Learning Probabilistic Structure Graphs for Classification and Detection of Object Structures

Johannes Hartz
Cognitive Systems Laboratory
Department of Informatics
University of Hamburg
Email: hartz[at]informatik.uni-hamburg.de

***Abstract*—This paper presents a novel and domain-independent approach for graph-based structure learning. The approach is based on solving the Maximum Common Subgraph-Isomorphism problem to generalise a model graph over a set of training examples. Then a full probabilistic model is assigned to the learnt graph. We call this approach Probabilistic Structure Graphs (PSGs). This article explains how PSG models are learnt and how probabilities for model instances are derived. It shows how to use PSG models to perform MAP classification, and presents evaluation of learnt models in the context of image understanding. Here, we classify observable object structures in the domain of building facade images (average classification rate $\approx 80\%$). Additionally, we present encouraging results from interpreting facade images, where we detect instances of learnt models in a set of cluttered objects. We show that bottom-up scene interpretation based solely on learnt models seems achievable, without any hand-crafted domain knowledge.**

I. Introduction

To allow a knowledge based system to perform tasks in the field of image understanding, structure models for different classes of meaningful object aggregations are needed. This paper presents an approach for supervised learning of such models. They are learnt from images, where meaningful object aggregations have been annotated in terms of set descriptions of their parts. From these annotations a graph representation is derived, to serve as an example of the object aggregation class. The learning approach is based on finding Maximum Common Attributed Subgraphs of these training examples, to determine a generalised model graph. To provide additional information, the projections of the examples on the generalised model are recorded in a node combination histogram and used later-on, to compute probabilities for model instances. Based on these probabilities, Maximum A-Posteriori Probability (MAP) classification is performed.

This work is part of the eTRIMS project[1], and is a follow-up to our previous approach, where we used a logic-based representation inside a Version Space framework, to learn structure models for high-level scene interpretation of building facade scenes [1], [2], [3]. The results made clear that albeit the fact that crisp representations work for carefully selected examples, they are not suitable for application in complex real-world domains with highly heterogeneous data. In contrast, PSG models do not pose any restrictions on the heterogeneity of the data and show good performance.

In the last decade, probabilistic models have been a primary research field for structure modelling and interpretation (e.g. [4], [5], [6]). Unfortunately, probabilistic models for complex domains suffer from the curse of dimensionality [7], which shows itself in interconnected problems: (1) Joint probability distributions are needed to observe the dependencies between objects, but have exponential space complexity. So in practice, for observations made in a complex domain, the method is very costly. (2) To make statistically profound predictions from a probabilistic model, a broad data basis is required. For the scenario of structure learning as for example in the eTRIMS project, a suitable data basis is not acquirable, because the amount of manually annotated training data needed to sufficiently cover the example space is too large. (3) Learning the structure of probabilistic models like Bayesian Networks or Markov Random Fields is still an open questions, which has only been solved under specific constraints.

We have developed the PSG model approach to tackle the above problems in two ways: To reduce the dimensionality of the training data and to make structure models learnable (allowing arbitrary heterogeneity of training examples), we first use a graph-based representation to generalise over all training examples. Then we treat the projections of the examples on the graph model as observations for the class to be learnt. This means we do not base the probabilistic model on the real-world observations (with high dimensionality), but on the indirect observations inside the generalised model, which makes the dimensionality tractable. The problem of a sufficient data basis also arises for PSG models. To counter this problem, we present a new method to generalise over the observations made. Contrary to a pure frequentist probability approach, we interpret single examples as observations for different subparts of the model. We show empirically that these *generalised observations* allow for better predictions than frequentist probabilities, when the training set is small. Thus we are able to make reasonable predictions from models with small training sets.

The outline of the article is as follows: In the next section we describe the PSG model representation. Section III shows how PSG models are learnt over a set of examples. Section IV then explains how probabilites for model instances are derived,

[1]This research has been supported by the European Community under the grant IST 027113, eTRIMS - eTraining for Interpreting Images of Man-Made Scenes

and how learnt models are applied in image understanding. In Section VII we evaluate learnt PSGs in the buildings domain and present the results we have achieved in structure classification and structure instance detection. Section VIII finally gives a summary and an outlook on future work.

