Test-time active feature selection through tractable acquisition functions

Athresh Karanam[1] **Sriraam Natarajan**[1]

[1]University of Texas at Dallas

Abstract

Real-life test-time decision making frequently necessitates the acquisition of additional features, which can often be costly. For instance, diagnosing a patient may require conducting laboratory tests or performing imaging scans. The acquisition of these potentially expensive features on a per-instance basis enables the efficient and effective allocation of resources. However, existing acquisition functions often pose computational challenges, necessitating the use of approximations. To address this issue, we introduce MEASURES, a framework for test-time active feature selection. Our approach harnesses the tractability and expressive efficiency of Probabilistic Circuits (PCs), a class of deep tractable probabilistic models, to compute otherwise intractable acquisition functions. Our experiments demonstrate the superior effectiveness of tractable acquisition functions compared to existing approaches.

1 INTRODUCTION

Consider the task of diagnosing a patient with diseases. Healthcare personnel typically rely on easily acquirable attributes of the patient, such as age, body mass index (BMI), and current symptoms, to initiate the diagnostic process. Subsequently, they might conduct specific medical examinations or ask targeted questions about the patient's condition to assist in the diagnosis. These medical examinations and questions are tailored to each patient, enabling efficient diagnoses without the need for wasteful examinations or irrelevant questions. Moreover, obtaining all possible features, including examinations and questions, would be both prohibitively expensive and potentially unnecessary. Taking inspiration from this scenario, the test-time active feature selection (AFS) [9] problem aims to automate this tailored feature selection process while adhering to a predefined budget.

In this work, we propose a novel test-time AFS framework, MEASURES, based on the expected variance ($\mathbb{EV}$) of a classifier w.r.t to a joint distribution over the complete set of observable features. While the computation of this metric is intractable for arbitrary pairs of classifiers and joint models, we leverage the tractability and expressiveness of Probabilistic Circuits (PCs) - a class of deep tractable probabilistic models - to render this computation tractable. Our contributions can be summarized as follows: (1) We propose $\mathbb{EV}$ as a novel acquisition function for test-time AFS. (2) We propose an algorithm for exactly computing variance $\mathbb{VAR}$ of an omni-compability probabilistic classifier w.r.t a smooth and decomposable PC (see sections 3 and 4). (3) We evaluate our proposed approach on the MNIST dataset and show its superior effectiveness over four simple baselines.

2 RELATED WORK

Active Feature Selection (AFS): Active feature selection is an important problem that has been studied extensively along with related problems such as Active Feature Acquisition (AFA) [13] and Active Feature Elicitation (AFE) [10]. In the AFS setting, the goal is to select features to acquire during test-time that optimize an objective that is often used as a proxy for test set accuracy. Notably, we assume the presence of an immutable classifier initially, and our focus is solely on selecting features that would most effectively enhance the model's prediction. Lewenberg et. al. [7] tackle the active survey problem, wherein the task is to sequentially select the next survey question to ask. Ma et. al. [9] tackle the problem of active variable selection, wherein they sequentially select the next variable based on the current (partial) set of observed variables. Our work differs from this in that we propose a greedy feature selection algorithm instead of employing an RL-based dynamic feature selector.

Probilistic Circuits(PC): Probabilistic circuits (PCs) [2]

Accepted for the 6th Workshop on Tractable Probabilistic Modeling at UAI (TPM 2023).

are computational graphs with sums, products and tractable distributions as operations. Imposing structural constraints on PCs allows for tractable computation of various probabilistic queries such as marginals (MAR), conditionals (CON), maximum a posteriori probability (MAP) etc. They have garnered great interest over the last decade due to their expressiveness and tractability. Recently, Vergari et. al. [14] proposed a characterization of the tractability of simple PC transformations –sums, products, quotients, powers, logarithms, and exponentials–in terms of sufficient structural constraints of the circuits. Our work builds upon this to characterize and construct tractable PCs for tractable computation of acquisition functions for the test-time active feature selection (AFS) task. In this regard, our work closely relates to Khosravi et. al. [3]. They consider the necessary structural constraints for tractable computation of moments of a PC w.r.t another PC. Our work differs from theirs in that we consider different structural constraints and apply it to the test-time AFS task.

