

Semi-Qualitative Probabilistic Networks in Computer Vision Problems

Cassio P. de Campos, *Rensselaer Polytechnic Institute,*
Troy, NY 12180, USA.

Email: decamc@rpi.edu

Lei Zhang, *Rensselaer Polytechnic Institute,*
Troy, NY 12180, USA.

Email: zhangl2@rpi.edu

Yan Tong, *Rensselaer Polytechnic Institute,*
Troy, NY 12180, USA.

Email: tongy2@rpi.edu

Qiang Ji, *Rensselaer Polytechnic Institute,*
Troy, NY 12180, USA.

Email: jiq@rpi.edu

*Received: *** Revised: ****

Abstract

This paper explores the application of semi-qualitative probabilistic networks (SQPNs) that combine numeric and qualitative information to computer vision problems. Our version of SQPN allows qualitative influences and imprecise probability measures using intervals. We describe an Imprecise Dirichlet model for parameter learning and an iterative algorithm for evaluating posterior probabilities, maximum a posteriori and most probable explanations. Experiments on facial expression recognition and image segmentation problems are performed using real data.

AMS Subject Classification: 68T37, 62F15, and 68T30.

Keywords: Probabilistic networks; Qualitative relations; Imprecise probabilities; Computer vision applications.

1 Introduction

Graphical models such as Bayesian Networks are becoming increasingly popular in many applications. Such models provide compact structures to encode joint probabil-

ity distributions for their variables, relying on a factorization of its conditional probability distributions. Qualitative probabilistic networks (QPNs) relax the precise probability values that are mandatory in Bayesian networks. No quantitative information is demanded, and joint distributions are defined through qualitative relations among variables and probability values (Druzdzel and Henrion, 1993; Wellman, 1990).

Semi-qualitative probabilistic networks (SQPNs) combine properties of standard Bayesian networks and QPNs (Parsons and Dohnal, 1993; Renooij and van der Gaag, 2002; de Campos and Cozman, 2005a). The aim of this paper is to present an SQPN formulation based on qualitative influences (Wellman, 1990) and probability intervals (Walley, 1991), and to discuss its application to two computer vision problems: facial expression recognition and image segmentation.

Although there are efficient algorithms for QPNs (Druzdzel and Henrion, 1993), exact inferences in SQPNs are hard (de Campos and Cozman, 2005a). We introduce an algorithm based on iterative search that converges to a local optimal solution. Parameter learning is also considered, when we want to estimate the probability measures of conditional probability distributions given the structure of the network. We discuss the ideas of de Campos and Cozman (2005a), where an SQPN prior is combined to data through an Imprecise Dirichlet Model, and describe how to apply those ideas to our computer vision domains.

Experiments on computer vision problems indicate the benefits of using the imprecise probabilities of SQPNs instead of the classical solutions. We show satisfactory results while recognizing facial expressions and segmenting images. In both cases we deal with real data and qualitative relations elicited by experts.

The paper is divided as follows. Section 2 reviews the basics of Bayesian networks and QPNs, and presents the definition of SQPN adopted in this work. Section 3 describes a fast and accurate algorithm for performing inferences, including posterior probability queries, maximum a posteriori and most probable explanations, while Section 4 explain the parameter learning procedure based on the Imprecise Dirichlet Model. Computer vision experiments with real data are detailed in Section 5. Section 6 concludes the paper and indicates some paths for future work.

2 Semi-qualitative probabilistic networks

The objective of this section is to define our semi-qualitative probabilistic networks. First, we review the ideas of Bayesian networks and QPNs that are used in the SQPNs. Let $\mathcal{X} = \{X_1, \dots, X_n\}$ be a set of n categorical random variables such that X_i takes values in the finite set $\{x_i^1, \dots, x_i^{r_i}\}$. For $W \subseteq \mathcal{X}$, the lowercase w represents a configuration for the variables in W , that is, $w = \{x_i^{k_i} | X_i \in W\}$, where indices k_i are known. We also assume an ordering for the categories of each variable, that is, for each X_i , we have $x_i^1 < x_i^2 < \dots < x_i^{r_i}$.

A Bayesian network is a pair (G, \mathcal{P}) , where: $G = (V_G, E_G)$ is a directed acyclic graph (DAG), with V_G a collection of vertices associated to random variables of \mathcal{X} (a node per variable), and E_G a collection of arcs; \mathcal{P} is a collection of conditional probability distributions $p(X_i | PA_i)$ where PA_i are the parents of X_i in the graph (PA_i may be empty), respecting the relations of E_G . In a Bayesian network every variable

is conditionally independent of its non-descendants given its parents (Markov condition). This structure induces a single joint probability distribution by the expression $p(\mathcal{X}) = \prod_i p(X_i|PA_i)$. For ease of expose, variables and nodes of the graph are used interchanged, as there is an one-to-one relation between them.

A QPN is similar to a Bayesian network, but instead of probability distributions $p(X_i|PA_i)$ we have a collection of *qualitative* constraints on probability values (no numerical values are defined). The same Markov condition applies. Qualitative constraints are defined by *influences* and *synergies* (Wellman, 1990). In this paper we only consider influences, as they are the most used type of qualitative constraint in the literature and they are easy to acquire from experts. Qualitative influences define some knowledge about the category of a variable given the category of another.

