Bayesian Reasoning based on Predictive and Contextual Feature Selection

Vagan Terziyan

Industrial Ontologies Group, University of Jyvaskyla, Finland
e-mail: vagan@it.jyu.fi

Abstract

Bayesian Networks are proven to be a comprehensive model to describe causal relationships among domain attributes with probabilistic measure of appropriate conditional dependency. However, depending on task and context, many attributes of the model might not be relevant. If a Bayesian Network has been learned across multiple contexts then all uncovered conditional dependencies are averaged over all contexts and cannot guarantee high predictive accuracy when applied to a concrete case. We are considering a context as a set of contextual attributes, which are not directly effect probability distribution of the target attributes, but they effect on a "relevance" of the predictive attributes towards target attributes. In this paper we use the Bayesian Metanetwork vision to model such context-sensitive feature relevance. Such model assumes that the relevance of predictive attributes in a Bayesian network might be a random attribute itself and it provides a tool to reason based not only on probabilities of predictive attributes but also on their relevancies. According to this model, the evidence observed about contextual attributes is used to extract a relevant substructure from a Bayesian network model and then the predictive attributes evidence is used to reason about probability distribution of the target attribute in the extracted sub-network. We provide the basic architecture for such Bayesian Metanetwork, basic reasoning formalism and some examples. Distinguishing between relevant and irrelevant features of the domain objects is extremely important for the decision making, however another problem, to sort relevant features either to contextual or to predictive ones, is as much important too. In this paper we also consider three strategies of extracting context from relevant features, which are based on: *part_of* context, role-based context and interface-based context.

1. Introduction

A *Bayesian network* is a valuable tool for reasoning about probabilistic (causal) relationships [1]. A Bayesian network for a set of attributes $X = \{X1, \ldots, Xn\}$ is a directed acyclic graph with a network structure S that encodes a set of conditional independence assertions about attributes in X, and a set P of local probability distributions associated with each attribute [2].

An important task in learning Bayesian networks from data is model selection [3]. The models-candidates are evaluated according to measured degree to which a network structure fits the prior knowledge and data. Than the best structure is selected or several good structures are processed in model averaging. Each attribute in ordinary Bayesian network has the same status, so they are just combined in possible models-candidates to encode possible conditional dependencies however many modifications of Bayesian networks require distinguishing between attributes, e.g. as follows:

- *Target attribute*, which probability is being estimated based on set of evidence.
- *Predictive attribute*, which values being observed and which influences the probability distribution of the target attribute(s).
- *Contextual attribute*, which has not direct visible effect to target attributes but influences relevance of attributes in the predictive model. A contextual attribute can be conditionally dependent on some other contextual attribute.

Causal independence in a Bayesian network refers to the situation where multiple causes provided by predictive attributes contribute independently to a common effect on a target attribute. Context specific independence refers to such dependencies that depend on particular values of contextual attributes.

In [4], Butz exploited contextual independencies based on assumption that while a conditional independence must hold over all contexts, a contextual independence need only hold for one particular context. He shows how contextual independencies can be modeled using multiple Bayesian networks. Boutilier et al. [5] presents two algorithms to exploit context specific independence in a Bayesian network. The first one is network transformation and clustering. The other one is a form of cutset conditioning. This is done using reasoning by cases, where each case is a possible assignment to the variables in the cutset. The results of inference for all cases are combined to give the final answer to the query. Zhang [6] presents

a rule-based contextual variable elimination algorithm. Contextual variable elimination represents conditional probabilities in terms of generalized rules, which capture context specific independence in variables. Geiger and Heckerman [7] present another method to exploit context specific independence. With the notion of similarity networks, context specific independencies are made explicit in the graphical structure of a Bayesian network.

Bayesian Multi-nets were first introduced in ([8]) and then studied in ([9]) as a type of classifiers. A Bayesian multi-net is composed of the prior probability distribution of the class node and a *set* of local networks, each corresponding to a value that the class node can take. A Bayesian multi-net allows the relations among the features to be different – i.e., for different values the class node takes, the features can form different local networks with different structures. In a sense, the class node can be also viewed as a parent of all the feature nodes since each local network is associated with a value of the class node. A recursive Bayesian multinet was introduced by Pena et al [10] as a decision tree with component Bayesian networks at the leaves and was applied to a geographical data-clustering problem. The key idea was to decompose the learning Bayesian network problem into learning component networks from incomplete data.

In our previous work [11, 12], is the multilevel probabilistic meta-model (Bayesian Metanetwork), has been presented, which is an extension of traditional BN and modification of recursive multinets. It assumes that interoperability between component networks can be modeled by another BN. Bayesian Metanetwork is a set of BN, which are put on each other in such a way that conditional or unconditional probability distributions associated with nodes of every previous probabilistic network depend on probability distributions associated with nodes of the next network. We assume parameters (probability distributions) of a BN as random variables and allow conditional dependencies between these probabilities. Algorithms for learning Bayesian Metanetworks were discussed in [13].

As our main goal in this paper, we are presenting another view to the Bayesian Metanetwork by presenting the concept of attribute "relevance" as additional (to an attribute value probability) computational parameter of a Bayesian Network. Based on computed relevance only a specific sub-network from the whole Bayesian Network will be extracted and used for reasoning.

The rest of paper organized as follows. In Section 2 we first provide basic architecture of the Bayesian Metanetwork for managing Attribute Relevance. In Section 3 we provide the reasoning formalism and few examples. Very general concept of a Relevance Bayesian Metanetwork is given in Section 4. This paper is an extended version of [18] invited to be submitted to this special issue. Section 5 provides additional discussion comparably to [18] and related to strategies of contextual features selections for Bayesian Metanetwork reasoning. We conclude in Section 6.

2. Bayesian Metanetwork for Managing Attributes' Relevance

Relevance is a property of an attribute as a whole, not a property of certain values of an attribute (see Fig. 1). This makes a difference between relevance and probability, because the last one has as many values as an attribute itself. Another words, when we say probability, we mean probability of the value of the attribute, when we say relevance, we mean relevance (*probability to be included to the model*) of the attribute as whole.

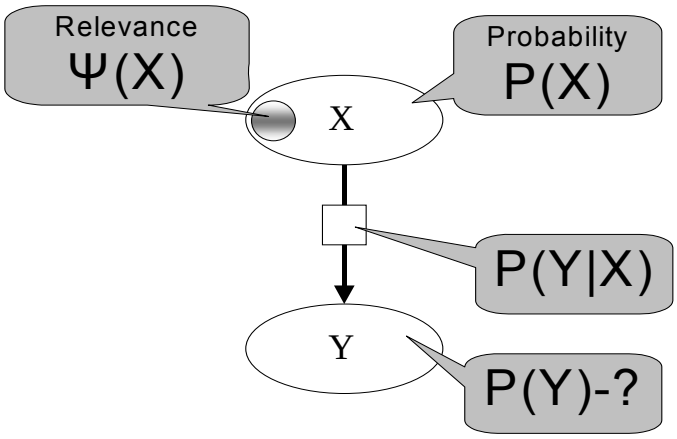


Fig. 1. The relevance of an attribute in a Bayesian Network

Bayesian Network in Fig. 1 actually includes two following subnetworks (see Fig. 2), which illustrate the definition of a “relevance”.

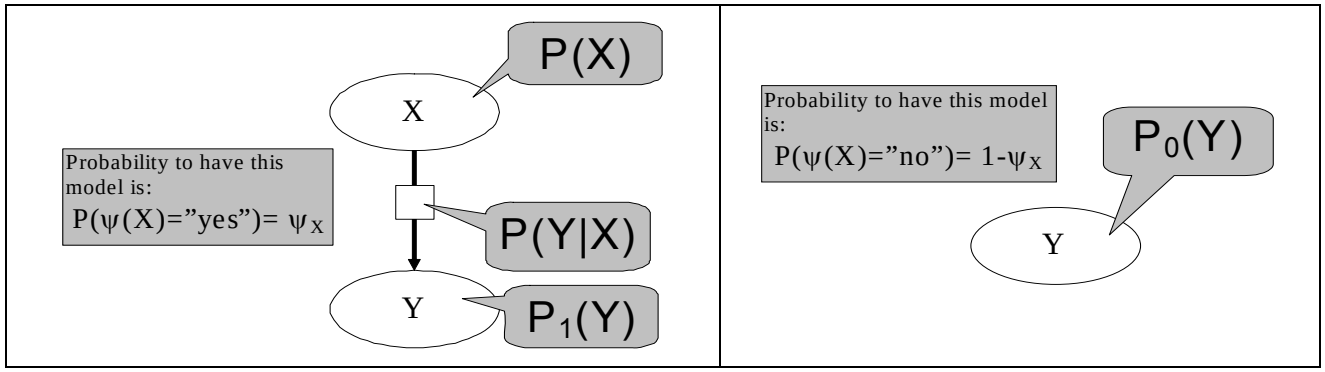


Fig. 2. Two valid subnetworks based on the relevance of an attribute

In the network from Fig. 1 and Fig. 2 we have:

Attributes: predictive attribute X with values $\{x_1,x_2,\ldots,x_{nx}\}$, target attribute Y with values $\{y_1,y_2,\ldots,y_{ny}\}$.
Probability distributions of the attributes: P(X), P(Y|X).
Posteriori probability distributions of the target attribute: $P_1(Y)$ and $P_0(Y)$ for the two cases (X – relevant or X irrelevant) of the valid subnetwork respectively.

Relevance predicate:
ψ(X) = “yes”, if parameter X is relevant;
ψ(X) = “no”, if parameter X is not relevant.

Relevance value: $\psi_X = P(\psi(X) = \text{“yes”})$.

