Active Search in Intensionally Specified Structured Spaces

Dino Oglic [† ‡]
dino.oglic@uni-bonn.de
[†]Institut für Informatik III
Universität Bonn, Germany

Roman Garnett [♯]
garnett@wustl.edu
[♯]Dep. of Computer Science & Eng.
Washington University in St. Louis, USA

Thomas Gärtner [‡]
tg@thomasgaertner.org
[‡]School of Computer Science
The University of Nottingham, UK

Abstract

We consider an active search problem in intensionally specified structured spaces. The ultimate goal in this setting is to discover structures from structurally different partitions of a fixed but unknown target class. An example of such a process is that of computer-aided *de novo* drug design. In the past 20 years several Monte Carlo search heuristics have been developed for this process. Motivated by these hand-crafted search heuristics, we devise a Metropolis–Hastings sampling scheme where the acceptance probability is given by a probabilistic surrogate of the target property, modeled with a max entropy conditional model. The surrogate model is updated in each iteration upon the evaluation of a selected structure. The proposed approach is consistent and the empirical evidence indicates that it achieves a large structural variety of discovered targets.

1 Introduction

We consider an active classification problem in structured spaces, where the goal is not to learn a hypothesis but to discover a diverse set of structures exhibiting a target property. A variant of this problem where the only goal is to discover targets is known as active search (Garnett et al. 2012).

In the applications we consider, the search space is specified only intensionally and its cardinality is at least exponential in the size of its combinatorial objects (e.g., number of edges in a graph). Thus, the extension of the search space can neither be completely stored on a disk nor enumerated in feasible time. The structures we aim to discover are characterized by a target property that is a priori not known for any structure and is expensive to evaluate on each structure. The evaluation process can be noisy and it is simulated with an oracle. The structures exhibiting the target property are typically rare and we can not assume that they are concentrated in a small region of the search space. We are thus interested in finding a diverse set of candidates that spans the whole space and is likely to exhibit the target property.

Taking drug discovery as our main motivating example, several problems have been identified as the cause for the huge cost associated with attrition (Scannell et al. 2012; Schneider and Schneider 2016), i.e., drug candidates failing later stages of the development process, and increased use of algorithmic support has been proposed as a remedy (Woltosz 2012). In particular, (i) the *chemspace*, i.e., the space of potentially synthesizable compounds, is huge—estimates are often larger than 10^{60}; (ii) there are many *activity cliffs*, i.e., small changes in structure can have large effects on pharmaceutical activity, and (iii) existing compound libraries focus on a very restricted area of the chemspace. *De novo design* approaches (Schneider and Fechner 2005) aim to overcome these problems by constructing desired molecular structures from scratch. In the past 20 years, several Monte Carlo search heuristics have been developed for de novo design of drug-like molecules (Schneider and Fechner 2005). A common property of these search heuristics is the generation of molecular structures using Markov chains. Several search heuristics incorporate an additional scoring step in which the generated structures are accepted/rejected with a probability based on a hand-crafted energy-based scoring function. The whole process can be seen as Metropolis sampling from an expert-designed distribution. Throughout the constructive process this designed distribution is either kept static or manually updated as the process evolves.

Motivated by these hand-crafted search heuristics, we propose a data-driven approach that learns the target class of desired structures as it observes the results of new experiments. To deal with the intensionally specified search space, we assume that a *proposal generator* can be constructed which is specific to the application domain and has support on all parts of the space that contain the targets. Similar to the described Monte Carlo search heuristics, we model this proposal generator with a Markov chain given by its transition kernel. The transition kernel can be either conditional or independent and in the latter case the proposal generator is an uninformed sampler. As the target structures are typically rare and expensive to evaluate, the cost per discovered structure would be prohibitively high for plain Monte Carlo search performed by evaluating each proposed structure. To overcome this, our approach relies on a max-entropy conditional model that acts as a probabilistic surrogate for the oracle evaluations. This conditional model is updated in each iteration upon the evaluation of a selected structure. As this changes the distribution of the Metropolis sampler in the following discovery step, we can not assume that the sampled structures are drawn independently from identical distributions.

We analyze the theoretical properties of this process in Section 3 where we show its consistency and bound the mixing

Algorithm 1 DE-NOVO-DESIGN

Input: target property $y^* \in \mathcal{Y}$, conditional exponential family model $p(y \mid x, \theta)$ with a regularization parameter $\lambda > 0$, proposal generator $\mathcal{G}$, evaluation oracle $\mathcal{O}$, and budget $B \in \mathbb{N}$
Output: list of structures $x_1, x_2, \ldots, x_B \in \mathcal{X}^B$

1: $\theta_1 \leftarrow \mathbf{0}$
2: **for** $t = 1, 2, \ldots, B$ **do**
3: $\quad x_t \sim \mathcal{G}$
4: $\quad$ **repeat**
5: $\quad\quad x \sim \mathcal{G}$ and $u \sim \mathcal{U}[0, 1]$
6: $\quad\quad$ **if** $u < p(y^* \mid x, \theta_t)/p(y^* \mid x_t, \theta_t)$ **then** $x_t \leftarrow x$ **end if**
7: $\quad$ **until** CHAIN MIXED
8: $\quad y_t \leftarrow \mathcal{O}(x_t)$ and $w_t \leftarrow {}^1\!/_{p(y^* \mid x_t, \theta_t)}$
9: $\quad \theta_{t+1} \leftarrow \arg\min_\theta -\frac{1}{t}\sum_{i=1}^{t} w_i \ln p(y_i \mid x_i, \theta) + \lambda \|\theta\|^2_{\mathcal{H}}$
10: **end for**

time of the Metropolis–Hastings chain with an independent proposal generator. To study the empirical performance *in silico*, i.e., without conducting lab experiments, we design synthetic testbeds that share many characteristics with drug design (Section 4). In particular, instead of the chemspace, we consider the space of all graphs of a given size and aim at constructing graphs with rare and structurally non-smooth properties such as having a Hamiltonian cycle or being connected and planar. We conclude with a discussion where we contrast our approach to other related approaches (Section 5).

2 Algorithm

Algorithm 1 gives a pseudo-code description of our approach. To model the evaluation of the target property, our algorithm takes as input an oracle which outputs a label for a given structure. To reflect the expensiveness of these evaluations, the oracle can be accessed a number of times that is limited by a budget. Other parameters of the algorithm are the proposal generator, target property, and parameters specifying a set of models from the conditional exponential family. In the next section, we demonstrate that for this choice of a conditional model the probabilistic surrogate for the oracle evaluations is a max-entropy model subject to constraints on the first moments of the sample. Denote the space of candidate structures $\mathcal{X}$, the space of properties $\mathcal{Y}$, and a Hilbert space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$. The parameter set $\Theta \subseteq \mathcal{H}$ is usually a compact subset of the Hilbert space and together with the sufficient statistics $\phi : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ of $y \mid x$ specifies the set of conditional exponential models as

$$p(y \mid x, \theta) = \exp\big(\langle \phi(x, y), \theta \rangle - A(\theta \mid x)\big), \tag{1}$$

where $A(\theta \mid x) = \ln \int_{\mathcal{Y}} \exp(\langle \phi(x, y), \theta \rangle)$ and $\theta \in \Theta$. In practice, we do not directly specify the parameter set Θ but instead simply regularize the importance weighted negative log-likelihood of the sample by adding the term $\|\theta\|^2_{\mathcal{H}}$. To account for this, the algorithm takes as input a hyperparameter which controls the regularization.

The constructive process is initialized by setting the parameter vector of the conditional exponential family to zero (line 1). This implies that the first sample is unbiased and uninformed. Then, the algorithm starts iterating until we deplete the oracle budget B (line 2). In the initial steps of each iteration (lines 3–7), the Metropolis–Hastings algorithm (Metropolis et al. 1953) is used to sample from the posterior $p(x \mid y^*, \theta_t) = \frac{p(y^* \mid x, \theta_t) p_0(x)}{p_0(y^*)}$, where $p_0(y^*)$ is the marginal probability of $y^* \in \mathcal{Y}$ and $p_0(x)$ is the stationary distribution of the proposal generator $\mathcal{G}$ defined with a transition kernel g for which the detailed balance condition holds (Andrieu et al. 2003). Thus, to obtain samples from the posterior $p(x \mid y^*, \theta_t)$, the Metropolis–Hastings acceptance criterion is

$$\frac{p(y^* \mid x', \theta_t)}{p(y^* \mid x_t, \theta_t)} \cdot \frac{p_0(x') \cdot g(x' \to x_t)}{p_0(x_t) \cdot g(x_t \to x')} = \frac{p(y^* \mid x', \theta_t)}{p(y^* \mid x_t, \theta_t)}, \tag{2}$$

where x' is the proposed candidate, x_t is the last accepted state, θ_t is the parameter vector of the conditional exponential family model, and $g(x_t \to x')$ denotes the probability of the transition from state x_t to state x'. After the Metropolis–Hastings chain has mixed (line 7), the algorithm outputs its last accepted state x_t as a candidate structure and presents it to an evaluation oracle (line 8). The oracle evaluates it providing feedback y_t to the algorithm. The labeled pair (x_t, y_t) is then added to the training sample and an importance weight is assigned to it (line 8). The importance weighting is needed for the consistency of the algorithm because the samples are neither independent nor identically distributed. Finally, the conditional exponential family model is updated by optimizing the weighted negative-log likelihood of the sample (line 9). This model is then used by the algorithm to sample a candidate structure in the next iteration. The optimization problem in line 9 is convex in θ and the representer theorem (Wahba 1990) guarantees that it is possible to express the solution θ_{t+1} as a linear combination of sufficient statistics, i.e., $\theta_{t+1} = \sum_{i=1}^{t} \sum_{c \in \mathcal{Y}} \alpha_{ic} \phi(x_i, c)$ for some $\alpha_{ic} \in \mathbb{R}$. Hence, a globally optimal solution can be found and a set of conditional exponential family models can be specified using only a joint input–output kernel and a regularization parameter.

3 Theoretical analysis

In this section, we first show that in Algorithm 1 a max-entropy conditional model is used as a probabilistic surrogate for the oracle. We then prove that Algorithm 1 is consistent and analyze the mixing time of an independent Metropolis–Hastings chain for sampling from the posterior $p(x \mid y^*, \theta)$.

3.1 Max-entropy probabilistic surrogate

In previous work it was shown that exponential family models are max-entropy models subject to constraints on the first moments of the sample (Jaynes 1957). The following proposition is an adaptation of this max-entropy result to *conditional* exponential family models. For the sake of completeness, a proof is provided in Appendix A.

Proposition 1. *Let $\mathcal{P}$ denote the set of all conditional distributions that have square integrable densities with respect to a base measure defined on the domain of a sufficient statistic $\phi(x, y)$ and support on the entire domain of $\phi(x, y)$. A max-entropy conditional distribution from $\mathcal{P}$ that satisfies a set of constraints on the first moments of the sample can be represented as a conditional exponential model. To specify this distribution it is sufficient to find the maximum a posteriori estimator from the conditional exponential family of models.*

This proposition guarantees that conditional exponential family models are objectively encoding the information from

the sample into the model. In fact, any other choice of the conditional model makes additional assumptions about the samples that reduce the entropy and introduces a potentially undesirable bias into the process.