II. Model Representation

A learnt Probabilistic Structure Graph model is comprised of two components: A structure graph, to represent the structure model generalised over the examples, and a joint probability distribution, which predicts the probabilties for node combinations. To derive this distribution, we record which part of the model m corresponds to each of the examples e_i, in terms of the projection of the nodes of e_i on m. We record this statistic in a matching table, to derive a node combination histogram for the model. This two-fold representation enables us to have a fully generalised structure model, while retaining the information which sub-models have stronger or weaker support from the examples.

A. Structure graph

The structure graph $G = (N, E)$ in a PSG model is a directed graph with two edges in opposite directions for each pair of adjacent nodes. The nodes $N = \{n_1, \ldots, n_j\}$ are labeled with object types. In the eTRIMS domain, which is concerned with modelling building facades, this includes different objects like *Windows, Doors, Railings, Stairs, Signs, Vegetation, Cars, Canopies, Dormers,* etc.... The edges $E = \{e_1, \ldots, e_k\}$ are labeled with symbolic relation descriptors, to characterise the relations between the objects in the graph

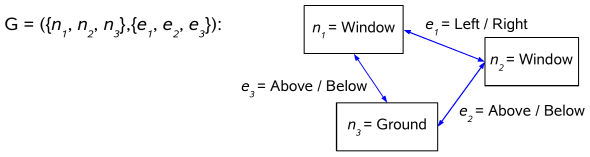

Fig. 1. Structure graph example (pairs or directed edges are contracted)

(see Figure 1). The basic edge types used for modelling this domain are a superset of RCC5 [13], including spatial relation descriptors for overlapping and non-overlapping objects. For non-overlapping relations the descriptors for both 8 and 4-neighbourhood are used (e.g. *Left*, *Above-Right*, ...). For overlapping relations the different types of overlap (*Inside*, *Outside*, *Overlap*) are observed individually for each image axis (e.g. *InsideX-OutsideY*). Additionally, we observe alignment relations between the objects (*AlignedTop, AlignedBottom, AlignedLeft, AlignedRight*); and all spatial relation descriptors have a fuzzy variant (e.g. *Fuzzy-Below-Right*). The basic relation descriptors and their combinations form a taxonomy of spatial relations, as shown in Figure 2 (three dots indicate that (x) more relation combinations are possible, fuzzified relations are omitted at all). The total number of possible different spatial relation descriptors is larger than 150.

We have devised the basic relation descriptors in such a way, that they are widely overlapping. If several of these descriptors hold true for an observation, they are combined to a more specific one (e.g. *Above* and *Left* are combined to *Above-Left*). Hence the basic relation descriptors and their combinations form a taxonomy of possible relations for objects in the application domain (see Figure 2). The basic descriptors are the root nodes of this taxonomy, because they are not orderable in terms of generality. The combinations of relation descriptors constitute more specific relation types. The taxonomy of relation descriptors is used to determine the general-specific ordering of relation types during learning and for classification.

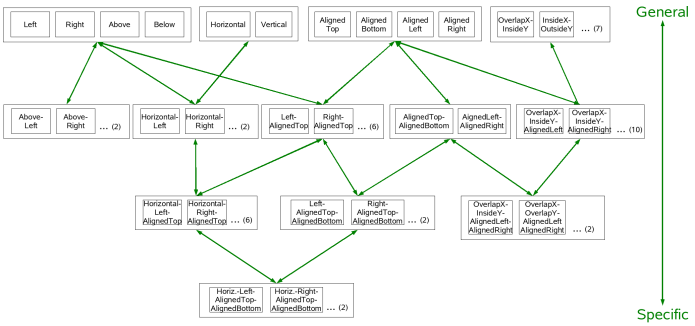

Fig. 2. Spatial relation taxonomy for the eTRIMS domain

B. Joint probability distribution

To derive the joint probability distribution for the co-existence of nodes, we use a node combination histogram which we record during learning. More precisely, we record the different node sets of the model on which the examples have been projected. To do this, we use a matching table that holds the labels of all nodes in the structure graph as columns (see Figure 3). The number of columns (i.e. the number of nodes in the structure graph) is denoted l. The number of rows (i.e. the number of training examples) is denoted r. How exactly projections are recorded, is presented in the next section. How we derive a joint probability distribution for the existence of nodes from this data, is presented in Section IV.