Let’s estimate P(Y) based on Bayesian reasoning:

$$P(Y) = \psi_X \cdot P_1(Y) + (1-\psi_X) \cdot P_0(Y), \quad (1)$$

where:

$$P_1(Y) = \sum_{i=1}^{nx} P(Y \mid X = x_i) \cdot P(X = x_i). \quad (2)$$

$P_0(Y)$ can be calculated based on P(Y|X) knowing that in that case Y is not depending on X, because X is considered as not relevant:

$$P_0(Y) = \frac{1}{nx} \sum_{i=1}^{nx} P(Y \mid X = x_i). \quad (3)$$

Substituting (2) and (3) to (1) we obtain:

$$P(Y) = \psi_X \cdot \sum_{i=1}^{nx} [P(Y \mid X = x_i) \cdot P(X = x_i)] + (1-\psi_X) \cdot \frac{1}{nx} \cdot \sum_{i=1}^{nx} P(Y \mid X = x_i) =$$

$$= \sum_{i=1}^{nx} P(Y \mid X = x_i) \cdot [\psi_X \cdot P(X = x_i) + \frac{(1-\psi_X)}{nx}],$$

which is in a more compact form is:

$$P(Y) = \frac{1}{nx} \cdot \sum_X P(Y \mid X) \cdot [nx \cdot \psi_X \cdot P(X) + (1 - \psi_X)]. \quad (4)$$

Consider example, where the attribute X will be “state of whether” and attribute Y, which is influenced by X, will be “state of mood”. Let the values of the attributes and appropriate prior probabilities will be as follows:

X (“state of weather”) ={“sunny”, “overcast”, “rain”}

P(X=”sunny”) = 0.4;
P(X=”overcast”) = 0.5;
P(X=”rain”) = 0.1;

Y (“state of mood”) ={“good”, “bad”};

P(Y=”good” | X=”sunny”) = 0.7;
P(Y=”good” | X=”overcast”) = 0.5;
P(Y=”good” | X=”rain”) = 0.2;

Let conditional probability, which links X and Y will be as follows:

P(Y=”bad” | X=”sunny”) = 0.3;
P(Y=”bad” | X=”overcast”) = 0.5;
P(Y=”bad” | X=”rain”) = 0.8;

Assume the value of relevance for the attribute X is known and equal: ψ_X=0.6.

Now, according to (4) we have:

$$P(Y = "good") = \frac{1}{3} \cdot \{P(Y = "good" \mid X = "sunny") \cdot [1.8 \cdot P(X = "sunny") + 0.4] +$$
$$+ P(Y = "good" \mid X = "overcast") \cdot [1.8 \cdot P(X = "overcast") + 0.4] +$$
$$+ P(Y = "good" \mid X = "rain") \cdot [1.8 \cdot P(X = "rain") + 0.4]\} = \frac{1}{3} \cdot [0.7 \cdot (1.8 \cdot 0.4 + 0.4) +$$
$$+ 0.5 \cdot (1.8 \cdot 0.5 + 0.4) + 0.2 \cdot (1.8 \cdot 0.1 + 0.4)] = \frac{1}{3} \cdot [0.7 \cdot 1.12 + 0.5 \cdot 1.3 + 0.2 \cdot 0.58] =$$
$$= \frac{1}{3} \cdot [0.784 + 0.65 + 0.116] = 0.517;$$
$$P(Y = "bad") = \frac{1}{3} \cdot \{P(Y = "bad" \mid X = "sunny") \cdot [1.8 \cdot P(X = "sunny") + 0.4] +$$
$$+ P(Y = "bad" \mid X = "overcast") \cdot [1.8 \cdot P(X = "overcast") + 0.4] +$$
$$+ P(Y = "bad" \mid X = "rain") \cdot [1.8 \cdot P(X = "rain") + 0.4]\} = \frac{1}{3} \cdot [0.3 \cdot (1.8 \cdot 0.4 + 0.4) +$$
$$+ 0.5 \cdot (1.8 \cdot 0.5 + 0.4) + 0.8 \cdot (1.8 \cdot 0.1 + 0.4)] = \frac{1}{3} \cdot [0.3 \cdot 1.12 + 0.5 \cdot 1.3 + 0.8 \cdot 0.58] =$$
$$= \frac{1}{3} \cdot [0.336 + 0.65 + 0.464] = 0.483.$$

One can also notice that these values belong to the intervals created by the two extreme cases, when parameter X is not relevant at all or it is fully relevant:

$$0.467 \approx P_0(Y = "good")|_{\psi_X=0} < P(Y = "good")|_{\psi_X=0.6} < P_1(Y = "good")|_{\psi_X=1} = 0.55;$$

$$0.45 = P_1(Y = "bad")|_{\psi_X=1} < P(Y = "bad")|_{\psi_X=0.6} < P_0(Y = "bad")|_{\psi_X=0} \approx 0.533.$$

3. General Formalism and Samples

More complicated case is the management of relevance in the following situation (see Fig. 3):

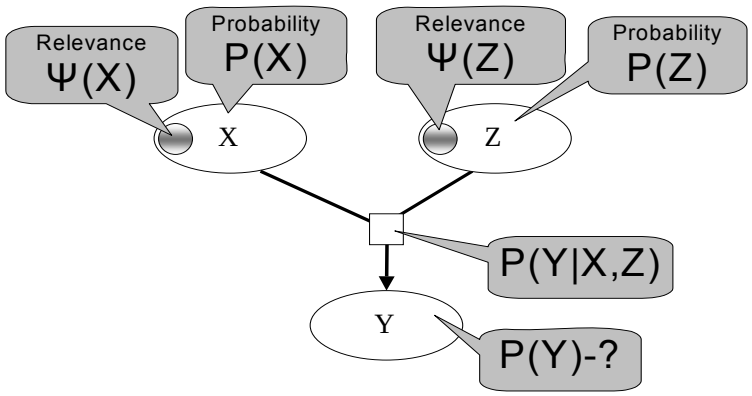


Fig. 3. Relevance management with two predictive attributes

Here we have 4 following subnetworks depending on the relevance (see Fig. 4).

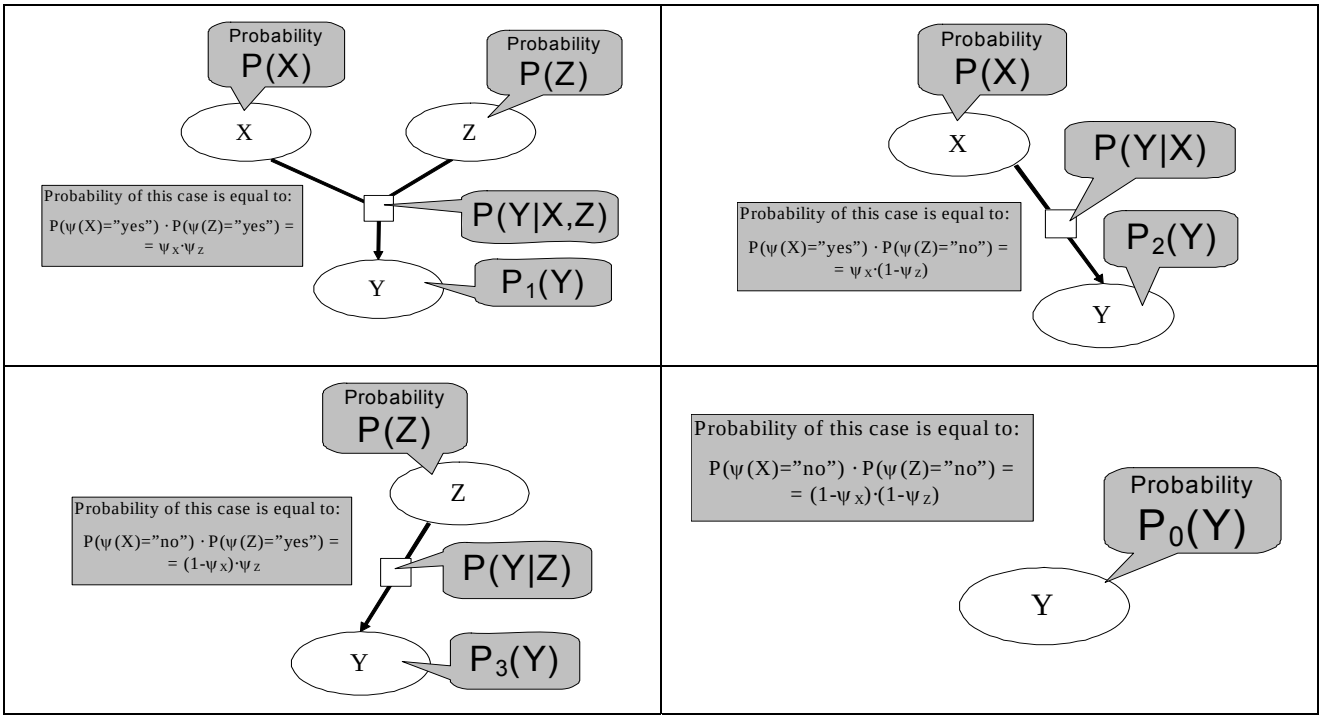


Fig. 4. Subnetworks for the case with two predictive attributes

In the above case we have:

Attributes:
X with values $\{x_1, x_2, \ldots, x_{nx}\}$;
Z with values $\{z_1, z_2, \ldots, z_{nz}\}$;
Y with values $\{y_1, y_2, \ldots, y_{ny}\}$.

Probabilities: P(X), P(Z), P(Y|X,Z).

Relevance predicate:
$\psi(X)$ = “yes”, if parameter X is relevant;
$\psi(X)$ = “no”, if parameter X is not relevant.