3.2 Consistency

In this section, we show that Algorithm 1 converges in probability to the best model from a parameter set Θ. For this, we assume that Θ is a compact subset of a Euclidean space and that there exist constants $R, r > 0$ such that $\|\theta\| \leq R$ for all $\theta \in \Theta$ and $\|\phi(x,y)\| = \sqrt{k\big((x,y),(x,y)\big)} \leq r$ for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$. In finite dimensional Euclidean spaces closed spheres are compact sets and, in line with our previous assumption, we can take Θ to be the sphere of radius R centered at the origin. In infinite dimensional spaces closed spheres are not compact sets and in this case it is possible to find an approximate finite dimensional basis of the kernel feature space using the Cholesky decomposition of the kernel matrix (Fine and Scheinberg 2002) and define Θ as in the finite dimensional case. We note that this is a standard step for many kernel based approaches in machine learning (Bach 2007).

Given the stationary distribution $p_0(x)$ of the proposal generator and the conditional label distribution of the evaluation oracle $p_0(y \mid x)$, the *latent data-generating distribution* is $p_0(x,y) = p_0(y \mid x)\, p_0(x)$. We measure the difference between this data-generating distribution and our conditional exponential family model, parameterized with a vector θ, using the Kullback–Leibler divergence (Akaike 1973; White 1982). Eliminating the parameter-free terms from this divergence measure, we obtain the loss function of θ,

$$L(\theta) = -\int_{\mathcal{X}\times\mathcal{Y}} p_0(x,y) \ln p(y \mid x, \theta)\,.$$

We assume that there exists a unique minimizer of the loss function $L(\theta)$ in the interior of the parameter set Θ and denote this minimizer with θ^*. If the optimal parameter vector $\theta^* \in \Theta$ satisfies $\mathbb{E}_{p_0(y|x)}[\phi(x,y)] = \mathbb{E}_{p(y|x,\theta^*)}[\phi(x,y)]$ for all $x \in \mathcal{X}$, it is said that the model is *well-specified*.

In our case, sample points are obtained from a query distribution that *depends* on previous samples, i.e., $x_i \sim q(x \mid x_1, \ldots, x_{i-1})$, but labels are still obtained from the conditional label distribution $y_i \sim p_0(y \mid x_i)$ independent of x_j $(j < i)$. The main difficulty in proving the consistency of the approach in the general case where the queried structures are neither independent nor identically distributed comes from the fact that standard concentration bounds do not hold for this setting. A workaround frequently encountered in the literature is to assume that the model is well-specified as in this case the sampling process is consistent irrespective of the query distribution. Before proving convergence in the general case, we first briefly consider the cases of independent samples and well-specified models.

For the common case in which the training sample is drawn *independently* from a distribution $q(x)$, let

$$\hat{\theta}_n = \arg\max_{\theta\in\Theta} \frac{1}{n} \sum_{i=1}^{n} \frac{p_0(x_i)}{q(x_i)} \ln p(y_i \mid x_i, \theta)\,. \tag{3}$$

The sequence of optimizers $\{\hat{\theta}_n\}_{n\in\mathbb{N}}$ converges to the optimal parameter vector θ^* (White 1982; Shimodaira 2000). For $q(x) = p_0(x)$, $\hat{\theta}_n$ is the *maximum likelihood* estimate of θ^* over an i.i.d. sample $\{(x_i,y_i)\}_{i=1}^n$. Moreover, for $\Theta = \{\theta \mid \|\theta\| \leq R\}$ the latter optimization problem is equivalent to finding the *maximum a posteriori* estimator with a Gaussian prior on θ (Altun, Smola, and Hofmann 2004).

In the case of a well-specified model, for all $x \in \mathcal{X}$, it holds $\mathbb{E}_{p_0(y|x)}[\phi(x,y)] = \mathbb{E}_{p(y|x,\theta^*)}[\phi(x,y)]$. Thus, for all marginal distributions $p_0(x)$, the gradient of the loss is zero at θ^*, i.e., $\nabla L(\theta^*) = \int_{\mathcal{X}} p_0(x) \int_{\mathcal{Y}} \phi(x,y)\,(p(y \mid x, \theta^*) - p_0(y \mid x)) = 0$. In other words, if the model is well-specified, the maximum likelihood estimator is consistent for all query distributions.

We now proceed to the general case for which we do not make the assumption that the model is well-specified and again show that the optimizer θ_t converges to the optimal parameter vector θ^*. At iteration t of Algorithm 1 an instance is selected by sampling from the query distribution $q(x \mid \mathcal{D}_{t-1}) = p(x \mid y^*, \theta_t)$, where θ_t denotes a parameter vector from Θ which is completely determined by the previously seen data $\mathcal{D}_{t-1}$. Thus, a candidate sampled at iteration t depends on previous samples through the parameter vector and the independence between input–output pairs within the sample is lost. As a result of this, the convergence of the sequence $\{\theta_t\}_{t\in\mathbb{N}}$ to θ^* for the general case of misspecified model cannot be guaranteed by the previous results relying on the independence assumption (Shimodaira 2000).

To show the consistency in this general case, we first rewrite the objective which is optimized at iteration t of Algorithm 1. For a fixed target property y^*, the parameter vector θ_{t+1} is obtained by solving the following problem:

$$\min_{\theta} \frac{1}{t} \sum_{i=1}^{t} \frac{A(\theta \mid x_i) - \langle \phi(x_i, y_i), \theta \rangle}{p(y^* \mid x_i, \theta_i)} + \lambda \|\theta\|^2\,. \tag{4}$$

Assuming the parameter set is well behaved (Theorem 2), the objective in Eq. (4) is convex and can be optimized using standard optimization techniques. Before we show that the sequence of optimizers θ_t converges to the optimal parameter vector θ^*, let us formally define the empirical loss of a parameter vector θ given the data $\mathcal{D}_t$ available at iteration t,

$$L(\theta \mid \mathcal{D}_t) = \frac{1}{t} \sum_{i=1}^{t} \frac{p_0(y^*)\,\big(A(\theta \mid x_i) - \langle \phi(x_i, y_i), \theta \rangle\big)}{p(y^* \mid x_i, \theta_i)}.$$

The following theorem and corollary show that Algorithm 1 is consistent in the general case for misspecified models and a sample of structures which are neither independent nor identically distributed. The proofs are provided in Appendix A.

Theorem 2. *Let $p(y \mid x, \theta)$ denote the conditional exponential family distribution parameterized with a vector $\theta \in \Theta$, where Θ is a compact subset of a d dimensional Euclidean space $\mathbb{R}^d$. Let $p_0(x,y)$ denote a latent data generating distribution such that, for all $x \in \mathcal{X}$, the support of the likelihood function $p_0(y \mid x)$ is contained in the support of $p(y \mid x, \theta)$ for all $\theta \in \Theta$. Let $\big|\ln p(y \mid x, \theta)\big| \leq h(x,y)$ for all $\theta \in \Theta$ and some function $h(x,y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which is Lebesque integrable in the measure $p_0(x,y)$. Then for all $0 < \varepsilon, \delta < 1$ there exists $N(\varepsilon,\delta) \in \Omega\big(\frac{1}{\varepsilon^2}\big(d \ln \frac{1}{\varepsilon} + \ln \frac{1}{\delta}\big)\big)$ such that for all $t \geq N(\varepsilon,\delta)$ we have $P\big(\sup_{\theta\in\Theta} |L(\theta) - L(\theta \mid \mathcal{D}_t)| \leq \varepsilon\big) \geq 1 - \delta$.*

Corollary 3. *The sequence of estimators $\{\theta_t\}_{t\geq 1}$ converges in probability to $\theta^* \in \Theta$.*

3.3 Mixing time analysis

Having shown the consistency of Algorithm 1, we proceed to bound the mixing time of the Metropolis–Hastings chain. For that, we consider an independent proposal generator $\mathcal{G}$ and provide a simple *coupling* analysis to bound the worst case mixing time of an independent Metropolis–Hastings chain for sampling from the posterior $p(x \mid y^*, \theta_t)$ (Vembu, Gärtner, and Boley 2009). This allows us to utilize perfect sampling algorithms such as coupling from the past (Propp and Wilson 1996) to draw samples from the posterior. We assume $|\mathcal{X}|$ parallel and identical chains are started from all possible states $x \in \mathcal{X}$ and an identical random bit sequence is used to simulate all the chains. Thus, whenever two chains move to a common state, all the future transitions of the two chains are the same. From that point on it is sufficient to track only one of the chains. This is called a *coalescence* (Huber 1998). Propp and Wilson (1996) have shown that if all the chains were started at time $-\mathcal{T}$ and have coalesced to a single chain at step $-T$ with $\mathcal{T} > T > 0$, then samples drawn at time 0 are exact samples from the stationary distribution.

For conditional exponential family models $p\,(y \mid x, \theta) > 0$, the lower bound can be controlled with the regularization parameter. Thus, there will always be a path with non-zero probability between any two target structures. As it is the case with other Metropolis algorithms, for difficult problems where clusters of targets are far apart in the search space, the mixing will be slower as the model becomes more confident. The following proposition (a proof is provided in Appendix A) gives a worst case bound on the mixing time of an independent Metropolis–Hastings chain for sampling from the posterior distribution $p(x \mid y^*, \theta_t)$.

Proposition 4. *For all $0 < \varepsilon < 1$, with probability $1 - \varepsilon$, the mixing time $\tau(\varepsilon)$ of an independent Metropolis–Hastings chain for sampling from the posterior distribution $p(x \mid y^*, \theta_t)$ is bounded from above by $\lceil \ln \varepsilon / \ln\big(1 - \exp(-4r\,\|\theta_t\|)\big) \rceil$.*

4 Experiments

Having provided theoretical justification for our approach in the previous section, here we evaluate its effectiveness with a series of synthetic experiments that are designed to mimic the construction of cocktail recipes and graphs with desired properties. The main reason for not evaluating the approach on a real-world problem is not the lack of a proposal generator for that domain but the lack of a suitable experimental set up (the usual retrospective analysis on labeled data is not suitable for active search in intensionally specified structured spaces). For instance, to apply the approach to the design of molecules – our main motivating example – an independent proposal generator can be used (Goldberg and Jerrum 1997), as well as numerous samplers outlined in Schneider & Fechner (2005).

In the first set of experiments, we design cocktails of different flavors – dry, creamy, and juicy. The recipes are represented as sparse real-valued vectors such that the non-zero values in these vectors indicate the proportions of the respective ingredients (i.e., the vectors are normalized). In the second set of experiments, the goal is to design Hamiltonian and connected planar graphs, as well as the respective complements of these classes. As we can not expect to be able to perfectly distinguish each of these classes from its complement due to the hardness of complete graph kernels (Gärtner, Flach, and Wrobel 2003), we can not expect to learn to perfectly generate these concepts. The main objective of these experiments is to demonstrate that our approach can discover a diverse set of target-structures in non-smooth problems which act as *in silico proxies* for the drug design task. In particular, in the construction of Hamiltonian graphs and complements of these, there are numerous Hamiltonian graphs which become non-Hamiltonian with a removal of a single edge. Such graphs are structurally very similar and close in the design space. Thus, these testbeds can mimic well the activity cliffs specific to drug design where very similar structures have different protein binding affinities.

