Online Forest Density Estimation

Frédéric Koriche
CRIL, CNRS UMR 8188
Université d'Artois, France
koriche@cril.fr

Abstract

Online density estimation is the problem of predicting a sequence of outcomes, revealed one at a time, almost as well as the best expert chosen from a reference class of probabilistic models. The performance of each expert is measured with the log-likelihood loss. The class of experts examined in this paper is the family of discrete, acyclic graphical models, also known as *Markov forests*. By coupling Bayesian mixtures with symmetric Dirichlet priors for parameter learning, and a variant of "Follow the Perturbed Leader" strategy for structure learning, we derive an online forest density estimation algorithm that achieves a regret of $\tilde{O}(\sqrt{T})$, with a per-round time complexity that is quasi-quadratic in the input dimension. Using simple and flexible update rules, this algorithm can be easily adapted to predict with Markov trees or mixtures of Markov forests. Empirical results indicate that our online algorithm is a practical alternative to the state-of-the-art batch algorithms for learning tree-structured graphical models.

1 INTRODUCTION

Graphical models have attracted considerable interest in AI, computational statistics, and machine learning (Wainwright and Jordan, 2008; Koller and Friedman, 2009). One of the key virtues of these models is to allow a separation between qualitative, structural aspects of uncertain knowledge, and quantitative, parametric aspects of uncertainty. As such, graphical models are able to represent, in a compact and intelligible way, high-dimensional probability distributions, using local interactions between variables. For *undirected* graphical models, also known as *Markov networks*, the structure is an undirected graph G, and the parameters are grouped into a set $\boldsymbol{\theta}$ of factors associated with the cliques of G. The probability $\mathbb{P}_M(\boldsymbol{x})$ assigned to an outcome $\boldsymbol{x}$ by a model $M = (G, \boldsymbol{\theta})$ is given by the product of factors in $\boldsymbol{\theta}$ which are activated by $\boldsymbol{x}$, divided by a normalization constant, known as the *partition function*.

A fundamental problem in graphical models is to extract from a series of observed outcomes, the structure and the parameters of a model that accurately predicts future, unseen, outcomes. This learning problem, which can be generalized to arbitrary probabilistic models, is often referred to as *density estimation* in the literature (Grünwald, 2007; Rissanen, 2012). In the *batch* density estimation setting, it is assumed that outcomes are sampled independently from a fixed (but unknown) target distribution. The data samples, available ahead of time, are separated into a training set for learning the model, and a test set for evaluating its performance. Contrastingly, in the *online* density estimation setting, there are no statistical assumptions about the series of outcomes (Merhav and Feder, 1998; Cesa-Bianchi and Lugosi, 2006). The learner receives inputs sequentially, and its performance is measured over all the observed sequence. The absence of statistical assumption makes online algorithms applicable in adaptive or "dynamic" environments, where the target distribution is allowed to arbitrarily change in response to various events, including the learner's decisions. Even in "static" environments, online algorithms can provide a practical alternative to batch algorithms, by processing only one outcome at a time. They are indeed particularly suited to handle streaming applications, where all the data is not available in advance, or large-scale domains with massive amounts of data.

Conceptually, online density estimation with graphical models can be viewed as a repeated game between the learner and its environment. The parameters of the game are an outcome space $\mathcal{X}$ and a class $\mathcal{M}$ of graphical models over $\mathcal{X}$, called *experts*. During each trial t of the game, the learner selects (possibly at random) a model $M^t \in \mathcal{M}$, the environment responds by an outcome $\boldsymbol{x}^t \in \mathcal{X}$, and the learner incurs the log-likelihood loss (or *log-loss*, for short) $\ell(M^t, \boldsymbol{x}^t) = -\ln \mathbb{P}_{M^t}(\boldsymbol{x}^t)$. The quality of an online learning algorithm is measured according to two standard metrics. The first, called *regret*, measures the difference in cumulative loss between the algorithm and the best expert in $\mathcal{M}$. Borrowing the terminology of game theory, an online learning algorithm is called *Hannan-consistent* if its regret over any possible sequence of T outcomes is only sublinear in T. The second metric is computational complexity, i.e. the amount of resources required to compute M^t at each round t, given the sequence of outcomes observed so far.

In this paper, we examine the problem of online density estimation for the class of *(discrete) Markov forests*, which represent discrete multivariate probability distributions where interdependencies are restricted to an acyclic graph. Markov forests are endowed with two remarkable properties, namely, (i) they can be factorized into a *closed form* which does not involve a partition function, and (ii) the space of all acyclic graphs upon which a Markov forest can be constructed is a *matroid*. As observed in (Pearl, 1988; Lauritzen, 1996), the closed-form expression of the probability distribution $\mathbb{P}_M$ associated with an n-dimensional Markov forest $M = (F, \boldsymbol{\theta})$ is given by

$$\mathbb{P}_M(\boldsymbol{x}) = \prod_{i=1}^{n} \theta_i(x_i) \prod_{(i,j)\in F} \frac{\theta_{ij}(x_i, x_j)}{\theta_i(x_i)\theta_j(x_j)} \tag{1}$$

where $\theta_i(x_i)$ and $\theta_{ij}(x_i, x_j)$ are the marginal densities of the node i and the edge (i, j), respectively. Based on (1), probabilistic inference in Markov forests can be performed in linear time. Moreover, the matroid associated with the structure space of Markov forests allows linear optimization to be performed in low-polynomial time, using the greedy matroid algorithm.

Based on these properties, the "batch" forest density estimation problem can be solved in quasi-quadratic time (in the input dimension n), by finding a maximum weight spanning tree in the complete graph of order n, whose edges are weighted according to the empirical bivariate marginals measured on the training set. This simple and elegant strategy, due to Chow and Liu (1968), is the blueprint of more sophisticated algorithms for learning other tree-structured graphical models, such as constrained Markov forests (Liu et al., 2011; Tan et al., 2011), and mixtures of Markov trees (Meila and Jordan, 2000; Kumar and Koller, 2009). Beyond Markov forests and their variants, the problem of finding the structure and the parameters of a maximum likelihood graphical model is, in general, NP-hard (Chickering, 1995), even for the restricted classes of Bayesian polytrees (Dasgupta, 1999) and Markov networks of bounded treewidth (Srebro, 2003).

Our Results. The challenge of the "online" forest density estimation problem lies in the fact that outcomes are revealed only one at a time, thus forcing the learner to iteratively update both the structure and the parameters of a Markov forest, so as to minimize the cumulative log-loss over the sequence of outcomes observed so far. This difficulty naturally raises the question of whether there exist Hannan-consistent forest density estimation algorithms with a total runtime complexity comparable to that of batch learning algorithms. By exploiting the closed form of Markov forests and the matroid of their structure space, we answer this question in the affirmative using easily implementable update strategies.

The key point of our online learning algorithm and its regret analysis lies in the fact that the parameters and the structure of a forest can be updated in an *independent* way. Indeed, in light of the closed-form expression (1), the log-likelihood loss of a Markov forest can be additively decomposed to the nodes and the edges of a forest, in such a way that the contribution of the local components are independent of the forest structure. Thus, the regret of any algorithm producing the sequence $M^1, \cdots, M^T$ of Markov forests can be decomposed into a telescopic sum of two regret expressions, namely, a "parametric" part defined over the forest parameters $\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^T$, and a "structural" part defined over the forest structures $F^1, \cdots, F^T$.

By exploiting the additive decomposition of the log-loss, the parametric part can, in turn, be decomposed into a sum of "local" regrets defined over node and edge parameters. This observation naturally suggests the use of Bayesian mixtures under Dirichlet priors for estimating univariate marginals and bivariate marginals. Such mixtures, which have been extensively studied in the literature of density estimation (see e.g. Cesa-Bianchi and Lugosi (2006); Grünwald (2007)), can be implemented using very simple update rules. Namely, by selecting Jeffreys mixtures for univariate and bivariate marginal estimators, our strategy achieves a regret that is logarithmic in the number T of rounds, with a per-round time complexity that is quadratic in the input dimension n.

Concerning the structural part of the regret, the log-loss is an affine function of the forest structure. This, together with the matroid property of forest structures, opens up the possibility of using various online combinatorial optimization algorithms (see e.g. Koolen et al. (2010); Audibert et al. (2011)). Here, our structure-update strategy is based on the well-known *Follow the Perturbed Leader* (FPL) algorithm (Hannan, 1957; Kalai and Vempala, 2005), which essentially adds a random perturbation to the total loss observed so far, and selects the forest structure that minimizes the resulting cost function. In order to attenuate the possibly unstable effects of perturbations, our strategy uses a convex combination of forest structures, coupled with a swap-rounding method (Chekuri et al., 2010) for generating, at each iteration, a forest that is consistent with the current convex mixture. By ignoring logarithmic factors, this strategy achieves a regret of $\tilde{O}(\sqrt{T})$, with a quadratic per-round time complexity.

