Learning from Plane Graphs

T. Fannes
Department of Computer Science
Katholieke Universiteit Leuven
Belgium
thomas.fannes@cs.kuleuven.be

J. Ramon
Department of Computer Science
Katholieke Universiteit Leuven
Belgium
jan.ramon@cs.kuleuven.be

ABSTRACT

Planar graphs form an interesting graph class as one encounters them in real world applications, e.g., the representation of images, construction plans and traffic maps. Despite the interest in this type of graphs, up to now there has been no in-depth study of the several learning tasks related to planar graphs and their tractability. In this paper, we make a first step in this direction. In particular, we present a number of learnability results for plane and planar graphs, we list a number of important prediction tasks in the context of applications such as image recognition and spatial reasoning, and we outline the future research needed to acquire a good understanding of these tasks, as well as possible solution methods.

Categories and Subject Descriptors

I.2.6 [**A.I.**]: Learning—*Concept Learning*

Keywords

Concept Learning, Plane & Planar Graphs, Isomorphism

1. INTRODUCTION

Planar graphs occur in real world applications, amongst others to represent images, construction plans and traffic maps. Planar graphs satisfy interesting properties which allow much more efficient processing than general graphs. However, in the literature little attention has been given to exploiting these properties in machine learning tasks. Up to the best of our knowledge, no computational learning theory results focus on the classes of planar and plane graphs. Moreover, in fields considering particular applications, only recently some machine learning approaches handle sufficiently complex patterns to make spatial reasoning useful.

Let us first consider two relevant application domains, image recognition and chemo-informatics (especially QSAR). First, in image recognition, up to recently the focus still was on recognizing pixel maps than on reasoning spatially and exploiting graph properties. For instance, a common strategy in image recognition [6] is to first segment the image into regions of pixels of similar color, obtaining a plane graph, and then to try to identify these regions (the faces of the plane graph).

A lot of work in the domain of image recognition focuses on identifying objects by recognizing pixel maps, e.g., by recognizing their color or texture. One strategy is to learn clusters of frequently occurring pixel patches, called words, and to attach to each image a bag of words like in the case of text documents. A recent example of such a dictionary learning approach is [11].

Even though current image processing systems are quite successful on relatively easy tasks, they fail at analyzing more complex scenes. While the art of recognizing image patches progresses well, much less attention has been paid to combining them and reason about their spatial relations.

A similar shortcoming may exist in current methods for 3D QSAR/QSPR. Quantitative structure-activity/property relationship models relate the structure of molecules with their properties. Since many years, such models consider simple distances [9] between atoms and force fields at certain points [4]. Still, integration of advanced spatial modeling and reasoning has been rather limited.

Recently, some algorithms have been proposed to mine relevant graph classes. Notable examples are found in the work on geometrical pattern mining [14] where frequent subgraphs are mined that also satisfy spatial constraints, and in plane graph mining [17] which mines frequent plane graphs.

To the best of our knowledge, there has been no in-depth study of the different learning tasks related to planar graphs and their tractability. We believe that such a study may be helpful, and combined with existing techniques from both computer vision and computational geometry [7], can significantly enhance the performance of image processing algorithms. We believe that this aproach will, next to improvements in insight and performance, result in more interpretable models, allowing to give an understandable description of the concepts learned.

In this paper, we make a first step in this direction. In particular, we present a number of learnability results for plane and planar graphs, we list a number of important prediction tasks in the context of applications such as image recognition and spatial reasoning, and we outline the future research needed to acquire a good understanding of these tasks, as well as possible solution methods. Our contributions are twofold. First, we present a number of learnability results for plane and planar graphs. Even though those results are preliminary, they form a first step in our analysis and their discussion can contribute to a better understanding of learning spatial shapes and relationships. Next, we list a number of important prediction tasks in the context of applications such as image recognition and spatial reasoning, and we outline a possible roadmap for future research towards to a good understanding of these tasks and possible solution methods.

This paper is organized as follows: In Section 2, we briefly review some basic concepts from graph theory and learning theory. In Section 3, we show that, in plane and planar subgraph isomorphism, the pattern graph is learnable if it has a bounded size. Section 4 deals with learning from ambiguous data. In Section 5, we show some interesting learning settings where the graphs represent two dimensional objects and where stacking objects is allowed. Section 6 sketches a roadmap where we consider learning in a three dimensional context as well as learning ambiguous concepts. Finally, in Section 7, we give some concluding remarks.

2. PRELIMINARIES

We now recall essential definitions from graph theory and computational learning theory needed in the remainder of this paper. For more details see [10] for graph theory and [13] for computational learning theory.

2.1 Graph Theory

Graphs. An undirected, simple graph G is a pair (V, E) where V is a finite set of vertices and $E \subseteq \{\{u, v\} \mid u, v \in V\}$ is a set of edges. We use $V(G)$ to denote the vertices and $E(G)$ the edges of a graph G. $\mathcal{G}$ is the family of all graphs. G' is a subgraph of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. A path is a sequence $v_1, v_2, \ldots, v_n$ of vertices such that there is an edge $\{v_i, v_{i+1}\}$ for all $1 \leq i < n$. A graph G is called connected if for every pair of vertices $v, w \in V(G)$ there exists a path in G connecting both vertices. Otherwise, the graph is disconnected. A graph is n-connected if there is no set of $n-1$ vertices whose removal disconnects the graph. A labeled graph is a triple (V, E, λ) such that (V, E) is a graph and $\lambda : V \cup E \to \Sigma$ is a mapping from its vertices and edges to labels from some alphabet Σ. To make the alphabet explicit, we say that (V, E, λ) is a Σ-labeled graph. In that case, $\lambda(v)$ represents the label of vertex v, and $\lambda(\{v, w\})$ that of edge $\{v, w\}$.