Relevancies:
$\psi_X = P(\psi(X) = \text{“yes”})$;
$\psi_Z = P(\psi(Z) = \text{“yes”})$.

Let’s estimate P(Y):

$$P(Y) = \psi_X \cdot \psi_Z \cdot P_1(Y) + \psi_X \cdot (1-\psi_Z) \cdot P_2(Y) + \\ + (1-\psi_X) \cdot \psi_Z \cdot P_3(Y) + (1-\psi_X) \cdot (1-\psi_Z) \cdot P_0(Y) \quad (5)$$

$$P_1(Y) = \sum_{i=1}^{nx} \sum_{k=1}^{nz} P(Y \mid X = x_i, Z = z_k) \cdot P(X = x_i) \cdot P(Z = z_k). \quad (6)$$

$$P_2(Y) = \sum_{i=1}^{nx} P(Y \mid X = x_i) \cdot P(X = x_i). \quad (7)$$

$$P_3(Y) = \sum_{k=1}^{nz} P(Y \mid Z = z_k) \cdot P(Z = z_k). \quad (8)$$

Now we should extract P(Y|X), P(Y|Z), $P_0(Y)$ from P(Y|X,Z) and P(Y|Z) from P(Y|X,Z), which is:

$$P(Y \mid X) = \frac{1}{nz} \cdot \sum_{k=1}^{nz} P(Y \mid X, Z = z_k), \quad (9)$$

$$P(Y \mid Z) = \frac{1}{nx} \cdot \sum_{i=1}^{nx} P(Y \mid X = x_i, Z), \quad (10)$$

$$P_0(Y) = \frac{1}{nx \cdot nz} \cdot \sum_{i=1}^{nx} \sum_{k=1}^{nz} P(Y \mid X = x_i, Z = z_k), \quad (11)$$

We can rewrite (7) using (9) as follows:

$$P_2(Y) = \frac{1}{nz} \cdot \sum_{i=1}^{nx} \sum_{k=1}^{nz} P(Y \mid X = x_i, Z = z_k) \cdot P(X = x_i). \quad (12)$$

We can also rewrite (8) using (10) as follows:

$$P_3(Y) = \frac{1}{nx} \cdot \sum_{i=1}^{nx} \sum_{k=1}^{nz} P(Y \mid X = x_i, Z = z_k) \cdot P(Z = z_k). \quad (13)$$

Finally we can substitute (6), (11), (12), (13) to (5):

$$P(Y) = \psi_X \cdot \psi_Z \cdot \sum_{i=1}^{nx}\sum_{k=i}^{nz} P(Y \mid X = x_i, Z = z_k) \cdot P(X = x_i) \cdot P(Z = z_k) +$$
$$+ \psi_X \cdot (1-\psi_Z) \cdot \frac{1}{nz} \cdot \sum_{i=1}^{nx}\sum_{k=i}^{nz} P(Y \mid X = x_i, Z = z_k) \cdot P(X = x_i) +$$
$$+ (1-\psi_X) \cdot \psi_Z \cdot \frac{1}{nx} \cdot \sum_{i=1}^{nx}\sum_{k=i}^{nz} P(Y \mid X = x_i, Z = z_k) \cdot P(Z = z_k) +$$
$$+ (1-\psi_X) \cdot (1-\psi_Z) \cdot \frac{1}{nx \cdot nz} \cdot \sum_{i=1}^{nx}\sum_{k=i}^{nz} P(Y \mid X = x_i, Z = z_k),$$

which is in a more compact form:

$$P(Y) = \frac{1}{nx \cdot nz} \cdot \sum_{X}\sum_{Z} P(Y \mid X, Z) \cdot [nx \cdot nz \cdot \psi_X \cdot \psi_Z \cdot P(X) \cdot P(Z) +$$
$$+ nx \cdot \psi_X \cdot (1-\psi_Z) \cdot P(X) + nz \cdot (1-\psi_X) \cdot \psi_Z \cdot P(Z) + (1-\psi_X) \cdot (1-\psi_Z)]. \quad (14)$$

Consider example:

Let we have the following set of data:

X	**Z**	**Y**
Sunny	Alone	Good
Overcast	With girlfriend	Bad
Overcast	With dog	Bad
Sunny	With dog	Good
Sunny	Alone	Good
Overcast	Alone	Bad
Rain	With girlfriend	Bad
Sunny	With dog	Good
Overcast	With dog	Bad
Sunny	With girlfriend	Bad
Overcast	With girlfriend	Good
Overcast	Alone	Bad
Overcast	With dog	Bad
Sunny	With girlfriend	Good
Overcast	With dog	Bad
Overcast	Alone	Bad
Sunny	Alone	Bad
Sunny	With dog	Bad
Rain	With girlfriend	Good
Overcast	With dog	Good

X (“state of weather”) ={“sunny”, “overcast”, “rain”}
Z (“companion”) ={“alone”, “girlfriend”, “dog”}
Y (“state of mood”) ={“good”, “bad”};

P(X=”sunny”) = 0.4;
P(X=”overcast”) = 0.5;
P(X=”rain”) = 0.1;

P(Z=”alone”) = 0.3;
P(Z=”girlfriend”) = 0.3;
P(Z=”dog”) = 0.4;

P(Y=”good”| X, Z)	Z = “alone”	Z=”girlfriend”	Z=”dog”
X = “sunny”	0.667	0.5	0.667
X = “overcast”	0	0.5	0.25
X = “rain”	0	0.5	0

P(Y=”bad”| X, Z)	Z = “alone”	Z=”girlfriend”	Z=”dog”
X = “sunny”	0.333	0.5	0.333
X = “overcast”	1	0.5	0.75
X = “rain”	1	0.5	1

According to (9):

P(Y | X)	X = “sunny”	X = “overcast”	X = “rain”
Y = “good”	0.611	0.25	0.167
Y = “bad”	0.389	0.75	0.833

According to (10):

P(Y | Z)	Z = “alone”	Z = “girlfriend”	Z = “dog”
Y = “good”	0.222	0.5	0.306
Y = “bad”	0.778	0.5	0.694

According to (11):

P_0(Y=”good”) = 0.3426
P_0(Y=”bad”) = 0.6574
Assume that relevancies of our parameters are as follows:
$\psi_X = 0.8$, $\psi_Z = 0.5$

Now we can estimate P(Y) based on (14):

$$P(Y = "good") = \frac{1}{9} \cdot [0.667 \cdot (3.6 \cdot 0.4 \cdot 0.3 + 1.2 \cdot 0.4 + 0.3 \cdot 0.3 + 0.1) +$$
$$+ 0.5 \cdot (3.6 \cdot 0.4 \cdot 0.3 + 1.2 \cdot 0.4 + 0.3 \cdot 0.3 + 0.1) + 0.667 \cdot (3.6 \cdot 0.4 \cdot 0.4 + 1.2 \cdot 0.4 + 0.3 \cdot 0.4 + 0.1) +$$
$$+ 0.5 \cdot (3.6 \cdot 0.5 \cdot 0.3 + 1.2 \cdot 0.5 + 0.3 \cdot 0.3 + 0.1) + 0.25 \cdot (3.6 \cdot 0.5 \cdot 0.4 + 1.2 \cdot 0.5 + 0.3 \cdot 0.4 + 0.1) +$$
$$+ 0.5 \cdot (3.6 \cdot 0.1 \cdot 0.3 + 1.2 \cdot 0.1 + 0.3 \cdot 0.3 + 0.1)] \approx 0.3773.$$

$$P(Y = "bad") \approx 0.6227.$$

One can also notice that these values belong to the interval created by the two extreme cases, when parameters are not relevant at all or they are fully relevant:

$$0.3426 \approx P_0(Y = "good")|_{\psi_X=0,\psi_Z=0} < P(Y = "good")|_{\psi_X=0.8,\psi_Z=0.5} < P_1(Y = "good")|_{\psi_X=1,\psi_Z=1} \approx 0.3867;$$

$$0.6133 \approx P_1(Y = "bad")|_{\psi_X=1,\psi_Z=1} < P(Y = "bad")|_{\psi_X=0.8,\psi_Z=0.5} < P_0(Y = "bad")|_{\psi_X=0,\psi_Z=0} \approx 0.6574.$$

Consider the general case of managing relevance (Fig. 5):

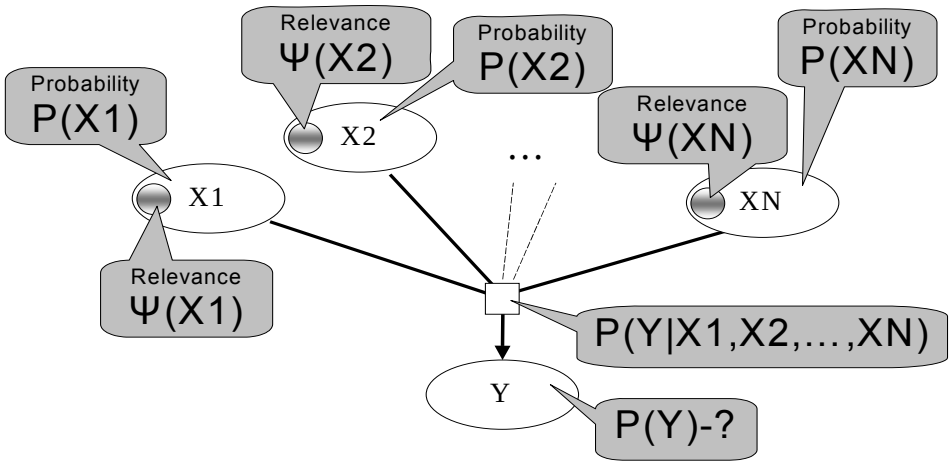


Fig. 5. General case of relevance management

In this case we have the following:

Predictive attributes:

X1 with values $\{x1_1, x1_2, \ldots, x1_{nx1}\}$;
X2 with values $\{x2_1, x2_2, \ldots, x2_{nx2}\}$;
…
XN with values $\{xn_1, xn_2, \ldots, xn_{nxn}\}$;

Target attribute:

Y with values $\{y_1, y_2, \ldots, y_{ny}\}$.