In our empirical evaluation, we compare Algorithm 1 to k-NN active search with 1- and 2-step look-ahead (Garnett et al. 2012) and a greedy method which discovers structures by repeatedly performing *argmax* search over samples from a proposal generator using the learned conditional label distribution (selected structures are labeled by an oracle and the model is updated in each iteration). In the first step of this evaluation, we measure the improvement of each of the considered approaches over plain Monte Carlo search performed with a proposal generator. We assess the performance of the approaches with correct-construction curves which show the cumulative number of distinct target structures discovered as a function of the budget expended. To quantify the improvement of the approaches over plain Monte Carlo search, we measure the lift of the correct-construction curves. In particular, for sampling from the minority class of a proposal generator the lift is computed as the ratio between the number of distinct structures from this class generated by an algorithm and the number of such structures observed in a sample (of the same size) from the distribution of the proposal generator. In the second step of our empirical evaluation, we assess the structural diversity between the targets discovered by an algorithm. We do this by incorporating diversity into the correct-construction curves. Namely, we take a sample of $50\,000$ structures from the proposal generator and filter out targets. We consider these as undiscovered targets and compute the average distance between an undiscovered structure and a subsample of budget size from this set of structures. With this average distance as radius we circumscribe a sphere around each of the undiscovered targets. Then, instead of construction-curves defined with the number of discovered targets, we use the construction-curves defined with the number of the spheres having a target structure within them. To quantify the effectiveness of the considered algorithms in discovering structurally diverse targets, we normalize these sphere based construction-curves with one such curve corresponding to an ideal algorithm that only generates targets – the output of this algorithm can be represented with a subsample of budget size from the undiscovered target structures.

Implementation details for all algorithms are provided in Appendix C. We have simulated Algorithm 1 with the uniform proposal generator over the space of graphs with 7 and 10 nodes (Wormald 1987). For the space of cocktails, we have developed a frequency based sampler from a small set of cocktails collected from `www.webtender.com`. This

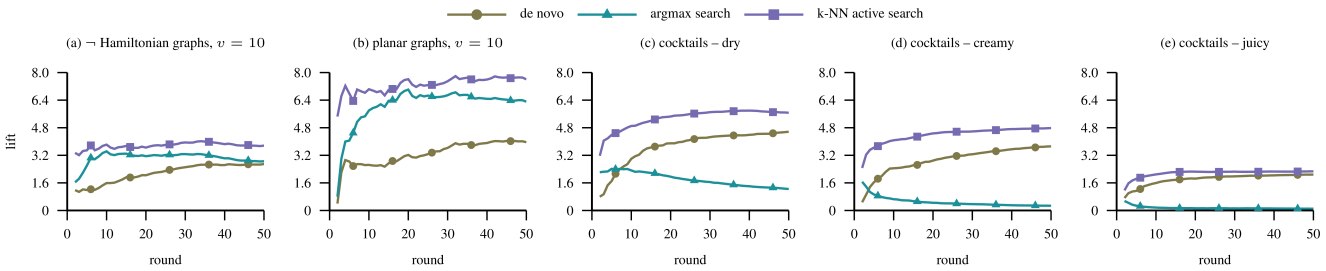

Figure 1: The figure shows the lift of correct-construction curves for considered graph and cocktail concepts. The lift indicates how much more likely it is to see a target compared to the Monte Carlo search with a proposal generator.

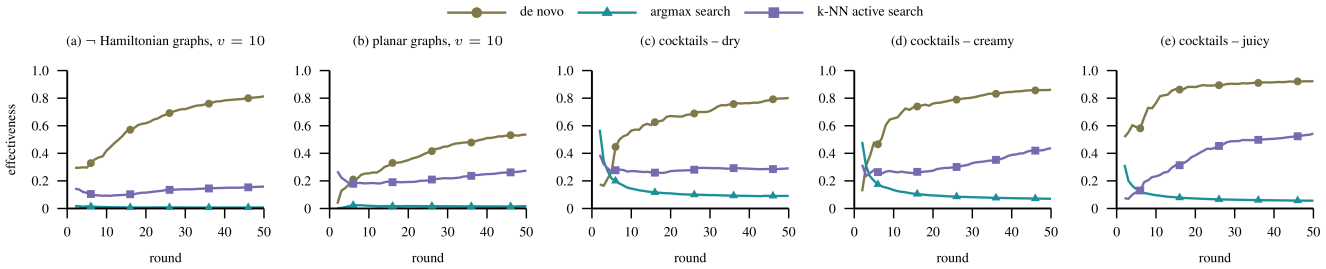

Figure 2: The figure shows the dispersion of discovered targets relative to an algorithm with the identical proposal generator that outputs only targets. The reported curves can be seen as the percentage of discovered target class partitions given a budget.

sampler generates cocktail recipes by first sampling the number of ingredients from the Poisson distribution and then it selects the recipe ingredients based on their co-occurrence frequency in the collected data set. The parameters of this proposal generator are moment-matched with respect to the collected cocktail data set. As this proposal generator almost always samples recipes with 2-10 ingredients, for n possible ingredients the number of different ingredient combinations is $\sum_{k=2}^{10} \binom{n}{k}$ (approximately n^{10}). As the sampler is developed based on a set of cocktails with 335 ingredients there are approximately 10^{24} different combinations of ingredients in this search space. Thus, this is a huge search space that can provide an insight into the properties of the discovery process on large scale problems. To label the cocktails generated by this proposal generator we have trained decision trees for each of the considered flavor profiles using a labeling of these cocktails according to flavor. All the reported results were obtained by averaging over 5 runs of the algorithm. The Metropolis–Hastings sampling was performed with a burn-in sample of 50 000 proposals and sampling was done for 50 rounds/batches. In each round we take 10 i.i.d. samples by running 10 Metropolis–Hastings chains in parallel (note that samples from different rounds are dependent). To allow for models of varying complexity, we have estimated the conditional exponential family regularization parameter in each round using 5-fold stratified cross-validation. As the competing approaches – argmax and k-NN active search (Garnett et al. 2012) – are not designed to search for targets without an a priori provided labeled structures, we have made a minor modification to our problem setting and warm-started each method with a random sample of 5 target and the same number of non-target structures. For graphs these were chosen uniformly from the search space and for cocktails uniformly from the available sample of cocktails. Note that without this warm-start the argmax search estimates the distribution of target structures with a single peak around the first discovered target. Moreover, k-NN probabilistic model cannot learn a property until it sees more than k labeled structures and it is unlikely to observe a target in k successive samples from a proposal generator.

In Figure 3.3, we show the lift of the correct-construction curves for all the considered approaches. We have defined these correct-construction curves by considering isomorphic graphs and cocktails with equal sets of ingredients (ignoring portions of each ingredient) as identical structures. The plots indicate that our approach and k-NN active search are able to emphasize the target class in all the domains for all the considered properties. Moreover, for our approach the magnitude of this emphasis is increasing over time and it is more likely to generate a target as the process evolves. In all domains and for all properties, k-NN active search discovers more target structures than our approach. For graph properties, we see that argmax search also discovers more targets than our approach. For cocktails, argmax search discovers many cocktails with identical sets of ingredients and different portions of these (such cocktails are considered identical in the correct-construction curves). Thus, if we are only interested in discovering target structures without considering structural diversity between them, our empirical evaluation indicates that it is better to use k-NN active search than Algorithm 1.

In Figure 3.3, we show the dispersion of target structures discovered by each of the considered approaches. The plots

indicate that our approach achieves a large structural variety of discovered targets. In all domains and for all properties, our approach outperforms both k-NN active and greedy argmax search. These experiments also indicate that k-NN active search explores more than argmax search. In some of the plots, a dip can be observed in the curves for k-NN active and argmax search. This can be explained by the exploitative nature of these algorithms and the fact that the search is focused to a small region of the space until all the targets from it are discovered. In contrast to this, our approach discovers targets from the whole space and can cover a large number of spheres centered at undiscovered samples with a relatively small number of targets. Thus, if we are interested in discovering diverse target structures, our results indicate that it is better to use Algorithm 1 than k-NN active or argmax search.

5 Discussion

Active search with k-NN probabilistic model (Garnett et al. 2012) is a related approach with the problem setting similar to that of de novo design. The key distinction between the investigated problem setting and k-NN active search is in the requirement to discover structures from the whole domain. Garnett et al. (2012) assume that an extensional description in the form of a finite subset of the domain is explicitly given as input to the algorithm. In this work we require only an intensional description of the domain. For instance, for the domain of graphs on $n \in \mathbb{N}$ vertices, the intensional description is just that of the number of vertices, while the extensional one consists of a list of all graphs on n vertices. In many cases, considering intensional descriptions is much more promising because an algorithm with an extensional description of an exponentially large or uncountable search space can only consider small and often arbitrary subsets of this space. The second key distinction between k-NN active search and de novo design is in the assessment of their outcomes. In particular, both approaches try to find, as soon as possible, as many as possible target structures. However, k-NN active search is designed to only discover members of a target class and Algorithm 1 is designed to find members of distinct structural partitions of a target class. This is very useful in domains where there are numerous isofunctional structures and in which k-NN active search outputs structures from small number of structural partitions of a target class.

Recently, active search has been applied to a problem related to our cocktail construction task – interactive exploration of patterns in a cocktail dataset (Paurat, Garnett, and Gärtner 2014). The difference between our setting and that of Paurat et al. (2014) is in the requirement to generate novel and previously unseen cocktails exhibiting a target property rather than searching for patterns in an existing cocktail dataset. In addition to this, active search has been applied to real-world problems where the search space is given by a single combinatorial graph, and some subset of its nodes is interesting (Wang, Garnett, and Schneider 2013). This is different from applications we consider here and for which the search space is the space of all graphs of a given size.

As the investigated problem setting can be seen as a search in structured spaces, our approach is, with certain distinctions, closely related to structured output prediction (Tsochantaridis et al. 2004; Daumé III, Langford, and Marcu 2009). In structured output prediction the goal is to find a mapping from an instance space to a 'structured' output space. A common approach is to find a joint scoring function, from the space of input–output pairs to the set of reals, and to predict the output structure which maximizes the scoring function for each test input. Finding a good scoring function can often be cast as a convex optimization problem with exponentially many constraints. It can be solved efficiently if the so-called *separation* and/or *decoding* sub-problems can be solved efficiently. One difference between the investigated setting and structured output prediction is in the assumption how input–output pairs are created. In particular, structured output prediction assumes that the provided outputs are optimal for the given inputs. In many de novo design problems, it is infeasible to find the best possible output for a given input. For de novo drug design this assumption implies that we would need to know the best molecule—from the space of all synthesizable molecules—with respect to different properties, such as binding affinity to specific protein sites. Moreover, as the decoding problem is designed assuming that the input–output pairs are optimal the greedy *argmax* approach to solving this problem does not incorporate exploration. As a result of this, similar to argmax search these methods generate structures from a very small number of structural partitions of the target class. Other differences are in the iterative nature of de novo design and in the hardness of the separation or decoding sub-problems that most structured output prediction approaches need to solve. Another related sub-problem is that of finding *preimages* (Weston, Schölkopf, and Bakir 2004) which is typically also hard in the context of structured domains except for some special cases such as strings (Giguère et al. 2015).