In a nutshell, our online forest density estimation algorithm achieves Hannan-consistency with a cumulative runtime complexity that is comparable to that of the Chow-Liu algorithm. Furthermore, our algorithm can be easily adapted to predict with Markov trees, and mixtures of Markov forests (with shared parameters). Experiments conducted on several real-world datasets support our theoretical approach. Notably, our online learning algorithm rapidly converges to the estimations of the state-of-the-art batch algorithms for Markov trees (Chow and Liu, 1968), and thresholded Markov forests (Tan et al., 2011).

Paper Structure. After introducing the necessary background in forest polytopes (Section 2) and Markov forests (Section 3), we present our online forest density estimation algorithm in Section 4. Its regret analysis is detailed in Section 5, and its experimental validation is presented in Section 6. Finally, Section 7 concludes with some related work in online learning, together with several perspectives of further research.

2 FOREST POLYTOPES

We start with some notations and definitions which will be used throughout the paper. Let $[n]$ denote the set $\{1, \cdots, n\}$, and $\binom{[n]}{2}$ denote the set $\{(i, j) \in [n] \times [n], i < j\}$. To simplify notation, we use the abbreviation $\boldsymbol{x}^{1:t}$ to designate any sequence of vectors $\boldsymbol{x}^1, \cdots, \boldsymbol{x}^t$. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and $p \in [1, \infty]$, we use $\|\boldsymbol{x}\|_p$ to denote the L_p norm of $\boldsymbol{x}$, and we use$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ to denote the scalar product of $\boldsymbol{x}$ and $\boldsymbol{y}$. For a subset $\boldsymbol{X} \subseteq \mathbb{R}^n$, we denote by $\operatorname{conv} \boldsymbol{X}$ the convex hull of $\boldsymbol{X}$.

In what follows, we shall adopt set and vector notations interchangeably for describing graphs over the node set $[n]$; in the set notation, G is a subset of $\binom{[n]}{2}$, and in the vector notation $\boldsymbol{g}$ is a vector in $\{0, 1\}^{\binom{n}{2}}$, such that $g_{ij} = 1$ if and only if $(i, j) \in G$. As usual, a *forest* is an acyclic graph, and a *spanning tree* is an acyclic, connected graph that spans $[n]$. The spaces of all forests and all spanning trees of order n are denoted $\mathbf{F}_n$ and $\mathbf{T}_n$, respectively. It is well-known that the set of all forests of order n defines a *matroid* over the ground set $\binom{[n]}{2}$, where $\mathbf{F}_n$ is the collection of independent sets, and $\mathbf{T}_n$ is the collection of bases. The rank of this matroid is $n - 1$, which corresponds to the size of any spanning tree over $[n]$.

The convex hull of $\mathbf{F}_n$, where elements are viewed in vector notation, is called the *forest polytope*. This polyhedron of dimension $n - 1$ is characterized by the system of inequalities:

$$\operatorname{conv} \mathbf{F}_n = \left\{ \boldsymbol{p} \in \mathbb{R}_+^{\binom{n}{2}} \colon \langle \boldsymbol{p}, \boldsymbol{g} \rangle \leq n-1, \text{ for all } \boldsymbol{g} \in \{0,1\}^{\binom{n}{2}} \right\}$$

Such inequalities are often referred to as *acyclicity constraints* in the literature (Shrijver, 2003). The convex hull of $\mathbf{T}_n$, called the *spanning tree polytope*, is the subset of $\operatorname{conv} \mathbf{F}_n$ formed by the points $\boldsymbol{p}$ satisfying the equality $\langle \boldsymbol{p}, \mathbf{1} \rangle = n - 1$, where $\mathbf{1}$ is the all-ones vector in $\mathbb{R}^n$. By Carathéodory's theorem, any point $\boldsymbol{p} \in \operatorname{conv} \mathbf{F}_n$ (resp. $\boldsymbol{p} \in \operatorname{conv} \mathbf{T}_n$) can be represented as a convex combination of $t \leq n - 1$ forests (resp. spanning trees), i.e. $\boldsymbol{p} = \sum_{\tau=1}^{t} \alpha_\tau \, \boldsymbol{f}^\tau$, where $\boldsymbol{\alpha} \in \mathbb{R}_+^t$ and $\|\boldsymbol{\alpha}\|_1 = 1$.

Although the forest polytope and the spanning tree polytope are characterized by an exponential number N of acyclicity constraints, linear optimization under these combinatorial structures can be performed in low polynomial time. Indeed, by Edmond's theorem (1970), both $\operatorname{conv} \mathbf{F}_n$ and $\operatorname{conv} \mathbf{T}_n$ are totally dual-integral, and hence, any minimizer $\boldsymbol{p}^*$ of a linear objective $\langle \boldsymbol{w}, \boldsymbol{p} \rangle$ subject to $\boldsymbol{p} \in \operatorname{conv} \mathbf{F}_n$ (resp. $\boldsymbol{p} \in \operatorname{conv} \mathbf{T}_n$) is an extreme point in $\mathbf{F}_n$ (resp. $\mathbf{T}_n$). This point $\boldsymbol{p}^*$ can be found in $O(n^2 \log n)$ time, using the greedy matroid algorithm. Specifically, for the forest polytope, the greedy algorithm first sorts the $\binom{n}{2}$ edges in decreasing order according to the linear objective $\boldsymbol{w}$, next keeps the first m edges with non-positive weight, and then iteratively finds a minimum cost forest F over these m ordered edges. For the spanning tree polytope, the greedy algorithm coincides with Kruskal's method, which also sorts the edges according to $\boldsymbol{w}$, but uses (in the worst case) all the $\binom{n}{2}$ ordered edges for generating a minimum cost spanning tree.

Finally, in order to round fractional points in matroid polytopes, we shall focus on SWAP method proposed by Chekuri et al. (2010). This algorithm takes as input a fractional point $\boldsymbol{p} \in \operatorname{conv} \mathbf{F}_n$ (resp. $\boldsymbol{p} \in \operatorname{conv} \mathbf{T}_n$), given as a convex combination $\boldsymbol{p} = \sum_{\tau=1}^{t} \alpha_\tau \boldsymbol{f}^\tau$ of forests (resp. spanning trees), and iteratively generates the sequence of points $\boldsymbol{p}^1, \cdots, \boldsymbol{p}^t$, such that $\boldsymbol{p}^1 = \boldsymbol{p}$, $\boldsymbol{p}^t$ is an extreme point in $\mathbf{F}_n$ (resp. $\mathbf{T}_n$), and $\mathbb{E}[\boldsymbol{p}^\tau] = \boldsymbol{p}$ for all $\tau \in [t]$. Each point $\boldsymbol{p}^{\tau+1}$ is obtained from $\boldsymbol{p}^\tau$ by arbitrarily choosing two components $\alpha_i \boldsymbol{f}^i$ and $\alpha_j \boldsymbol{f}^j$ in $\boldsymbol{p}^\tau$, and by replacing them with a new component $(\alpha_i + \alpha_j)\boldsymbol{f}'$. Here, $\boldsymbol{f}'$ is generated in $O(n^2)$ time using random base exchanges. So, $\boldsymbol{p}$ can be rounded using $t - 1$ quadratic-time operations. Importantly, SWAP can be interrupted at any iteration τ to give a convex combination $\boldsymbol{p}^\tau$ of $t+1-\tau$ graphical structures. In what follows, we use $\text{SWAP}_k(\boldsymbol{p})$ to denote the application of at most $\tau = t + 1 - k$ iterations of the SWAP algorithm, which returns a convex combination including at most k components.

3 MARKOV FORESTS

The graphical models examined in this paper are defined over a set $\{X_1, \cdots, X_n\}$ of multinomial random variables, each taking values over a finite alphabet $\{1, \cdots, m\}$, with $m \geq 2$.