Algorithm 1 MEASURES

Input: Observed variables x, classifier $\mathcal{F}$, PC P modeling probability distribution Pr over variables X, budget B

Output: Set of features Z to observe

```
$Z \leftarrow \emptyset$
for i=1...B do
    $z_i \leftarrow \arg\min_z ComputeEV(\mathbf{x}, \mathcal{F}, \mathcal{P}, z, S)$
    $Z \leftarrow Z \cup z_i$
    $E_u \leftarrow E_u \backslash z_i$
    $E_o \leftarrow E_o \cup z_i$
end for
```

3 PRELIMINARIES

In this section, we introduce PCs and their relevant structural properties.

We adapt the following definition of PCs from Peharz et. al. [11].

Definition 1. *(Probabilistic Circuit). Given a set of random variables $\mathbf{X}$, a PC is a tuple($\mathcal{G}, \psi$), where $\mathcal{G}$ is a directed acyclic graph (DAG) with vertices V and edges E and $\psi : V \rightarrow 2^{\mathbf{X}}$ is the **scope function**, a mapping from each node to a subset of $\mathbf{X}$. The scope of any internal node $N \in V$ equals the union of the scopes of its children, i.e. $\psi(N) = \cup_{N' \in ch(N)} \psi(N')$. Each leaf node of $\mathcal{G}$ computes a probability density over its scope. All internal nodes of $\mathcal{G}$ are either sum nodes (S) or product nodes (P). A sum node S computes a convex combination of its children, i.e. $S = \sum_{N' \in ch(S)} w_{S,N'} N'$ and $\forall N' \in ch(S) w_{S,N'} > 0$. A sum node is said to be normalized if the weights of its children add up to 1, i.e. $\sum_{N' \in ch(S)} w_{S,N'} = 1$. A product node P computes a product of its children, i.e. $P = \prod_{N' \in ch(P)} N'$.*

PCs can be interpreted as neural networks where input layers compute non-linear functions as probability densities over subsets of $\mathbf{X}$ and all internal nodes compute either affine combinations or products of their inputs. Inputs to each node N are limited to its scope $\psi(N)$. Imposing additional structural constraints on the computational graph $\mathcal{G}$ of a PC allows tractable computation of various probabilistic queries such as marginals, conditionals, Maximum A Posteriori (MAP) estimates as well as tractable representations of circuit operations such as products of PCs. Next, we will introduce the structural constraints relevant to our work.

Definition 2. *(Decomposability). A product node P is decomposable if the scope of its children are pairwise disjoint i.e. $\forall N, N' \in ch(P), N \neq N'$ it holds that $\psi(N) \cap \psi(N') = \emptyset$. A PC is decomposable if all its product nodes are decomposable.*

Decomposability is a desirable property as it allows for tractable computation of marginals [12, 2]. Essentially, decomposability allows for integrals at product nodes to be computed as the product of integrals of its children. This, when applied recursively, necessitates integrals to be computed only at the leaves, which we assume to be tractable.

Definition 3. *(Smoothness). A sum node S is smooth if the scope of its children are the same i.e. $\forall N, N' \in ch(S)$ it holds that $\psi(N) = \psi(N')$. A PC is smooth if all its sum nodes are smooth.*

While smoothness is not required for tractable computation of marginals, smooth and normalized sum nodes afford well-defined probabilistic semantics to PCs. Smooth and decomposable PCs are also known as sum-product networks (SPNs) [12].

Smoothness and decomposability ensure that PCs can compute marginals tractably but more advanced circuit operations such as tractable representation of products of PCs, which is required for computation of moments of distributions, as we consider in this work, still remain intractable for arbitrary smooth and decomposable PCs.