Probabilities: P(X1), P(X2),…, P(XN); P(Y|X1,X2,…,XN).

Relevancies:
ψ_{X1} = P(ψ(X1) = “yes”);
ψ_{X2} = P(ψ(X2) = “yes”);
…
ψ_{XN} = P(ψ(XN) = “yes”);

Let’s estimate P(Y).

Generalizing (4) and (14) to the case of N predictive variables we finally obtain:

$$P(Y) = \frac{1}{\prod_{s=1}^{N} nxs} \cdot \sum_{X1} \sum_{X2} \ldots \sum_{XN} [P(Y \mid X1, X2, \ldots XN) \cdot \prod_{\forall r(\psi(Xr)="yes")} nxr \cdot \psi_{Xr} \cdot P(Xr) \cdot \prod_{\forall q(\psi(Xq)="no")} (1 - \psi_{Xq})].$$

4. A Relevance Metanetwork

Relevance Bayesian Metanetwork can be defined on a given predictive probabilistic network as it shown in Fig. 6. It encodes the conditional dependencies over the relevancies. Relevance metanetwork contains prior relevancies and conditional relevancies. Considering such definition of relevance metanetwork over the predictive network it is clear that the strict correspondence between nodes of both network exists but the arcs do not need to correspond (as shown on Fig. 6). It means that relevancies of two variables can be dependent, although their values are conditionally independent and vice versa (as shown on Fig. 7). So, the topologies of the networks are different in general case.

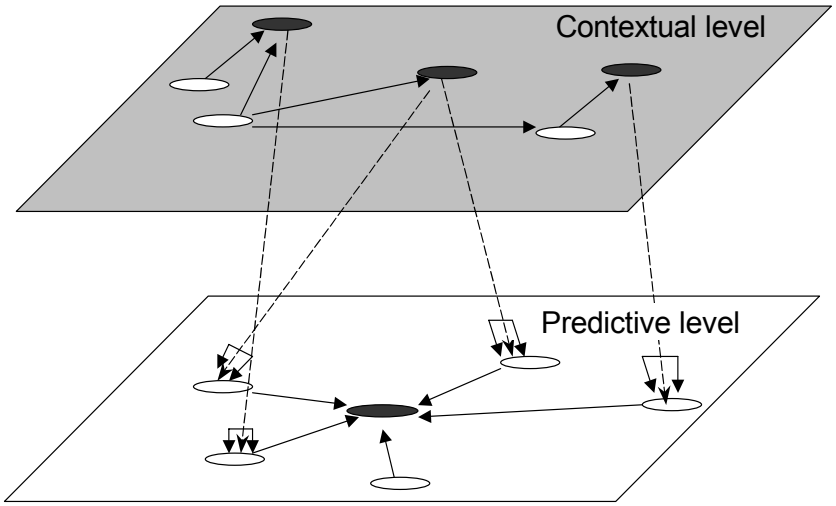


Fig. 6. Relevance network defined over the predictive network

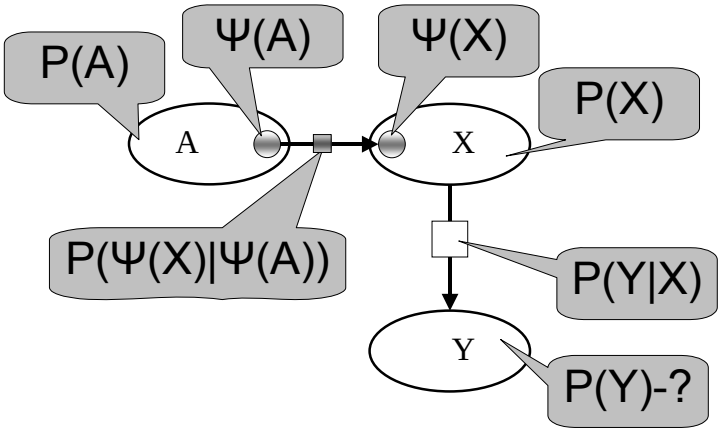


Fig. 7. Architecture of a simple relevance metanetwork

In a relevance network the relevancies are considered as random variables between which the conditional dependencies can be learned. For example in Fig. 7, the probability of target attribute Y can be computed as follows:

$$P(Y) = \frac{1}{nx} \cdot \sum_{X} \{P(Y \mid X) \cdot [nx \cdot P(X) \cdot$$

$$\cdot \sum_{\psi_A} P(\psi_X \mid \psi_A) \cdot P(\psi_A) + (1 - \psi_X)]\}.$$

More complicated example of a Bayesian metanetwork completed from predictive and relevance networks is shown in Fig. 8. The graph of the Metanetwork in the figure consists of two subgraphs (a) predictive network layer and (b) relevance network layer. The challenge here is that the relevance network

subgraph models the relevance conditional dependency and in the same time the posteriori relevance values calculated with this graph effect the calculations at the basic predictive subgraph as it was shown above.

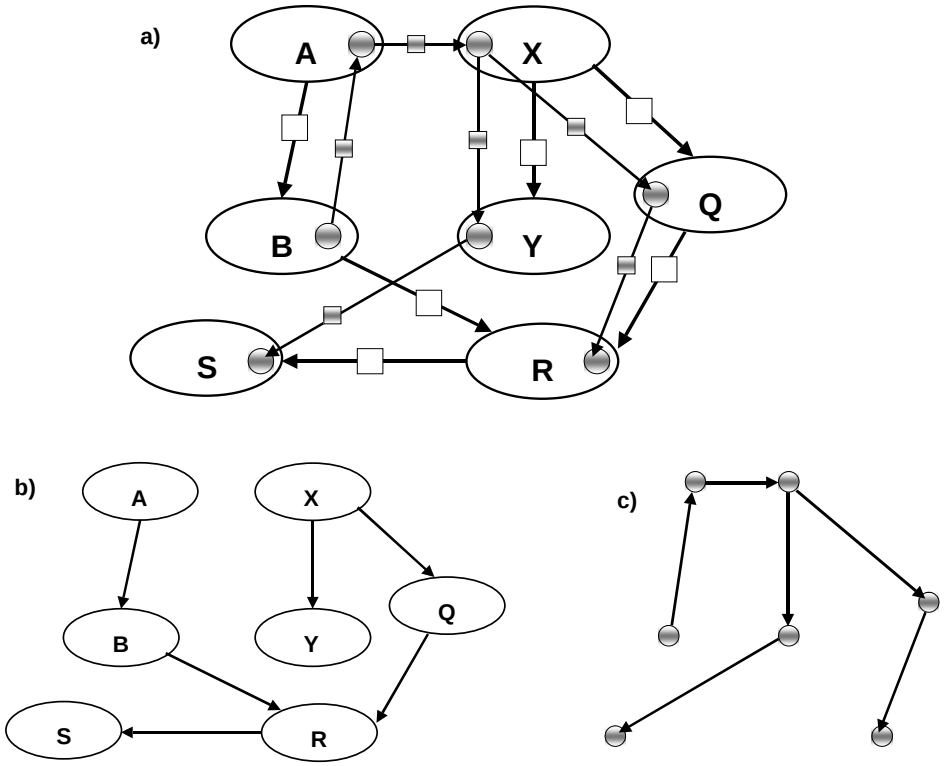


Fig. 8. Example of Bayesian metanetwork (a), consisting of the predictive network level (b) and relevance network level (c). Predictive and relevance networks have corresponding nodes, but different topologies.

5. Multilevel Context Extraction for Bayesian Metanetworks

Distinguishing between relevant and irrelevant features of the domain objects is, of course, extremely important for the decision making within that domain. However another problem, to sort relevant features either to contextual or to predictive ones, is as much important too. As we can see from e.g. Bayesian Metanetworks above, contextual and predictive features have different roles in the model and present on different levels of its organization.

The theories of context according to [14] can be divided into two general types: the first, which sees context as a way of partitioning a global model of the world into smaller and simpler pieces; the second, which sees context as a local theory of the world in a network of relations with other local theories, can be considered as more general than the first one. On the other hand, contexts can be considered as local (i.e. *not shared*) models that encode a party's subjective view of a domain [15]. This makes contexts comparable and in some sense opposite to ontologies, which are considered as *shared* models of some domain that encode a view which is common to a set of different parties [16]. Contexts and ontologies have both strengths and weaknesses. It was argued in [17] that the strengths of ontologies are the weaknesses of contexts and vice versa. In [17] the attempt was made to contextualize the ontologies by acquiring certain useful properties that a pure shared approach cannot provide. The result is *Context OWL (C-OWL)*, a language whose syntax and semantics have been obtained by extending the OWL to allow for the representation of contextual ontologies.

The above definitions are giving some hints on how to split the domain description (without complex mathematical processing) to predictive and contextual features, assuming that the goal is to enable reliable decision making based on Bayesian Metanetwork within that domain. In this chapter we consider three strategies of extracting context from relevant features, which are based on: *part_of* context, role-based context and interface-based context.

5.1. *Part_of* context extraction

It is known that it is more reliable to make decisions concerning any domain object if to take into account the environment within which this object is placed. For example in industrial applications related to condition monitoring, remote diagnostics, predictive maintenance, etc., it is really important to sense not only parameters of the machine (device) in question but also to measure the environmental conditions in which this machine is operating (See Fig. 9).