A *probability table* for a random variable X_i, is a vector $\boldsymbol{\theta}_i$ in the m-dimensional probability simplex, where $\theta_i(u)$ denotes the probability that $X_i = u$. Similarly, a probability table for a pair of random variables (X_i, X_j) is a vector $\boldsymbol{\theta}_{ij}$ in the m^2-dimensional probability simplex, where $\theta_{ij}(u, v)$ indicates the probability that $X_i = u$ and $X_j = v$. By $\boldsymbol{\Theta}_{m,n}$, we denote the set of all mappings $\boldsymbol{\theta}$ that assign a probability table $\boldsymbol{\theta}_i$ to each $i \in [n]$, and a probability table $\boldsymbol{\theta}_{ij}$ to each $(i, j) \in \binom{[n]}{2}$, while satisfying the *marginalization constraints*:

$$\sum_{u=1}^{m} \theta_{ij}(u, v) = \theta_j(v) \text{ and } \sum_{v=1}^{m} \theta_{ij}(u, v) = \theta_i(u) \qquad (2)$$

Note that the dimension of $\boldsymbol{\theta}$ is $d = mn + m^2\binom{n}{2}$. The class of *m-ary n-dimensional Markov forests* is defined as $\mathcal{F}_{m,n} = \mathbf{F}_n \times \boldsymbol{\Theta}_{m,n}$, and the class of *$m$-ary n-dimensional Markov trees* is given by $\mathcal{T}_{m,n} = \mathbf{T}_n \times \boldsymbol{\Theta}_{m,n}$. For a class $\mathcal{M} \in \{\mathcal{F}_{m,n}, \mathcal{T}_{m,n}\}$, we denote by $\boldsymbol{P}(\mathcal{M})$ the matroid polytope associated with the structure space of $\mathcal{M}$, i.e. $\boldsymbol{P}(\mathcal{M}) = \operatorname{conv} \mathbf{F}_n$ if $\mathcal{M} = \mathcal{F}_{m,n}$, and $\boldsymbol{P}(\mathcal{M}) = \operatorname{conv} \mathbf{T}_n$ if $\mathcal{M} = \mathcal{T}_{m,n}$.

By taking into account the acyclicity constraints of $\mathbf{F}_n$ and the marginalization constraints of $\boldsymbol{\Theta}_{m,n}$, the probability distribution $\mathbb{P}_M$ over $\mathcal{X} = [m]^n$ represented by a Markov forest $M = (\boldsymbol{f}, \boldsymbol{\theta})$ is given by the closed-form expression (1). More generally, if $(\boldsymbol{p}, \boldsymbol{\theta})$ is a pair of vectors in $\boldsymbol{P}(\mathcal{M}) \times \boldsymbol{\Theta}_{m,n}$, then the corresponding distribution is given by

$$\mathbb{P}_{\boldsymbol{p},\boldsymbol{\theta}}(\boldsymbol{x}) = \prod_{i \in [n]} \theta_i(x_i) \prod_{(i,j) \in \binom{[n]}{2}} \left(\frac{\theta_{ij}(x_i, x_j)}{\theta_i(x_i)\theta_j(x_j)} \right)^{p_{ij}} \qquad (3)$$

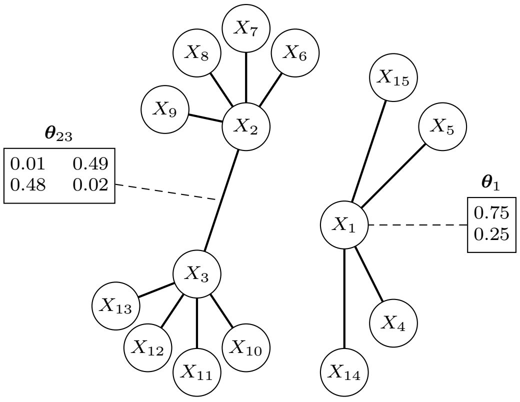

Figure 1: A binary Markov forest.

When $\boldsymbol{p}$ is described as a convex combination of forests (resp. trees), the pair $(\boldsymbol{p}, \boldsymbol{\theta})$ can be viewed as a *mixture* of Markov forests (resp. Markov trees) sharing the same parameters.

4 ONLINE MARKOV FORESTS

Recall that online learning can be viewed as a repeated game between a learning algorithm $\mathcal{A}$ and its environment. During each trial $t \in [T]$, the learner $\mathcal{A}$ starts by choosing (possibly at random) a model $M^t \in \mathcal{M}$, where $\mathcal{M}$ is a class of graphical models. Next, the environment responds by supplying an outcome $\boldsymbol{x}^t \in \mathcal{X}$, and then, $\mathcal{A}$ incurs the log-loss $\ell(M^t, \boldsymbol{x}^t) = -\ln \mathbb{P}_{M^t}(\boldsymbol{x}^t)$. The *(expected) regret* of the learning algorithm $\mathcal{A}$ with respect to the sequence of outcomes $\boldsymbol{x}^{1:T} = (\boldsymbol{x}^1, \cdots, \boldsymbol{x}^T)$ is given by

$$R_{\boldsymbol{x}^{1:T}}(\mathcal{A}) = \sum_{t=1}^{T} \mathbb{E}\left[\ell(M^t, \boldsymbol{x}^t)\right] - \min_{M \in \mathcal{M}} \sum_{t=1}^{T} \ell(M, \boldsymbol{x}^t) \quad (4)$$

where the expectation is taken with respect to the learner's internal randomization. By extension, the *minimax regret* of $\mathcal{A}$ at horizon T, denoted $R_T(\mathcal{A})$, is the maximum of $R_{\boldsymbol{x}^{1:T}}(\mathcal{A})$ over any sequence $\boldsymbol{x}^{1:T}$ in $\mathcal{X}^T$. $\mathcal{A}$ is called *Hannan-consistent* if its minimax regret is sublinear in T, or equivalently, if its average minimax regret $R_T(\mathcal{A})/T$ vanishes as $T \to \infty$.

The classes of experts investigated in this study are Markov forests and Markov trees, that is, $\mathcal{M} \in \{\mathcal{F}_{m,n}, \mathcal{T}_{m,n}\}$. The log-loss can be extended to $\boldsymbol{P}(\mathcal{M}) \times \boldsymbol{\Theta}_{m,n} \times \mathcal{X} \to \mathbb{R}$, using $\ell(\boldsymbol{p}, \boldsymbol{\theta}, \boldsymbol{x}) = -\ln \mathbb{P}_{\boldsymbol{p},\boldsymbol{\theta}}(\boldsymbol{x})$, where $\mathbb{P}_{\boldsymbol{p},\boldsymbol{\theta}}$ is defined according to the closed-form expression (3). Interestingly, we can observe that ℓ is an affine function of $\boldsymbol{p}$, given by

$$\ell(\boldsymbol{p}, \boldsymbol{\theta}, \boldsymbol{x}) = \psi(\boldsymbol{x}) + \langle \boldsymbol{p}, \boldsymbol{\phi}(\boldsymbol{x}) \rangle \quad (5)$$

where

$$\psi(\boldsymbol{x}) = \sum_{i \in [n]} \ln \frac{1}{\theta_i(x_i)} \text{ and } \phi_{ij}(x_i, x_j) = \ln\left(\frac{\theta_i(x_i)\theta_j(x_j)}{\theta_{ij}(x_i, x_j)}\right)$$

It is important to keep in mind that the sign of the components in the vector $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{\binom{n}{2}}$ can be positive or negative. A negative weight $\phi_{ij}(x_i, x_j)$ can be interpreted as a positive contribution (or gain) in favor of using the edge (i, j) in the graphical structure. Contrarily, a positive weight $\phi_{ij}(x_i, x_j)$ provides a negative contribution to the candidate edge (i, j).

With these notions in hand, we are now in position to examine the online forest density estimation (OFDE) algorithm. As specified in Algorithm 1, OFDE takes as input a class of experts $\mathcal{M} \in \{\mathcal{F}_{m,n}, \mathcal{T}_{m,n}\}$, and an upper bound k on the number of candidate structures maintained by its mixture. During each trial t, the learner maintains a pair $(\boldsymbol{p}^t, \boldsymbol{\theta}^t) \in \boldsymbol{P}(\mathcal{M}) \times \boldsymbol{\Theta}_{m,n}$, where $\boldsymbol{p}^t$ is a convex combination of at most k structures. The model $M^t = (\boldsymbol{f}^t, \boldsymbol{\theta}^t)$ used to predict the outcome $\boldsymbol{x}^t$ is obtained by generating $\boldsymbol{f}^t$ at random according to $\boldsymbol{p}^t$ (Line 4). After observing $\boldsymbol{x}^t$ (Line 5), the learner updates its parameters and its structure according to Lines 6-7 and Lines 8-12, respectively.

The vector of parameters $\boldsymbol{\theta}^t$ is updated by applying the Jeffreys (1946) rule to the probability table of each node $i \in [n]$ and each candidate edge $(i, j) \in \binom{[n]}{2}$. Here, t_u (resp. t_v) is the number of u's (resp. v's) in the sequence $x_i^1, \cdots, x_i^t$, and t_{uv} is the number of occurrences of (u, v) in $(x_i, x_j)^1, \cdots, (x_i, x_j)^t$.