Isomorphism and subgraph isomorphism. Let $G, H \in \mathcal{G}$. Then H is isomorphic to G if there exists a bijection $\phi : V(H) \to V(G)$ such that for every $v, w \in V(H)$: $\{v, w\} \in E(H)$ iff $\{\phi(v), \phi(w)\} \in E(G)$. Labeled graphs G and H are isomorphic if additionally the map ϕ preserves labels of vertices and edges, i.e. $\forall x \in V(G) \cup E(G) : \lambda(\phi(x)) = \lambda(x)$. H is subgraph isomorphic to G, denoted as $H \preceq G$, if H is isomorphic to a subgraph of G. In subgraph isomorphisms $H \preceq G$, we will often refer to H as the pattern and to G as the example or database graph. In general, deciding subgraph isomorphism is known to be NP-complete [3] while graph isomorphism is currently not known to be NP-complete or in P.

Planar graphs. Informally, a graph is planar if it can be drawn in the plane in such a way that no two edges intersect except on a vertex. A planar embedding of a graph G is an injective mapping $\psi : V(G) \to \mathbb{R}^2$ which maps every vertex $v \in V(G)$ to a point in 2-dimensional space ($\mathbb{R}^2$) and every edge to a continuous curve between its endpoints. $\psi(G)$ denotes the planar embedding of G. A graph is planar if it admits a planar embedding such that no two embedded edges intersect. $\mathcal{G}_p$ denotes the class of all planar graphs. Unless stated otherwise, we will restrict our discussion to 2-connected planar graphs.

Plane graphs. A plane graph is a triple $G = (V, E, \psi)$ such that (V, E) is a planar graph and ψ is a planar embedding such that the image of no two edges intersect. A face of G is the region S bounded by a subset of edges of G such that for any $x, y \in S$, x and y can be joined by a curve that does not meet any edge of the embedding (except possibly at x and y). A face is uniquely defined by its boundary, i.e. the cycle of edges that form the perimeter of the face. $F(G)$ denotes the set of all faces of G. A labeled plane graph is a quadruple (V, E, λ, ψ), where (V, E, ψ) is a plane graph and $\lambda : V(G) \cup E(G) \cup F(G) \to \Sigma$ is a labeling function assigning labels from an alphabet Σ to vertices, edges and/or faces of G. The class of all plane graphs is denoted by $\mathcal{G}_e$.

Plane isomorphism and plane subgraph isomorphism. Let $G, G' \in \mathcal{G}_e$. G' is a plane subgraph of G if G' is a subgraph of G and they share the same embedding. For $G, H, H' \in \mathcal{G}_e$, H is plane isomorphic to G if there exists a mapping from H to H' such that H' is a plane subgraph of G and the mapping preserves the faces (and in the labeled case also the labels) in H and H'. We denote this relationship using $H \sqsubseteq G$. For a pattern H of bounded vertex size, planar subgraph isomorphism can be decided in polynomial time (see e.g. the $\mathcal{O}\left(|V(G)||V(H)|^{3|V(H)|}\right)$ algorithm in [8]). Plane subgraph isomorphism has a time complexity of $\mathcal{O}\left(|V(H)||V(G)|\right)$ [5].

In the sequel, we will slightly abuse terminology by using "graph" (resp. "plane graph") to refer to equivalence classes under isomorphism (resp. plane isomorphism) when it is clear from the context.

2.2 Computational Learning Theory

Computational learning theory deals with the question of deciding whether an unknown concept can be learned from a set of labeled examples. The set of all possible instances, called the instance space, is denoted by χ. A concept is a subset of χ. Alternatively, a concept can be defined as a boolean mapping $c : \chi \to \{0, 1\}$ where $c(x) = 1$ indicates that x is a positive example of c and $c(x) = 0$ a negative example. An instance labeled together with its concept value is called a labeled example and is denoted by $e = \langle x, c(x) \rangle$. A concept class $\mathcal{C}$ over χ is a collection of concepts over χ.

Let $EX(c, \mathcal{D})$ be a procedure that on each call returns a labeled example $\langle x, c(x) \rangle$, where x is drawn randomly and independently according to a distribution $\mathcal{D}$. A learning algorithm is said to learn a concept c if, when given as input some labeled examples drawn by calls to $EX(c,\mathcal{D})$, it infers a hypothesis h. The set of all possible hypothesis is called the hypothesis class $\mathcal{H}$. Unless stated otherwise, $\mathcal{H}$ is equal to $\mathcal{C}$.

Definition 1 (PAC learning). *A concept class $\mathcal{C}$ over χ is* Probably Approximately Correct (PAC) learnable *if there exists an algorithm A that satisfies the following: for every $c \in \mathcal{C}$, for every distribution $\mathcal{D}$ on χ, and for all $\epsilon, \delta \in]0, \frac{1}{2}[$, if A is given access to $EX(c,\mathcal{D})$ and inputs ϵ and δ, then with probability at least $1-\delta$, A outputs a hypothesis h satisfying error$(h) \leq \epsilon$ with error(h) the probability that a random instance is labeled wrongly by the hypothesis. This probability is taken over random examples drawn by calls to $EX(c,\mathcal{D})$.*

The input ϵ will be referred to as the error parameter, and δ as the confidence parameter.