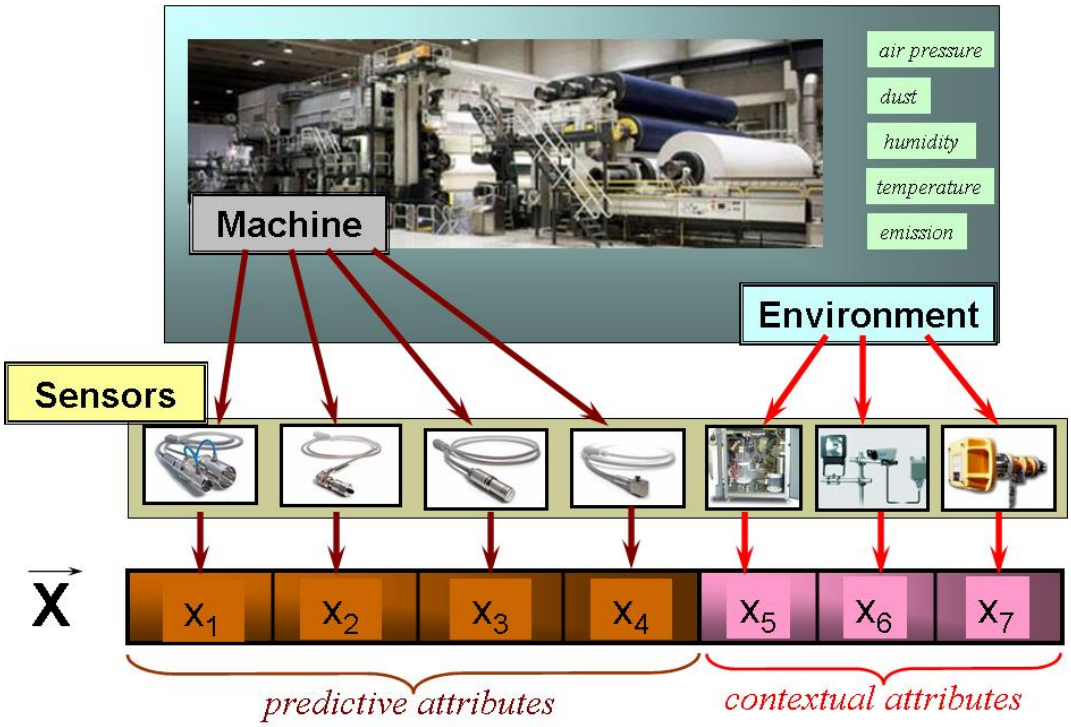


Fig. 9. To make diagnostics or to predict performance of some industrial machine it is reasonable to collect both: parameters measured directly from the machine and also parameters of the working environment of the machine.

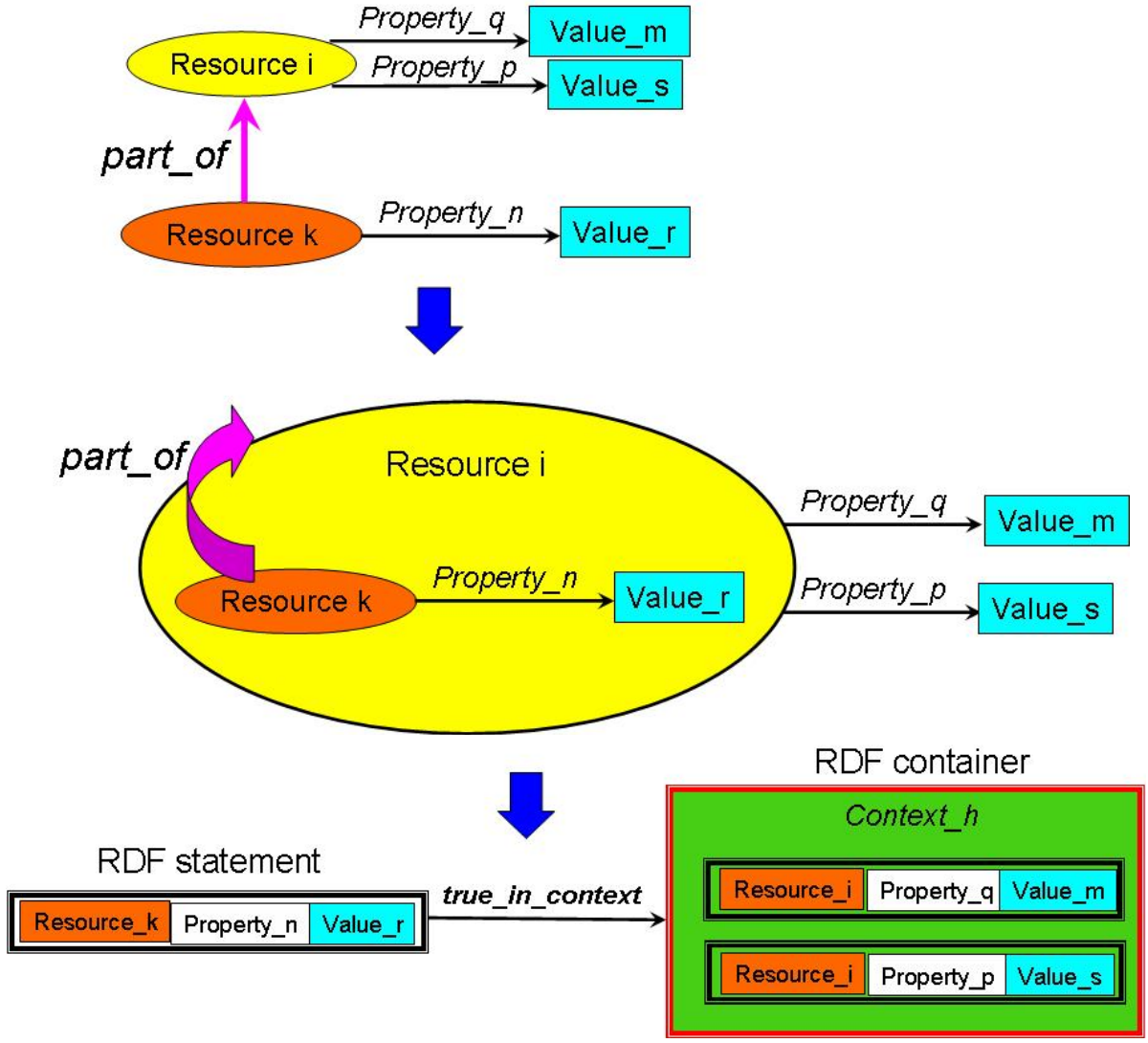
The attributes of the object and the attributes of its environment have different role in decision making process. If the first ones usually directly affect on the outcome (diagnosis, prediction, etc) and can be called "predictive" attributes, but alternatively the second ones most likely affect on the choice of right decision model for the diagnostics or prediction and can be called "contextual" attributes.

In general, the environment for any domain object is one or several other objects, which include this domain object as their part. For example, a department has some faculty as an environment, a wheel has some car as an environment, an arm has some body as an environment, player Andriy Shevchenko has "Chelsea" football club as an environment, etc.

The idea of the *part_of* context extraction is based on known hierarchy of the nested domain objects. If object A is part of object B (i.e. connected with *part_of* relation on a semantic network), then all predictive attributes of object B will be contextual attributes for object A. This is illustrated in Fig. 10, where a sample of domain model represented by RDF[1]-based semantic network is shown. Also it is shown nested view to *part_of* relation, which is also often used to visualize nested hierarchies. Using terminology of Semantic Web[2], in this example we have two resources: (a) Resource *k*, which is part of Resource *i* , has two datatype properties (property *q* with value *m* and property *p* with value *s*), and (b) Resource *i* itself with property *n* with value *r*. Actually we have four RDF statements: two about resource *k* based on two properties *part_of* and *property_n* , and two about resource *i* based on two properties *property_q* and *property_p*. In [19] the extension of RDF called CDF (Context Description Framework) is considered that allows making RDF statements in a context of some other RDF statements using for that *true_in_context* property for RDF statements (a kind of reification), and the value of this property is generally a contained of RDF statements. The CDF graph for the example above in also presented in Fig. 10. In the table from Fig.10 one can see a

[1] http://www.w3.org/RDF/
[2] http://www.w3.org/2001/sw/

separation between predictive and contextual features of the Resource *k*, which is based on *part_of* relation. Thus possible Bayesian Metanetwork to model such sample will place predictive attributes to predictive level of the network and the contextual features to the contextual level (see Section 4).