Related to the proposed approach are also methods for *interactive learning and optimization* as well as *Bayesian optimization*. Interactive learning and optimization methods implement a two-step iterative process in which an agent interacts with a user until a satisfactory solution is obtained. Some well-known interactive learning and optimization methods tackle problems in information retrieval (Yue and Joachims 2009; Shivaswamy and Joachims 2012) and reinforcement learning (Wilson, Fern, and Tadepalli 2012; Jain et al. 2013). However, these methods are only designed to construct a single output from the domain of real-valued vectors and can not be directly applied to structured domains. Bayesian optimization (Brochu, Cora, and de Freitas 2010; Shahriari et al. 2015), on the other hand, is an approach to sequential optimization of an expensive, black-box, real-valued objective. Rather than seeking a set of high-quality items, Bayesian optimization focuses on finding the single highest-scoring point in the domain. We, in contrast, consider discrete labels and wish to maximize the number of diverse targets found in an intensionally specified structured space. In drug design, this emphasis on exploring all parts of the search space is known as *scaffold-hopping* (Schneider and Fechner 2005) and it is related to the problem of attrition (Schneider and Schneider 2016). Namely, in order to address this problem it is not sufficient to search for a molecule with the highest activity level as it can be toxic or bind to an undesired protein in addition to the target protein. If attrition is

to be reduced an algorithm needs to find a number of structurally different molecules binding to a target protein. As our approach achieves a large structural variety of discovered targets, it has a potential to tackle this difficult problem.

Acknowledgments

We are grateful for access to the University of Nottingham High Performance Computing Facility. Part of this work was also supported by the German Science Foundation (grant GA 1615/1-1).

References

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*. Akadémiai Kiado.

Aldous, D. 1983. Random walks on finite groups and rapidly mixing Markov chains. *Séminaire de Probabilités XVII*.

Altun, Y.; Smola, A. J.; and Hofmann, T. 2004. Exponential families for conditional random fields. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*.

Andrieu, C.; de Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to MCMC for machine learning. *Machine Learning*.

Azuma, K. 1967. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*.

Bach, F. 2007. Active learning for misspecified generalized linear models. In *Advances in Neural Information Processing Systems 19*.

Beygelzimer, A.; Dasgupta, S.; and Langford, J. 2009. Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning*.

Borgwardt, K. M. 2007. *Graph kernels*. Ph.D. Dissertation, Ludwig Maximilians University Munich.

Brochu, E.; Cora, V. M.; and de Freitas, N. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Cameron, P. J. 1998. *Introduction to Algebra*. Oxford Univ. Press.

Carl, B., and Stephani, I. 1990. *Entropy, Compactness, and the Approximation of Operators*. Cambridge University Press.

Daumé III, H.; Langford, J.; and Marcu, D. 2009. Search-based structured prediction. *Machine Learning*.

Dixon, J. D., and Wilf, H. S. 1983. The random selection of unlabeled graphs. *Journal of Algorithms*.

Fine, S., and Scheinberg, K. 2002. Efficient SVM training using low-rank kernel representations. *Journ. of Machine Learning Research*.

Garnett, R.; Krishnamurthy, Y.; Xiong, X.; Schneider, J.; and Mann, R. P. 2012. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Conference on Machine Learning*.

Gärtner, T.; Flach, P. A.; and Wrobel, S. 2003. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory*.

Gelfand, I. M., and Fomin, S. V. 1963. *Calculus of variations*. Prentice-Hall Inc.

Giguère, S.; Rolland, A.; Laviolette, F.; and Marchand, M. 2015. Algorithms for the hard pre-image problem of string kernels and the general problem of string prediction. In *Proceedings of the 32nd International Conference on Machine Learning*.

Goldberg, L. A., and Jerrum, M. 1997. Randomly sampling molecules. In *Proceedings of the 8th ACM SIAM Symposium on Discrete Algorithms*.

Guruswami, V. 2000. Rapidly mixing Markov chains: A comparison of techniques (survey). Technical report, MIT.

Huber, M. 1998. Exact sampling and approximate counting techniques. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*.

Jain, A.; Wojcik, B.; Joachims, T.; and Saxena, A. 2013. Learning trajectory preferences for manipulators via iterative improvement. In *Advances in Neural Information Processing Systems 26*.

Jaynes, E. T. 1957. Information theory and statistical mechanics. *Physical Review*.

Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; and Teller, E. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*.

Paurat, D.; Garnett, R.; and Gärtner, T. 2014. Interactive exploration of larger pattern collections: A case study on a cocktail dataset. In *Proceedings of KDD IDEA*.

Propp, J. G., and Wilson, D. B. 1996. Exact sampling with coupled Markov chains and applications to statistical mechanics. In *Proc. of the 7th International Conference on Random Struct. and Algorithms*.

Scannell, J. W.; Blanckley, A.; Boldon, H.; and Warrington, B. 2012. Diagnosing the decline in pharmaceutical R&D efficiency. *Nature Reviews Drug Discovery*.

Schneider, G., and Fechner, U. 2005. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*.

Schneider, P., and Schneider, G. 2016. De novo design at the edge of chaos. *Journal of Medicinal Chemistry*.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and de Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. Technical report, Universities of Harvard, Oxford, Toronto, and Google DeepMind.

Shimodaira, H. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*.

Shivaswamy, P., and Joachims, T. 2012. Online structured prediction via coactive learning. In *Proceedings of the 29th International Conference on Machine Learning*.

Tsochantaridis, I.; Hofmann, T.; Joachims, T.; and Altun, Y. 2004. SVM learning for interdependent and structured output spaces. In *Proc. of the 21st International Conference on Machine Learning*.

Vembu, S.; Gärtner, T.; and Boley, M. 2009. Probabilistic structured predictors. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*.

Wahba, G. 1990. *Spline models for observational data*. SIAM.

Wang, X.; Garnett, R.; and Schneider, J. 2013. Active search on graphs. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Weston, J.; Schölkopf, B.; and Bakir, G. H. 2004. Learning to find pre-images. In *Adv. in Neural Information Processing Systems 16*.

White, H. 1982. Maximum likelihood estimation of misspecified models. *Econometrica*.

Wilson, A.; Fern, A.; and Tadepalli, P. 2012. A Bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems 25*.

Woltosz, W. S. 2012. If we designed airplanes like we design drugs... *Journal of Computer-Aided Molecular Design*.

Wormald, N. C. 1987. Generating random unlabelled graphs. *SIAM Journal on Computing*.

Yue, Y., and Joachims, T. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th International Conference on Machine Learning*.

A Proofs

Proposition 1. *Let $\mathcal{P}$ denote the set of all conditional distributions that have square integrable densities with respect to a base measure defined on the domain of a sufficient statistic $\phi(x,y)$ and support on the entire domain of $\phi(x,y)$. A max-entropy conditional distribution from $\mathcal{P}$ that satisfies a set of constraints on the first moments of the sample can be represented as a conditional exponential model. To specify this distribution it is sufficient to find the maximum a posteriori estimator from the conditional exponential family of models.*

Let $f, g \in \mathcal{P}$ and $\langle f, g\rangle = \int_{\mathcal{X}\times\mathcal{Y}} f(y \mid x)\, g(y \mid x)$ be the dot product defined on this space. For a marginal distribution of structures $\tilde{p}(x)$, the conditional entropy of a distribution $p \in \mathcal{P}$ is defined as

$$H(p \mid \tilde{p}) = -\int_{\mathcal{X}} \tilde{p}(x) \int_{\mathcal{Y}} p(y \mid x) \ln p(y \mid x).$$

Now, let ϕ_i denote the ith component of the feature map ϕ. From the available sample it is possible to estimate the empirical value of these component-statistics. In particular, we can denote with $\alpha_i = \frac{1}{n}\sum_{j=1}^{n} \phi_i(x_j, y_j)$, where $(x_j, y_j) \sim p_0(x,y)$. Then, a max-entropy distribution from $\mathcal{P}$ satisfying a set of constraints on the first moments of the sample would be a solution of the following optimization problem

$$\begin{aligned} &\arg\max_{p\in\mathcal{P}} H(p \mid p_0) \\ &\text{s.t.} \quad \int_{\mathcal{X}\times\mathcal{Y}} \phi_i(x,y)\, p(y \mid x)\, p_0(x) \le \alpha_i, i \in \mathcal{I}. \end{aligned}$$

Proof. Gathering all the constraints and forming the Lagrangian we get

$$\begin{aligned} \mathcal{L}(p,\lambda) = \quad & -H(p \mid p_0) + \int_{\mathcal{X}} \lambda(x) \left(\int_{\mathcal{Y}} p(y \mid x) - 1 \right) + \\ & \int_{\mathcal{X}\times\mathcal{Y}} \lambda(x,y)\, p(y \mid x) + \\ & \sum_{i\in\mathcal{I}} \lambda_i \left(\int_{\mathcal{X}\times\mathcal{Y}} \phi_i(x,y)\, p_0(x)\, p(y \mid x) - \alpha_i \right), \end{aligned}$$

where $\lambda_i \ge 0$ for all $i \in \mathcal{I}$, $\lambda(x) \ge 0$ for all $x \in \mathcal{X}$, and $\lambda(x,y) \le 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

For a functional $F(p)$, the functional gradient ∇F at p is the principal linear term in the change of F after it is perturbed by ε in the direction of q (Gelfand and Fomin 1963, Section 3)

$$F(p + \varepsilon q) = F(p) + \varepsilon \langle \nabla F, q\rangle + \mathcal{O}(\varepsilon^2).$$

Applying this derivation rule to the entropy term, we deduce $\nabla_p H = -p_0(x)(\ln p(y \mid x) + 1)$. The functional gradient of the Lagrangian is then given by

$$\begin{aligned} \nabla_p \mathcal{L} = \quad & p_0(x) \left(1 + \ln p(y \mid x) + \sum_{i\in\mathcal{I}} \lambda_i \phi_i(x,y) \right) + \\ & \lambda(x) + \lambda(x,y). \end{aligned}$$

Setting the functional gradient to zero, we obtain a max-entropy distribution satisfying the constraints,

$$p^*(y \mid x) = \frac{\exp(\langle \theta^*, \phi(x,y)\rangle)}{\exp\left(\frac{\lambda^*(x) + \lambda^*(x,y)}{p_0(x)}\right)},$$

where $\phi = \operatorname{vec}\left(1, \{\phi_i\}_{i\in\mathcal{I}}\right)$ and $\theta^* = -\operatorname{vec}\left(1, \{\lambda_i^*\}_{i\in\mathcal{I}}\right)$.