The mixture $\boldsymbol{p}^t$ is updated using the following operations: first, apply the FPL strategy to produce an intermediate structure $\boldsymbol{f}^{t+\frac{1}{2}}$ (Lines 9-10); next, combine this structure with $\boldsymbol{p}^t$ to yield a new intermediate mixture $\boldsymbol{p}^{t+\frac{1}{2}}$ (Line 11), and then use SWAP$_k$ to build a new mixture with at most k components (Line 12). The values of the hyperparameters α_t and β_t, used to generate mixtures and perturbations, will be derived from regret analysis. Note that the same algorithmic scheme is used to learn with Markov forests and Markov trees. The key difference lies in the behavior of the greedy matroid algorithm; as mentioned above, the greedy algorithm uses only non-positive weights in $\phi^t(\boldsymbol{x}^t)$ to find an optimal point in $\operatorname{conv} \mathbf{F}_n$ for the linear objective $\boldsymbol{w}^t$, while it uses all weights in $\phi^t(\boldsymbol{x}^t)$ to produce an optimal point in $\operatorname{conv} \mathbf{T}_n$ for $\boldsymbol{w}^t$.

Theorem 1. The per-round time complexity of the OFDE algorithm is in $O\big(n^2m^2 + n^2 \log n + kn^2\big)$.

Proof. Based on the Jeffreys rule, the parameters $\boldsymbol{\theta}^{t+1}$ are computed in $O(d)$ time, where $d = mn + m^2\binom{n}{2}$. Furthermore, $\boldsymbol{f}^{t+\frac{1}{2}}$ is obtained in $O(n^2 \log n)$ time by applying the greedy matroid algorithm, and by exploiting the fact that the objective $\boldsymbol{w}^t$ can be maintained in $O(n^2)$ time per trial using $\boldsymbol{w}^0 = \mathbf{0}$ and $\boldsymbol{w}^t = \boldsymbol{w}^{t-1} + \phi^t(\boldsymbol{x}^t)$. Since $\boldsymbol{p}^{t+\frac{1}{2}}$ includes at most $k+1$ components, the updated mixture $\boldsymbol{p}^{t+1}$ and the updated forest $\boldsymbol{f}^{t+1}$ are obtained using SWAP in $O(n^2)$ time and $O(kn^2)$ time, respectively. □

Note that the time complexity of the Chow-Liu algorithm for a training set of size T is in $O(Tm^2n^2 + n^2 \log n)$. So, if k is constant or logarithmic in n, then the overall complexity of OFDE at horizon T is comparable to that of the Chow-Liu algorithm.

```
Algorithm 1: Online Forest Density Estimation (OFDE)

Input:
a class of experts $\mathcal{M} \in \{\mathcal{F}_{m,n}, \mathcal{T}_{m,n}\}$, and mixture size $k$
Initialization step
1  $\theta_i^1(u) \leftarrow \frac{1}{m}$ for all $i \in [n], u \in [m]$
2  $\theta_{ij}^1(u,v) \leftarrow \frac{1}{m^2}$ for all $(i,j) \in \binom{[n]}{2}, u, v \in [m]$
3  $\boldsymbol{p}^1 \leftarrow \mathbf{0}$
Trials
foreach $t \leftarrow 1, \dots$ do
4      Play $M^t \leftarrow (\boldsymbol{f}^t, \boldsymbol{\theta}^t)$ where $\boldsymbol{f}^t = \textsc{Swap}_1(\boldsymbol{p}^t)$
5      Receive $\boldsymbol{x}^t$
       Parameter update
6      $\theta_i^{t+1}(u) \leftarrow \dfrac{t_u + \frac{1}{2}}{t + \frac{m}{2}}$ for all $i \in [n], u \in [m]$
7      $\theta_{ij}^{t+1}(u,v) \leftarrow \dfrac{t_{uv} + \frac{1}{2}}{t + \frac{m^2}{2}}$ for all $(i,j) \in \binom{[n]}{2}, u, v \in [m]$
       Structure update
8      Choose $\alpha_t$ and $\beta_t$ in $(0,1)$
9      Draw $\boldsymbol{r}^t$ in $\left[0, \frac{1}{\beta_t}\right]^{\binom{n}{2}}$ uniformly at random
10     $\boldsymbol{f}^{t+\frac{1}{2}} \leftarrow \mathrm{argmin}_{\boldsymbol{p} \in \boldsymbol{P}}(\boldsymbol{w}^t)$ where
           $\boldsymbol{w}^t \leftarrow \boldsymbol{r}^t + \sum_{\tau=1}^{t} \phi^\tau(\boldsymbol{x}^\tau)$
11     $\boldsymbol{p}^{t+\frac{1}{2}} \leftarrow \alpha_t \boldsymbol{p}^t + (1-\alpha_t)\boldsymbol{f}^{t+\frac{1}{2}}$
12     $\boldsymbol{p}^{t+1} \leftarrow \textsc{Swap}_k(\boldsymbol{p}^{t+\frac{1}{2}})$
```

5 REGRET ANALYSIS

Based on the decomposable form of the log-loss (5), the regret of the OFDE algorithm can be expressed as a telescopic sum of two separate components, namely, a "parametric" regret with fixed structure and varying parameters, and a "structural" regret, with fixed parameters and varying structure. Formally, let $(\boldsymbol{p}, \boldsymbol{\theta})^{1:T} = ((\boldsymbol{p}^1, \boldsymbol{\theta}^1), \cdots, (\boldsymbol{p}^T, \boldsymbol{\theta}^T))$ be the sequence generated by the algorithm during T rounds. Then,

$$R_{\boldsymbol{x}^{1:T}}[(\boldsymbol{p}, \boldsymbol{\theta})^{1:T}] = R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T}) + R_{\boldsymbol{x}^{1:T}}(\boldsymbol{p}^{1:T})$$

where

$$R_{\boldsymbol{x}^{1:T}}(\boldsymbol{p}^{1:T}) = \sum_{t=1}^{T} \ell(\boldsymbol{p}^t, \boldsymbol{\theta}^t, \boldsymbol{x}^t) - \ell(\boldsymbol{p}^*, \boldsymbol{\theta}^t, \boldsymbol{x}^t), \quad (6)$$

$$R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T}) = \sum_{t=1}^{T} \ell(\boldsymbol{p}^*, \boldsymbol{\theta}^t, \boldsymbol{x}^t) - \ell(\boldsymbol{p}^*, \boldsymbol{\theta}^*, \boldsymbol{x}^t) \quad (7)$$

and where $(\boldsymbol{p}^*, \boldsymbol{\theta}^*)$ is any minimizer in $\boldsymbol{P}(\mathcal{M}) \times \boldsymbol{\Theta}_{m,n}$ of the cumulative log-loss $\sum_{t=1}^{T} \ell(\boldsymbol{p}, \boldsymbol{\theta}, \boldsymbol{x}^t)$. The rest of this section is devoted to the analysis of each separate part (7) and (6), and the unification of our results.

5.1 PARAMETRIC REGRET

For the analysis of the parametric regret $R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T})$, we consider the problem of online density estimation problem with the class of experts $(\{\boldsymbol{p}\}, \boldsymbol{\Theta}_{m,n})$, where $\boldsymbol{p}$ is an arbitrary point in $\boldsymbol{P}$. As mentioned above, $\boldsymbol{p}$ can be viewed as a convex combination $\boldsymbol{p} = \mathbb{E}[\boldsymbol{f}]$ of graphical structures $\boldsymbol{f} \in \mathcal{M}$. Using the additive decomposition (5) and the linearity of expectations, we have

$$\begin{aligned} R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T}) &= \mathbb{E}\left[\sum_{t=1}^{T} \ell(\boldsymbol{f}, \boldsymbol{\theta}^t, \boldsymbol{x}^t) - \ell(\boldsymbol{f}, \boldsymbol{\theta}^*, \boldsymbol{x}^t)\right] \\ &= \mathbb{E}\left[\ln \prod_{t=1}^{T} \frac{\mathbb{P}_{\boldsymbol{f}, \boldsymbol{\theta}^*}(\boldsymbol{x}^t)}{\mathbb{P}_{\boldsymbol{f}, \boldsymbol{\theta}^t}(\boldsymbol{x}^t)}\right] \end{aligned} \quad (8)$$

In light of the closed-form expression (1), the logarithmic term inside the expectation in (8) can be reformulated as

$$\begin{aligned} \ln \prod_{t=1}^{T} \frac{\mathbb{P}_{\boldsymbol{f}, \boldsymbol{\theta}^*}(\boldsymbol{x}^t)}{\mathbb{P}_{\boldsymbol{f}, \boldsymbol{\theta}_t}(\boldsymbol{x}^t)} &= \sum_{i=1}^{n} \ln \frac{\theta_i^*(x_i^{1:T})}{\theta_i^{1:T}(x_i^{1:T})} + \\ \sum_{(i,j) \in F} \ln \frac{\theta_{ij}^*(x_{ij}^{1:T})}{\theta_{ij}^{1:T}(x_{ij}^{1:T})} &+ \sum_{(i,j) \in F} \ln \frac{\theta_i^{1:T}(x_i^{1:T})}{\theta_i^*(x_i^{1:T})} \frac{\theta_j^{1:T}(x_j^{1:T})}{\theta_j^*(x_j^{1:T})} \end{aligned} \quad (9)$$

where

$$\theta_i^*(x_i^{1:t}) = \prod_{\tau=1}^{t} \theta_i^*(x_i^\tau), \qquad \theta_i^{1:t}(x_i^{1:t}) = \prod_{\tau=1}^{t} \theta_i^\tau(x_i^\tau)$$

$$\theta_{ij}^*(x_{ij}^{1:t}) = \prod_{\tau=1}^{t} \theta_{ij}^*(x_i^\tau, x_j^\tau), \quad \theta_{ij}^{1:t}(x_{ij}^{1:t}) = \prod_{\tau=1}^{t} \theta_{ij}^\tau(x_i^\tau, x_j^\tau)$$