Example # \ Model node	n_1	n_2	n_3	n_4	n_5	n_6
1	1	1	1	1	0	0
2	0	1	1	0	1	0
3	0	0	1	1	0	1
4	1	0	0	0	1	1

Fig. 3. Matching table example

III. Model Learning

A PSG learner L consists of several PSG models m_{c_i} for different structure classes $c_1, \ldots, c_n$. In the eTRIMS domain, these are classes like *Entrance, Balcony* and *Window-Array*. Since all models are learnt independently, we refer to learning one model m in the following section. All training examples for m are presented as PSG structure models with the most

specific edge descriptors applicable. Whenever a new example e is presented, there are different courses of action depending on r (the number of examples presented already):

- If $r = 0$: L adopts e as its model m
- If $r > 0$: L generalises its model m *minimally* to fit e

Whenever a PSG model m is generalised, we demand this generalisation to result in a *minimal* model, in the sense that (1) the generalised model has as few nodes as possible, and (2) the edge generalisation cost is minimised. The main goal here is to reduce model complexity while trying to keep the model as specialised as possible. There are three steps to generate the *minimal generalisation(s)* of a PSG model, which are explained in detail in the next three sections:

1) Determine *Generalisable Subgraph-Isomorphism* to find the **largest** common generalisable subgraph(s) of m and e.
2) If there are several largest common generalisable subgraphs, we choose the one which minimises the *generalisation cost* as basis for the model generalisation.
3) Perform structure graph generalisation and update the matching table.

If there are cases in which several different generalisations are considered *minimal* (i.e. there are several largest common generalisable subgraphs and their generalisation cost is equal), we retain all of them, because we cannot decide for a single model regarding the criterion of *minimal generalisation*. If a multiple internal model representation is not suitable for later application, we record all further generalisation costs for each of the different models, and decide for the one with the smallest generalisation cost sum over all examples.

A. Generalisable Subgraph-Isomorphism

Subgraph-Isomorphism is defined as a bijection between two graphs that preserves the edge structure regarding the adjacency of nodes. This means G is *Subgraph-Isomorph* to m' and e', if and only if there are bijections $m' \leftrightarrow G$ and $e' \leftrightarrow G$ such that nodes that are adjacent in m' are also adjacent in G and nodes that are adjacent in e' are also adjacent in G.

We define *Generalisable Subgraph-Isomorphism* as a bijection between two graphs that is *subgraph-isomorph*, and allows edges to be matched only if they have a common generalisation, according to a given taxonomy of edge types. Nodes can only be matched if they have the same node type. Following from this definition, *Generalisable Subgraph-Isomorph* is a specialisation of *Subgraph-Isomorphism*, because the structure and the edge types of the graph are considered. For the examples from the buildings domain, we evaluate the generalisability of edge types through the taxonomy presented in Figure 2.

A graph G is called a common generalisable subgraph of m and e if it is *Generalisable Subgraph-Isomorph* to a subgraph m' of m and a subgraph e' of e.

B. Edge generalisation and cost function

The generalisation cost function is devised to prefer the most specific edge type generalisation. To determine the generalisation cost for a common generalisable subgraph, we evaluate the cost for generalising the edges $E = \{e_1, \ldots, e_n\} \in m'$ with the matching edges $F = \{f_1, \ldots, f_n\} \in e'$. The generalisation cost c_{ij} for generalising $e_i \in m'$ with $f_j \in e'$ is determined as:

$$c_{ij} = \frac{(d_i - d_{gen}) + (d_j - d_{gen})}{2 \cdot d_{max}} \,. \tag{1}$$

where d_k is the depth of the edge type e_k in the edge taxonomy, d_{gen} is the depth of the generalised edge type and d_{max} is the maximum depth of the taxonomy. If there are several possible edge type generalisations, the one with minimal costs is chosen. We accumulate the costs for generalising all edges $E = \{e_1..e_n\} \in m'$ to the matching cost sum c.