Definition 1 Let X_a, X_b be variables such that $X_a \in PA_b$ (X_a is parent of X_b). We say that X_a influences X_b positively if

$$\forall k_a > k'_a, k_b = 1, \dots, r_b - 1 : \sum_{k > k_b} p(x_b^k | x_a^{k_a}, pa'_b) \geq \sum_{k > k_b} p(x_b^k | x_a^{k'_a}, pa'_b) + \delta, \quad (1)$$

where $\delta \geq 0$ is a constant and pa'_b ranges over all configurations of parents except for X_a , that is, Equation (1) must hold for each configuration pa'_b of $PA_b \setminus \{X_a\}$.

This roughly means that observing a greater category for X_a makes more likely to have greater categories in X_b . If these constraints hold for $\delta > 0$, this influence is said *strong* with threshold δ (Renoij and van der Gaag, 1999). Otherwise, it is said *weak* for δ . A negative influence is obtained replacing the inequality operator \geq by \leq and the sign of δ to negative. A zero influence is obtained replacing the inequality operator \geq by an equality operator. When applied to binary variables, summations of Equation (1) disappear and we have a simpler formulation: $p(x_b^2 | x_a^2, pa'_b) \geq p(x_b^2 | x_a^1, pa'_b) + \delta$, for all pa'_b (greater value of X_a leads to a greater value of X_b).

Besides standard influences, we may deal with non-monotonic relations (Renoij and van der Gaag, 2000). They happen when constraints hold only for some configurations of parents (but not all of them). For example, suppose we have three binary variables such that X_b has X_a and X_c as parents and that $p(x_b^2 | x_a^2, x_c^1) \geq p(x_b^2 | x_a^1, x_c^1)$ holds, but $p(x_b^2 | x_a^2, x_c^2) \geq p(x_b^2 | x_a^1, x_c^2)$ can not be stated. Hence we do not have a positive influence of X_a on X_b , because it would be necessary to have both constraints to ensure that X_a influences X_b positively. In this case, the category of X_c (the other parent of X_b concerning a possible influence from X_a) is relevant and determines when the constraint holds. In this case, a non-monotonic influence of X_a on X_b that holds when X_c is x_c^1 (but not x_c^2) might be stated.

In SQPNs, we allow qualitative relations to co-exist with probability intervals. One may define that the probability measure of a node given its parents is within some range:

Definition 2 Let X_i be a network variable, x_i^k one of its categories, pa_i a configuration of X_i parents, $l_{i(pa_i)k}$ and $u_{i(pa_i)k}$ constant numbers. A range constraint for x_i^k given pa_i is

$$l_{i(pa_i)k} \leq p(x_i^k | pa_i) \leq u_{i(pa_i)k}. \quad (2)$$

The values $p(x_i^k|pa_i)$ are the parameters of the network. Networks with imprecise probability parameters (including this interval case) are known as credal networks (Cozman, 2000). Their nodes are associated to random variables and conditional probability distributions are specified using closed convex sets of probability measures. We adopt a definition of SQPN that combines influences with probability intervals:

Definition 3 An SQPN is a pair (G, \mathcal{C}) , where: $G = (V_G, E_G)$ is a DAG, with V_G a collection of vertices associated to random variables of \mathcal{X} (a node per variable), and E_G a collection of arcs; \mathcal{C} is a finite collection of qualitative influences and range constraints defined over conditional probabilities $p(x_i^k|pa_i)$, where x_i^k is a category of variable X_i and pa_i is a configuration of the parents of X_i in G .

An SQPN represents a set of joint probability distributions that satisfy all constraints in the network. We assume that every joint probability distribution in the convex hull of this set satisfy the Markov condition: variables are conditionally independent of non-descendants given their parents in the graph, which results in the same factorization as in Bayesian networks: $p(\mathcal{X}) = \prod_i p(X_i|PA_i)$. This concept is known as *strong independence* in the theory of sets of probabilities (Couso et al., 1999; Cozman, 2000). Our SQPN is in fact a sub-case of a credal network under strong independence. However, sets of conditional probability distributions are not separately specified, as usually assumed in credal networks (da Rocha and Cozman, 2002). We instead have *influences* that involve parameters of different distributions (although in the same node).

3 Inferences in SQPNs

A belief updating in an SQPN can be formulated as a multi-linear programming problem. The goal is to minimize/maximize a marginal or joint probability subject to influences and ranges.

Definition 4 Let \mathcal{X} be the set of variables in an SQPN, $X_q \subseteq \mathcal{X}$ be a set of query variables with x_q as its configuration, and $X_e \subseteq \mathcal{X} \setminus X_q$ be the set of observed variables with x_e as the observations. The belief updating query is the task of finding $\text{argmin} \min_p p(x_q|x_e)$ and $\text{argmax} \max_p p(x_q|x_e)$, where p is a probability distribution constrained by the SQPN.