Definition 4. *(Compatability). Two smooth and decomposable circuits P and Q are **compatible** if any pair of product nodes $N \in P$ and $M \in Q$ with the same scope can be rearranged into compatible binary products that decompose the same way i.e. $\psi(N_i) = \psi(M_i)$, N_i and M_i are compatible for some rearrangement of the children of N into N_1, N_2 and M into M_1, M_2. A smooth and decomposable circuit is **omni-compatible** if it is combatible with all decomposable circuits.*

4 MEASURES

In our AFS setting, we are given two models over variables $(\mathbf{X}, y)$ 1. a generative model $\mathcal{P}$ modeling the joint probability $Pr(\mathbf{X}, y)$ and 2. a classifier $\mathcal{F}$ modeling $Pr(y|\mathbf{X})$. At test-time, we are given a data instance with partially observed set of features $\mathbf{x} \subset \mathbf{X}$ and we aim to select the next feature to observe such that an acquisition function is maximized. We will introduce other notations in the remainder of the paper, as necessary.

First, we introduce the notion of variance of $\mathcal{F}$ w.r.t $\mathcal{P}$ when $\mathbf{x}$ is observed for a data instance:

Definition 5. *Variance of a classifier $\mathcal{F}$ w.r.t a joint distribution Pr when a partial set of features $\mathbf{x}$ are observed is defined as follows:*

$$\begin{aligned}\mathbb{VAR}_{Pr}[\mathcal{F}, \mathbf{x}] = E_{m \sim Pr(m|x)}[\mathcal{F}(\mathbf{x}m)^2] \\ - [E_{m \sim Pr(m|x)}\mathcal{F}(\mathbf{x}m)]^2\end{aligned}$$

We propose selecting the feature that maximizes the reduction in the variance of $\mathcal{F}$ after it is observed. Intuitively, continuing with the medical example, we want to perform the laboratory test on a patient, knowing the results of which, a diagnosis can be made irrespective of the results of any other tests.

Given: A classifier $\mathcal{F}$, a probability distribution Pr over $\mathbf{X}$ and a set of partially observed features $\mathbf{x}$ for an instance

To Do: Identify the next feature $z \in \mathbf{X} \backslash \mathbf{x}$ that leads to the largest decrease in the variance of the prediction for zx.

$$\begin{aligned}\arg\max_z \mathbb{VAR}_{Pr}[\mathcal{F}, \mathbf{x}] - \mathbb{VAR}_{Pr}[\mathcal{F}, \mathbf{x}z] \\ = \arg\min_z \mathbb{VAR}_{Pr}[\mathcal{F}, \mathbf{x}z]\end{aligned}$$

However, in real-world situations we do not know the value of z while computing the above metric for each candidate feature being considered. So, we propose to use its expected value instead, as defined below,

Definition 6. *Expected variance ($\mathbb{EV}$) of a classifier $\mathcal{F}$ w.r.t a joint distribution Pr for an unobserved feature z is defined as follows*

$$\mathbb{EV}_{Pr}[\mathcal{F}, \mathbf{x}, z] = E_{z \sim Pr(z|\mathbf{x})}\mathbb{VAR}_{Pr}[\mathcal{F}, \mathbf{x}z]$$

Thus, our objective becomes,

$$\begin{aligned}\arg\max_z \mathbb{VAR}_{Pr}[\mathcal{F}, \mathbf{x}] - \mathbb{EV}_{Pr}[\mathcal{F}, \mathbf{x}, z] \\ = \arg\min_z \mathbb{EV}_{Pr}[\mathcal{F}, \mathbf{x}, z]\end{aligned}$$