C. Model generalisation procedure

To generalise a PSG model, the structure graph and the matching table are adapted. The action taken to generalise the structure graph in m with an example e depends on the size of the largest common generalisable subgraph(s) of m and e:

- If e is *Generalisable Subgraph-Isomorph* to m:
  e is matched on m and the edges $E \in m$ are generalised with the edges $F \in e$.
- If a subgraph e' of e is *Generalisable Subgraph-Isomorph* to m:
  e' is matched on m and the edges $E \in m$ are generalised with the edges $F \in e'$. Nodes and edges in e but not in e' are added to m preserving the structure they had in e.
- If no subgraph e' of e is *Generalisable Subgraph-Isomorph* to m:
  Nodes and edges in e are added to m preserving the structure they had in e. Note that this leads to unconnected subgraphs in the generalised structure model.

When the structure graph has been generalised, the matching table needs to be updated. In the matching table we record the node set of the generalised model m, on which the node set of example e has been projected. For each example, we update the matching table such that we first add a new row. For all the nodes $n_{i..k} \in e$ that could be matched on m before generalisation, we record this in the existing columns in the matching table. For all nodes $n_{j..l} \in e$ that could not be matched on m before generalisation, we add a column to the table and record their first occurence in the structure model.

IV. Deriving probabilities

In this section we present a procedural description of how probabilities for possible instances of the structure model are derived. Generally, when we apply PSG models, we call a set of objects to be evidence E. To decide which probability is assigned to evidence E having been caused by model m, we first determine if E can be classifed by the structure graph of m. If not, the probability is zero.

To assess structure graph classification, we check if the structure graph of model m allows for matching of E, by evaluating *Subsuming Subgraph-Isomorphism* between m and E. In practice, *Subsuming Subgraph-Isomorphism* ensures that all restrictions in terms of graph structure and edge descriptors imposed by the model, are fulfilled by the matched evidence.

We define *Subsuming Subgraph-Isomorphism* as a bijection between two graphs g_1 and g_2 that is *subgraph-isomorph*, and allows edges of g_2 to be matched only if the respective edge types of g_1 are equal or more general, according to a given taxonomy of edge types. Nodes can only be matched if they have the same node type.

If the evaluation of *Subsuming Subgraph-Isomorphism* is successful, it results in at least one projection $E \rightarrow m' \subseteq m$. If several projections are possible, we decide for the one with minimal costs, according to Equation 1. In the next step, we now evaluate the probability for the subgraph m' to be matched, given the observations in the matching table. We propose two different methods for this, which are presented in Section IV-A and Section IV-B. The first approach directly translates the frequency of recorded projections (i.e. observed node combinations), into a distribution of the joint probability of model nodes to exist. The second approach assumes that one observation can raise the frequency of other observations, weighted by the size of their intersection. This means that since we know where in the model graph each example has been projected, we use these observations to interpolate the probability for not observed instances. This method allows us to learn models from small training sets ($\approx$ 40 examples), which make meaningful predictions of a high dimensional probability space.

A. Frequentist observations

Given a generalised structure model m and its matching table t with r entries, we interpret each row of the matching table (i.e. the node set of m on which example e_i has been projected) as an example for the distribution D over all possible node combinations. We then use distribution D to derive probabilities for possible model instances, which are constituted by a set of nodes and their spatial relations described in the structure graph. The overall process is as follows:

To determine the probability for evidence E, we check if evidence E can be matched on m according to *Subsuming Subgraph-Isomorphism*. If so, we retain the resulting projection $E \rightarrow m'$, to evaluate how often the exact node set of m' has been matched during training. This indicates the probability for the evidence given the training examples. We do this by counting the respective rows in t:

$$o = \sum_{i=1}^{r} \begin{cases} 1 & \text{if } e_i \rightarrow m' \\ 0 & \text{else} \end{cases} . \tag{2}$$