Definition 2 (Efficient PAC learning). *A concept class $\mathcal{C}$ is efficiently PAC learnable if if there is an algorithm that runs in time polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.*

The following theorem relates the size of a hypothesis class to the PAC learnability of a concept class [13].

Theorem 1. *Let $\mathcal{C}$ be a concept class of bounded size over χ. If A is an algorithm that, for any concept $c \in \mathcal{C}$, given as input a set S of m labeled examples from c, outputs a hypothesis h consistent with S, then $\mathcal{C}$ is PAC learnable. Furthermore, A is guaranteed to find a hypothesis with a certain confidence and error as in Definition 1, if the set of labeled examples has a size m with*

$$m \geq \frac{1}{2\log(2)\,\epsilon}\left(\log|\mathcal{H}| + \log\left(\frac{1}{\delta}\right)\right).$$

Definition 3 (VC dimension). *Let S be a set of m instances. S is said to be shattered by the concept class $\mathcal{C}$ if, for every boolean vector $b \in \{0,1\}^m$, there exists a concept $c \in \mathcal{C}$ such that $b = c(S)$. The* Vapnik-Chervonenkis (VC) dimension *of $\mathcal{C}$, denoted as VCD($\mathcal{C}$), is the cardinality of the largest set S shattered by $\mathcal{C}$, and is ∞ if S can be arbitrarily large.*

The following theorem links the VC dimension of a concept class to its PAC learnability [2].

Theorem 2. *Consider a concept class $\mathcal{C}$ over χ. $\mathcal{C}$ is PAC learnable if and only if its VC dimension is finite.*

3. LEARNING PLANE AND PLANAR GRAPH CONCEPTS

In this section, we will consider some basic learning settings: the goal is to find the conditions under which a pattern H is learnable in both plane and planar subgraph isomorphism. We will also provide settings under which a pattern H is not learnable.

Definition 4. *Let H be a graph. The concept $c_{H\preceq}$ over $\chi = \mathcal{G}$ defines the subset of graphs G for which H is subgraph isomorphic to G, i.e. for all $G \in \mathcal{G}$: $c_{H\preceq}(G) = 1$ iff $H \preceq G$. The class of all subgraph isomorphism concepts is denoted $\mathcal{C}_{\mathcal{G},\preceq}$. When we restrict χ to planar graphs, the class of all concepts is denoted by $\mathcal{C}_{\mathcal{G}_p,\preceq}$. The class of all planar subgraph isomorphism concepts where the pattern H has a vertex set of size bounded by $n \in \mathbb{N}$ is denoted $\mathcal{C}_{\mathcal{G}_p\preceq,n}$.*

Definition 5. *Let H be a plane graph. We define the concept $c_{H\sqsubseteq}$ over $\chi = \mathcal{G}_e$, in a way similar as Definition 4, as the subset of plane graphs G for which H is plane subgraph isomorphic to G. The class of all concepts $c_{H\sqsubseteq}$ is denoted $\mathcal{C}_{\mathcal{G}_e,\sqsubseteq}$. In particular, we will also study the class of plane subgraph isomorphism concepts where the pattern H has a maximum of n vertices. We denote this concept class $\mathcal{C}_{\mathcal{G}_e\sqsubseteq,n}$.*

We note that the number of different planar and plane graphs with limited number of vertices is bounded, although exponential in the number of vertices.

Definition 6. *For any $s \in \mathbb{N}$, let $\mathcal{L}_{\mathcal{G}_e,s} \subset \mathcal{G}_e$ denote the set of 2-connected (non-isomorphic) plane graphs with s or less vertices, and let $\mathcal{L}_{\mathcal{G}_p,s} \subset \mathcal{G}_p$ denote the set of 2-connected (non-isomorphic) planar graphs with s or less vertices.*

Lemma 1. *Both $|\mathcal{L}_{\mathcal{G}_e,s}|$ and $|\mathcal{L}_{\mathcal{G}_p,s}|$ are bounded by functions exponential in s.*

Proof. In [18] the authors prove that the number of triangulations of a two dimensional point set with s elements, is bounded by 59^s. Each plane graph on s vertices can be represented as a triangulation of these s points, with 0 or more edges removed. Let n_v, n_e and n_f represent the number of vertices, edges and faces in a plane graph and $n_{f\times e}$ the number face-edge-pairs. Every face of a graph has at least three edges, so $n_{f\times e} \geq 3n_f$. And every edge has two faces, so $n_{f\times e} = 2n_e$. In combination with Euler's formula, $n_v + n_f = n_e + 2$, this implies that $n_e \leq 3n_v - 6$. We can conclude that $|\mathcal{L}_{\mathcal{G}_e,s}| \leq 59^s 2^{3s-6} < 472^s$. By definition, a graph is planar if it has at least one embedding in the plane, so we can conclude that the $|\mathcal{L}_{\mathcal{G}_e,s}| \leq |\mathcal{L}_{\mathcal{G}_p,s}|$, and therefore we have also obtained an upper bound on the number of planar graphs. ☐

Even though the classic bound on the number of graphs in general, $2^{|V(G)|^2}$ is better for small graphs, this bound for plane and planar graphs is exponential in s rather than in s^2.