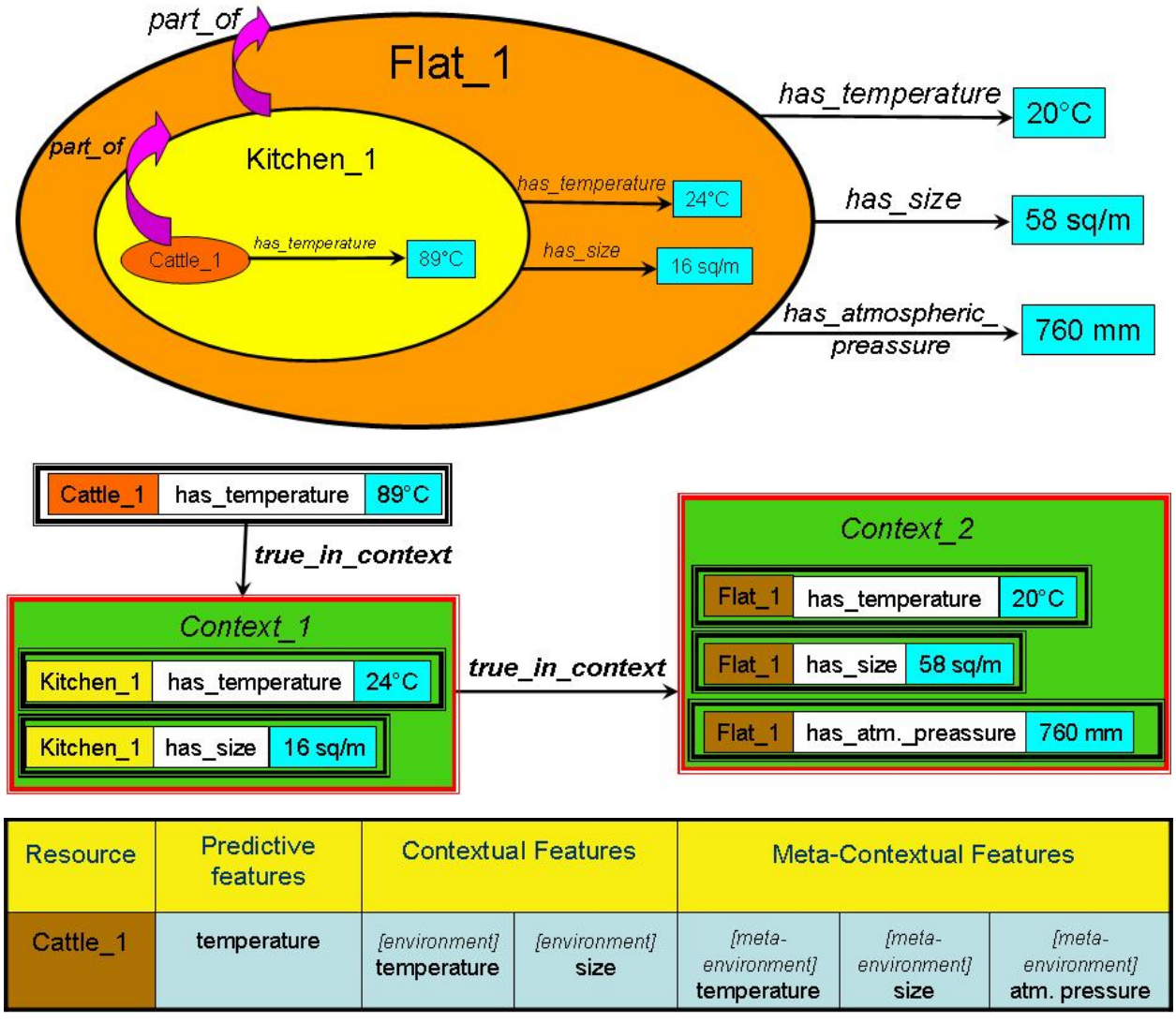
Resource	Predictive features	Contextual Features	
Resource_k	Property_n	Property_q	Property_p

Fig. 10. The sample of *part_of* context: (a) RDF graph view, (b) nested graph view, (c) CDF graph view, (d) table shows the separation of predictive and contextual features for Resource *k*.

The approach for feature separation described above is naturally recursive due to nested hierarchy of the domain provided by *part_of* relation. If object A is part of object B and B is part of object C, then according to previous definitions it is true that: (a) predictive attributes of object B are in the same time contextual attributes of object A; (b) predictive attributes of object C are in the same time contextual attributes of object B. The above implies that the attributes of object C are in the same time *meta*-contextual attributes of object A.

See example in Fig. 11. Here we have a nested graph of simple domain with three objects: a *cattle*, which is part of some *kitchen*, which is part of some *flat*. It is shown in CDF graph that measured parameters from the cattle are considered in the context of measured parameters from the kitchen, which are also considered in the context of measured parameters from the flat. For this case the table shows the separation of all

parameters of the cattle among three sets: predictive (direct measurements from the cattle), contextual (measurements taken from the kitchen) and meta-contextual (measurements taken from the flat). The more deep hierarchy the domain description has the more precise definition we can get about each domain object and we can easily separate this definition to predictive, contextual, meta-contextual, meta-meta-contextual, etc features and the more precise and reliable diagnoses and predictions we can make based on it.



Resource	Predictive features	Contextual Features		Meta-Contextual Features		
Cattle_1	temperature	[environment] temperature	[environment] size	[meta-environment] temperature	[meta-environment] size	[meta-environment] atm. pressure

Fig. 11. The example of three nested objects (cattle, kitchen, flat), corresponding CDF graph and feature separation table.

In case if such multilevel domain models should be presented in form of Bayesian Metanetwork, then the above approach allows easily distribute all nodes (domain parameters) among different levels of the Metanetwork prior to learning. Consider example in Fig 12. It is shown that some domain description (represented by RDF graph) has been distributed among the levels of the model according to hierarchy of the context and then transformed to a sample of Relevance Bayesian Metanetwork. Concrete structure and parameters of the Metanetwork should be learned separately for each level. In the example the domain parameters has been distributed among three levels. Possibly learned Bayesian Metanetwork can be as follows (Fig. 12.). On the predictive level parameter u is conditionally dependent on parameters t and v, the relevance of parameter t is dependent on parameter p from the context level and the relevance of parameter v is dependent on parameter s from the context level. On the context level parameter p is conditionally dependent on parameters r and s, the relevance of parameter s is dependent on parameter n from the meta-context level. Finally on the meta-context level parameter n is conditionally dependent on parameter m.

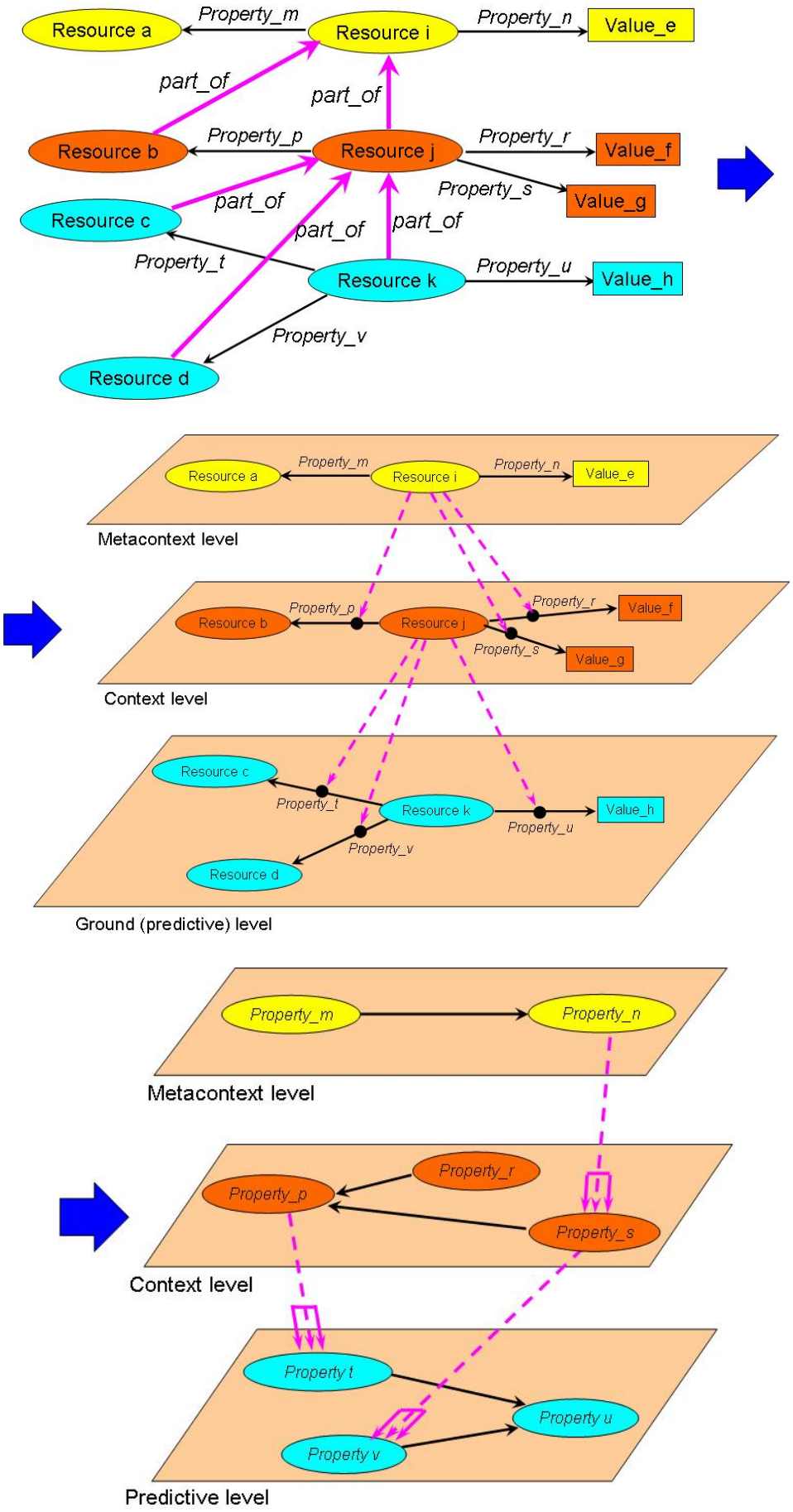


Fig. 12. The example of RDF graph, first distributed among three levels based on context hierarchy and finally transformed to a Relevance Bayesian Metanetwork based on learning.

A domain object generally can be part of several other objects. In this case its context should integrate all properties of its "parents". In the example from Fig.13, object John is part of two objects "Golf Club" and "Symphonic Orchestra". Thus the properties of John (e.g. age) should be considered in the context of all properties of the Golf Club and of the Symphonic Orchestra.