The maximum number of examples for distribution D is the number of rows in t:

$$o_{max} = r . \tag{3}$$

Then to derive the probability for evidence E to be of class c, we compute:

$$P(V = v|C = c) = \frac{o}{o_{max}} . \tag{4}$$

B. Generalised observations

This section describes how we generalise over the node combination histogram recorded for model m, to be able to make more extensive interpretations of unseen evidence. This can be paraphrased as the process of generating synthetic examples for the node combination distribution D, guided by the observations that we have made. In practice, we interpret each projection recorded in t as an observation for different subparts of the model (contrary to the *frequentist observations* approach, which counts each projection as exactly one observation). More precisely, we do not just count occurences of projections $e_i \rightarrow m'$ in t (where m' is the part of the model matched by E), but also all projections $e_i \rightarrow m''_c$, where the node set N'' of m''_c has an intersection with the node set N' of m'. This means we also derive probabilities from observations, if they only intersect with the part of the model matched by the evidence E. We weigh these observations by the size of the intersection of the node sets of m''_c and m', to be > 0 and ≤ 1. Note that when we employ *generalised observations* to generate distribution D, we generate a complete joint probability distribution over all node combinations in the graph with probabilities > 0.

Let N' be the node set of m' (the part of the model matched by E), and N_i be the node set of m''_c (the part of the model matched by example e_i). To derive the sum of all *generalised observations* from the examples we compute:

$$o = \sum_{i=1}^{r} \frac{(\mid (N' \cap N_i \mid)^2}{\mid N' \mid \cdot \mid N_i \mid} . \tag{5}$$

Since we count each example as an observation for multiple subsets of the model, we now have to compute the sum of all possible *generalised observations*, to derive a normalised probability for evidence E. The sum is computed using the following equation, where N_i is the node set of the respective example e_i:

$$o_{max} = \sum_{i=1}^{r} \sum_{n=1}^{l} \sum_{e=max(1,n-l+|N_i|)}^{min(n,|N_i|)} \binom{\mid N_i \mid}{e} \cdot \frac{e}{\mid N_i \mid} \cdot \binom{l- \mid N_i \mid}{n-e} \cdot \frac{e}{n} . \tag{6}$$

Then to derive the probability for evidence E to be of class c, we compute:

$$P(V = v|C = c) = \frac{o}{o_{max}} . \tag{7}$$

To assess the prediction performance of *generalised observations*, we have compared the *frequentist observations* and the *generalised observations* scheme empirically. For this experiment, we have drawn samples from a synthetic distribution modelling the joint probability over all possible node combinations of a graph with 6 nodes (using unique node labels and neglecting edge labels). We have used these

samples to train a PSG model with them. For the training we have ensured perfect matching of the samples, to neglect the matching problem which is inherent in the learning process. Then we predicted the unknown distribution from the learnt model, using the *frequentist observations* and the *generalised observations* method. We have compared the maximum error of the predictions made w.r.t. to the number of samples used for training. The results shown in Figure 4 are averaged over 100 validations of this experiment. They show that *generalised observations* reveal a better prediction performance for an unknown node combination distribution than *frequentist observations*, if the number of examples is < 200. Further experiments have even shown that this number significantly increases, when the number of nodes in the PSG model is increased.

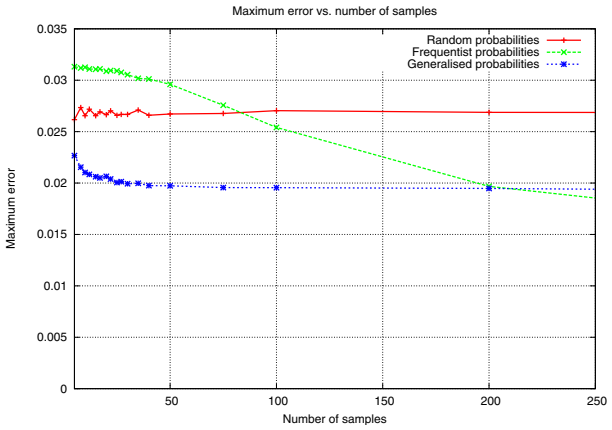

Fig. 4. Prediction performance of frequentist-like and generalised probabilities

V. MAP CLASSIFICATION OF OBJECT STRUCTURES

To classify evidence v that we know to be of one of the classes $c_1, \ldots, c_n$, we use the normal Bayesian formulation to derive the maximum a-posteriori probability (MAP):