In words, we want to find one joint probability distribution that satisfy all SQPN constraints and that minimizes the probability of x_q given x_e (the same idea for the maximization). This query can be written using a multi-linear expression¹ from Bayesian network theory:

$$p(x_q|x_e) = \frac{p(x_q, x_e)}{p(x_e)} = \frac{\sum_{X_i \notin X_q \cup X_e} \prod_i p(x_i^{k_i}|pa_i)}{\sum_{X_i \notin X_e} \prod_i p(x_i^{k_i}|pa_i)}, \quad (3)$$

¹We assume that the probability of evidence is greater than zero so as Equation (3) can be multiplied by $p(x_e)$ to obtain a multi-linear equation.

where $x_i^{k_i}$ and pa_i must comply with the corresponding x_q and x_e . Expression (3) together with influences and ranges constraints of the SQPN form a multi-linear programming problem. Given the potentially huge number of multi-linear terms in Expression (3) (the number of terms is exponential on the size of the network) and the fact that probability values $p(x_i^{k_i}|pa_i)$ are not precisely known but defined through linear constraints of the SQPN, exact inferences are NP^{PP}-complete (de Campos and Cozman, 2005b) and will eventually fail to obtain an answer in reasonable time.

One idea to solve this inference problem is to treat an SQPN as a (potentially large) set of Bayesian networks over the same graph (but with distinct probability values) and locally search into this set for the best configuration (da Rocha et al., 2003). Suppose we want to maximize Expression (3), that is, $\max_p p(x_q|x_e)$, subject to the SQPN constraints (p must comply with the SQPN constraints). If we fix the probabilities in each conditional distribution of each node of the SQPN, we obtain a Bayesian network and Expression (3) can be evaluated using standard Bayesian network techniques. The maximum value over all such choices is the global optimum solution. However, the number of Bayesian networks that an SQPN represents is huge and it is not possible to look at all of them.

A simple but accurate idea is to perform an iterative local search (ILS). Start from an initial feasible guess (it is possible to use linear programming to take a collection of conditional distributions $p(x_i|pa_i)$ that satisfy the SQPN constraints) and evaluate the query using any Bayesian network inference method (this is the initial value). Now, choose a node (using some deterministic criterion like running cyclically amongst nodes) and for the distributions of this node, allow the probability values $p(x_i|pa_i)$ to vary (still according to the SQPN constraints) and take the best solution. Note that, when we fix all distributions but those of a single node, Expression (3) becomes a division of linear functions, because probability values of a single node never appear multiplied together. As constraints (influences and ranges) are already linear, we can apply fractional linear programming (there is no need to enumerate vertices). This is particularly important when many constraints are specified, as the number of vertices may be exponential in the number of constraints. So, we keep a current best solution and each time a better value is locally found, we update the current solution. We stop the process when there is no possible improvement among all nodes (taken one at a time). The ILS algorithm finitely converges to a local optimum solution of the belief updating query, as the feasible region induced by the network is a polytope (although the objective function is multi-linear, all constraints are linear) and there are a finite number of vertices (note that ILS walks through vertices). Taking care of ties in the linear programs, the characteristics of the multi-linear program imply that the solution is a local optimum (Lukatskii and Shapot, 2001).

This procedure is similar to the method proposed by da Rocha et al. (2003), but our version is more general: we do not need to enumerate vertices neither assume that sets of conditional distributions are separately specified. We also note that this procedure works even with qualitative synergies as well as credal networks in general (the only eventual change is to replace linear programming by convex programming if non-linear convex constraints are used, but that does not change the characteristics of the procedure as the latter shares all important features of the former).

Another possible implementation of the ILS algorithm is obtained using a symbolic variable elimination, where one collects all generated multi-linear constraints to use in a multi-linear solver (de Campos and Cozman, 2004). Prior to the iterative process, a symbolic variable elimination is executed supposing all probability values are unknown. Then the procedure iteratively chooses nodes (one at a time) and replaces the unknown probability values of the multi-linear program by fixed values, except for values of the chosen node (that remain unknown and will be optimized). Now, the only unfixed/unknown probability values pertain to a single node, which implies that the multi-linear program becomes linear and can be solved as before. Eventually we converge to a local optimum. The advantage of this implementation is that a single symbolic variable elimination is performed, and then the multi-linear program is reused at each iteration to create the local linear programs that are solved.

Now suppose we want to find the categories of variables in X_q that maximizes their joint probability given observations x_e :

Definition 5 Let \mathcal{X} be the set of variables in an SQPN, $X_q \subseteq \mathcal{X}$ be a set of query variables, and $X_e \subseteq \mathcal{X} \setminus X_q$ be the set of observed variables with x_e as the observations. The maximum a posteriori query is the task of finding $\operatorname{argmax} \max_{X_q} \max_p p(X_q | x_e)$, where p is constrained by the SQPN.

We call this problem MPE (most probable explanation) when $X_q = \mathcal{X}$ (maximum a posteriori or MAP is the general case). The ILS algorithm can also be used to solve MAP and MPE queries. We just need to apply a simple reformulation (which we call *MAP-reformulation*): for each $X_i \in X_q$, create a new binary node Y_i with X_i as its sole parent such that $\sum_{X_i} p(y_i^1 | X_i) = 1$ (this is the only constraint besides simplex constraints that restrict the imprecise probabilities of node Y_i). Although the reformulated network has constraints $\sum_{X_i} p(y_i^1 | X_i) = 1$ that are not influences neither ranges, they are still linear and involve only probability values inside a single node. Thus we use ILS to solve the query $\max_p p(y_q | x_e)$, where y_q is the set $\{y_i^1 | X_i \in X_q\}$. In the solution of this belief updating query, $p(y_i^1 | X_i)$, for all i , act as indicator functions of which categories the variables in X_q take ($p(y_i^1 | X_i) = 1$ exactly when X_i takes the desired MAP category, and zero otherwise).