As $p(y \mid x) > 0$ for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$, it follows from the complementary slackness that $\lambda^*(x,y) = 0$. Now, taking $\lambda^*(x) = p_0(x) \ln \int_{\mathcal{Y}} \exp(\langle \theta^*, \phi(x,y)\rangle)$, we see that the max-entropy conditional distribution is defined as

$$p^*(y \mid x) = \frac{\exp(\langle \theta^*, \phi(x,y)\rangle)}{\int_{\mathcal{Y}} \exp(\langle \theta^*, \phi(x,y)\rangle)}.$$

Let us verify that this is indeed the desired distribution. For any other feasible conditional distribution $r \in \mathcal{P}$ we have

$$\begin{aligned} & H(r \mid p_0) = -\int_{\mathcal{X}\times\mathcal{Y}} p_0(x)\, r(y \mid x) \ln r(y \mid x) \frac{p^*(y \mid x)}{p^*(y \mid x)} \le \\ & -\operatorname{KL}(p^*, r) - \int_{\mathcal{X}\times\mathcal{Y}} p_0(x)\, r(y \mid x) \ln p^*(y \mid x) \le \\ & -\int_{\mathcal{X}\times\mathcal{Y}} p_0(x)\, r(y \mid x) \left(-1 - \frac{\lambda^*(x)}{p_0(x)} - \sum_{i\in\mathcal{I}} \lambda_i^* \phi_i(x,y) \right) \le \\ & 1 + \int_{\mathcal{X}} \lambda^*(x) + \sum_{i\in\mathcal{I}} \lambda_i^* \alpha_i = \\ & -\int_{\mathcal{X}\times\mathcal{Y}} p_0(x)\, p^*(y \mid x) \left(-1 - \frac{\lambda^*(x)}{p_0(x)} - \sum_{i\in\mathcal{I}} \lambda_i^* \phi_i(x,y) \right) = \\ & -\int_{\mathcal{X}\times\mathcal{Y}} p_0(x)\, p^*(y \mid x) \ln p^*(y \mid x) = H(p^* \mid p_0), \end{aligned}$$

where we have used the complementary slackness to transform the equation from line 4 to the one in line 5. In the second line, KL denotes the Kullback–Leibler divergence.

As already pointed out, the coefficients $\{\theta_i^*\}_{i\in\mathcal{I}}$ are chosen to satisfy the moment constraints. Assuming these coefficients have been computed for the moment constraints defined with equalities, the negative entropy of $p^*(y \mid x)$ with respect to the empirical version of $p_0(x)$, $p_0^{emp}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{x=x_i}$, is given by

$$\begin{aligned} & -H(p^* \mid p_0^{emp}) = \frac{1}{n}\sum_{i=1}^{n} \int_{\mathcal{Y}} p^*(y \mid x_i) \ln p^*(y \mid x_i) = \\ & \frac{1}{n}\sum_{i=1}^{n} \int_{\mathcal{Y}} p^*(y \mid x_i) \left(\langle \phi(x_i, y), \theta^*\rangle - A(\theta^* \mid x_i) \right) = \\ & \langle \alpha, \theta^*\rangle - \frac{1}{n}\sum_{i=1}^{n} A(\theta^* \mid x_i) = \\ & \frac{1}{n}\sum_{i=1}^{n} \langle \phi(x_i, y_i), \theta^*\rangle - A(\theta^* \mid x_i), \end{aligned}$$

where $A(\theta^* \mid x) = \ln \int_{\mathcal{Y}} \exp\left(\langle \phi(x,y), \theta^*\rangle\right)$. Thus, to find a max entropy distribution it is sufficient to solve the following non-constrained convex optimization problem

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{n} \langle \phi(x_i, y_i), \theta\rangle - A(\theta \mid x_i).$$

Setting the gradient of this objective function to zero, we verify that θ^* satisfies the set of first moment constraints on the sample. Hence, the conditional exponential family model specified with the maximum a posteriori estimate of the parameter vector is a max entropy conditional model subject to constraints on the first moments of the sample. □

Remark 1. *We note here that Proposition 1 is just a minor adjustment of the classical result for exponential family models (Jaynes 1957) and as such it should not be judged as a contribution of this paper.*

Lemma A.1. *For all* $0 < \varepsilon < 1$ *and* $\theta_1, \theta_2 \in \Theta$ *such that* $\|\theta_1 - \theta_2\| < \frac{2p_{\min}\varepsilon}{r+\sqrt{r^2+2\Lambda p_{\min}\varepsilon}}$, *it holds* $|L(\theta_1) - L(\theta_2)| < \varepsilon$ *and* $|L(\theta_1 \mid \mathcal{D}_t) - L(\theta_2 \mid \mathcal{D}_t)| < \varepsilon$.

Proof. Performing the Taylor expansion of the log-likelihood around θ_1 we get

$$\ln p(y \mid x, \theta_2) \leq \ln p(y \mid x, \theta_1) + \mathbb{E}_{y \sim p(y|x,\theta_1)} \left[\phi(x,y)^\top (\theta_2 - \theta_1)\right] + \frac{\Lambda}{2} \|\theta_2 - \theta_1\|^2.$$

Now, applying the Cauchy-Schwartz inequality to the right hand-side and using the condition $\|\theta_1 - \theta_2\| < {2p_{\min}\varepsilon\Lambda}/{r+\sqrt{r^2+2\Lambda p_{\min}\varepsilon}}$ the claim follows, i.e.,

$$|L(\theta_1) - L(\theta_2)| \leq \|\theta_1 - \theta_2\| \left(r + \frac{\Lambda}{2}\|\theta_1 - \theta_2\|\right) < \varepsilon,$$

$$|L(\theta_1 \mid \mathcal{D}_t) - L(\theta_2 \mid \mathcal{D}_t)| \leq \frac{\|\theta_1 - \theta_2\|\left(r + \frac{\Lambda}{2}\|\theta_1 - \theta_2\|\right)}{p_{\min}} < \varepsilon.$$

□

Lemma A.2. *Let* $\nu = {p_{\min}\varepsilon}/{\left(2r+\sqrt{4r^2+2\Lambda p_{\min}\varepsilon}\right)}$ *and let* $B_1, \ldots, B_{\mathcal{N}(\Theta,\nu)}$ *be an* ν*-cover of the set* Θ. *Then*

$$P\left(\sup_{\theta\in\Theta} |L(\theta) - L(\theta \mid \mathcal{D}_t)| \leq \varepsilon\right) > 1 - \mathcal{N}(\Theta,\nu) \sup_{s=1,\ldots,\mathcal{N}(\Theta,\nu)} P\left(|L(\theta_s) - L(\theta_s \mid \mathcal{D}_t)| > \frac{\varepsilon}{2}\right),$$

where θ_s *denotes the center of the ball* B_s.

Proof. From the assumptions of the lemma it follows that $\sup_{\theta\in\Theta} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon$ if and only if there exists $1 \leq s \leq \mathcal{N}(\Theta,\nu)$ such that $\sup_{\theta\in B_s} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon$. Applying the union bound we get

$$P\left(\sup_{\theta\in\Theta} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon\right) \leq \sum_{s=1}^{\mathcal{N}(\Theta,\nu)} P\left(\sup_{\theta\in B_s} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon\right). \tag{5}$$

On the other hand, we have

$$|L(\theta_i) - L(\theta_i \mid \mathcal{D}_t) - L(\theta) + L(\theta \mid \mathcal{D}_t)| < |L(\theta_i) - L(\theta)| + |L(\theta_i \mid \mathcal{D}_t) - L(\theta \mid \mathcal{D}_t)|.$$

From the last equation and Lemma A.1 for θ_i center of B_i and all $\theta \in B_i$ we get

$$|L(\theta) - L(\theta \mid \mathcal{D}_t)| - |L(\theta_i) - L(\theta_i \mid \mathcal{D}_t)| < \frac{\varepsilon}{2}.$$

As this holds for all $0 < \varepsilon < 1$ and $\theta \in B_i$ we get that $\sup_{\theta\in B_i} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon$ implies $|L(\theta_i) - L(\theta_i \mid \mathcal{D}_t)| > \frac{\varepsilon}{2}$. From here it follows that

$$P\left(\sup_{\theta\in B_s} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon\right) < P\left(|L(\theta_s) - L(\theta_s \mid \mathcal{D}_t)| > \frac{\varepsilon}{2}\right). \tag{6}$$

Combining the results from Eq. (5) and (6) the claim follows. □

Proposition A.3. *(Carl and Stephani 1990) Let* $\mathcal{B}$ *be a finite dimensional Banach space and let* B_R *be the ball of radius* R *centered at the origin. Then, for* $d = \dim(\mathcal{B})$, *it holds*

$$\mathcal{N}(B_R, \varepsilon, \|\cdot\|) \leq \left(\frac{4R}{\varepsilon}\right)^d.$$

Theorem 2. *Let* $p(y \mid x, \theta)$ *denote the conditional exponential family distribution parameterized with a vector* $\theta \in \Theta$, *where* Θ *is a compact subset of a* d *dimensional Euclidean space* $\mathbb{R}^d$. *Let* $p_0(x, y)$ *denote a latent data generating distribution such that, for all* $x \in \mathcal{X}$, *the support of the likelihood function* $p_0(y \mid x)$ *is contained in the support of* $p(y \mid x, \theta)$ *for all* $\theta \in \Theta$. *Let* $\left|\ln p(y \mid x, \theta)\right| \leq h(x, y)$ *for all* $\theta \in \Theta$ *and some function* $h(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ *which is Lebesque integrable in the measure* $p_0(x, y)$. *Then for all* $0 < \varepsilon, \delta < 1$ *there exists* $N(\varepsilon, \delta) \in \Omega\left(\frac{1}{\varepsilon^2}\left(d \ln \frac{1}{\varepsilon} + \ln \frac{1}{\delta}\right)\right)$ *such that for all* $t \geq N(\varepsilon, \delta)$ *we have* $P\left(\sup_{\theta\in\Theta} |L(\theta) - L(\theta \mid \mathcal{D}_t)| \leq \varepsilon\right) \geq 1 - \delta$.

According to our assumptions, the parameter set Θ is a compact subset of a finite dimensional Euclidean space and $p(y \mid x, \theta)$ is bounded away from zero for all $x \in \mathcal{X}$, for all $y \in \mathcal{Y}$, and for all $\theta \in \Theta$. Thus, we can assume that there exists a constant $p_{\min} > 0$ such that $p(y \mid x, \theta) \geq p_{\min}$. Let $\Lambda = \max_{\theta\in\Theta} \lambda_1(\theta)$, where λ_1 denotes the largest eigenvalue of the Hessian matrix of the importance-weighted negative log-likelihood objective function. As the set Θ is compact and the likelihood is a continuous function for all $x \in \mathcal{X}$, the eigenvalues of the Hessian matrix are bounded. Therefore, there exists a finite maximizer Λ. From the compactness of the set Θ, it also follows there exists a finite ε-cover of this set for all $\varepsilon > 0$. Before we proceed with the proofs, for the purpose of clarity, we review the notation introduced in Section 3. We denote with $R > 0$ the radius of a ball containing the set Θ in its interior, i.e., $\forall\theta \in \Theta$ it holds $\|\theta\| < R$. Similarly, $r > 0$ denotes the radius of a ball containing the mapped features in its interior, i.e., $\forall x \in \mathcal{X}, y \in \mathcal{Y}$ it holds $\|\phi(x, y)\| < r$.