THEOREM 3. *Let s be a positive integer. The concept class $\mathcal{C}_{\mathcal{G}_e \sqsubseteq, s}$ of all concepts $c_{H\sqsubseteq}$ where pattern H has at most s vertices, is PAC learnable.*

PROOF. The concept class $\mathcal{C}_{\mathcal{G}_e \sqsubseteq, s}$ is bounded, so we can apply Theorem 1. We describe a trivial algorithm A_e that, given a set S of m labeled examples, outputs a hypothesis consistent with S. The algorithm will first construct the set $\mathcal{L}_{\mathcal{G}_e, s}$ in an arbitrary order. Because each concept $c_{H\sqsubseteq} \in \mathcal{C}_{\mathcal{G}_e \sqsubseteq, s}$, H will also be in $\mathcal{L}_{\mathcal{G}_e, s}$ and at least one valid hypothesis will be found. Following Theorem 1 and Lemma 1, the concept class $\mathcal{C}_{\mathcal{G}_p \preceq, s}$ is PAC learnable and A_e will output a consistent hypothesis with respect to ϵ and δ when the sample set S contains m samples with

$$m \geq \frac{1}{2\log(2)\,\epsilon}\left(s\log(472) + \log\frac{1}{\delta}\right),$$

which is linear in the maximum number of vertices in the concept pattern graph. □

In practice, instead of exhaustively considering all possible graphs, a more efficient approach would be to take a (maximal) common subgraph of the training examples which does not occur in any negative example, e.g., by starting from the smallest positive example and iterating over its subgraphs.

THEOREM 4. *Let s be a positive integer. The concept class $\mathcal{C}_{\mathcal{G}_p \preceq, s}$, i.e. the set of all concepts $c_{H\preceq}$ where $|V(H)| \leq s$, is PAC learnable.*

PROOF. The proof is analogous to that of Theorem 3: there exists an upper bound on the number of planar graphs with bounded vertex set so we can apply Theorem 1. Therefore $\mathcal{C}_{\mathcal{G}_p \preceq, s}$ is PAC learnable and the size of the required sample set is linear in the maximum number of vertices in the concept graph. □

COROLLARY 1. *Let s be a positive integer. Both $\mathcal{C}_{\mathcal{G}_p \preceq, s}$ and $\mathcal{C}_{\mathcal{G}_e \sqsubseteq, s}$ are efficiently PAC learnable.*

PROOF. Since s is a constant, constructing the set of graphs in A_e and A_p can be done in constant time. Both plane and planar subgraph isomorphism can be decided in time linear in the size of the training examples [5, 8], so the total time complexity for each algorithm is

$$\mathcal{O}\left(\frac{1}{\epsilon}\log\left(\frac{1}{\delta}\right)v_{\overline{G}}\right),$$

linear in the size of the example graphs. □

We have shown that for a bounded pattern size, both plane and planar subgraph isomorphism are efficiently PAC learnable. Note that s is a constant, and thus the algorithms supplied in Corollary 1 are not polynomial in s. In many computer vision applications the learning database contains hundreds of examples and the concept can be a graph of dozens of vertices. For example, in [16], the author uses a synthetical video with an object appearing in each frame. These images are transformed into plane graphs by triangulation and the largest frequent subgraph, the object to track, has 22 vertices. We can achieve a 99% probability of an error smaller than 33% by selecting and labeling 306 frames from that video.

If we consider the class of all graphs instead of plane and planar graphs, learning patterns of bounded size is also efficiently PAC learnable but differs greatly: Firstly, the number of unrestricted graphs is bounded by 2^{s^2}, so the required number of samples is quadratic in the maximum number of vertices. Secondly, its runtime becomes

$$\mathcal{O}\left(\frac{1}{\epsilon}\log\left(\frac{1}{\delta}\right)v_{\overline{G}}^{\,s}\right),$$

so the runtime is no longer linear in the average number of vertices in the example graph. The maximum number of vertices in the pattern determines the degree of the polynomial.

In the next part of this section, we consider learning subgraph isomorphism for unbounded patterns.

THEOREM 5. *Consider the set of all plane graphs $\mathcal{G}_e$. The concept class $\mathcal{C}_{\mathcal{G}_e, \sqsubseteq}$ of all concepts $c_{H\sqsubseteq}$ is not PAC learnable.*

PROOF. We show that the VC dimension of $\mathcal{C}_{\mathcal{G}_e, \sqsubseteq}$ is infinite and by Theorem 2, $\mathcal{C}_{\mathcal{G}_e, \sqsubseteq}$ is not PAC learnable. To do this, we construct an arbitrary large set S shattered by $\mathcal{C}_{\mathcal{G}_e, \sqsubseteq}$, based on the following transformation that maps binary numbers to plane graphs.

Let $g = g_1 g_2 \ldots g_k$ be a binary number of length k (leading zero's not removed). The plane graph G represented by this number is constructed as follows: we start from a cycle of length $k+1$ and transform that cycle by position markers on an edge. A n-marker at edge l is defined by adding $n-1$ new edges to the exterior of the cycle in such a way that together with l, they form the boundary of one face, namely a polygon with n vertices.

To define a unique orientation, we label edge 0 with a 4-marker and number the edges in clockwise order. For the other other edges 1 to k, we will mark edge j iff $g_j = 0$. Figure 1 shows the plane graphs corresponding to two numbers. Let g and h be two binary numbers with the same length and G and H their corresponding graphs. By construction, H is plane subgraph isomorphic to G iff $g_i = 1$ implies $h_i = 1$, in other words $H \sqsubseteq G \Leftrightarrow h \,\&\, g = g$ [1].