We may observe that (9) is essentially a composition of local regrets defined over univariate density estimators $\theta_i^{1:T}(x_i^{1:T})$ and bivariate density estimators $\theta_{ij}^{1:T}(x_{ij}^{1:T})$. Notably, for each edge $(i,j) \in F$, the regret of the bivariate estimator $\theta_{ij}^{1:T}(x_{ij}^{1:T})$ is compensated by the relative gains of the univariate estimators $\theta_i^{1:T}(x_i^{1:T})$ and $\theta_j^{1:T}(x_j^{1:T})$. Such a decomposition motivates the use of well-known Bayesian mixtures with Dirichlet priors for specifying the estimators. We focus here on *symmetric* Dirichlet priors, given by

$$p_\mu(\boldsymbol{\lambda}) = \frac{\Gamma(m\mu)}{\Gamma(\mu)^m} \prod_{v=1}^{m} (\lambda(v))^{\mu - 1}$$

where $\boldsymbol{\lambda}$ is a vector in the m-dimensional probability simplex, $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ is the gamma function, and $\mu \in [0,1]$ is a hyperparameter. The corresponding Bayesian mixture $\lambda(x^{1:t})$ for the sequence $x^{1:t} = (x^1, \cdots, x^t)$ is given by

$$\int \prod_{\tau=1}^{t} \mathbb{P}_{\boldsymbol{\lambda}}(x^\tau) p_\mu(\boldsymbol{\lambda}) d\boldsymbol{\lambda} = \frac{\Gamma(m\mu)}{\Gamma(\mu)^m} \frac{\prod_{v=1}^{m} \Gamma(t_v + \mu)}{\Gamma(t + m\mu)} \quad (10)$$

where t_v is the number of v's in $x^{1:t}$. Thus, by applying (10) with $\mu = 1/2$ to the estimators $\theta_i^t(x_i^{1:t})$ and $\theta_{ij}^{1:t}(x_{ij}^{1:t})$, we derive the update rules specified by Lines 6-7 of the OFDE algorithm.

Before deriving a bound for the parametric regret (7), we present two useful double inequalities for log-gamma functions, summarized in the next lemma.

Lemma 1. Let m be a positive integer. Then for any $t > 0$,

$$-\ln\sqrt{2} \le \ln\Gamma\left(t+\frac{1}{2}\right) - t\ln t + t - \ln\sqrt{2\pi} \le 0 \quad (11)$$

$$0 \le \ln\Gamma\Big(t+\frac{m}{2}\Big) - \ln\Gamma\Big(t+\frac{1}{2}\Big) - \frac{m-1}{2}\ln t \le o(1) \quad (12)$$

Proof. (11) is a reformulation of Lemma 1 in (Watanabe and Roos, 2015), and the right-hand inequality of (12) follows from the classical asymptotic relation (see e.g. Qi and Luo (2013)):

$$\lim_{t\to\infty}\left[t^{b-a}\frac{\Gamma(t+a)}{\Gamma(t+b)}\right] = 1$$

using $a = m/2$ and $b = 1/2$. For the left-hand inequality of (12), we can observe that

$$\ln\frac{\Gamma\left(t+\frac{m}{2}\right)}{\Gamma\left(t+\frac{1}{2}\right)} = \ln\frac{\Gamma\left(z+\frac{m-1}{2}\right)}{\Gamma\left(z+k\right)} + \ln\frac{\Gamma\left(z+k\right)}{\Gamma\left(z\right)} \quad (13)$$

where $z = t + 1/2$ and $k = \lfloor\frac{m-1}{2}\rfloor$. Based on the identity $\ln\Gamma(z+k) = \ln\Gamma(z) + \sum_{i=0}^{k-1}\ln(z+i)$, the second term in the right-hand side of (13) is lower bounded by $k\ln z$. So, if m is odd, then $k = \frac{m-1}{2}$, and hence, (13) is lower bounded by $\frac{m-1}{2}\ln t$, as desired. Now, if m is even, then using $z' = t + k$, we can observe that the first term in the right-hand side of (13) can be rewritten as the log-ratio of $\Gamma(z'+1)$ to $\Gamma(z'+\frac{1}{2})$. Thus, by Wendel's inequality (1948), we have

$$\frac{1}{2}\ln z' \le \ln\frac{\Gamma(z'+1)}{\Gamma(z'+\frac{1}{2})} \le \frac{1}{2}\ln\left(z'+\frac{1}{2}\right)$$

By combining the lower bounds $k\ln z$ and $\frac{1}{2}\ln z'$, it follows that (13) is lower bounded by $(k+\frac{1}{2})\ln t = \frac{m-1}{2}\ln t$, which again yields the desired result. □

With these inequalities in hand, we can derive "sandwiching" bounds for the regret of the Jeffreys mixture. Specifically, using the Bayes mixture (10) with $\mu = 1/2$, the regret expression $\ln\theta^*(x^{1:T}) - \ln\theta^{1:T}(x^{1:T})$ is equal to

$$\ln\left[\prod_{u=1}^{m}\left(\frac{t_u}{T}\right)^{t_u}\right] + \ln\frac{\Gamma\left(T+\frac{m}{2}\right)}{\prod_{u=1}^{m}\Gamma\left(t_u+\frac{1}{2}\right)} + C_m \quad (14)$$

where $C_m = m\ln\Gamma(\frac{1}{2}) - \ln\Gamma(\frac{m}{2})$. By coupling the double inequalities (11) and (12), we can deduce that

$$-\ln\sqrt{2} \le \ln\Gamma\left(T+\frac{m}{2}\right) + T - T\ln T - \frac{m-1}{2}\ln T - \ln\sqrt{2\pi} \le o(1)$$

Similarly, using the double inequality (11) and summing over m values, we can infer that

$$-m\ln\sqrt{2} \le \ln\prod_{u=1}^{m}\Gamma\left(t_u+\frac{1}{2}\right) - \sum_{u=1}^{m}t_u\ln t_u + T - m\ln\sqrt{2\pi} \le 0$$

Now, using the fact that the first term in (14) is equal to $\sum_{u=1}^{m}t_u\ln t_u - T\ln T$, we can combine the above two double inequalities to derive the sandwiching bounds:

$$-\ln\sqrt{2} \le \ln\frac{\theta^*(x^{1:T})}{\theta^{1:T}(x^{1:T})} - \frac{m-1}{2}\ln\frac{T}{2\pi} - C_m \le m\ln\sqrt{2} + o(1) \quad (15)$$

Unsurprisingly, the right-hand inequality of (15) coincides with the regret bound of the Jeffreys mixture established by Xie and Barron (2000). As shown below, the left-hand inequality of (15) will also prove useful for bounding the regret expression (9).

Lemma 2. The parametric regret $R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T})$ of the OFDE algorithm is upper bounded by

$$\frac{n(m-1)+(n-1)(m-1)^2}{2}\ln\frac{T}{2\pi} + C_{m,n} + o(m^2 n)$$

where $C_{m,n} = nC_m + (n-1)(C_{m^2} - 2C_m)$.