Proof. We define all random variables with respect to a probability space $(\Omega, \mathcal{D}, \mathbb{P})$, where Ω is a state space, $\mathcal{D}$ is a σ-algebra of Ω, and $\mathbb{P}$ a probability measure of $\mathcal{D}$. The sampling process is performed using an external source of randomness which we model with an i.i.d. sequence of random variables $\{U_t\}_{t\in\mathbb{N}}$. We fix the filtration $\{\mathcal{D}_t\}_{t\in\mathbb{N}}$ where $\mathcal{D}_t \subset \mathcal{D}$ is the σ-algebra generated by $\{(U_1, \theta_1, x_1, y_1), \ldots, (U_t, \theta_t, x_t, y_t)\}$.

The input-output pair (x_{t+1}, y_{t+1}) is measurable with respect to the σ-algebra generated by $(\mathcal{D}_t, U_{t+1})$. In other words, given the history of observations the pair is random only with respect to U_{t+1}.

Having defined our random variables, we proceed with the proof. In a part of the proof we use some of the standard techniques from the theory of martingales and follow the same principle as the proof of the importance weighted active learning (Beygelzimer, Dasgupta, and Langford 2009). In the first step, we show that $\mathbb{E}_{\mathcal{D}_t}\left[L\left(\theta \mid \mathcal{D}_t\right)\right] = L\left(\theta\right)$. In particular, it holds

$$\mathbb{E}\left[L\left(\theta \mid \mathcal{D}_t\right)\right] = \frac{1}{t}\sum_{i=1}^{t}\int \frac{p_0\left(y^*\right)}{p\left(y^* \mid x_i, \theta_i\right)} l\left(x_i, y_i, \theta\right) \mathbb{P}\left(\mathcal{D}_t\right) =$$

$$\frac{1}{t}\sum_{i=1}^{t}\int \frac{p_0\left(y^*\right)}{p\left(y^* \mid x_i, \theta_i\right)} p\left(x_i \mid y^*, \theta_i\right) p_0\left(y_i \mid x_i\right) l\left(x_i, y_i, \theta\right) \cdot$$

$$\underbrace{\int \mathbb{P}\left(\mathcal{D}_{t-1} \mid x_i, y_i, \theta_i\right)}_{=1} = \frac{1}{t}\sum_{i=1}^{t}\int l\left(x_i, y_i, \theta\right) p_0\left(x_i, y_i\right) =$$

$$L\left(\theta\right),$$

where $\ell\left(x, y, \theta\right) = A\left(\theta \mid x\right) - \langle \phi\left(x, y\right), \theta \rangle$.

In the second step of the proof, we bound the discrepancy between the empirical and the expected loss. As there is a dependence within the sample, we cannot rely on the concentration bounds requiring the independence assumption. Therefore, we introduce a sequence for which we prove it is a martingale and then proceed with bounding the discrepancy using a martingale concentration inequality.

Let W_j, $j = 1, \ldots, t$, be a sequence of random variables such that

$$W_j = -w_j \ln p\left(y_j \mid x_j, \theta\right) - L\left(\theta\right), \tag{7}$$

where $w_j = \frac{p_0\left(y^*\right)}{p\left(y^* \mid x_j, \theta_j\right)}$. According to the assumptions, $p\left(y \mid x, \theta\right)$ is bounded away from zero for all $x \in \mathcal{X}$, for all $y \in \mathcal{Y}$, and for all $\theta \in \Theta$. Thus, it holds $\sup_{\theta \in \Theta, x \in \mathcal{X}, y \in \mathcal{Y}} \left|\ln p\left(y \mid x, \theta\right)\right| < -\ln p_{\min}$. From here it implies that $|W_j| \leq -\frac{\ln p_{\min}}{p_{\min}} < \infty$ and $\mathbb{E}\left[|W_j|\right] < \infty$.

We now show that the sequence $Z_t = \sum_{j=0}^{t} W_j$, with $W_0 = 0$, is a martingale. In particular,

$$\mathbb{E}\left[Z_t \mid Z_{t-1}, \ldots, Z_0\right] =$$
$$Z_{t-1} + \mathbb{E}_{x_t, y_t \mid \mathcal{D}_{t-1}}\left[w_t l\left(x_t, y_t, \theta\right)\right] - L\left(\theta\right) = Z_{t-1}.$$

On the other hand, it holds $|Z_t - Z_{t-1}| = |W_t| \leq -\frac{\ln p_{\min}}{p_{\min}}$. From here using the inequality for martingales by Azuma (1967) we deduce

$$P\left(\left|L\left(\theta \mid \mathcal{D}_t\right) - L\left(\theta\right)\right| > \frac{\varepsilon}{2}\right) =$$
$$P\left(|Z_t| > \frac{t\varepsilon}{2}\right) < 2\exp\left(-\frac{t\varepsilon^2 p_{\min}^2}{8\left(\ln p_{\min}\right)^2}\right). \tag{8}$$

As this holds for all $\theta \in \Theta$, applying Lemma A.2 for $\nu = \frac{p_{\min}\varepsilon}{2r + \sqrt{4r^2 + 2\Lambda p_{\min}\varepsilon}}$ we get

$$P\left(\sup_{\theta \in \Theta}\left|L\left(\theta\right) - L\left(\theta \mid \mathcal{D}_t\right)\right| > \varepsilon\right) <$$
$$2\mathcal{N}\left(\Theta, \frac{p_{\min}\varepsilon}{2r + \sqrt{4r^2 + 2\Lambda p_{\min}\varepsilon}}\right)\exp\left(-\frac{t\varepsilon^2 p_{\min}^2}{8\left(\ln p_{\min}\right)^2}\right).$$

From the last equation and Proposition A.3 we get

$$\ln\frac{\delta}{2} \geq d\ln\frac{4R\left(2r + \sqrt{4r^2 + 2\Lambda p_{\min}\varepsilon}\right)}{p_{\min}\varepsilon} - \frac{t\varepsilon^2 p_{\min}^2}{8\left(\ln p_{\min}\right)^2} \Longrightarrow$$
$$t\left(\frac{p_{\min}}{\ln p_{\min}}\right)^2 \varepsilon^2 \in \Omega\left(d\ln\frac{1}{\varepsilon} + \ln\frac{1}{\delta}\right) \Longrightarrow$$
$$t \in \Omega\left(\left(\frac{\ln p_{\min}}{p_{\min}}\right)^2 \frac{1}{\varepsilon^2}\left(d\ln\frac{1}{\varepsilon} + \ln\frac{1}{\delta}\right)\right)$$

Hence, we have shown that there exists a positive integer $N \in \Omega\left(\frac{1}{\varepsilon^2}\left(d\ln\frac{1}{\varepsilon} + \ln\frac{1}{\delta}\right)\right)$ such that for all $0 < \varepsilon, \delta < 1$, and all $t > N$ the claim holds.

□

Corollary 3. *The sequence of estimators $\{\theta_t\}_{t \geq 1}$ converges in probability to $\theta^* \in \Theta$.*

Proof. First note that from the compactness of Θ, it follows that the Hessian of the negative log-likelihood is strictly positive definite and, therefore, there exist unique minimizers of the loss functions $L\left(\theta\right)$ and $L\left(\theta \mid \mathcal{D}_t\right)$. From Theorem 2, we have that for sufficiently large t with probability $1 - \delta$ it holds that $L\left(\theta^* \mid \mathcal{D}_t\right) \leq L\left(\theta^*\right) + \varepsilon$ and $L\left(\theta_t\right) \leq L\left(\theta_t \mid \mathcal{D}_t\right) + \varepsilon$. From the strict convexity of the optimization objective $L\left(\cdot \mid \mathcal{D}_t\right)$ it follows that $L\left(\theta_t \mid \mathcal{D}_t\right) \leq L\left(\theta^* \mid \mathcal{D}_t\right)$. Hence, with probability $1 - \delta$

$$L\left(\theta_t\right) - L\left(\theta^*\right) \leq \left|L\left(\theta_t\right) - L\left(\theta_t \mid \mathcal{D}_t\right)\right| +$$
$$L\left(\theta_t \mid \mathcal{D}_t\right) - L\left(\theta^* \mid \mathcal{D}_t\right) + \left|L\left(\theta^* \mid \mathcal{D}_t\right) - L\left(\theta^*\right)\right| \leq 2\varepsilon.$$

From here it follows that the sequence of estimators $\{\theta_t\}_{t \geq 0}$ converges in probability to the optimal parameter θ^*. □

Definition A.1. *Let $\mathcal{M}$ be a finite, ergodic Markov chain defined on a state space Ω with transition probabilities $p(x \to x')$. A* coupling *is a joint process $(\mathcal{A}, \mathcal{B}) = (A_t, B_t)$ on $\Omega \times \Omega$ such that each of processes $\mathcal{A}$, $\mathcal{B}$, considered marginally, is a faithful copy of $\mathcal{M}$.*

Lemma A.4. *(Aldous 1983) Let $\mathcal{M}$ be a finite, ergodic Markov chain, and let (A_t, B_t) be a coupling for $\mathcal{M}$. Suppose that $P(A_{t(\varepsilon)} \neq B_{t(\varepsilon)}) \leq \varepsilon$, uniformly over the choice of initial state (A_0, B_0). Then the mixing time $\tau(\varepsilon)$ of $\mathcal{M}$ (starting at any state) is bounded from above by $t(\varepsilon)$.*

Proposition 4. *For all $0 < \varepsilon < 1$, with probability $1 - \varepsilon$, the mixing time $\tau(\varepsilon)$ of an independent Metropolis–Hastings chain for sampling from the posterior distribution $p(x \mid y^*, \theta_t)$ is bounded from above by $\lceil \ln\varepsilon / \ln\left(1 - \exp(-4r\left\|\theta_t\right\|)\right) \rceil$.*

Proof. As $\min_{x\in\mathcal{X}} p\left(y^* \mid x, \theta_t\right) \leq \max_{x\in\mathcal{X}} p\left(y^* \mid x, \theta_t\right)$, the lower bound on the Metropolis–Hastings acceptance criterion is never greater than 1. Then, from Eq. (2) and (1) it follows that, for a finite space $\mathcal{Y}$, the transition probability from a state x to a state x' satisfies

$$p(x \to x') \geq \frac{\exp\Big(\big\langle \phi(x', y^*), \theta_t \big\rangle - A(\theta_t \mid x')\Big)}{\exp\Big(\big\langle \phi(x, y^*), \theta_t \big\rangle - A(\theta_t \mid x)\Big)} = \frac{\sum_{\overline{y}\in\mathcal{Y}} \exp\big(\langle \phi\left(x', y^*\right) + \phi\left(x, \overline{y}\right), \theta_t\rangle\big)}{\sum_{\overline{y}\in\mathcal{Y}} \exp\big(\langle \phi\left(x, y^*\right) + \phi\left(x', \overline{y}\right), \theta_t\rangle\big)}.$$

Now, we can lower bound the transition probability by

$$p(x \to x') \geq \frac{|\mathcal{Y}| \exp\big(2 \cdot \langle \phi\left(x_\downarrow, y_\downarrow\right), \theta_t\rangle\big)}{|\mathcal{Y}| \exp\big(2 \cdot \langle \phi\left(x_\uparrow, y_\uparrow\right), \theta_t\rangle\big)} \geq \exp\Big(-2 \cdot \big|\big\langle \phi(x_\downarrow, y_\downarrow) - \phi(x_\uparrow, y_\uparrow), \theta_t \big\rangle\big|\Big),$$

where $\langle \phi\left(x_\downarrow, y_\downarrow\right), \theta_t\rangle$ and $\langle \phi\left(x_\uparrow, y_\uparrow\right), \theta_t\rangle$ are the minimum and maximum values of the dot products appearing in the numerator and denominator of $p(x \to x')$, respectively.