Algorithm 2 ComputeEV

```
Input: Observed variables $\mathbf{x}$, classifier $\mathcal{F}$ represented as a PC, PC $\mathcal{P}$ modeling probability distribution $Pr$ over variables $\mathbf{X}$, candidate feature $z$, sample size $S$

Output: $\mathbb{EV}_{Pr}[\mathcal{F}, \mathbf{x}, z]$

$EV \leftarrow 0$
for i=1...S do
    $z \sim Pr(z|\mathbf{x})$
    $PC_{M_1} \leftarrow \mathcal{MULTIPLY}(\mathcal{P}, \mathcal{F})$
    $PC_{M_2} \leftarrow \mathcal{MULTIPLY}(PC_{M_1}, \mathcal{F})$
    $Var \leftarrow \int_{m \sim Pr(m|\mathbf{x}z)} PC_{M_2}(\mathbf{x}zm)dm - [\int_{m \sim Pr(m|\mathbf{x}z)} PC_{M_1}(\mathbf{x}zm)dm]^2$
    $EV \leftarrow EV + Var$
end for
$EV \leftarrow \frac{EV}{S}$
Return: $EV$
```

Unfortunately, computation of $\mathbb{VAR}[\mathcal{F}, \mathbf{x}]$ for arbitrary $\mathcal{F}$ and $\mathcal{P}$ is intractable [4]. This is true for even relatively simple models, such as when $\mathcal{P}$ is a naive Bayes model and $\mathcal{F}$ is a logistic regression model. Khosravi et. al. [4] propose using compatible logical circuits and regression circuits to ensure tractable computation of higher moments of $\mathcal{F}$ w.r.t. $\mathcal{P}$. In this work, we propose using an omni-compatible circuit $\mathcal{F}$ and a smooth and decomposable circuit $\mathcal{P}$ instead.

Theorem 1. *The α^{th} moment of an omni-compatible circuit $\mathcal{F}$ over $\mathbf{X}$ comprising a sum unit with k inputs w.r.t a smooth and decomposable circuit $\mathcal{P}$ can be computed in $\mathcal{O}(k^{\alpha}|p|)$ time and space.*

Corollary 1.1. *Variance of an omni-compatible circuit $\mathcal{F}$ over $\mathbf{X}$ comprising a sum unit with k inputs w.r.t a smooth and decomposable circuit $\mathcal{P}$ can be computed in $\mathcal{O}(k^2|p|)$.*

This follows direcly from Theorem 1.

Theorem 2. *Computing the expected variance of an omni-compatible circuit $\mathcal{F}$ over $\mathbf{X}$ comprising a sum unit with k inputs w.r.t a smooth and decomposable circuit $\mathcal{P}$ is #P-hard.*

We present all proofs in the appendix A.1.

In order to address the intractability of computing $\mathbb{EV}$, we propose sampling $x_i \sim Pr(x_i|x)$ and using the sample $\mathbb{EV}$ as an approximation for it. Algorithm 1 presents MEASURES. It takes as input x, $\mathcal{F}$, $\mathcal{P}$ and budget B. Then, in each iteration, it greedily acquires feature z_i that minimizes $\mathbb{EV}_{Pr}[\mathcal{F}, \mathbf{x}, z_i]$ and adds it to the observed set of features. Algorithm 2 is used to compute approximate $\mathbb{EV}_{Pr}[\mathcal{F}, \mathbf{x}, z_i]$ by sampling $z \sim Pr(z|\mathbf{x})$ S times and computing $\mathbb{VAR}$ exactly in each sampling iteration. For details on the $\mathcal{MULTIPLY}$ algorithm we defer the readers to Vergari et. al. [14].

5 EXPERIMENTAL EVALUATION

Our experiments explicitly aim at answering the following questions,

Q1: Does the proposed MEASURES approach effectively select features for each instance at test-time?

Q2: How do MEASURES variants of acquisition functions compare to their sampled approximations?

We evaluate our proposed approach on the MNIST dataset [6] where bottom half of each image at test-time is observed and the task is to select the best set of features from the top-half of the image while adhering to a budget. We evaluate the approaches using 3 different selection budgets of 0.1, 0.3 and 0.5 as proportion of unobserved features.