We now construct a sequence S of arbitrary size k as follows: the i-th example will be the graph representation of the binary number consisting of all zero's and a single one at position i. For instance, the first element will have markers on each edge except on the first one. For each binary number h of size k, there exists a concept $c_{H\sqsubseteq}$ such that $c_{H\sqsubseteq}S = h$. The concept can be realized by defining H as the graph representation of h: by construction, H will only be plane subgraph isomorphic to the elements at position i where h_i is equal to one. Therefore S is shattered by $\mathcal{C}_{\mathcal{G}_e, \sqsubseteq}$

[1] & represents the bitwise AND operation

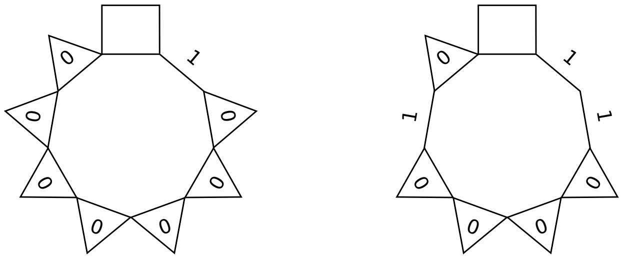

Figure 1: The plane graph representation of 10000000 **and** 11000010**. By construction the right graph is plane subgraph isomorphic to the left one as** 11000010 & 10000000 = 10000000**.**

and thus $\mathcal{C}_{\mathcal{G}_e,\sqsubseteq}$ has an infinite VC dimension. By Theorem 2, $\mathcal{C}_{\mathcal{G}_e,\sqsubseteq}$ is not PAC learnable. □

A similar argument can be used to show that $\mathcal{C}_{\mathcal{G}_p,\preceq}$ is not PAC learnable either.

THEOREM 6. *Consider the set of all plane graphs $\mathcal{G}_e$. Let S be a set of examples labeled by some unknown concept class $c_{H\sqsubseteq} \in \mathcal{C}_{\mathcal{G}_e,\sqsubseteq}$. The problem of deciding whether a consistent plane subgraph exists (CPS) is NP-complete.*

PROOF. We prove that CPS is NP-hard by reducing 3-SAT to CPS. Recall that 3-SAT deals with the question "given a finite set of clauses, each containing the disjunction of exactly three literals, is there some truth assignment for the variables which satisfies all of the clauses?" Let $x_1, \ldots, x_k$ be the variables in an instance of 3-SAT. In this proof we use markers, as defined in Theorem 5.

We start by describing a mapping from a literal to a graph C. C consists of a cycle with $2k+1$ edges. Edge 0 is marked with a 4-marker on the interior of the cycle and edges 1 to $2k$ are marked in clockwise order to define a unique orientation. For the remainder of the edges, edge $2i-1$ represents variable x_i, edge $2i$ represents $\neg x_i$. Each edge will be marked with a 3-marker, except the edge representing the negation of the literal. The graph in figure 2 represents the literal x_1, so only edge $\neg x_1$ is not marked.

We now transform 3-SAT to CPS in such a way that if there exists a consistent hypothesis, it will be a cycle as described above and the solution for 3-SAT can be derived from the markers on the cycle. For each clause, we create a positively labeled graph as follows: for each literal, we create a cycle from that literal. These cycles are marked at every position except at the negation of that literal. We connect these 3 cycles on a unique base, as shown in figure 3. By choosing a unique base, the hypothesis is restricted to a cycle (with markers) only.[2]. The intuition behind this construction is that a consistent hypothesis will have to be subgraph isomorphic to at least one of the three cycles in the graph. For each variable x_i, we create a negatively labeled graph as follows: the graph is a cycle described as above, having 3-markers everywhere but at edge x_i and $\neg x_i$. Hereby we restrict the solution to having at least one 3-marker on each edge pair $(x_i, \neg x_i)$.

[2]If the 3-SAT problem has only one clause, we can duplicate that clause without altering the problem. Hereby we ensure that the base will not be a part of the hypothesis.

Clearly, transforming a 3-SAT problem into a CPS-problem can be done in polynomial time, and so we only need to show that 3-SAT is satisfiable iff CPS has a consistent hypothesis.

- $\Rightarrow$: Suppose that a set of clauses has a satisfiable assignment for all variables. Let H be the cycle graph created by a conjunction of literals containing x_i if the assignment has x_i equal to 1, $\neg x_i$ if it has x_i equal to 0. H will not be isomorphic to any of the negative examples: either the x_i-edge or the $\neg x_i$-edge is labeled with a 3-marker. We now show that H is subgraph isomorphic to all of the positively labeled examples. Suppose there exist a positively labeled graph for which H is not a consistent hypothesis. Let $(z_1 \vee z_2 \vee z_3)$ be that clause. H should have a 3-marker at edges $\neg z_1$, $\neg z_2$ and $\neg z_3$. Then by construction, the assignment should have z_1, z_2 and z_3 set to zero. But this means that the assignment cannot satisfy the clause $(z_1 \vee z_2 \vee z_3)$. Therefore if a 3-SAT problem is satisfiable, H is valid hypothesis.
- $\Leftarrow$: Suppose that H is a valid hypothesis. We will now show that H represents an assignment that satisfies the 3-SAT problem. By the construction of the positive examples, H can only be a marked cycle, and by the construction of the negative examples, at least one of the edges $x_i, \neg x_i$ in each edge pair should be marked. We assign values for the variables in 3-SAT as follows: if edge x_i is marked and $\neg x_i$ not (or vice versa), assign the value 1 (or 0) to the variable x_i. If both edges are marked, choose an arbitrary value for that variable. We now show that this assignment is valid. Suppose that there exists a clause $(z_1 \vee z_2 \vee z_3)$ for which the assignment is not valid, so all the literals are assigned 0. By the construction of the assignment, H should have edges $\neg z_1$, $\neg z_2$ and $\neg z_3$ marked. But the graph for this clause will contain 3 cycles, each missing only one 3-marker, at positions $\neg z_1$, $\neg z_2$ and $\neg z_3$ respectively. Therefore H cannot be subgraph isomorphic to that graph, which contradicts the initial assumptions. Therefore, H is a valid hypothesis and will represent an assignment that satisfies the 3-SAT problem.