$$P(C = c|V = v) = \frac{P(V = v|C = c) * P(C = c)}{P(V = v)} \; . \quad (8)$$

where $P(V = v|C = c)$ is computed as shown in either Section IV-A or IV-B. To compute the class probability we use:

$$P(C = c) = \frac{o_{max}}{\sum_{i=1}^{n} o_{max_{c_i}}} \; . \quad (9)$$

where $o_{max_{c_i}}$ is the maximum number of observations for class c_i. To compute the probability for evidence v to be observed at all, we use:

$$P(V = v) = \sum_{i=1}^{n} P(V = v|C = c_i) * P(C = c_i) \; . \quad (10)$$

To evaluate a set of learnt models, we use these formulae to perform MAP classification of object structures on a test set of instances. We have performed this evaluation for the models from the eTRIMS domain and present the results later in this article.

VI. SCENE INTERPRETATION

Scene interpretation is a term commonly used for vision tasks going beyond object recognition, such as the detection of object structures and the prediction of missing objects. As explicated in [8], scene interpretation can be modelled formally as a knowledge-based process. The burden of the interpretation process lies on the conceptual models, and the richer a domain, the more demanding is the task of learning these models. In our work, an object structure model specifies a set of objects with certain properties and relations which together constitute a meaningful scene entity. Figure 5 shows an example, where a conceptual model for the aggregate "Window Array" (trained over a set of examples) is applied to automatically detected scene objects [9], [1], [10]. The resulting scene interpretation shows that the structure model rejects false positive recognitions of windows, and leads to the prediction of unrecognised windows.

We are currently working on the application of PSG models

Fig. 5. Detected objects and resulting scene interpretation

to perform scene interpretation in the eTRIMS project. We use PSG models here to explain a set of cluttered objects, in terms of imposing the most probable composition of structures on them. To do this, we request several services that PSG models offer: We use them to detect possible structure model instances in the cluttered scene, to classify the detected instances and to compute the most probable combination of model instances, which is the most probable interpretation. Experiments have been carried out successfully and the results are shown in Figure 8, but for the sake of compactness, we cannot present the whole interpretation scheme here. A key requirement for using PSG models in scene interpretation is the need for a rest class, to cope with possibly incomplete models.

VII. PSG APPLICATION AND RESULTS

The data used in this work is a result of the efforts made in the eTRIMS project [11]. The used benchmark data set (to be published) is similar to [12]. It consists of 110 images of building facades with object annotations in seperate XML files. The annotations describe objects in the image by a class label and a polygon. This includes different objects like *Windows, Doors, Railings, Stairs, Signs, Vegetation, Cars, Canopies, Dormers,* etc.... Object structures are described by their label and a set description of their parts. This includes *Entrances, Balconies, Window-Arrays, Dormers, Facades, Buildings* amongst others.

Note that the parts of object structures might be scene objects or other object structures, which constitutes a compositional hierarchy of object structures and objects (examples of image annotations can be found in Figure 8).

We consider the annotations made by a single annotator to be a **gold standard**, because they are subjective and not based on pre-defined rules for annotating (other than the labels). It is not possible to prove or disprove them. This makes the learning task more difficult, because not all of the images are annotated consistently. Another source of error lies in the annotated image scale. The scale value is an estimate made by the image annotator, with an approx. error between $0-20\%$. We feel that the inaccuracies in the annotation are not a problem in itself, because they rather reflect the complexity of the application domain, and ensure that the learnt models are robust enough for the real-world.