Then, using the Cauchy–Schwarz inequality, we derive

$$p(x \to x') \geq \exp\big(-2\big\|\phi(x_\downarrow, y_\downarrow) - \phi(x_\uparrow, y_\uparrow)\big\|\|\theta_t\|\big).$$

From our assumptions we have that $\|\theta\| \leq R$ and $\|\phi(x, y)\| \leq r$. Thus, it holds that

$$p(x \to x') \geq \exp(-4r\|\theta_t\|) \geq \exp(-4Rr). \tag{9}$$

From Eq. (9) it follows that the probability of not coalescing for T steps is upper bounded by $\big(1 - \exp(-4r\|\theta_t\|)\big)^T$. Then for $t(\varepsilon) = \big\lceil \ln \varepsilon / \ln\big(1 - \exp(-4r\|\theta_t\|)\big)\big\rceil$, we have

$$P(A_{t(\varepsilon)} \neq B_{t(\varepsilon)}) \leq \big(1 - \exp(-4r\|\theta_t\|)\big)^{t(\varepsilon)} = \varepsilon,$$

and from the coupling lemma (Aldous 1983; Guruswami 2000, or see Lemma A.4/Definition A.1) we conclude that a chain has mixed after $t \geq t(\varepsilon)$ steps with probability $1 - \varepsilon$. □

Figure 3: The figure shows the lift of correct-construction curves for the considered concepts and their complements. The lift indicates that our approach is capable of emphasizing a target property, irrespective of its type – rare or dominant class.

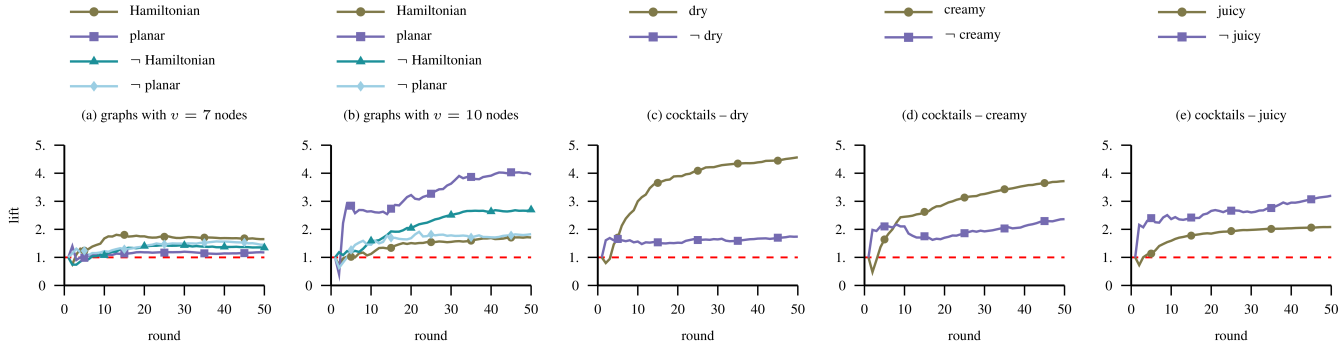

Table 1: We report the fraction of target structures observed within 50 000 samples from the proposal generators. The sampling was performed 5 times and the reported values are mean and standard deviation of the fractions computed over these runs.

Graphs, $v = 7$		Graphs, $v = 10$		Cocktails		
Hamiltonian	Connected Planar	Hamiltonian	Connected Planar	dry	creamy	juicy
36.68% (±0.24)	65.01% (±0.20)	77.45% (±0.28)	8.68% (±0.15)	11.27% (±0.14)	16.83% (±0.14)	34.54% (±0.13)

B Additional plots

In this appendix, we present additional findings from our synthetic experiments.

Emphasis of a target class. In Table 1, we give the distribution of the considered target classes and their complements. The table indicates that in our experiments we consider cases in which a target class is both rare and dominant. For graph concepts, we consider the spaces of unlabelled graphs with 7 and 10 vertices. The search space of graphs with 7 vertices contains only 1044 non-isomorphic graphs and it was used for sanity checks and tuning of our algorithm. The search space of graphs with 10 vertices, on the other hand, contains 12 005 168 graphs and on this space for minority class properties we have compared our approach to competing methods. The results of this comparison are presented in Section 4. Here, we present additional results indicating that our approach is able to emphasize a target class property even in cases when the property is dominant.

Similar to Section 4, to quantify the improvement of our approach over the plain Monte Carlo search performed with a proposal generator we measure the lift of its correct-construction curve. However, for sampling from the majority class the lift is defined as the ratio of the number of non-targets in a sample from the proposal generator and the number of such structures generated by our approach. In other words, for dominant properties the lift quantifies how much less likely it is to observe a non-target property when performing the discovery using Algorithm 1 compared to the Monte Carlo search. In Figure 3, we show that Algorithm 1 is able to emphasize the target classes in all domains, irrespective of their types – dominant or rare. Moreover, the magnitude of this emphasis is increasing over time and it is more likely to discover a target as the process evolves.

Designed cocktails. In Table 2, for each of the three cocktail property classes, we give five example cocktails constructed by our algorithm at the termination of the experiment. The color of each ingredient indicates relative proportion in the cocktail; more blue indicates a stronger presence. The evaluation oracles used in these experiments are described in Appendix C.4. It is possible to compare the constructed cocktail recipes to the real cocktails by looking up the displayed sets of ingredients at `http://bartenderapp.com`.

Table 2: Designed cocktails

dry	creamy	juicy
midori melon liqueur jagermeister tequila	creme de cacao cream bailey's irish cream	vodka orange juice orange soda
gin powdered sugar nutmeg lemon	bailey's irish cream green creme de menthe coffee heavy cream nutmeg	pineapple juice tequila blue curacao vodka
vodka triple sec grenadine	gin green creme de menthe creme de cacao	sour mix lemon juice tequila
scotch jack daniels kahlua	bailey's irish cream kahlua brandy	gin sweet and sour orange juice
jagermeister malibu rum	milk kahlua sugar	cranberry juice peach schnapps

C Implementation details

In this appendix, we describe in details our experimental set up including the used proposal generators, kernel functions, and evaluation oracles. We first briefly review the competing approaches and specifics of their implementations.

C.1 Competing methods

k-NN active search. One limitation of k-NN active search (Garnett et al. 2012) is the requirement that a finite set of structures needs to be provided to the algorithm beforehand. In our experiments, we have simulated this approach with $50\,000$ structures sampled from the proposal generator and k-NN probabilistic model with $k = 50$ (ties are not possible due to the choice of the hyperparameters).

In order to apply this approach to large sets of structures with a different probabilistic model an efficient pruning strategy needs to be devised. In the original paper, the authors give the pruning rules only for k-NN probabilistic models. As it is non-trivial to come up with pruning rules for conditional exponential family models, a search with these models would be inefficient. For instance, for the investigated case of Hamiltonian graphs with 10 vertices and an extensional description in the form of a sample with less than 1% of all structures from this space the active search with 2-step look-ahead (Garnett et al. 2012) and a budget of 500 oracle evaluations requires more than 50 million parameter fittings for the search modeled with a conditional exponential family.

Argmax search. The argmax search works by taking a sample of structures from a proposal generator and then choosing a structure from this sample with the highest conditional probability of it being a target. The selected structure is then evaluated by the oracle and the conditional model is updated to account for this new observation. This method is designed to compensate for the fact that k-NN active search with 1-step look-ahead (Garnett et al. 2012) requires a finite sample of structures. For graphs, we use the uniform proposal generator in combination with this strategy and this is the most exploratory proposal generator for this greedy search strategy. The approach is simulated with the conditional exponential family model, kernels and proposal generators identical to the ones used by our approach.

C.2 Proposal generators

A proposal generator is used to propose structures from an intensionally given domain. To be able to discover structures with desired properties, the support of the proposal generator must contain them. The uniform proposal generator is generally a safe choice, as it guarantees that the entire space will eventually be discovered. In the absence of any other information about the desired structures, we advocate the uniform proposal generators from a maximum-entropy standpoint. However, when domain-specific knowledge is available, the proposal generator can be modified appropriately. This can be especially useful when the class of structures with desired properties is relatively small compared to the entire search space, in which case the uniform proposal generator might need many samples to see a positive candidate.

We note here that while our theoretical results hold for proposal generators modeled with conditional Markov chains, we only demonstrate the potential of our approach in the setting with uninformed samplers as proposal generators. The experiments with domain-specific conditional chains are deferred to a journal version of the paper.

Sampling sparse vectors. Here, we describe a method for sampling sparse vectors based on a small set of examples. The small set of examples is there to serve as side knowledge facilitating the design of a proposal generator. We use this method in Section 4 to propose cocktail recipes which are represented as sparse real-valued vectors.

In the first step, we ensure that the sampled construction is a sparse vector by setting the number of non-zero entries in it. This is achieved by sampling the number of non-zero components from a Poisson distribution whose mean parameter was obtained by moment-matching from data. This sampling process is repeated until we sample a number of non-zero components greater than one.

In the second step, we sample components of the construction and their values. In case we are interested in sampling only binary vectors, than the value sampling part can be skipped. To choose the components of a construction we need to make sure that components with high frequency of appearance in data and combinations of frequently co-occurring components are selected more often. We achieve this by first building a component graph or an ingredient network in the case of cocktails with edges between the nodes weighted in accordance with the co-occurrence frequency. In addition to this, we keep a vector with the frequencies of appearance of each component in the data. Then, the first non-zero component of a construction is chosen by sampling proportional to their appearance frequencies. Once the first component is sampled, we take its neighbors in the component graph and choose the next component by sampling proportional to the edge weights. Having sampled more than one component, we create a candidate frequency vector by merging the neighbor lists of the sampled components and accumulating the edge weights of components with multiple occurrences. In the last step of the process we sample the component values. We perform this step by taking samples from a triangular distribution. The mode and the value range parameters of these distributions are moment-matched from data (each component has a distribution).