We conclude that the CPS problem is NP-hard. A hypothesis can be validated in polynomial time, and therefore CPS is NP-complete. □

We had already shown that $\mathcal{C}_{\mathcal{G}_e,\sqsubseteq}$ was not PAC learnable in Theorem 5, and now Theorem 6 shows that, for a set of labeled examples, we cannot efficiently find a consistent hypothesis. We can use a similar transformation as that used above to show that finding a maximum common subgraph in a set of plane graphs is NP-complete.

We now consider another setting where the size of the pattern graph is unbounded but the graph is rooted on a face and a vertex. The concept graph will thus contain a *start*

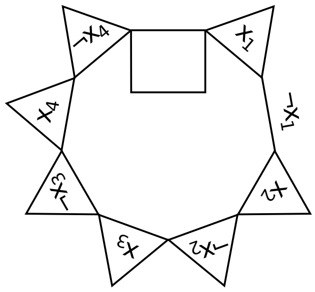

Figure 2: The cycle representing the boolean formula x_1. By construction, the graph will have no 3-marker at position $\neg x_1$.

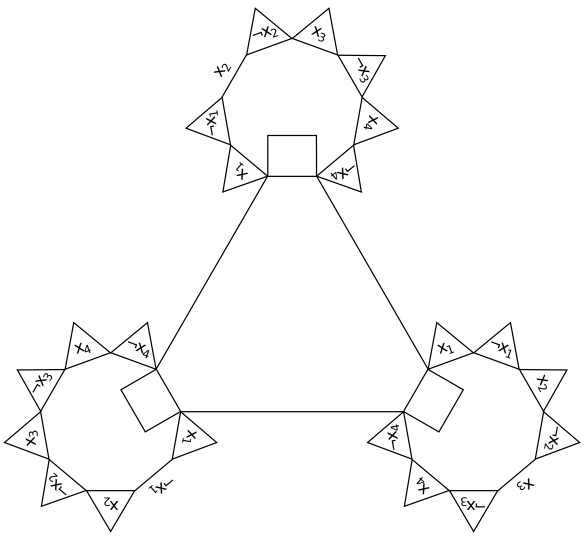

Figure 3: The positive example graph for the clause $(x_1 \vee \neg x_2 \vee \neg x_3)$, with three cycles for x_1 (left under), $\neg x_2$ (center top) and $\neg x_3$ (right under) and a unique base.

face and vertex, and on classifying an instance as positive, this face and vertex will be marked. For example, in the image recognition setting, consider a training set for learning the concept "bike". In each of the positive examples, the frame and the bike seat of a single occurrence of a bike will be tagged in addition to attaching a positive label. This setting is more realistic in tagged images datasets.

THEOREM 7. *Let f be a fixed face of H and v a fixed vertex of f. Consider the instance space $\langle G, f_G, v_G\rangle$ where $G \in \mathcal{G}_e$, $f_g \in F(G)$ and $v_G \in V(G)$. The concept class will label an instance as positive if f_G and v_G are images of f and v respectively, under some plane subgraph isomorphism mapping from H to G. We can learn a valid hypothesis h for a set of labeled examples in polynomial time.*

PROOF. Let H_0 be the graph consisting of only the face f. There is a plane subgraph isomorphism between H_0 and the graph G of each of the positive examples (G, f_G, v_G), mapping f on f_G and v on v_G. One can now extend H_0 by adding faces and keep doing so until no more face can be added while keeping a plane subgraph isomorphism between H_0 and the graph G of each of the positive examples $\langle G, f_G, v_G\rangle$, mapping f on f_G and v on v_G. When adding faces to H_0 there is always at most one choice for extending the plane subgraph isomorphism mappings between H_0 and the graphs of the examples. In this way, one constructs the maximal plane graph which is consistent with the positive examples. Therefore, this graph will not be plane subgraph isomorphic to any of the negative examples. Therefore, in time $\mathcal{O}\left(m v_{\overline{G}}\right)$ with m the sample size and $v_{\overline{G}}$ the average example graph size, one can find a consistent hypothesis. □

Note that this result does not imply that the setting in Theorem 3 is efficiently PAC learnable. Even though we have an algorithm that efficiently and consistently finds a hypothesis, the size of the hypothesis class is unbounded, and therefore we can not apply Theorem 1. Even more, we can use the same construction as in Theorem 5 to prove that it is not PAC-learnable.

4. LEARNING PLANE AND PLANAR GRAPH CONCEPTS FROM AMBIGUOUS DATA

In the previous section we were concerned with learnability from perfect information. In this section we consider learning concepts from ambiguous data. A typical example is the output of an image processing algorithm: A scene is transformed into a plane graph and the faces are labeled with several words. They are the possible interpretations of that face. For example, a classifier could tag one face as either a bike frame or a transparent triangle, an adjacent face as either a motorbike wheel or a bike wheel, etc. As we will see, learning from this ambiguous type of data is tractable too.

