. A framework to formally define the notion of data acquisition policy is proposed.

The next step will be concerned with the design and implementations of a working solutions as already started in [14]. Alternative hypotheses will be evaluated following the method illustrated in the last section. The datasets that we are processing in the SMAP project will enable a realistic experimentation of the relationship between model accuracy and acquisition costs. Moreover a direct comparison will be carried on between the innovative feature-based policy and the instance-based policies available in literature.

References

1. Blum, A.: Relevant examples and relevant features: Thoughts from computational learning theory. AAAI Fall Symposium on 'Relevance' (1994).
2. Blum, A., Langley, P.: Selection of Relevant Features and Examples in Machine Learning. Artificial Intelligence 97(1-2) (1997) 245-271.
3. Cohn, D.A., Atlas, L., Ladner, R.E.: Improving Generalization with Active Learning Machine Learning 15(2) (1994) 201–221.
4. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In: Proc. of International Conference on Machine Learning (1994) 121–129.
5. John, G.H., Langley, P.: Static Versus Dynamic Sampling for Data Mining. In Simoudis, E., Han, J., Fayyad, U.M. eds.: Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining, AAAI Press (1996) 367–370.
6. Kohavi, R., John, G.H.: Wrappers for Feature Subset Selection. Artificial Intelligence 97(1-2) (1997) 273–324.
7. Langley, P.: Selection of relevant features in machine learning. AAAI Fall Symposium on Relevance (1994) 140–144.
8. RayChaudhuri, T., Hamey, L.: An algorithm for active data collection for learning - feasibility study with neural network models. Report 95/173C, Department of Computing, School of MPCE, Macquarie University, Sydney, Australia (1995).
9. Saar-Tsechansky, M., Provost, F.J.: Active Sampling for Class Probability Estimation and Ranking. Machine Learning (2002).
10. Saar-Tsechansky, M., Provost, F.J.: Active Learning for Class Probability Estimation and Ranking. IJCAI (2001) 911–920.
11. Seung, H.S., Opper, M., Sompolinsky, H.: Query by Committee. Computational Learing Theory (1992) 287–294.
12. Turney, P.D.: Types of Cost in Inductive Concept Learning. In: Proc. of Workshop on Cost-Sensitive Learning at ICML (2000) 15–21.
13. Turney, P.D.: Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. Journal of Artificial Intelligence Research 2 (1995) 369–409.
14. Veeramachaneni, S., Avesani, P.: Active Sampling for Feature Selection. In: Proc. of IEEE International Conference on Data Mining (2003).