Lemma 6 The belief updating query after a MAP-reformulation $\max_p p(y_q | x_e)$, where $y_q = \{y_i^1 | X_i \in X_q\}$, also solves the MAP problem $\operatorname{argmax} \max_{X_q} \max_p p(X_q | x_e)$.

Proof. First suppose we have a solution of $\max_p p(y_q | x_e)$ where some $p(y_i^1 | X_i)$ does not concentrate in a single category of X_i , that is, there are x_i^j and x_i^k such that both $p(y_i^1 | x_i^j) > 0$ and $p(y_i^1 | x_i^k) > 0$. Now, note that $p(y_q | x_e) = \sum_{X_i} p(y_i^1 | X_i) \cdot p(X_i, y_q \setminus \{y_i^1\} | x_e)$, where the second factor of this multiplication does not depend on Y_i . Hence, if $p(y_i^1 | X_i)$ (which sums 1 over all X_i) does not concentrate on the greatest value of $p(X_i, y_q \setminus \{y_i^1\} | x_e)$, then it is possible to find a greater value for the query $p(y_q | x_e)$ by concentrating $p(y_i^1 | X_i)$ on that value, which would be a contradiction. So the only possible explanation is that $p(x_i^j, y_q \setminus \{y_i^1\} | x_e) = p(x_i^k, y_q \setminus \{y_i^1\} | x_e)$, that is, x_i^j and x_i^k achieve the same result. In this case, the solution that concentrates on either $p(y_i^1 | x_i^j)$ or $p(y_i^1 | x_i^k)$ can be used (both obtain the same maximum), and without loss of generality,

we assume that it concentrates on $p(y_i^1|x_i^j)$. Hence, because $p(y_i^1|X_i)$ concentrates in a single category of X_i (for all i), we have $\max_p p(y_q|x_e) = \max_p p(x_q|x_e) = \max_{X_q} \max_p p(X_q|x_e)$, where x_q is the solution of the MAP problem. \square

4 Learning with SQPNs and empirical data

In this section we briefly describe the learning procedure of de Campos and Cozman (2005a). Suppose we want to combine a domain knowledge elicited by an expert using an SQPN with a complete training data set. Let \mathcal{N} be an SQPN conveying the expert's prior beliefs and $D = \{D_1, \dots, D_N\}$ a complete data set, where $D_t = \{x_{1,t}^{k_1}, \dots, x_{n,t}^{k_n}\}$ is a sample of SQPN variables.

Our goal is to estimate the parameters of multinomial distributions on $X_i|pa_i$ using both \mathcal{N} and data. The Dirichlet distribution is a natural prior for $p(X_i|pa_i)$, because it is conjugate with the multinomial distribution. A possible parameterization is $p(X_i|pa_i) \propto \prod_k p(x_i^k|pa_i)^{s\tau_{ijk}-1}$ for $s \geq 0$ and $\forall ij \sum_k \tau_{ijk} = 1$, where the hyper-parameter s controls dispersion and hyper-parameters τ_{ijk} control location (Walley, 1996). The proposal is to view the content of \mathcal{N} as *constraints on the hyper-parameters* τ_{ijk} of Dirichlet distributions, that is, $\tau_{ijk} = p_{\mathcal{N}}(x_i^k|pa_i)$, with $j = pa_i$, subject to SQPN constraints of \mathcal{N} . This process is usually referred to as the *Imprecise Dirichlet Model* (IDM) (Bernard, 2005; Walley, 1996). Using expectation as estimation, we have

$$p(x_i^k|pa_i) = \frac{s\tau_{i(pa_i)k} + n_{i(pa_i)k}}{s + \sum_k n_{i(pa_i)k}}, \quad (4)$$

where $n_{i(pa_i)k}$ are the counts from the data set, that is, $n_{i(pa_i)k}$ is the number of samples in D that contains both x_i^k and pa_i . As Equation (4) is linear on parameters $p(x_i^k|pa_i)$ and $\tau_{i(pa_i)k}$, the result is a posterior credal network on which parameters $p(x_i^k|pa_i)$ must satisfy both Equation (4) and constraints of \mathcal{N} (de Campos and Cozman, 2005a). We do not have a posterior SQPN because, according to our definition of SQPN, Equation (4) is neither an influence nor a range constraint. Nevertheless, the inference algorithms of Section 3 can be directly applied to this posterior *credal* network. Note that we are globally dealing with constraints and parameters (Cano et al., 2007), as we do not estimate "local" bounds for $p(x_i^k|pa_i)$ but keep all relevant information in the model, which is then processed during inferences.

5 Computer Vision Applications

In this section we present two applications of SQPNs to computer vision problems. Both examples involve real data and indicate the benefits of using SQPNs in such domains. First we describe a recognition problem of facial expressions, where an SQPN models expert's prior beliefs. Parameter learning and belief updating queries are performed to achieve recognition results. Then we present an image segmentation problem where an SQPN is used to obtain a most probable explanation of edges and vertices that segment the image.