Proof. As specified by Equality (8), any upper bound on the minimax regret of (9) is an upper bound on $R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T})$. Based on this observation, consider the first term of (9), given by $\sum_{i=1}^{n}\ln[\theta_i^*(x_i^{1:T})/\theta_i^{1:T}(x_i^{1:T})]$. Using the right-hand inequality of (15) and summing over n nodes, this term is upper bounded by

$$\frac{n(m-1)}{2}\ln\frac{T}{2\pi} + nC_m + mn\ln\sqrt{2} + o(n)$$

Clearly, a similar strategy can be applied to the second term $\sum_{(i,j)\in F}\ln[\theta_{ij}^*(x_{ij}^{1:T})/\theta_{ij}^{1:T}(x_{ij}^{1:T})]$ of (9). Since $|F| \le n-1$, this term is upper bounded by

$$\frac{(n-1)(m^2-1)}{2}\ln\frac{T}{2\pi} + (n-1)C_{m^2} + (n-1)m^2\ln\sqrt{2} + o(n)$$

Finally, the third term of (9) can be reformulated as a sum over each $(i,j)\in F$ of two components: $\ln[\theta_i^{1:T}(x_i^{1:T})/\theta_i^*(x_i^{1:T})]$ and $\ln[\theta_j^{1:T}(x_j^{1:T})/\theta_j^*(x_j^{1:T})]$. By applying the left-hand side inequality of (15) to each component, and summing over at most $n-1$ edges, this term is upper bounded by

$$-(n-1)(m-1)\ln\frac{T}{2\pi} - 2(n-1)C_m + 2(n-1)\ln\sqrt{2}$$

By combining the above three bounds, rearranging terms, and taking into account the fact that $[(n-1)(m^2+2)+mn]\ln\sqrt{2}$ is in $o(m^2 n)$ (for $m \ge 2$), we get the desired result. □

In the binomial case ($m = 2$), it is easy to check that $C_{2,n} = n \ln \pi$. By reporting this result into Lemma 2, we may derive for binary Markov forests a parametric regret bound of the form:

$$R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T}) \leq \left(n - \frac{1}{2}\right) \ln T + o(n) \tag{16}$$

5.2 STRUCTURAL REGRET

Before deriving a bound for the structural part of the regret, we first examine some analytic properties of the loss function (5), specified as an affine function of the predicted structure.

Lemma 3. Given a class of models $\mathcal{M} \in \{\mathcal{F}_{m,n}, \mathcal{T}_{m,n}\}$, let $\boldsymbol{\theta}^{1:T}$ be the sequence of parameters in $\boldsymbol{\Theta}_{m,n}$ generated by the OFDE algorithm on the sequence of outcomes $\boldsymbol{x}^{1:T}$. Then, for any $t \in [T]$, any $\boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{P}(\mathcal{M})$, and any $\boldsymbol{x} \in \mathcal{X}$,

$$\left\|\boldsymbol{\phi}^t(\boldsymbol{x})\right\|_\infty \leq \ln\left(\frac{T}{2} + \frac{m^2}{4}\right)$$
$$\|\boldsymbol{p} - \boldsymbol{q}\|_1 \leq 2(n-1)$$

Proof. For the first property, consider two values $u, v \in [m]$. We may observe that $\phi^1_{ij}(u, v) = 0 < \ln(T/2 + m^2/4)$, for $m \geq 2$. Furthermore, using the Jeffreys update rule, we have

$$\begin{aligned}\phi^{t+1}_{ij}(u, v) &= \ln \frac{(t_u + 1/2)(t_v + 1/2)}{(t + m/2)^2} + \ln \frac{t + m^2/2}{t_{uv} + 1/2} \\ &\leq \ln \frac{1}{4} + \ln \frac{t + m^2/2}{t_{uv} + 1/2} \leq \ln\left(\frac{T}{2} + \frac{m^2}{4}\right)\end{aligned}$$

where the first inequality follows from the fact that the maximizer of $(t_u + 1/2)(t_v + 1/2)$ subject to the constraint $t_u + t_v \leq t$ is given by $t_u = t_v = \frac{t}{2}$. Concerning the second property, recall that the dimension of $\boldsymbol{P}(\mathcal{M})$ is $n-1$, which implies that $\|\boldsymbol{p}\|_1 \leq n-1$ for all $\boldsymbol{p} \in \boldsymbol{P}(\mathcal{M})$. This, together with the fact that $\|\boldsymbol{p} - \boldsymbol{q}\|_1 \leq \|\boldsymbol{p}\|_1 + \|\boldsymbol{q}\|_1$, implies the result. □

Based on these properties, we derive a regret bound for the structural part of the OFDE algorithm by analyzing the update rules in Lines 10-12. Specifically, the expression (6) is decomposed into the telescopic sum:

$$R_{\boldsymbol{x}^{1:T}}(\boldsymbol{p}^{1:T}) \leq \sum_{t=1}^{T} \langle \boldsymbol{p}^t, \boldsymbol{\ell}^t \rangle - \langle \boldsymbol{p}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle \tag{17}$$
$$+ \sum_{t=1}^{T} \langle \boldsymbol{p}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle - \langle \boldsymbol{f}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle \tag{18}$$
$$+ \sum_{t=1}^{T} \langle \boldsymbol{f}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle - \langle \boldsymbol{p}^*, \boldsymbol{\ell}^t \rangle \tag{19}$$

where $\boldsymbol{\ell}^t$ is used here as a shorthand of $\boldsymbol{\phi}^t(\boldsymbol{x}^t)$. Recall that each point $\boldsymbol{p}^t$ in the sequence $\boldsymbol{p}^{1:T}$ includes at most k forests. So, the difference (17) captures the regret of the OFDE algorithm with respect to its *unbounded* version, where the swap rounding step at Line 12 is omitted.

Lemma 4. The OFDE algorithm has no regret with respect to its unbounded version: $\sum_{t=1}^{T} \langle \boldsymbol{p}^t, \boldsymbol{\ell}^t \rangle - \langle \boldsymbol{p}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle = 0$.

Proof. Let $\tilde{\boldsymbol{p}}^{1:T}$ be the sequence of points given by the unbounded version of OFDE, that is, $\tilde{\boldsymbol{p}}^1 = \boldsymbol{p}^1$ and $\tilde{\boldsymbol{p}}^{t+1} = \alpha_t \tilde{\boldsymbol{p}}^t + (1 - \alpha_t)\boldsymbol{f}^{t+\frac{1}{2}}$ for each $t \in [T]$. We prove that $\mathbb{E}[\boldsymbol{p}^t] = \boldsymbol{p}^{t+\frac{1}{2}}$ by induction on $t \in [T]$. The case $t = 1$ follows from $\boldsymbol{p}^1 = \tilde{\boldsymbol{p}}^1$. Suppose by induction hypothesis that $\mathbb{E}[\boldsymbol{p}^t] = \tilde{\boldsymbol{p}}^t$. Using the SWAP algorithm, $\mathbb{E}[\boldsymbol{p}^{t+1}] = \alpha_t \boldsymbol{p}^t + (1 - \alpha_t)\boldsymbol{f}^{t+\frac{1}{2}}$. Furthermore, by induction hypothesis, we also know that $\mathbb{E}[\alpha_t \boldsymbol{p}^t + (1 - \alpha_t)\boldsymbol{f}^{t+\frac{1}{2}}] = \alpha_t \mathbb{E}[\boldsymbol{p}^t] + (1 - \alpha_t)\boldsymbol{f}^{t+\frac{1}{2}} = \tilde{\boldsymbol{p}}^{t+1}$. Since $\mathbb{E}[\mathbb{E}[\boldsymbol{p}^{t+1}]] = \mathbb{E}[\boldsymbol{p}^{t+1}]$, it follows that $\mathbb{E}[\boldsymbol{p}^{t+1}] = \tilde{\boldsymbol{p}}^{t+1}$, as desired. Based on this invariant, the result follows from the linearity of expectations: $\sum_{t=1}^{T} \langle \mathbb{E}[\boldsymbol{p}^t], \boldsymbol{\ell}^t \rangle - \langle \boldsymbol{p}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle = \sum_{t=1}^{T} \mathbb{E}\langle \boldsymbol{p}^{t+\frac{1}{2}} - \boldsymbol{p}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle = 0$. □

With this result in hand, the structural regret of the OFDE algorithm is reduced to the sum of (18) and (19). Using appropriate choices for the hyperparameters α_t and β_t we can derive sublinear regret bounds in both the horizon-dependent setting (where T is known) and the horizon-independent setting.

Lemma 5. Let $\gamma = \ln(T/2 + m^2/4)$. The structural regret $R_{\boldsymbol{x}^{1:T}}(\boldsymbol{p}^{1:T})$ of the OFDE algorithm is bounded by

- $n^2 \gamma \sqrt{2T}$ in the horizon-dependent case, using $0 < \alpha_t \leq \frac{1}{2\sqrt{2t}}$ and $\beta_t = \frac{1}{\gamma n}\sqrt{2/t}$;
- $n^2(\gamma + 1)^2 \sqrt{2T}$ in the horizon-independent case, using $0 < \alpha_t \leq \frac{1}{4\sqrt{2t}}$ and $\beta_t = \frac{1}{n}\sqrt{2/t}$.