We conclude the section with a formal description of the sampling process. A construction represented as a sparse vector is proposed with the following sequence of steps: (*i*) $m \sim \text{Poisson}(\eta)$, (*ii*) $c_1 \sim \pi$ (*iii*) $c_k \sim \tau_k = \sum_{s=1}^{k-1} \pi^{(c_s)}$ $(2 \leq k \leq m)$, (*iv*) $v_k \sim \text{Triangular}\,(u_{c_k}, v_{c_k}, \mu_{c_k})$ $(1 \leq k \leq m)$, where c_k denotes a non-zero component of the vector being sampled, τ_k the frequency vector such that the probability of sampling a component j is proportional to τ_{kj}, and v_k is a value assigned to the component c_k. The parameters η, π, $\pi^{(c_s)}$, u_{c_k}, v_{c_k}, and μ_{c_k} are moment-matched from data. In other words, given a data set with n sparse d-dimensional vectors we do the following: (*i*) $\eta = \frac{1}{n}\sum_{i=1}^{n} n_i$, where n_i denotes the number of non-zero components in the instance x_i; (*ii*)

$\pi_j = \frac{1}{n'} \sum_{i=1}^{n} \mathbf{1}_{x_{ij} \neq 0}$, where x_{ij} denotes the component j in the vector x_i, π_j its frequency of appearance in data, and $n' = \sum_{i=1}^{n} n_i$; (*iii*) $\pi_j^{(c_s)} = \frac{1}{n'_{c_s}} \sum_{i:x_{ic_s} \neq 0} \mathbf{1}_{x_{ij} \neq 0}$, where $n'_{c_s} = \sum_{i:x_{ic_s} \neq 0} (n_i - 1)$, $j \neq c_s$, $1 \leq j, c_s \leq d$, $\pi_{c_s}^{(c_s)} = 0$; (*iv*) $u_j = \max(0, \mu_j - 2 * \sigma_j)$, $v_j = \min(1, \mu_j + 2 * \sigma_j)$, and μ_j and σ_j are the mean and the standard deviation of the component j computed over the instances with non-zero values at this component. To enable the sampling of sparse vectors with combinations of non-zero components which are not appearing together in data we do the Laplace smoothing of vectors $\pi^{(c_s)}$ by adding $1/d$ to each of its components.

Sampling unlabelled graphs with *n* vertices. As the set of graphs is a complicated, combinatorial object, it can be difficult to design an efficient uninformed sampler. In general, to sample a random unlabelled graph it is common to use the Erdős–Rényi model with $p = 1/2$. This approach, however, samples some graphs too often and does not provide sufficient diversity to the constructive process (e.g., the probability of sampling an unlabelled path with n vertices is $\frac{n!}{2}$ times higher than the probability of sampling the complete graph with the same number of vertices). Instead, one could try to first sample the parameter p uniformly at random and then to sample a graph with edge probability p. The last method does not generate unlabelled graphs u.a.r., but it can be used to efficiently sample some graph concepts (e.g., acyclic graphs). In this paper we take the safest route and choose to propose graphs with n vertices using the uniform sampler. We now do a review of this sampler for unlabelled graphs with n vertices.

Let $\mathcal{G}_n$ denote the set of all canonically labelled graphs with n vertices. A *left action* of a group S on a set X is a function $\mu \colon S \times X \to X$ with the following two properties: (*i*) $(\forall x \in X)(\forall s, t \in S) : \mu(t, \mu(s, x)) = \mu(ts, x)$; (*ii*) $(\forall x \in X) : \mu(e, x) = x$ (where e is the identity element of the group S). If no confusion arises we write $\mu(s, x) = sx$. A group action defines the equivalence relation $\sim$ on a set X, i.e., $a \sim b \Leftrightarrow sa = b$ for some $s \in S$ and $a, b \in X$. The equivalence classes determined by this relation are called *orbits* of S in X. The number of orbits can be computed using the Frobenius–Burnside theorem (Cameron 1998).

Theorem C.1 (Frobenius–Burnside)**.** *Let X be a finite non-empty set and S be a finite group. If X is an S-set, then the number of orbits of S in X is equal to* $\frac{1}{|S|} \sum_{s \in S} |\operatorname{Fix}(s)|$ *, where* $\operatorname{Fix}(s) = \{x \in X \mid sx = x\}$.

To sample unlabelled graphs uniformly at random, Wormald (1987) proposed a rejection sampling method based on Theorem C.1. The idea is to consider the action of a symmetric group S_n over the set $\mathcal{G}_n$. Then, the orbits of S_n in the set $\mathcal{G}_n$ are non-isomorphic unlabelled graphs and to sample unlabelled graphs uniformly it suffices to uniformly sample the orbits (Dixon and Wilf 1983). Moreover, it is possible to show (Dixon and Wilf 1983; Wormald 1987) that uniform orbit sampling is equivalent to uniform sampling of an element from the set $\Gamma = \{(\pi, g) \mid \pi g = g; \pi \in S_n, g \in \mathcal{G}_n\}$.

According to Theorem C.1, an element $(\pi, g) \in \Gamma$ can be sampled u.a.r. by choosing a permutation π with probability proportional to $|\operatorname{Fix}(\pi)|$ and then choosing $g \in \operatorname{Fix}(\pi)$ u.a.r. Dixon & Wilf (1983) propose a more efficient sampling algorithm by partitioning the symmetric group into conjugacy classes $[\pi_i]$ $(1 \leq i \leq l)$ and sampling: (*i*) $[\pi_i] \sim {}^{|[\pi_i]||\operatorname{Fix}(\pi_i)|}/_{o_n|S_n|}$, (*ii*) $g \in \operatorname{Fix}(\pi_i)$ u.a.r.; where o_n denotes the number of non-isomorphic unlabelled graphs and π_i is a class representative for the class $[\pi_i]$. As it holds $|\operatorname{Fix}(\pi)| = |\operatorname{Fix}(\pi')|$ and $|\operatorname{Fix}(\pi) \cap [g]| = |\operatorname{Fix}(\pi') \cap [g]|$ for $\pi, \pi' \in [\pi_i]$ then (Wormald 1987)

$$P\big([g]\big) = \sum_{i=1}^{l} P\big([g], [\pi_i]\big) = \sum_{i=1}^{l} P\big([\pi_i]\big) P\big([g] \mid [\pi_i]\big) = \sum_{i=1}^{l} \frac{\big|[\pi_i]\big|\big|\operatorname{Fix}(\pi_i)\big|}{o_n|S_n|} \frac{\big|\operatorname{Fix}(\pi_i) \cap [g]\big|}{\big|\operatorname{Fix}(\pi_i)\big|} = \frac{1}{o_n}.$$

The problem with the approach is the fact that we need to know the exact number of non-isomorphic graphs with n vertices o_n to apply the algorithm and this number is not computable in polynomial time. To overcome this, Wormald (1987) partitions the elements of the group S_n into classes $[c_k] = \{\pi \in S_n \mid \operatorname{support}(\pi) = k\}$, $0 \leq k \leq n$, and upper bounds $\big|[c_i]\big|\big|\operatorname{Fix}(\pi_i)\big| \leq B_i$. The algorithm then samples an unlabelled graph u.a.r. as follows: (*i*) $[c_i] \sim {}^{B_i}/_{\sum_j B_j}$, (*ii*) $\pi_i \in [c_i]$ u.a.r., (*iii*) $g \in \operatorname{Fix}(\pi_i)$ u.a.r., and (*iv*) accept the sampled graph g with probability $B_i^{-1}\big|[c_i]\big|\big|\operatorname{Fix}(\pi_i)\big|$; otherwise, restart. On average, the method generates an unlabelled graph in time polynomial in the number of vertices.

C.3 Kernel functions

In this section, we describe application-specific kernel functions used in Section 4. For these kernel functions, we follow the standard procedure for tuple kernels and take kernels which factor into the product of domain kernels, $k\left((x, y), (x', y')\right) = k_{\mathcal{X}}(x, x') k_{\mathcal{Y}}(y, y')$, where $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are kernel functions over spaces $\mathcal{X}$ and $\mathcal{Y}$.

First we describe the property space kernel function $k_{\mathcal{Y}}$. In all the cases considered in Section 4 the property space $\mathcal{Y}$ is binary and we use the identity kernel. This is the space requiring the simplest feedback and the least effort from an evaluation oracle. In more complex experiments such as drug design, the evaluation oracle could output a structured label such as binary vector reflecting different aspects of the construction – price, binding properties, stability etc. In these cases, one could take the property space $\mathcal{Y}$ to be the power set of elementary properties and use the intersection kernel $k_{\mathcal{Y}}(y_i, y_j) = |y_i \cap y_j|$.

In the remainder of the section we describe the instance space kernels for the investigated domains. To apply our algorithm to the space of graphs we use the random walk kernel (Gärtner, Flach, and Wrobel 2003). The kernel performs random walks on both graphs and counts the number of matching walks. It can be computed as

$$k_{\mathcal{X}}(G_1, G_2) = \sum_{i,j=1}^{|\mathcal{V}_\times|} \sum_{n=0}^{\infty} [\lambda_n E_\times^n]_{ij}\,, \tag{10}$$

where $E_\times$ denotes the adjacency matrix of the product graph $G_1 \times G_2$ and $\{\lambda_n\}$ is a sequence of hyperparameters that needs to be set such that the sum in (10) converges for any pair of graphs G_1 and G_2. We apply the kernel with $\lambda_n = \lambda^n$ to

unlabelled graphs, and for this particular case $E_\times = E_1 \otimes E_2$. The kernel can be computed efficiently using the fixed-point method (Borgwardt 2007).

To apply our algorithm to the space of sparse real-valued vectors, we use the Gaussian kernel with diagonal relevance length scale matrix M, i.e., $k_{\mathcal{X}}(x, x') = \exp\left(-1/2(x - x')^\top M(x - x')\right)$. For each coordinate we set the relevance scale as

$$m_{jj} = \frac{2n}{\sqrt{\mathrm{nnz}}\left(\max_{i=1}^{n} x_{ij} - \min_{i=1}^{n} x_{ij}\right)},$$

where n denotes the number of instances, nnz the total number of non-zero entries in the data set, d dimension of the instances, and $1 \leq j \leq d$.

C.4 Evaluation oracles

DRY

[1 | jagermeister ≥ 0.225 ? DRY : **go to** 2]
[2 | gin ≥ 0.465639 ? DRY : **go to** 3]
[3 | jackdaniels ≥ 0.138889 ? DRY : **go to** 4]
[4 | 151 proof rum ≥ 0.291666 ? DRY : **go to** 5]
[5 | vodka ≥ 0.437037 ? DRY : NOT DRY]

CREAMY

[1 | bailey's irish cream ≥ 0.03324 ? CREAMY : **go to** 2]
[2 | creme de cacao ≥ 0.0059365 ? CREAMY : **go to** 3]
[3 | milk ≥ 0.21495 ? CREAMY : **go to** 4]
[4 | irish cream ≥ 0.006375 ? CREAMY : **go to** 5]
[5 | cream ≥ 0.014754 ? CREAMY : NOT CREAMY]

JUICY

[1 | orange juice ≥ 0.040152 ? JUICY : **go to** 2]
[2 | cranberry juice ≥ 0.084 ? JUICY : **go to** 3]
[3 | pineapple juice ≥ 0.183334 ? JUICY : **go to** 4]
[4 | sour mix ≥ 0.0625 ? JUICY : **go to** 5]
[5 | sweet and sour ≥ 0.274614 ? JUICY : NOT JUICY]