5.1 Facial Expression Recognition

Consider a problem of recognizing facial expressions from real image data (Pantic and Rothkrantz, 2000; Fasel and Luetttin, 2003). The Facial Action Coding System (Ekman and Friesen, 1978) is the most commonly used system for facial behavior analysis. Based on this system, facial expressions can be decomposed into a set of Action Units (denoted as AUs), which are related to contractions of specific sets of facial muscles. An automatic system for facial AU recognition has applications in human behavior science, human-computer interaction, interactive games, entertainment, and psychiatry. In this work, we intend to recognize 14 commonly occurring AUs .¹ Details about AUs and their definitions can be found in (Ekman and Friesen, 1978).

Based on the study in (Tong et al., 2007), there are semantic relationships among AUs . On the one hand, some AUs may always happen together to show a meaningful facial expression. For example, AU_6 (cheek raiser) tends to occur together with AU_{12} (lip corner puller) when someone is smiling. On the other hand, some AUs may be mutually exclusive. For instance, AU_{25} (lips part) never happens simultaneously with AU_{24} (lip presser) since they are activated by same facial muscles but their motion directions are opposite. Instead of recognizing each AU individually, a probabilistic network can be employed to explicitly model relationships among AUs .

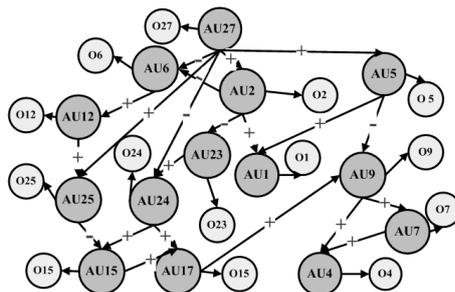


Figure 1: Network for the AU recognition problem.

We use an SQPN network with 28 binary nodes. Figure 1 depicts its structure. There are 14 gray nodes associated to AUs , and 14 bright nodes associated to computer vision measurements (one measurement node for each AU node). The category au_i^2 is the *true* category, meaning that AU_i is active (or present), while au_i^1 is *false* (AU_i is inactive or absent). Note that every link between AU nodes has a sign, which is provided by a domain expert. The sign indicates if there is a positive or negative qualitative influence between the corresponding AUs . For example, it is difficult to do AU_2 (outer brow raiser) without AU_1 (inner brow raiser), but we can do AU_1 without AU_2 . Thus there is

¹ AU_1 (Inner brow raiser), AU_2 (Outer brow raiser), AU_4 (Brow lowerer), AU_5 (Upper lid raiser), AU_6 (Cheek raiser and lid compressor), AU_7 (Lid tightener), AU_9 (Nose wrinkler), AU_{12} (Lip corner puller), AU_{15} (Lip corner depressor), AU_{17} (Chin raiser), AU_{23} (Lip tightener), AU_{24} (Lip Presser), AU_{25} (Lips part), and AU_{27} (Mouth stretch).

a positive influence of AU_2 on AU_1 . We also extract some generic constraints provided by domain experts: AU_{27} is not usual, so $P(au_{27}^2) \leq 0.5$; if a node AU_i has more than one parent node and all of them have positive influence, then $P(au_i^2|pa_i^2) \geq 0.8$, where pa_i^2 is the configuration where all parents are active; if a node AU_i has more than one parent node and all of them have negative influence, then $P(au_i^2|pa_i^2) \leq 0.2$.

Real data from the DFAT-504 database (Kanade et al., 2000) are used for training. This data set contains more than 8000 frames that are manually labeled into AU labels (presence or absence), frame by frame. Hence, we have 14 AU labels for each frame. Besides the manually coded AU labels, we also obtain the measurements of AUs using the computer vision techniques of Bartlett et al. (2006), which delivers 14 AU measurements for each frame. 500 frames are randomly selected from the data set for testing. From the remaining frames, two sample sets are taken, where 100 and 1000 frames are randomly chosen for training. With small amount of data, AU recognition tends to be harder. Thus, we can verify how effective is the SQPN in these challenging environments.

First, we use a Bayesian network with the same structure of our SQPN and we learn parameters using a standard maximum likelihood estimation. Standard inference techniques are performed for each AU in each frame of the testing set. We evaluate the probability of each AU given the observations from computer vision measurements. Then, our SQPN is employed on the same scenarios. The learning idea of Section 4 is followed by the inference procedure described in Section 3. For each AU_i of each frame, we find interval probabilities $[\min_p p(au_i^k|o), \max_p p(au_i^k|o)]$ for the presence ($k = 2$) and for the absence ($k = 1$) given the computer vision measurements o . Only cases with *interval dominance* are analyzed, that is, the maximum probability of presence is lesser than the minimum probability of absence (or vice-versa). In such cases, it is possible to decide between presence or absence without further considerations. The results for the Bayesian network are presented in Table 1, as well as results for the SQPN. Column 2 shows the percentage of cases in the testing set that the SQPN inference achieves interval dominance. Positive rate (columns 3 and 5) indicate the percentage of well recognized active AUs , while negative rate (columns 4 and 6) show the percentage of well classified inactive AUs . We must note that cases are partitioned by the SQPN inference into interval-dominance cases and uncertain cases (where the decision using the intervals generated by the SQPN is harder and depends on the strategy we choose). Results on interval-dominance cases show that recognition rate is better when the SQPN is employed, which can be viewed as the following: *the SQPN inference gives a measure of certainty about its answer. In cases where inference is "sure" (interval dominance happens), the result is generally better than that of a Bayesian network.* A study about cases where interval dominance does not hold and the best strategy to take is left for future work.