THEOREM 8. *Let s be a positive integer constant. Let Σ be a finite alphabet of labels. Consider the instance space $\mathcal{G}_{e,2^\Sigma}$ of all plane 2^Σ-labeled graphs, i.e. graphs whose elements are labeled with subsets of Σ. Consider also the concept class $\mathcal{C}_{\mathcal{G}_{e\sqsubseteq,\vee,s}}$ of Σ-labeled plane graph with at most* *s vertices. An instance G positively exemplifies a concept $\mathcal{C}_{H\sqsubseteq,\vee} \in \mathcal{C}_{\mathcal{G}_e\sqsubseteq,\vee,s}$ iff $(V(H), E(H))$ is plane subgraph isomorphic to $(V(G), E(G))$ under some mapping π such that for all $z \in V(G) \cup E(G) \cup F(G)$, $L(z) \in L(\pi(z))$. The concept class $\mathcal{C}_{\mathcal{G}_e\sqsubseteq,\vee,s}$ is PAC learnable from examples of $\mathcal{G}_{e,2^\Sigma}$.*

PROOF. Just like in Theorem 3 we provide an upper bound on the size of the concept class. We first prove a linear bound on the maximum number of elements in a plane graph. In Lemma 1, we have proved that for a graph H with $|V(H)| \leq s$, $|E(H)| \leq 3s-6$. If we inject these values in Euler's formula ($|F(H)|+|V(H)|-|E(H)| = 2$), we find $|F(H)| \leq 2s-4$. Hence, $|V(H)|+|E(H)|+|F(H)| \leq 6s-10$. Hence, a plane graph H with at most s vertices can have at most $|\Sigma|^{6s-10}$ different labelings. We can now bound the size of the concept class by

$$|\mathcal{C}| \leq |\mathcal{L}_{\mathcal{G}_e,s}|\,|\Sigma|^{6s-10} \leq 472^s|\Sigma|^{6s-10} \leq (472|\Sigma|^6)^s.$$

The required size of the sample set is then

$$m \geq \frac{1}{2\log(2)\,\epsilon}\left(s\log\left(472|\Sigma|^6\right) + \log\frac{1}{\delta}\right).$$

□

Let us consider an example: In [1], the author uses plane graphs to represent the structure of the façade of a house. The vertices represent entities such as a window or a door, and the edges represent relations and are labeled with “close in a vertical direction”, “close in a horizontal direction” and “touch in a horizontal direction”. The examples can be labeled ambiguously, e.g., the difference between “touch in a horizontal direction” and “close in a horizontal direction” is not that distinct so the edge receives both labels. By using a similar technique as in Theorem 8, we can learn the concept of a house with a probability of 90% and an error of 10% from 357 labeled examples.

Note that we assumed here that examples would only be positively labeled if the object H could be present. In fact, in practice it is possible that two structurally isomorphic plane graphs get the same sets of labels for each face, even though the correct labels on both graphs are different. In that case, it is impossible to distinguish between a positive and a negative example, and the error can not be made arbitrarily small. In such case, the best one can hope for is to perform as well as possible given those data characteristics. Learning theory can then give, e.g., a bound on testing data given the performance on the training data.

5. OVERLAPPING GRAPH OBJECTS

So far, we have focused on learning a concept graph embedded in other graphs. This section sketches strategies for learning in two-dimensional scenes: the scene as well as the objects are represented by plane graphs. If A is an object, G_A is the plane graph representing that object. We impose a z-ordering on these objects, e.g., each object appears to be drawn on a paper, cut out and then stacked on each other. Therefore, object A may partially hide object B and as a result, the scene graph contains all of G_A and only part of G_B. It is obvious that next to learning theory and graph theory, computational geometry will play an important role in analyzing such settings.

Learning a partially hidden object

In a first setting, the number of objects is limited to two, A and B, where A lies on top of B. The position of A is limited in such a way that no face of G_B is partitioned by G_A, i.e. the faces of G_B are either fully apparent or hidden. Knowing G_A can we effectively learn G_B? Intuitively we see that this will not be possible: for instance, suppose that G_A overlaps G_B completely in each scene, so actually no information on B is obtained.

There are amongst others the following two approaches to this problem: In the first one, an example is labeled positive iff a fraction of B is visible which is sufficiently large for the labeler to know that B is present in the scene. In combination with an upper bound on the size of G_B, this ensures a sufficient overlap between its subgraphs, from which the structure of B can be learned. Again, alternatively one could consider a second setting where perfect prediction is not possible because there is simply not sufficient information in the image to reveal the real class. In that case, one can bound the difference between performance on training set and test set based on sample set size.

More complex settings derived from overlapping graph objects could also be considered:

- Object A could partially cover B in such a way that it alters some faces of G_B.
- An example is labeled positive if both A and B are visible in the scene. There is no predefined z-order on A and B, so we do not known which object is hiding part of the other, and hence, in contrast to the previous cases, we cannot first learn A and then learn B knowing what part of the scene G_A is hiding.
- Instead of two, the scene contains multiple objects to be learned simultaneously.

In each of those cases, under quite general and realistic conditions one can derive, in a fashion similar to the results in previous sections, that a sample size linear in the size of the patterns to be learned is sufficient. In the latter case with more than two objects however, space and memory requirements of a naive algorithm will be exponential in the number of objects.

6. TOWARDS LEARNING FROM 2D PROJECTIONS

Up to now, we restricted ourselves to two dimensions, possibly with a z-ordering. A next step would be to consider a three-dimensional setting where a plane graph can be seen as a projection of a three dimensional scene. The unknown concept will not be a plane, but a three dimensional graph. As in the two-dimensional setting, certain geometrical properties, e.g., the convexity of an unknown concept, can determine whether a concept is learnable or not.