Proof. Consider the regret expression (18), and let $\alpha_t = \alpha'/\sqrt{t}$. Using the specification of $\boldsymbol{p}^{t+\frac{1}{2}}$ given at Line 11, we get

$$\begin{aligned}\sum_{t=1}^{T} \langle \boldsymbol{p}^{t+\frac{1}{2}} - \boldsymbol{f}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle &= \sum_{t=1}^{T} \alpha_t \langle \boldsymbol{p}^t - \boldsymbol{f}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle \\ &\leq 2(n-1)\gamma \sum_{t=1}^{T} \alpha_t \leq 4(n-1)\gamma\alpha'\sqrt{T}\end{aligned}$$

where the first inequality follows from Hölder's inequality $\langle \boldsymbol{p}^t - \boldsymbol{f}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle \leq \|\boldsymbol{p}^t - \boldsymbol{f}^{t+\frac{1}{2}}\|_1 \|\boldsymbol{\ell}^t\|_\infty$, and the application of Lemma 3. The last inequality follows from $\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}$. Now, observe that (19) is the regret of the FPL strategy. Let $\beta_t = \beta'/\sqrt{t}$. By applying Theorem 3.3 in (Kalai and Vempala, 2005), we can derive that

$$\begin{aligned}\sum_{t=1}^{T} \langle \boldsymbol{f}^{t+\frac{1}{2}} - \boldsymbol{p}^*, \boldsymbol{\ell}^t \rangle &\leq 2\beta' R A \sqrt{T} + \tfrac{D}{\beta'}\sqrt{T} \\ &\leq (n-1)n^2\gamma^2\beta'\sqrt{T} + \tfrac{2(n-1)}{\beta'}\sqrt{T}\end{aligned}$$

using the facts that $R = \max_{t \in [T]} \langle \boldsymbol{f}^{t+\frac{1}{2}}, \boldsymbol{\ell}^t \rangle \leq 2(n-1)\gamma$ from Hölder inequality, $A = \max_{t \in [T]} \|\boldsymbol{\ell}^t\|_1 \leq \gamma n^2/2$, and

$D = \max_{t\in[T]} \|\boldsymbol{f}^{t+\frac{1}{2}} - \boldsymbol{f}^{t-\frac{1}{2}}\|_1 \leq 2(n-1)$. Combining the derived bounds for (18) and (19), and rearranging, yields

$$R_{\boldsymbol{x}^{1:T}}(\boldsymbol{p}^{1:T}) \leq (n-1)\sqrt{T}\left(4\gamma\alpha' + n^2\gamma^2\beta' + \frac{2}{\beta'}\right)$$

In the horizon-dependent case, we can take $\beta' = \sqrt{2}/\gamma n$ to derive that $R_{\boldsymbol{x}^{1:T}}(\boldsymbol{p}^{1:T}) \leq \gamma(n-1)(4\alpha' + n\sqrt{2})\sqrt{T}$, which is bounded by $n^2\gamma\sqrt{2T}$ for $\alpha' \leq 1/2\sqrt{2}$. In the horizon-independent case, γ is unknown. So, using $\beta' = \sqrt{2}/n$, we get that $R_{\boldsymbol{x}^{1:T}}(\boldsymbol{p}^{1:T}) \leq (n-1)[4\gamma\alpha' + (n/\sqrt{2})(\gamma^2+1)]\sqrt{T}$, which is bounded by $n^2(\gamma+1)^2\sqrt{2T}$ for $\alpha' \leq 1/4\sqrt{2}$. □

5.3 MAIN RESULTS

We have now all ingredients in hand to prove the Hannan-consistency of our online learning algorithm. The next theorem is obtained by coupling Lemma 2 with Lemma 5. The corollary is simply derived by replacing the bound in Lemma 2 with (16).

Theorem 2. For the classes of Markov forests $\mathcal{F}_{m,n}$ and Markov trees $\mathcal{T}_{m,n}$, there exists an online density estimation algorithm achieving a minimax regret in $O(m^2 n \ln T + n^2 \ln T\sqrt{T})$ in the horizon-dependent case, and $O(m^2 n \ln T + n^2 (\ln T)^2 \sqrt{T})$ in the horizon-independent case.

Corollary 1. For the classes of binary Markov forests $\mathcal{F}_{2,n}$ and Markov trees $\mathcal{T}_{2,n}$, there exists an online density estimation algorithm that attains (in the horizon-dependent case) a minimax regret of $n^2 \ln(\frac{T}{2}+1)\sqrt{2T} + (n-\frac{1}{2})\ln T + o(n)$.

To conclude the theoretical part of this study, recall that the competitor used in both parts (6) and (7) of the regret analysis is an expert in $(\boldsymbol{P}(\mathcal{M}), \boldsymbol{\Theta}_{m,n})$. As mentioned above, such experts are tree-structured mixtures sharing the same parameters, which predict according to the probability distribution (3). As stated in Lemma 2, the parametric regret $R_{\boldsymbol{x}^{1:T}}(\boldsymbol{\theta}^{1:T})$ of the OFDE algorithm with respect to these experts is in $O(m^2 n \ln T)$. Since the regret bounds in Lemma 5 also hold for these competitors, it follows that OFDE is Hannan-consistent with respect to tree-structured mixtures.

6 EXPERIMENTS

In order to empirically evaluate our algorithm, we performed simulations on 4 publicly available datasets[1], listed in Table 1. Though all these datasets are binary-valued, they differ in the number of variables and the number of instances.

Our experimental objective was to compare the OFDE algorithm with respect to batch learning algorithms which have the benefit of hindsight for the train set. To this end, we used the Chow-Liu (CL) algorithm (Chow and Liu, 1968) that learns a Markov tree, and the Chow-Liu with Thresholding (CLT) algorithm (Tan et al., 2011), that learns a Markov forest by pruning the Chow-Liu tree. The CL and OFDE algorithms were trained without using the tune set. As CLT relies on a user-supplied threshold parameter $\epsilon \in (0,1)$, we performed experiments using several values $\{1/4, 1/2, 3/4\}$ for this parameter and kept in our results the best choice of ϵ measured on the tune set. In our implementation of CLT, we used a slight refinement of the original pruning rule: any edge (i,j) for which the empirical mutual information is lower than $n^{-\epsilon}$ is removed from the tree. Our OFDE algorithm was trained with both reference classes $\mathcal{F}_{2,n}$ and $\mathcal{T}_{2,n}$. Here, we denote by OFDE$_F$ (resp. OFDE$_T$) the instantiation of OFDE with Markov forests (resp. Markov trees). Both instances of OFDE were trained under the horizon-independent setting, using $k = \ln n$, $\alpha_t = \frac{1}{4\sqrt{2t}}$, and $\beta_t = \frac{1}{n}\sqrt{2/t}$.

Dataset	Train set	Tune set	Test set	Vars (n)
Abalone	3,134	417	626	31
Covertype	30,000	4,000	6,000	84
KDDCup 2000	180,092	19,907	34,955	64
MSNBC	291,326	38,843	58,265	17

Table 1: Dataset Characteristics.

The batch algorithms CL and CLT were trained on the whole train set, and their generalization performance was measured using the average log-loss evaluated on the test set. For the online learners OFDE$_F$ and OFDE$_T$, the instances were revealed only one at a time and, at the end of each iteration, the performance was measured by evaluating the average log-loss on the test set. The sequence of observations was generated by simply listing the instances of the train set.

The results, averaged over 10 experiments per dataset, are reported in Figure 2. Unsurprisingly, the performance of OFDE$_T$ is generally better than OFDE$_F$, since the batch tree learner CL outperforms its forest variant CLT. Yet, it is apparent that OFDE$_T$ and OFDE$_F$ respectively converge to the estimations of CL and CLT. The convergence rates are particularly remarkable for the datasets Covertype, KDDCup 2000, and MSNBC, where a logarithmic scale is used for the number of iterations.

Concerning runtimes, the three algorithms were implemented in C++ and tested on a Quad-core Intel XEON X5550. For all datasets, the per-round runtime of OFDE (using forests or trees) is less than 3 ms. This indicates that OFDE can be used as practical alternative to CL(T) for handling streaming applications.

7 DISCUSSION

As a fundamental result in universal prediction, it is known that the optimal solution achieving minimax regret for any class $\mathcal{M}$ of discrete probabilistic models is obtained by the *normalized maximum likelihood* strategy (Shtarkov, 1987). Unfortunately, for this optimal strategy, the time horizon T must be known in advance, and the computation of the log-loss at each round $t \in [T]$ requires the evaluation of exponentially many marginalization terms. Thus, one of the key challenges in online density estimation is to devise horizon-independent strategies that pro-

