1. Related Work. We will discuss related literature in several subfields.

Interpretable Models: There is a growing interest in interpretable (transparent, comprehensible) models because of practical societal importance(see **?????????**). There are now regulations on algorithmic decision-making in the European Union on the "right to an explanation" (**?**) that would legally require interpretability in predictions.

The body of work closest to ours is that on *optimal decision tree modeling*. There is work starting in the late 1990's on building optimal decision trees using optimization techniques (e.g., **???**), continuing until the present (**??**). A particularly interesting paper along these lines is that of **?**, who created a "bottom-up" way to form optimal decision trees. Their method performs an expensive search step, mining all possible leaves (rather than all possible rules), and using those leaves to form trees. Their method can lead to memory problems, but it is possible that these memory issues can be mitigated using the theorems in this paper. [1] Another work close to ours is that of **?**, who introduce an algorithm to generate more interpretable decision trees by allowing constraints to be placed on the size of the decision tree. Like us, they use a branch-and-bound technique to constrain the size of the search space and limit the eventual size of the decision tree. During tree construction, they bound the possible Minimum Description Length (MDL) cost of every different split at a given node. If every split at that node is more expensive than the actual cost of the current subtree, then that node can be pruned. In this way, they were able to prune the tree while constructing it instead of just constructing the tree and then pruning at the end. Hm? However, even with the added bounds, this approach does not generally yield globally optimal decision trees because they constrained the number of nodes in the tree.

On the other end of the spectrum from optimal decision tree methods are *greedy splitting and pruning* methods like CART and C4.5 . They do not perform exploration of the search space beyond greedy splitting.

There are *Bayesian tree and rule list methods* that aim to explore the space of trees **???**, however, the space of trees of a given depth is much larger than the space of rule lists of that same level of depth, and the trees within these algorithms are grown in a top-down greedy way. Because of this, the authors noted that the MCMC chain tends to reach only locally optimal solutions. This is why Bayesian rule-based methods (**??**) have tended to be more successful in escaping local minima. This work builds specifically on that of **?**. In particular, we use their fast bit-vector manipulations, and build on their bounds.

Rule learning methods: Most rule learning methods are not designed for optimality or interpretability, but mainly for computational speed and/or accuracy. In *associative classification*, classifiers are formed either greedily from the top down as rule lists, **????** or they are formed by taking the simple union of pre-mined rules **?**, whereby any observation that fits into any of the rules is classified as positive **?**. Inductive Logic Programming **?** algorithms form disjunctive normal form patterns via a set of operations (rather than using optimization). These approaches are not appropriate for obtaining a guarantee of optimality. Methods for decision list learning construct rule lists iteratively in a greedy way **??????**, which again have no guarantee on optimality, and tend not to produce optimal rule lists in general. Some methods allow for interpretations of single rules, without constructing rule lists (**?**).

There is a tremendous amount of related work in other subfields that are too numerous to discuss at length here. We have not discussed *rule mining* algorithms since they are part of an interchangeable preprocessing step for our algorithm, and are deterministically fast (that is, they will not generally slow our algorithm down). We also did not discuss methods that create disjunctive

[1] There is no public version of their code for distribution as of this writing.

normal form models, e.g. logical analysis of data, and many associative classification methods).

There are *related problems concerning interpretable lists of rules.* Rule lists of various flavors have been developed recently such as Falling Rule Lists **?**, which are constrained, as well as rule lists for dynamic treatment regimes **?** and cost-sensitive dynamic treatment regimes **?**. Both **?** and **?** use Monte Carlo searches through the space of rule lists. The method proposed in this work could potentially be adapted to handle these kinds of interesting problems. We are currently working on bounds for Falling Rule Lists **?** similar to those presented here.

References.