A. Structure classification experiments

Fig. 6. Example *Entrance*, *Balcony* and *Window-Array* instances with bounding boxes marking the parts

To evaluate the performance of learnt models, we perform MAP classification on a test set of instances of the classes *Entrance, Balcony* and *Window-Array*. Examples of these classes are shown in Figure 6. The classes are widely overlapping, because all instances are mainly composed of the same object types *Window, Door* and *Railing*. The object label *Railing* is used for both the railings of *Entrances* and *Balconies*, as is *Door* for both the doors of *Entrances* and *Balconies*. Hence the classes must be distinguished through the spatial structure of their parts.

All experiments have been carried out using 8-fold cross validation on 40 instances of each class, which have been selected randomly from those available in the benchmark dataset. To show the performance of PSG model learning, we present classification rates and error rates for each class in Table I. These numbers habe been averaged over 10 different random training sequences. We have evaluated the influence of the training sequence order separately, and present the results in the next section. To assess classification errors, we present the confusion matrix for a typical validation in Table II. To evaluate how the performance develops w.r.t. the number of training examples, we present learning and error rate curves in Figure 7 (the curves have been clipped where no trend is visible).

TABLE I
CLASSIFICATION AND ERROR RATES

	Entrance	*Balcony*	*Window-Array*	*Average*
Classification rate	0.82	0.75	0.95	0.83
Error rate	0.12	0.12	0.02	0.09

TABLE II
CONFUSION MATRIX

true \ predicted	*Entrance*	*Balcony*	*Window-Array*	*Rejected*
Entrance	**32**	4	0	4
Balcony	6	**30**	0	4
Window-Array	0	0	**38**	2

B. Results and evaluation

The classification experiments show that PSG models successfully classify instances of object structures in a complex domain with overlapping classes. The classes *Entrance* and *Balcony*, which have more heterogenous examples, show good results, and the class *Window-Array*, which is very homogeneous in terms of the spatial structure of the examples, shows a very good performance. The evaluation of different training sequence orders showed that, albeit the fact that the generalisation scheme is sensitive to the order, the results for different sequences do only differ by approx. $\pm 5\%$.

An interesting observation from the evaluation (see Table III), is that the coherency of the training examples (in terms of their composition and the spatial structure of their parts), has a wide influence on the number of nodes of the generalised structure graph, and also the degree of its completeness. The class *Entrance* has the largest variety of instances, and also shows a huge and rather incomplete structure graph. The class *Window-Array* on the other hand, is very homogeneous and shows a rather small and complete structure graph.

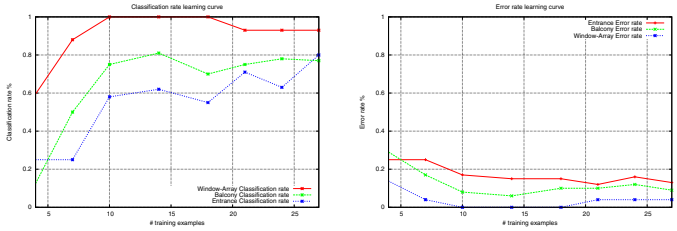

Fig. 7. Classification and error rate learning curves

C. Structure detection experiments

In Figure 8 you find results from structure detection experiments. The system was presented an unknown scene with objects of all types, and was asked to find the most probable interpretation in terms of the most probable structure model

TABLE III
VALIDATION OBSERVATIONS

	Entrance	*Balcony*	*Window-Array*
Avg. example ♯ nodes	3.81	4.14	6.38
Avg. structure graph ♯ nodes	15.6	7.10	8.75
Avg. structure graph completen.	0.38	0.68	1

Fig. 8. Complete annotated images alongside automatically detected structures of the classes *Entrance*, *Balcony* and *Window-Array*

instances. Because of the promising results, we are currently working on an in-depth evaluation of the scene intepretation approach with PSG models.