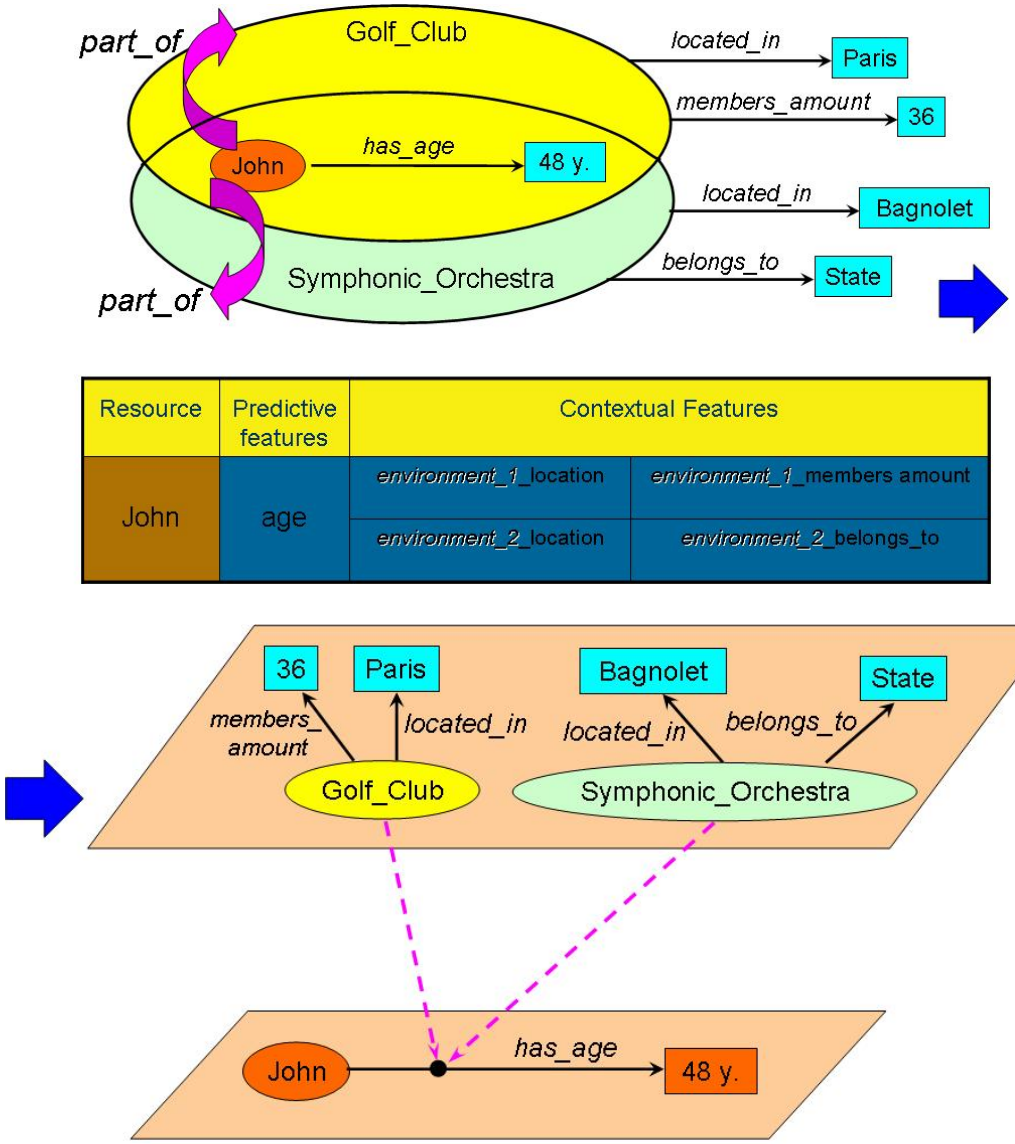


Fig. 13. The example of the domain is shown where an object is part of two of other domain objects. It can be seen that in this case the context for that object description formed as integration of the descriptions of its "parents".

5.2. *Role-based* context extraction

Another approach for context extraction is related to such domain objects, which are proactive components of some organizations or business processes. Most often this is applied to humans or intelligent agents. Such objects usually play certain roles in their organization or in their business process. The natural context for such objects individual descriptions can be the description of their current role (goals, duties, responsibilities, behavior, commitments, policies, etc.). In case if some object is in the same time member of several organizations (or processes) then all integrated duties should form the context of this object and possible contradictions should be resolved (see Fig. 14). The case from Fig. 14 has similar nature as the one from Fig.13. As we can see the lady from Fig. 14 is the member (i.e. part of) several organizations (family, office, volleyball team, women's club). According to part_of hierarchy the context for this lady description should include descriptions of all these organizations (similarly as in Fig. 13). However the important part of the context will be also the description of the roles and appropriate duties the lady plays in these organizations

(e.g. wife in the family, defender in the team, concursant in the club, manager in the office, etc.). The specific feature of the role-based context is that some commitments and duties related to someone's roles in different organizations can be contradictory and that is an important task of appropriate decision-making tools to resolve such contradictions.

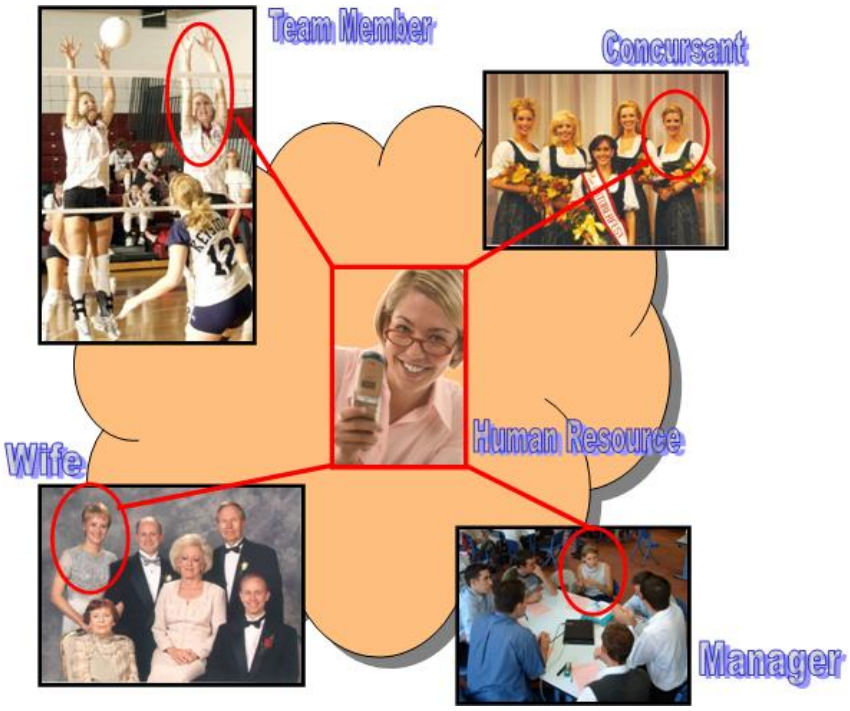


Fig. 14. The example of the proactive object (human resource), which is part of several organization and which is playing different roles in each of them. The context of this object should include the description of these roles (duties, commitments, responsibilities, etc).

Consider two challenges related to *part_of* hierarchies and appropriate contexts. The first one is the fact that the *part_of* domain structuring (clustering), as well as any other domain ontology engineering, is essentially subjective. This means that the same object described according to two different domain ontologies will have two different sets of not only predictive features but also contextual ones. The second challenge is that *part_of* hierarchies are generally dynamic and this result to the fact that the context is the function of time. For example certain object can proactively move from organization to organization, recreate commitments, change duties etc. This means that appropriate decision support system should take into account such temporal (and spatial also) dynamics of the contexts as well as its subjectivity.

5.3. *Interface-based* context extraction

Another interpretation of a context and its influence to relevance of the domain objects' features is related to domain objects visualization through graphical user interfaces. We base on assumption that each interface is designed to certain category of users to provide them access to certain information needed to perform certain goal-driven activity. This means that the information about the same domain object being shown in different interface should be selected according to the goals assumed by each particular interface. Thus each interface can be considered as a kind of context, which affect on the set of relevant features of domain objects to be visualized through it.

In the example in Fig. 15 we are considering aircraft as domain object and we have three interfaces (i.e. three contexts) for presenting aircraft information to the users. The first one is for representing spatial information (Google Maps), the second one is pilots' control panel for representing aircraft operational parameters during the flight; and the third one is the aircraft design e-manual for aircraft manufacturers. Each interface is considered as a context, which affect on which parameters of the aircraft is reasonable to show through this interface. It is evident that not all possible parameters of the aircraft are relevant for the presentation of the aircraft in each of these particular interfaces.