5.2 Image Segmentation

Image segmentation is a difficult low level problem in computer vision. It aims to partition an image into constituent regions of interest. Many approaches have been proposed to deal with this problem. Most of them are data-driven deterministic approaches, which may fail to produce satisfactory segmentation when there are shadow,

Dataset Size	Interval Dominance	Bayesian network		SQPN	
		Positive	Negative	Positive	Negative
100	49.2%	51.9%	98.1%	61.5%	97.4%
1000	54.8%	57.7%	99.4%	69.1%	99.8%

Table 1: Comparison between Bayesian network/max. likelihood and SQPN/Imprecise Dirichlet Model for AU recognition. Positive and negative rates are reported.

occlusion, cluttering, and/or noise in the image data (Gonzalez and Woods, 2002). We focus on edge-based segmentation methods that exploit edge information to partition the image (Caselles et al., 1997; Kass et al., 1988).

We present a framework based on SQPNs where image segmentation is performed using a most probable explanation inference. The SQPN considers uncertainties in the information that are not captured by other approaches. We use a two-layer SQPN based on edge and vertex information obtained with an over-segmented edge map. This edge map consists of edge segments and vertices, where a vertex is a position where three or more edges intersect each other. Figure 2(a) describes a simple case of edge map. The SQPN structure of Figure 2(b) is constructed from that edge map: parents of a vertex node are the edges that intersect to form the vertex. Edges are denoted by E_j (the whole set of edges is E) and vertices are denoted by V_t (the whole set is V). The shadowed nodes are related to computer vision measurements and have as parent the corresponding edges and vertices. They are denoted by M_{E_j} and M_{V_t} .

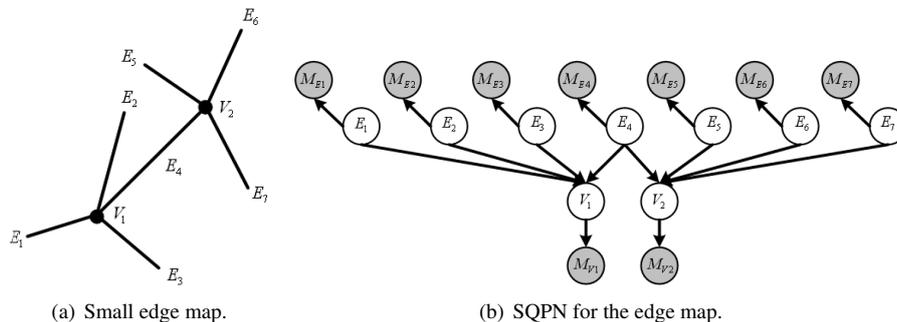


Figure 2: Edge map showing the relation between edges and vertices and the corresponding SQPN structure that is derived from the edge map.

Every node in our SQPN is binary: e_j^2 means the edge belongs to the object boundary that we are looking for, and e_j^1 means it does not. v_t^2 and v_t^1 have the same meaning for vertices. So, e_j^2 and v_t^2 are the *true* categories, and e_j^1 and v_t^1 are the *false* categories with respect to the boundary pertinence. Furthermore, shadowed nodes of Figure 2(b) are image measurements obtained from computer vision techniques for both edge and vertex nodes. The average intensity gradient magnitude is used as measurement for edge segments (nodes M_{E_j}) and the corner detector of Harris and Stephens (1988) is

employed to measure vertices (nodes M_{V_t}). The vertex measurement nodes are *discretized* according to the corner response. If the corner response is above a threshold and it is a local maximum, a corner is detected and the measurement node M_{V_t} is observed as $m_{V_t}^2$. Otherwise it is set to $m_{V_t}^1$.

Conditional probability distributions $p(M_{V_t}|V_t)$ describe the uncertainty between the category of a vertex V_t and its measurement M_{V_t} . They are defined as $p(m_{V_t}^2|v_t^2) = 0.99$ and $p(m_{V_t}^2|v_t^1) = 0.1$. This roughly means that if a corner is detected, the corresponding vertex node V_t has a high probability of being v_t^2 (which means it belongs to the boundary). Still a detected corner may not be part of the desired boundary.

In general, the boundary of an object should be simply connected, i.e., an edge segment should intersect with at most one edge segment at its end points. This constraint is imposed on conditional probability distributions between the edge nodes and the related vertex nodes as follows:

$$p(v_t^2|pa_{V_t}) = \begin{cases} \geq 0.5, & \text{if exactly two parent nodes are true;} \\ 0.3, & \text{if none of the parent nodes are true;} \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where pa_{V_t} represents a configuration for the parents of V_t . In words, if a corner is detected, the measurement M_{V_t} is $m_{V_t}^2$ (meaning that it is in the boundary). According to the conditional probability distribution of M_{V_t} given V_t , the vertex node has a high probability of being v_t^2 (belonging to the boundary). In such a case, it is most likely that exactly two parent nodes are true. This implies the simple connectivity of edges at this vertex. We set the value of $p(v_t^2|pa_{V_t})$ to 0.3 when vertices are detected in the background. In such a case, it is possible that none of the parent edge segments are true boundary. However, the conditional probability for this case shall be smaller than the case that exactly two parent nodes are true. On the other hand, if no corner is detected, then the vertex node V_t has a high probability to be false and it gives no additional information to infer the category of the parent edges.