[1] `alchemy.cs.washington.edu/papers/davis10a/`

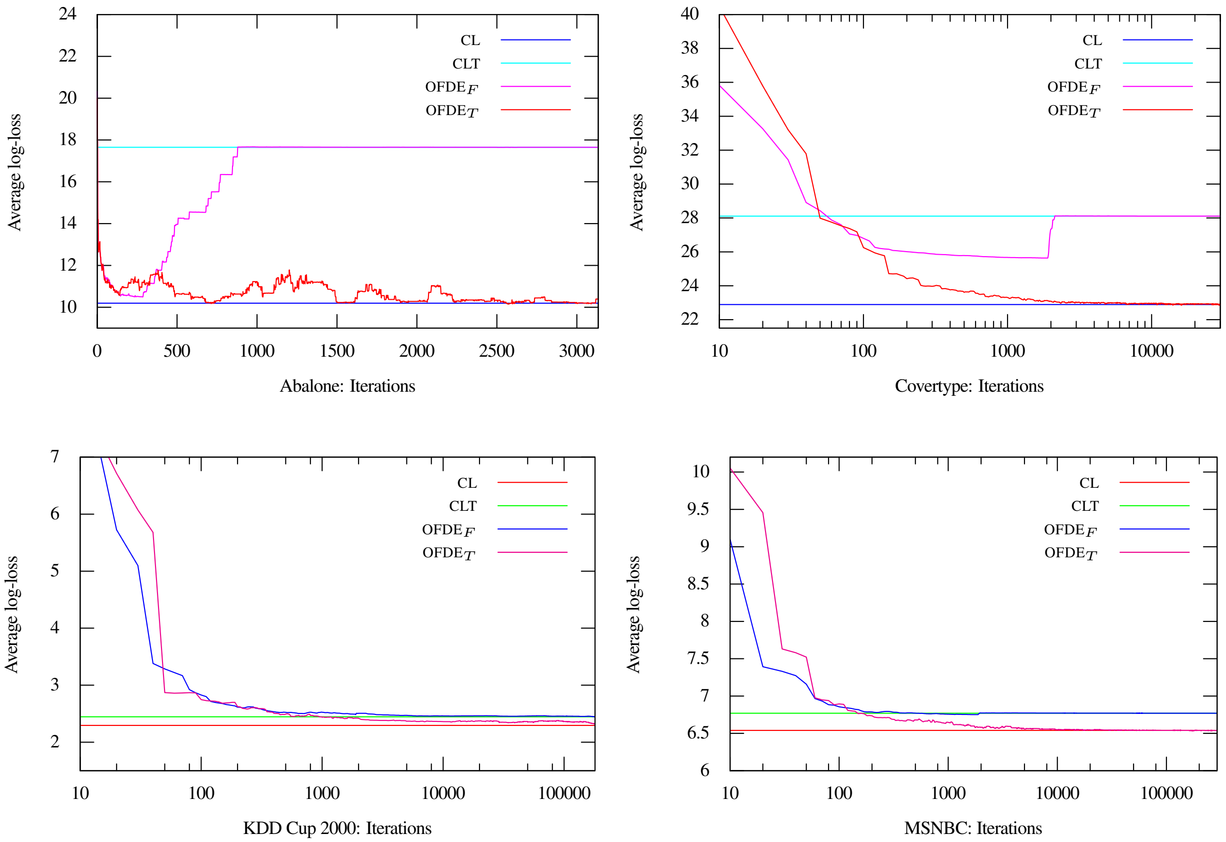

Figure 2: Comparison of OFDE$_T$ and OFDE$_F$ with CL and CLT on the four datasets.

vide a good compromise between minimax optimality and computational complexity. Several easily implementable nearly optimal strategies have been proposed for unidimensional probabilistic models, including binomial and multinomial families (Freund, 1996; Xie and Barron, 2000; Watanabe and Roos, 2015), and, more generally, univariate exponential families (Takimoto and Warmuth, 2000; Azoury and Warmuth, 2001; Kotłowski and Grünwald, 2011). Much less is known, however, about multidimensional families, especially the classes of graphical models characterized by multiple interdependencies between variables. A notable exception is the work by Bauer et al. (1997) for sequentially predicting the parameters of a Bayesian network. Yet, the target network structure is known in advance. To our knowledge, the present paper is one of the first studies that investigates both structural and parametric aspects of graphical models in online density estimation.

By considering classes of experts defined over varying structures, our study has intimate connections with *online combinatorial optimization*, a topic of online learning where the reference classes are combinatorial spaces. Several Hannan-consistent algorithms have been proposed in this setting, including the *Follow the Perturbed Leader* (FPL) strategy (Hannan, 1957; Kalai and Vempala, 2005), and the *Online Mirror Descent* (OMD) strategy (Koolen et al., 2010; Audibert et al., 2011; Rajkumar and Agarwal, 2014). Though OMD is known to achieve better regret bounds than FPL, it relies on a projection step performed at each iteration, in order to maintain the current estimate in the convex hull of the combinatorial space. The computational complexity of this projection step is typically much worse than the cost of linear optimization, especially when the combinatorial space is a matroid. The FPL strategy, advocated in this study, provides a reasonable compromise between optimality and computational complexity. Yet, alternative strategies can be devised in our setting such as, for example, online Franck-Wolfe optimization methods (Hazan and Kale, 2012).

A natural perspective of research that emerges from our study is to devise *lower bounds* for the minimax regret of forest density estimators. In a related setting, Kveton et al. (2014) have recently shown that such lower bounds are essentially logarithmic in T for the reference class of partition matroids. We conjecture that similar bounds holds for graphical matroids, and more generally for the classes $\mathcal{F}_{m,n}$ and $\mathcal{T}_{m,n}$. Finally, our work is also related to mixtures of trees (Meila and Jordan, 2000; Kumar and Koller, 2009). To this point, we have shown that OFDE is Hannan-consistent with mixtures of forests (or trees) sharing the *same* parameters. An interesting open question is to determine whether *arbitrary* mixtures of trees are learnable in the online density estimation setting.

References

J-Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. In *Proc. of COLT*, pages 107–132, 2011.

K. Azoury and M. Warmuth. Relative loss bounds for online-learning density estimation with the exponential family of distributions. *Machine Learning*, 43:211–246, 2001.

E. Bauer, D. Koller, and Y. Singer. Update rules for parameter estimation in Bayesian networks. In *Proc. of UAI*, pages 3–13, 1997.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proc. of FOCS*, pages 575–584, 2010.

D. M. Chickering. Learning Bayesian networks is NP-complete. In *Proc. of AISTATS*, pages 121–130, 1995.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

S. Dasgupta. Learning polytrees. In *Proc. of UAI*, pages 134–141, 1999.

J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Proc. Calgary International Conference on Combinatorial Structures and their Applications*, pages 69–87, 1970.

Y. Freund. Predicting a binary sequence almost as well as the optimal biased coin. In *Proc. of COLT*, pages 89–98, 1996.

P. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

J. Hannan. Approximation to Bayes risk in repeated plays. In *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.

E. Hazan and S. Kale. Projection-free online learning. In *Proc. of ICML*, 2012.

H. Jeffreys. An invariant form for the prior probability in estimation problems. In *Proc. of Royal Society London*, number 186 in A, pages 453–461, 1946.

A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

W. Koolen, M. Warmuth, and J. Kivinen. Hedging structured concepts. In *Proc. of COLT*, pages 93–105, 2010.

W. Kotłowski and P. Grünwald. Maximum likelihood vs. sequential normalized maximum likelihood in on-line density estimation. In *Proc. of COLT*, pages 457–476, 2011.

M. P. Kumar and D. Koller. Learning a small mixture of trees. In *Proc. of NIPS*, pages 1051–1059, 2009.

B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *Proc. of UAI*, pages 420–429, 2014.

S. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

H. Liu, M. Xu, H. Gu, A. Gupta, J. Lafferty, and L. Wasserman. Forest density estimation. *Journal of Machine Learning Research*, 12:907–951, 2011.

M. Meila and M. I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.

N. Merhav and M. Feder. Universal prediction. *IEEE Transactions on Information Theory*, 48:1947–1958, 1998.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

F. Qi and Q-M. Luo. Bounds for the ratio of two Gamma functions: from Wendel's asymptotic relation to Elezović-Giordano-Pečarić's theorem. *Journal of Inequalities and Applications*, 2013(1):1–20, 2013.

A. Rajkumar and S. Agarwal. Online decision-making in general combinatorial spaces. In *Proc. of NIPS*, pages 3482–3490, 2014.

J. Rissanen. *Optimal Estimation of Parameters*. Cambridge, 2012.

A. Shrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume B. Springer, 2003.

Y. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3):175–186, 1987.

N. Srebro. Maximum likelihood bounded tree-width Markov networks. *Artificial Intelligence*, 143(1):123–138, 2003.

E. Takimoto and M. Warmuth. The last-step minimax algorithm. In *Proc. of ALT*, pages 279–290, 2000.

V. Tan, A. Anandkumar, and A. Willsky. Learning high-dimensional Markov forest distributions: Analysis of error rates. *Journal of Machine Learning Research*, 12: 1617–1653, 2011.

M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

K. Watanabe and T. Roos. Achievability of asymptotic minimax regret by horizon-dependent and horizon-independent strategies. *Journal of Machine Learning Research*, 16:1–48, 2015.

J. Wendel. Note on the Gamma function. *The American Mathematical Monthly*, 55(9):563–564, 1948.

Q. Xie and A. Barron. Asymptotic minimax regret for data compression, gambling, and prediction. *IEEE Transactions on Information Theory*, 46(2):431–445, 2000.