As shown in Vergari et. al.[14], an additive ensemble of regression trees can be represented as an omni-compatible circuit. We choose an ensemble of gradient boosted trees as $\mathcal{F}$. Additionally, we choose Einsum Networks [11], a class of smooth and decomposable PCs, as $\mathcal{P}$. This combination allows for expressive models while retaining tractability of $\mathbb{VAR}$ computation.

Table 1 presents prediction accuracy on MNIST dataset, comparing our proposed approach MEASURES to 4 baselines, 1. Random feature selection, 2. Using a fixed set of best features to acquire for all instances, as determined by feature importance w.r.t the classifier. For gradient boosted trees the feature importance is considered proportional to the total number of times a feature is split on in the ensemble of trees, 3. KLD - Features z_i with highest KL-divergence [5] between $Pr(m|\mathbf{x})$ and $Pr(m'|\mathbf{x}z_i)$, where $m = \mathbf{X}\backslash\mathbf{x}$ and $m' = \mathbf{X}\backslash\{\mathbf{x} \cup z_i\}$ and 4. $SampleEV$ - Features with the lowest sample expected variance in the output of the classifier. Clearly, MEASURES performs significantly better than all our baselines across the 3 budgets. We speculate that the poor performance of $SampleEV$ is due to the sampling required for computation of sample variance which MEASURES overcomes by enabling tractable computation of variance.

Strategy	0.1	0.3	0.5
Random	0.68	0.83	0.94
SING	0.72	0.86	0.95
KLD	0.79	0.88	0.96
SampleEV	0.67	0.83	0.92
MEASURES (Ours)	0.81	0.91	0.96

Table 1: Prediction accuracy on MNIST dataset using a greedy selection strategy (when applicable) and for 3 acquisition budgets: 0.1, 0.3 and 0.5. We compare our proposed approach MEASURES to 4 baselines: random feature acquisition (Random), single best set of features to acquire for all instances (SING), feature z_i that maximize the expected KL-Divergence between $Pr(m|\mathbf{x})$ and $Pr(m'|\mathbf{x}z_i)$, where m and m' represent the set of unobserved features and sample variance based feature acquisition (SampleEV)

6 CONCLUSION

In this work we present MEASURES, a novel strategy for test-time AFS that involves exact computation of the variance of a probabilistic classifier w.r.t a tractable joint model - in this case, a smooth and decomposable PC. Through our experiments on the MNIST dataset across 3 different selection budgets we show that MEASURES outperforms 4 baselines significantly. As far as we are aware, this is the first work that leverages tractable computation of advanced probabilistic queries such as variance of classifiers through PCs for the task of test-time AFS.

Our future research directions include validating our proposed approach on additional datasets, including real-world medical datasets. We also plan to compare our method against stronger AFS baselines [9, 1], explore the use of tractable computation for measures like KL-divergence in the context of test-time AFS, and compare them against their approximate variants. Additionally, we aim to incorporate MEASURES within existing RL-based AFS frameworks [13, 8].

Acknowledgements

The authors acknowledge the support by the NIH grant R01HD101246 and ARO award W911NF2010224.

References

[1] Muhammed Fatih Balin, Abubakar Abid, and James Y. Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 444–453. PMLR, 2019.

[2] Y Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf*, 2020.

[3] Pasha Khosravi, YooJung Choi, Yitao Liang, Antonio Vergari, and Guy Van den Broeck. On tractable computation of expected predictions. In *NeurIPS*, pages 11167–11178, 2019.

[4] Pasha Khosravi, YooJung Choi, Yitao Liang, Antonio Vergari, and Guy Van den Broeck. On tractable computation of expected predictions. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[5] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[6] Y. LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[7] Yoad Lewenberg, Yoram Bachrach, Ulrich Paquet, and Jeffrey S. Rosenschein. Knowing what to ask: A bayesian active learning approach to the surveying problem. In *AAAI*, pages 1396–1402. AAAI Press, 2017.