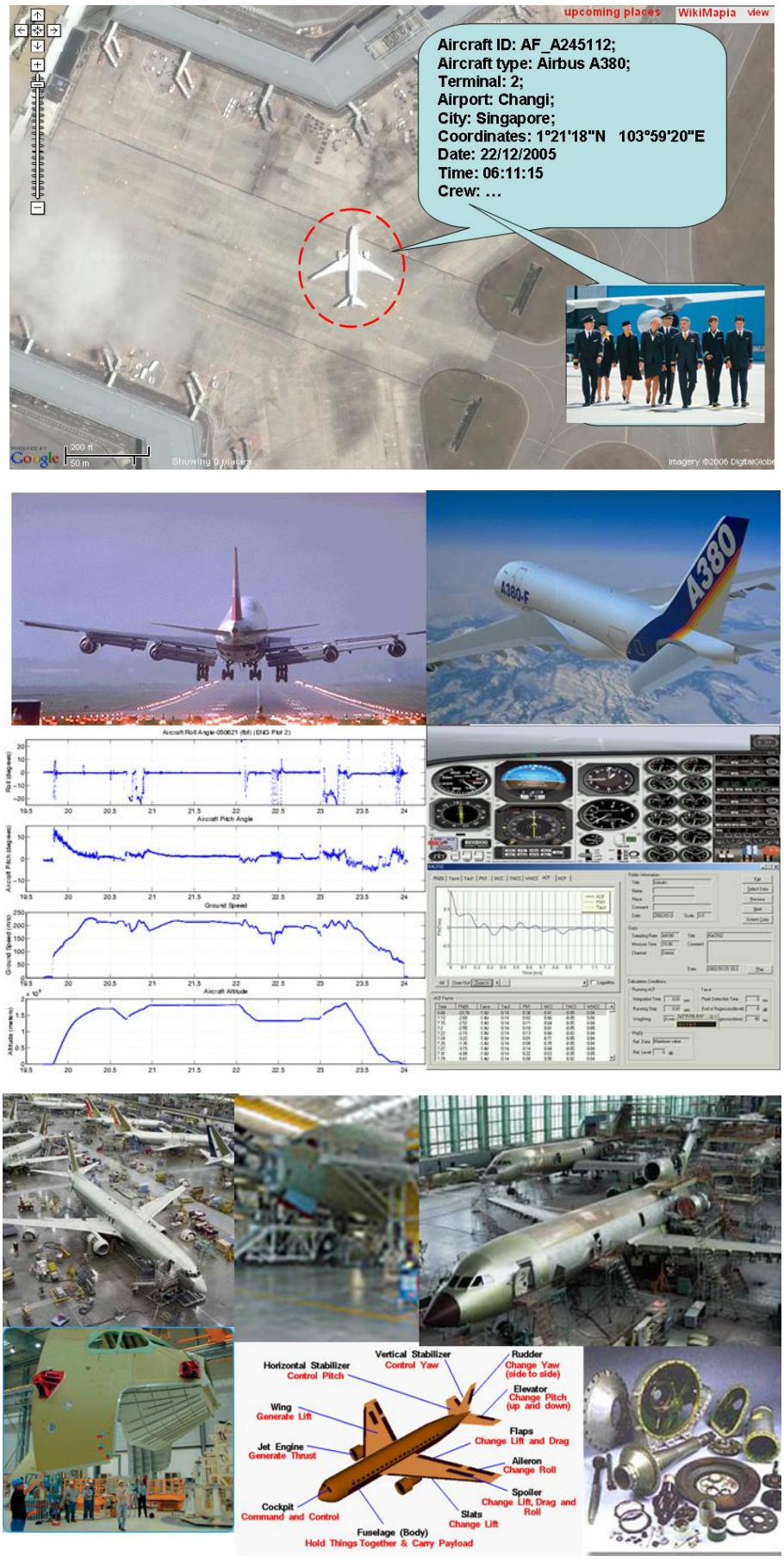


Fig. 15. The example of the domain object (aircraft) is shown in different interfaces: (a) Google Maps; (b) pilots' control panel; (c) manufacturing design e-manual. Each interface is considered as a context, which affect on which parameters of the aircraft is reasonable to show through this interface.

One of specific features of such context-based visualization can be also "zooming relevance", which means that zooming of the interface screen (e.g. map view) may also lead to changes of parameters relevancy for the same domain objects on the screen.

6. Conclusions

Bayesian Networks are proven to be a comprehensive model to describe causal relationships among domain attributes with probabilistic measure of appropriate conditional dependency. However, depending on task and context, many attributes of the model might not be relevant. If a Bayesian Network has been learned across multiple contexts then all uncovered conditional dependencies are averaged over all contexts and cannot guarantee high predictive accuracy when applied to a concrete case. We are considering a context as a set of contextual attributes, which are not directly effect probability distribution of the target attributes, but they effect on a "relevance" of the predictive attributes towards target attributes. In this paper we use the Bayesian Metanetwork vision to model such context-sensitive feature relevance. Such model assumes that the relevance of predictive attributes in a Bayesian network might be a random attribute itself and it provides a tool to reason based not only on probabilities of predictive attributes but also on their relevancies. According to this model, the evidence observed about contextual attributes is used to extract a relevant substructure from a Bayesian network model and then the predictive attributes evidence is used to reason about probability distribution of the target attribute in the extracted sub-network. Such models will be useful in cases when the relevance of the attributes essentially depends on the context.

Distinguishing between relevant and irrelevant features of the domain objects is extremely important for the decision making, however another problem, to sort relevant features either to contextual or to predictive ones, is as much important too. In this paper we consider three strategies of extracting context from relevant features, which are based on: *part_of* context, role-based context and interface-based context. The two challenges has been mention related to these strategies. The first one is the fact that domain models (providing the *part_of* hierarchies), or organizational roles distribution, or interface modeling, etc., are essentially subjective. This means that the same object described according to two different domain ontologies will have two different sets of not only predictive features but also contextual ones. The second challenge is that such contexts are generally dynamic. These challenges require from appropriate decision support system (e.g. based on Bayesian reasoning) to take into account such temporal (and spatial also) dynamics of the contexts as well as its subjectivity.

Acknowledgement

Author is grateful to Tekes (National Technology Agency of Finland) and cooperating companies (Agora Center, TeliaSonera, TietoEnator, Metso Automation, ABB, JSP) for the grant supporting activities of SmartResource project.

References

1. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, (Morgan Kaufmann, 1988).
2. M. Henrion, Some Practical Issues in Constructing Belief Networks, In: *Proceedings of the 3-rd Annual Conference on Uncertainty in Artificial Intelligence*, (Elsevier, 1989), pp. 161-174.
3. D. Heckerman, A Tutorial on Learning with Bayesian Networks, *Technical Report MSR-TR-95-06*, (Microsoft Research, March 1995).
4. C. J. Butz, Exploiting Contextual Independencies in Web Search and User Profiling, In: *Proceedings of the World Congress on Computational Intelligence*, (Hawaii, USA, 2002), pp. 1051-1056.
5. C. Boutiler, N. Friedman, M. Goldszmidt and D. Koller, Context-Specific Independence in Bayesian Networks, In: *Proceedings of the 12-th Conference on Uncertainty in Artificial Intelligence*, (Portland, USA, 1996), pp. 115-123.

6. N.L. Zhang, Inference in Bayesian networks: The Role of Context-Specific Independence, *International Journal of Information Technology and Decision Making*, **1**(1) 2002, 91-119.

7. D. Geiger and D. Heckerman, Knowledge Representation and Inference in Similarity Networks and Bayesian Multinets, *Artificial Intelligence*, Vol. 82, (Elsevier, 1996), pp. 45-74.

8. N. Friedman, D. Geiger, and M. Goldszmidt, Bayesian Network Classifiers, *Machine Learning*, **29**(2-3), (Kluwer, 1997), pp. 131-161.

9. J. Cheng and R. Greiner, Learning Bayesian Belief Network Classifiers: Algorithms and System, In: *Proceedings of the 14-th Canadian Conference on Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 2056, (Springer-Verlag Heidelberg, 2001), pp. 141-151.

10. J. Pena, J. A. Lozano, and P. Larranaga, Learning Bayesian Networks for Clustering by Means of Constructive Induction, *Machine Learning*, **47**(1), (Kluwer, 2002), pp. 63-90.

11. V. Terziyan, A Bayesian Metanetwork, *International Journal on Artificial Intelligence Tools*, **14**(3), (World Scientific, 2005), pp. 371-384.

12. V. Terziyan and O. Vitko, Bayesian Metanetwork for Modelling User Preferences in Mobile Environment, In: *Proceedings of KI 2003: Advances in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Vol. 2821, ed. A. Gunter, R. Kruse and B. Neumann, (Springer-Verlag, 2003), pp.370-384.

13. V. Terziyan and O. Vitko, Learning Bayesian Metanetworks from Data with Multilevel Uncertainty, In: M. Bramer and V. Devedzic (eds.), *Proceedings of the First IFIP International Conference on Artificial Intelligence and Innovations (AIAI-2004)*, Toulouse, France, (Kluwer, 2004), pp. 187-196.

14. P. Bouquet, C. Ghidini, F. Giunchiglia, and E. Blanzieri, Theories and Uses of Context in Knowledge Representation and Reasoning, In: V. Akman and C. Bazzanella, (eds.), Special Issue on Context, *Journal of Pragmatics*, Elsevier, Vol. 35, No. 3, 2003, pp. 455-484.

15. C. Ghidini and F. Giunchiglia, Local Models Semantics, or Contextual Reasoning = Locality + Compatibility, *Artificial Intelligence*, Vol. 127, No. 2, 2001, pp. 221–259.

16. P.F. Patel-Schneider, P. Hayes, and I. Horrocks, Web Ontology Language (OWL) Abstract Syntax and Semantics, Technical report, W3C, www.w3.org/TR/owl-semantics/, February 2003.

17. P. Bouquet, F. Giunchiglia, F. Van Harmelen, L. Serafini, and H. Stuckenschmidt , Contextualizing Ontologies, *Journal of Web Semantics* , Vol. 26, 2004, pp. 1-19.

18. Terziyan V., Bayesian Metanetwork for Context-Sensitive Feature Relevance, In: G. Antoniou et al. (eds.), Advances in Artificial Intelligence, *Proceedings of the 4-th Hellenic Conference on Artificial Intelligence (SETN 2006)*, Lecture Notes in Artificial Intelligence, Vol. 3955, 2006, pp. 356-366.

19. Khriyenko O., Terziyan V., A Framework for Context-Sensitive Metadata Description, *International Journal of Metadata, Semantics and Ontologies*, Vol. 1, No. 2, 2006, pp. 154-164.