The probability distributions $p(M_{E_j}|E_j)$ are continuous, but are always observable. We have learned Gaussian distributions from training images and their manual labeling, where we classified edge segments into two sets: boundary edges and non-boundary edges. We fit a Gaussian distribution for the distribution of edge measurements in each set. We do not discuss further on these measurements nodes, as we focus on the advantages of employing SQPN against the standard Bayesian approach, and the same procedure was performed in both cases. Nevertheless, we note that all continuous nodes are observable and we can easily obtain a similar discrete model.

Given the SQPN constructed as in Figure 2(b) and its parameters, the goal of image segmentation may be achieved by inferring the most probable categories of the variables given the observations (measurements), that is, we look for the categories of E, V, M_E, M_V that maximizes

$$\max_p p(E, V, M_E, M_V) = \prod_t p(V_t|pa_{V_t})p(M_{V_t}|V_t) \prod_j p(E_j)p(M_{E_j}|E_j).$$

The most probable categories can be found using the algorithm of Section 3. Figure 3 shows the segmentation results obtained after applying the SQPN described here (right

column). On the left column of Figure 3, the solution with standard Bayesian networks is presented for comparison. In those cases, the probability measure $p(v_t^2 | pa_{V_t})$ was set to 1 if exactly two parent nodes are true, because we want to force continuous contours for images (we were not sure if 1 was the best value for that measure. That is one of the motivation for the use of SQPNs). Although the decision of which result is better might be subjective, we visually consider that the solution with SQPN is superior to the solution with a Bayesian network. In the fish image, the contour of the fish is better detected. The SQPN used for this image has 84 nodes, of which 16 are vertex nodes, 26 are edge nodes and others are measurement nodes. In the cow image, the contour of the cow is closed and less edges are produced by the SQPN version. The employed SQPN has 280 nodes, of which 56 are vertex nodes, 84 are edge nodes, and others are measurement nodes. In the mushroom image, the segmentation result using the SQPN has also less edges and induced regions, so it is smoother. The SQPN used for the mushroom image has 440 nodes, of which 85 are vertex nodes, 135 are edge nodes and others are measurement nodes. Finally, we note that the only distinction between the model we employed here and a standard Bayesian network approach relies on the definition of interval probabilities in some nodes. We have kept our model close to the standard network to clearly show the difference in the results. Further investigation on other constraints and intervals is intended in a future work.

6 Conclusion

We present a definition of SQPN where range constraints are allowed together with usual qualitative influences. We permit monotonic and non-monotonic relations, and although we have not worked with other usual qualitative constraints found in the literature such as synergies, their inclusion is straightforward. For instance, no changes are needed to described learning and inference methods.

An inference algorithm for belief updating, most probable explanations and maximum a posteriori queries is discussed. It converges to a local optimum solution and has achieved good empirical results. We have also explored the combination of SQPNs and empirical data, an issue of clear practical importance. Two computer vision problems are treated. First, a facial action unit recognition where learning and reasoning are performed using an SQPN. Recognition rates have improved when compared to a standard Bayesian network in cases of interval dominance. Then, an SQPN is constructed for image segmentation. The segmentation is solved using a most probable explanation, and visual results indicate benefits of this approach.

We emphasize that the application of SQPNs to computer vision problems is the most important contribution of this work. Although the experiments conducted here were far from real computer vision challenges, good results obtained for the two problems indicate that the use of imprecise probability techniques such as SQPNs in real computer vision is promising. As future work, we intend to improve the models for recognition and segmentation, increasing the use of qualitative relations. Furthermore, SQPNs can be applied to other computer vision problems, such as object and human body tracking. Future research is also considered in this direction.