VIII. SUMMARY AND OUTLOOK

In this article we have presented a novel learning approach for graph-based structure models. The approach combines a well founded graph generalisation scheme with a probabilistic model, to represent the dependencies inside the graph. We have presented two approaches to model probabilties from observing the matchings of training examples on the model. The first approach interprets the observations frequentist-like, but has the drawback to assign probabilities only to a subset of all possible subgraphs of a PSG model. The *generalised observations* approach on the other hand, allows to derive a full joint probability distribution over all nodes in the structure graph (for an arbitrary low number of examples). We have shown empirically that these *generalised observations* yield better prediction performance than frequentist-like observations, for small training sets ($<$ 200 examples). Furthermore, we have shown that the devised probabilistic model can be succesfully employed to perform MAP classification of object structures.

PSG models are learnt for real-world application, which is performing image understanding tasks. The learnt models lend themselves to this task nicely: They allow for structure classification, which has to be performed regularly during the interpretation process. We have shown that PSG models can successfully solve this task, and we have presented comprehensive evaluation of their performance. Furthermore, PSG models allow structure detection in cluttered scenes, which could be shown by examples from scene interpretation in the eTRIMS domain. These results suggest that PSG models are a step towards full bottom-up scene interpretation, based solely on learnt models, without any hand-crafted domain knowledge. A further task in image interpretation would be the prediction of scene objects, to compensate misdetections from low-level image processing modules. Since PSG models feature an underlying probabilistic model, we can use this to predict parts which make a model instance more probable. This is ongoing work in the eTRIMS project. As a basic feature, the developed model representation and learning scheme for Probabilistic Structure Graphs is designed domain-independent. The only domain-dependant part is the edge type descriptors used to describe the relations between the objects in the domain (which could be any kind of descriptors, as long as a general-specific ordering can be imposed on them). Hence we are planning to evaluate PSG models in various other domains, like content based image retrieval, molecule discovery in chemo-informatics or the classification of sentences in natural language processing.

REFERENCES

[1] J. Hartz, B. Neumann, "Learning a Knowledge Base of Ontological Concepts for High-Level Scene Interpretation", *IEEE Proceedings of the Sixth International Conference on Machine Learning and Applications*, pp. 436–443, 2007.

[2] J. Hartz, L. Hotz, B. Neumann, K. Terzic, "Automatic Incremental Model Learning for Scene Interpretation", to appear in *Proceedings of the International Conference on Computational Intelligence (IASTED CI-2009)*, Honolulu (Hawaii, USA), Aug 2009.

[3] T.M. Mitchell, "Version Spaces: An Approach to Concept Learning", PhD thesis, Stanford University, Cambridge, MA, 1978.

[4] K. Murphy, A. Torralba, and W. T. Freeman, "Using the forest to see the trees: A graphical model relating features, objects, and scenes", Proc. of Neural Information Processing Systems, 2003.

[5] K. Sage, J. Howell, H. Buxton, "Recognition of Action", Activity and Behaviour in the ActIPret Project, Künstliche Intelligenz, 2/2005, BöttcherIT Verlag, Bremen, pp. 30–33.

[6] M. Boutell, J. Luo, "Scene parsing using region-based generative models", IEEE Transactions on Multimedia 9(1), pp. 136-146, 2007.

[7] R.E. Bellman, "Dynamic Programming", Princeton University Press, Princeton NJ, 1957.

[8] B, Neumann, "A Conceptual Framework for High-level Vision", Technical report FBI-HH-B-241/02, Universität Hamburg, 2002.

[9] J. Šochman, J. Matas, "WaldBoost - Learning for Time Constrained Sequential Detection", *Proc. of the Conference on Computer Vision and Pattern Recognition*, pp. 150–157, 2005.

[10] L. Hotz, B. Neumann, "Scene Interpretation as a Configuration Task", Künstliche Intelligenz, 3/2005, BöttcherIT Verlag, Bremen, pp. 59–65.

[11] eTRIMS, "E-Training for Interpreting Images of Man-Made Scenes", `http://www.ipb.uni-bonn.de/projects/etrims/`.

[12] F. Korc, W. Förstner, "eTRIMS Image Database for Interpreting Images of Man-Made Scenes", Technical Report TR-IGG-P-2009-01, University of Bonn, 2009.

[13] D.A. Randell, Z. Cui, A. Cohn, "A spatial logic based on regions and connection", *Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, pp. 165-176, 1992.