The simplest setting is to consider a transparent, convex polyhedron. If we limit the projections to those preserving the faces of the polyhedron, the learning question is straightforward: each convex polyhedron has a unique embedding on the unit sphere and, in a projection on the plane preserving faces, one of the faces becomes the outer face on the plane. Hence, the hypothesis class can be partitioned in equivalence classes of plane graphs which can be obtained from each other by choosing a different outer face. Once the correct hypothesis has been identified, we can directly derive the shape of the polyhedron from one of its plane projections.

We therefore will consider in future work two more challenging problem classes: Firstly, a harder-to-learn concept class is obtained by changing the isomorphism requirements. Up to now we used combinatorial isomorphism, where two graphs are isomorphic if they preserve faces. A next step could be to use the actual coordinates of each vertex. In [12], the authors list three other types of isomorphism, each more strict than the previous: projective, affine and congruent isomorphic. They require a projective, affine or orthogonal, respectively, mapping between coordinates. Secondly, a more realistic but also harder setting is when objects are non-transparant and/or the projection does not preserve faces.

An even more advanced step would be to learn three dimensional concepts from ambiguous data or overlapping objects. Positive results in this learning setting would be immediately applicable: most images are projections of three-dimensional scenes, they contain multiple objects and different feature extraction tools give way to ambiguous labeling.

Next to considering the ambiguous instances, be it in two or three dimensions, we could also consider learning ambiguous concepts: a concept can classify examples as positive if one of several different patterns is present, e.g., the classifier *vehicle* will accept all pictures containing a car, a bike or a bus.

7. DISCUSSION

Throughout this paper we have shown that learning a common pattern from plane and planar graph examples is possible if the pattern size is limited. We have shown that it is possible to learn concepts from ambiguous data. In Sections 5 and 6 we considered some interesting learning settings in two and three dimensions which lead directly to more realistic applications. Next to learning theory and graph theory, computational geometry will play an important role in a more in-depth study.

Even though this paper is only a first step in this direction, we can draw some interesting general conclusions. Firstly, from the properties of planar graphs it follows that the number of edges and faces are linear in the number of vertices, which makes more favourable learning results possible. Secondly, it is possible to construct rather complex and realistic scenario's, while keeping both sample size and computational complexity tractable.

The field of computer vision has developed many techniques for interpreting visual scenes, be it low- to medium-level, such as geometric primitives, patches, point clouds, and invariant features [19], but also medium- to high-level, such as composing hierarchical structures and graph-like representations [15]. We believe that the scene interpretation will be improved by adding concept learning in plane graphs to these computer vision techniques. An possible advantage is that the (possibly heavily decorated) plane concepts we consider are more interpretable than many of the current, often subsymbolic, methods. In future work we intend to extend the setting from recognizing objects to reasoning about spatial relationships, giving access to a whole new range of possible applications.

8. ACKNOWLEDGEMENTS

This research has been supported by ERC Starting Grant 240186 "MiGraNT: Mining Graphs and Networks: a Theory-based approach". We thank Elisa Fromont and Laura Antanas for supplying their datasets and for the fruitful discussions.

9. REFERENCES

[1] L. Antanas, M. van Otterlo, J. Oramas, T. Tuytelaars, and L. D. Raedt. Not far away from home: A relational distance-based approach to understanding images of houses. In *Inductive Logic Programming - 20th International Conference*, ILP '10, pages 22–29, 2010.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM*, 36:929–965, October 1989.

[3] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, STOC '71, pages 151–158, 1971.

[4] R. Cramer, D. Patterson, and J. Bunce. Recent advances in molecular field analysis (comfa). *Progress in Clinical and Biological Research*, 291:161–165, January 1989.

[5] G. Damiand, C. de la Higuera, J. Janodet, E. Samuel, and C. Solnon. A polynomial algorithm for submap isomorphism. In *Graph-Based Representations in Pattern Recognition*, volume 5534 of *Lecture Notes in Computer Science*, pages 102–112. Springer Berlin / Heidelberg, 2009.

[6] E. Davies. *Machine Vision : Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers, 2005.

[7] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and applications*. Springer, 2008.

[8] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, SODA '95, pages 632–640, 1995.

[9] P. Finn, S. Muggleton, D. Page, and A. Srinivasan. Pharmacophore discovery using the inductive logic programming system PROGOL. *Machine Learning*, 30:241 – 270, February 1998.

[10] F. Haray. *Graph Theory*. Narosa, 1969.

[11] R. Jenatton, G. O. J. Mairal, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the International Conference on Machine Learning*, ICML '10, 2010.

[12] V. Kaibel and A. Schwartz. On the complexity of polytope isomorphism problems. *Graphs and Combinatorics*, 19:215–230, June 2003.

[13] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1944.

[14] M. Kuramochi and G. Karypis. Discovering frequent geometric subgraphs. *Information Systems*, 32:1101–1120, December 2007.

[15] A. Pinz, H. Bischof, W. Kropatsch, G. Schweighofer, A. Opelt, and A. Ion. Representations for cognitive vision:a review of appearance-based, spatio-temporal, and graph-based approaches. *Electronic Letters on Computer Vision and Image Analysis*, 7:35–61, June 2008.

[16] A. Prado, B. Jeudy, E. Fromont, and F. Diot. Mining spatiotemporal patterns in dynamic plane graphs. In *Proceedings of the Spring workshop on Machine Learning*, SML '11, 2011.

[17] E. Samuel, C. de la Higuera, and J. Janodet. Extracting plane graphs from images. In *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, SSPR 10, pages 233–243, 2010.

[18] F. Santos and R. Seidel. A better upper bound on the number of triangulations of a planar point set. *Journal of Combinatorial Theory, Series A*, 102:186–193, April 2003.

[19] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3:177–280, July 2008.