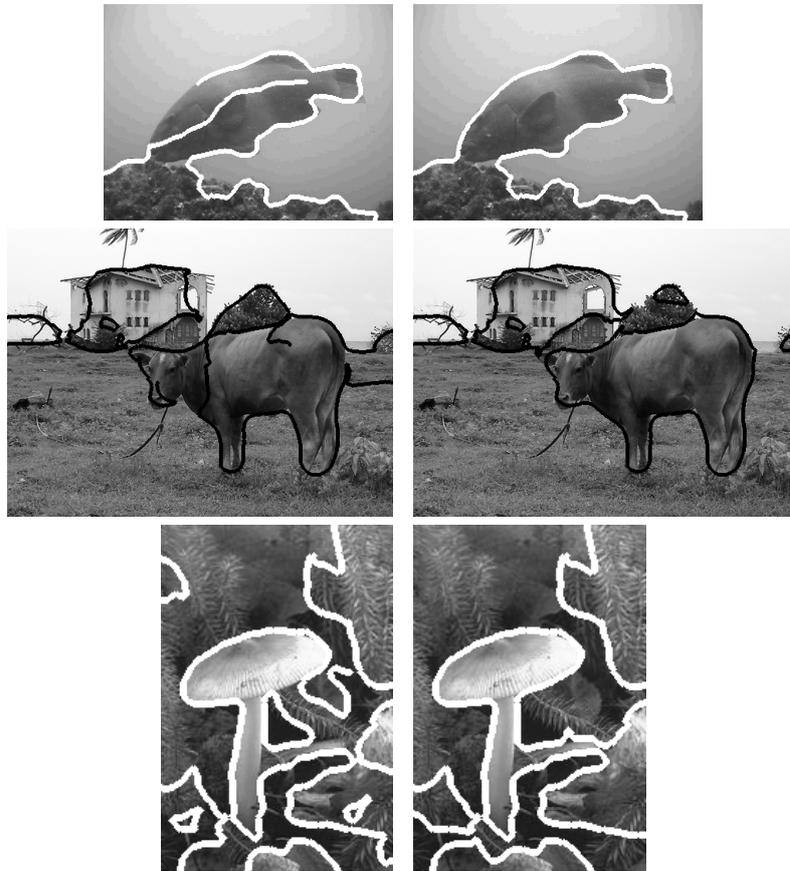


Figure 3: The left column contains the solutions of image segmentation using Bayesian networks. The right column contains the corresponding solutions with SQPNs.

Acknowledges

We thank the anonymous reviewers for valuable comments and suggestions.

References

- Bartlett M.S., Littlewort G.C., Frank M.G., Lainscsek C., Fasel I.R., Movellan J.R., 2006. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia*, 1(6), 22–35.
- Bernard J.M., 2005. An introduction to the imprecise dirichlet model for multinomial data. *International Journal of Approximate Reasoning*, 39(2-3), 123–150.
- Cano A., Gómez M., Moral S., 2007. Credal nets with probabilities estimated with

- an extreme imprecise dirichlet model. In *International Symposium on Imprecise Probabilities: Theory and Applications*, 57–66.
- Caselles V., Kimmel R., Sapiro G., 1997. Geodesic active contours. *International Journal of Computer Vision*, 22(1), 61–79.
- Couso I., Moral S., Walley P., 1999. Examples of independence for imprecise probabilities. In *International Symposium on Imprecise Probabilities and Their Applications*, 121–130. Ghent, Belgium.
- Cozman F.G., 2000. Credal networks. *Artificial Intelligence*, 120, 199–233.
- da Rocha J.C.F., Cozman F.G., 2002. Inference with separately specified sets of probabilities in credal networks. In *Conference on Uncertainty in Artificial Intelligence*, 430–437. Morgan Kaufmann, San Francisco.
- da Rocha J.C.F., Cozman F.G., de Campos C.P., 2003. Inference in polytrees with sets of probabilities. In *Conference on Uncertainty in Artificial Intelligence*, 217–224. Morgan Kaufmann, New York.
- de Campos C.P., Cozman F.G., 2004. Inference in credal networks using multilinear programming. In *Second Starting AI Researcher Symposium*, 50–61. IOS Press, Valencia.
- de Campos C.P., Cozman F.G., 2005a. Belief updating and learning in semi-qualitative probabilistic networks. In *Conference on Uncertainty in Artificial Intelligence*, 153–160.
- de Campos C.P., Cozman F.G., 2005b. The inferential complexity of Bayesian and credal networks. In *International Joint Conference on Artificial Intelligence*, 1313–1318.
- Druzdzel M.J., Henrion M., 1993. Efficient reasoning in qualitative probabilistic networks. In *AAAI Conference on Artificial Intelligence*, 548–553.
- Ekman P., Friesen W.V., 1978. *Facial action coding system: A technique for the measurement of facial movement*. Consulting Psychologists Press, Palo Alto, CA.
- Fasel B., Luetttin J., 2003. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36, 259–275.
- Gonzalez R.C., Woods R.E., 2002. *Digital Image Processing (2nd Edition)*. Addison-Wesley.
- Harris C., Stephens M.J., 1988. A combined corner and edge detector. In *Alvey Vision Conference*, 147–151. Manchester.
- Kanade T., Cohn J.F., Tian Y., 2000. Comprehensive database for facial expression analysis. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 46–53.

- Kass M., Witkin A., D.Terzopoulos, 1988. Snakes: Active contour models. In *International Journal of Computer Vision*, volume 1, 321–331.
- Lukatskii A.M., Shapot D.V., 2001. Problems in multilinear programming. *Computational Mathematics and Mathematical Physics*, 41(5), 638–648.
- Pantic M., Rothkrantz L.J.M., 2000. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 1424–1445.
- Parsons S., Dohnal M., 1993. A semiquantitative approach to reasoning in probabilistic networks. *Applied Artificial Intelligence*, 7, 223–235.
- Renooij S., van der Gaag L.C., 1999. Enhancing QPNs for trade-off resolution. In *Conference on Uncertainty in Artificial Intelligence*, 559–566.
- Renooij S., van der Gaag L.C., 2000. Exploiting non-monotonic influences in qualitative belief networks. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, 1285–1290. Madrid.
- Renooij S., van der Gaag L.C., 2002. From qualitative to quantitative probabilistic networks. In *Conference on Uncertainty in Artificial Intelligence*, 422–429. Morgan Kaufmann Publishers, San Francisco.
- Tong Y., Liao W., Ji Q., 2007. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1683–1699.
- Walley P., 1991. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London.
- Walley P., 1996. Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society B*, 58(1), 3–57.
- Wellman M.P., 1990. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3), 257–303.