[8] Yang Li and Junier Oliva. Active feature acquisition with generative surrogate models. In *International Conference on Machine Learning*, pages 6450–6459. PMLR, 2021.

[9] Chao Ma, Sebastian Tschiatschek, Konstantina Palla, José Miguel Hernández-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. In *International Conference on Machine Learning*, 2018.

[10] Sriraam Natarajan, Srijita Das, Nandini Ramanan, Gautam Kunapuli, and Predrag Radivojac. On whom should i perform this lab test next? an active feature elicitation approach. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3498–3505. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/486.

[11] Robert Peharz, Steven Lang, Antonio Vergari, Karl Stelzner, Alejandro Molina, Martin Trapp, Guy Van den Broeck, Kristian Kersting, and Zoubin Ghahramani. Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 7563–7574. PMLR, 2020.

[12] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690, 2011. doi: 10.1109/ICCVW.2011.6130310.

[13] Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[14] Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In *Advances in Neural Information Processing Systems*, volume 34, pages 13189–13201. Curran Associates, Inc., 2021.

A APPENDIX

A.1 THEORETICAL RESULTS

Theorem 1. *The α^{th} moment of an omni-compatible circuit $\mathcal{F}$ over $\mathbf{X}$ comprising a sum unit with k inputs w.r.t a smooth and decomposable circuit $\mathcal{P}$ can be represented as a smooth and decomposable circuit constructed in $\mathcal{O}(k^{\alpha}|p|)$ time and space.*

Proof. We use the known result that the product of an omni-compatible circuit $\mathcal{F}$ over $\mathbf{X}$ comprising a sum unit with k inputs w.r.t a smooth and decomposable circuit $\mathcal{P}$ can be represented as a smooth and decomposable circuit constructed in $\mathcal{O}(k|p|)$ time and space, where $|p|$ is the size of the circuit $\mathcal{P}$[14]. Subsequently, $\mathcal{F}^{\alpha}.\mathcal{P}$ can be represented as a smooth and decomposable circuit constructed in $\mathcal{O}(k^{\alpha}|p|)$ time and space. Finally, the α-th moment of $\mathcal{F}$ can be computed through a marginal query over $\mathcal{F}^{\alpha}.\mathcal{P}$ in linear time in the size of the circuit. □

Theorem 2. *Computing the expected variance of an omni-compatible circuit $\mathcal{F}$ over $\mathbf{X}$ comprising a sum unit with k inputs w.r.t a smooth and decomposable circuit $\mathcal{P}$ is #P-Hard.*

Proof. Let $\mathcal{F}$ be an omni-compatible circuit over $\mathbf{X}$ comprising a sum unit with k inputs and $\mathcal{P}$ be a smooth and decomposable PC. We denote the α-th moment of $\mathcal{F}$ w.r.t $\mathcal{P}$ as $\mathcal{M}^{\alpha}_{\mathcal{P}}(\mathcal{F})$ and $\mathcal{F}^{\alpha}.\mathcal{P}$ as $PC_{M_{\alpha}}$. Using Theorem 1 we know that $PC_{M_{\alpha}}$ is smooth and decomposable.

Now, consider the following $\#P-Hard$ problem: Let p and q be two decomposable circuits over variables $\mathbf{X}$. Computing their product $m(\mathbf{X}) = p(\mathbf{X}).q(X)$ as a decomposable circuit is $\#P - Hard$.

Using this result, we see that representing the product of marginal PCs $Mar(PC_{M_1}).Mar(PC_{M_1})$ as a smooth and decomposable PCs, which is required for computing $\mathbb{EV}_{Pr}(\mathcal{F}, \mathbf{x}, z)$, is $\#P - Hard$, rendering our expected variance problem $\#P - Hard$. □