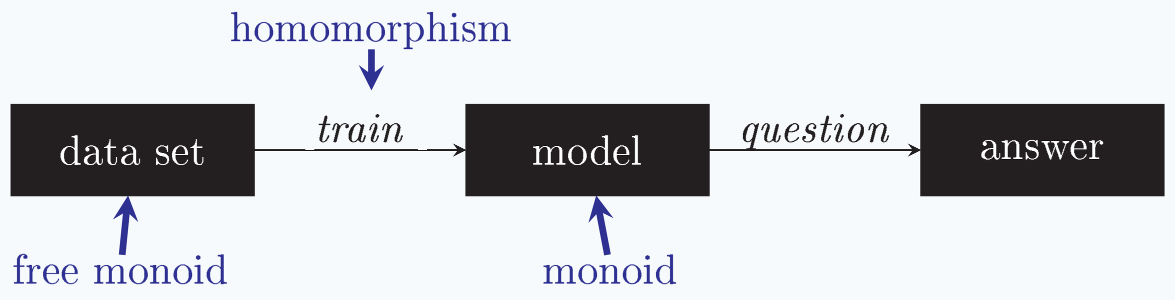
ALGEBRAIC CLASSIFIERS

Mike Izbicki

UCR

OVERVIEW

When our learning models have algebraic structure:



We get three important algorithms "for free"

DEFINITIONS

A **monoid** consists of a set $\mathcal{M}$, an associative binary operation $\diamond : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$, and a special identity element ϵ such that for all $m \in \mathcal{M}$,

$$\epsilon \diamond m = m \diamond \epsilon = m$$

If the model $\mathcal{M}$ we are trying to learn forms a monoid, then our batch trainer $T : \mathcal{D} \to \mathcal{M}$ is a **homomorphism** if for all data sets $d_1, d_2 \in \mathcal{D}$:

$$T(d_1 \diamond d_2) = T(d_1) \diamond T(d_2)$$

THE HLEARN LIBRARY

HLearn is a Haskell based implementation of these ideas. Source code and documentation is online at:

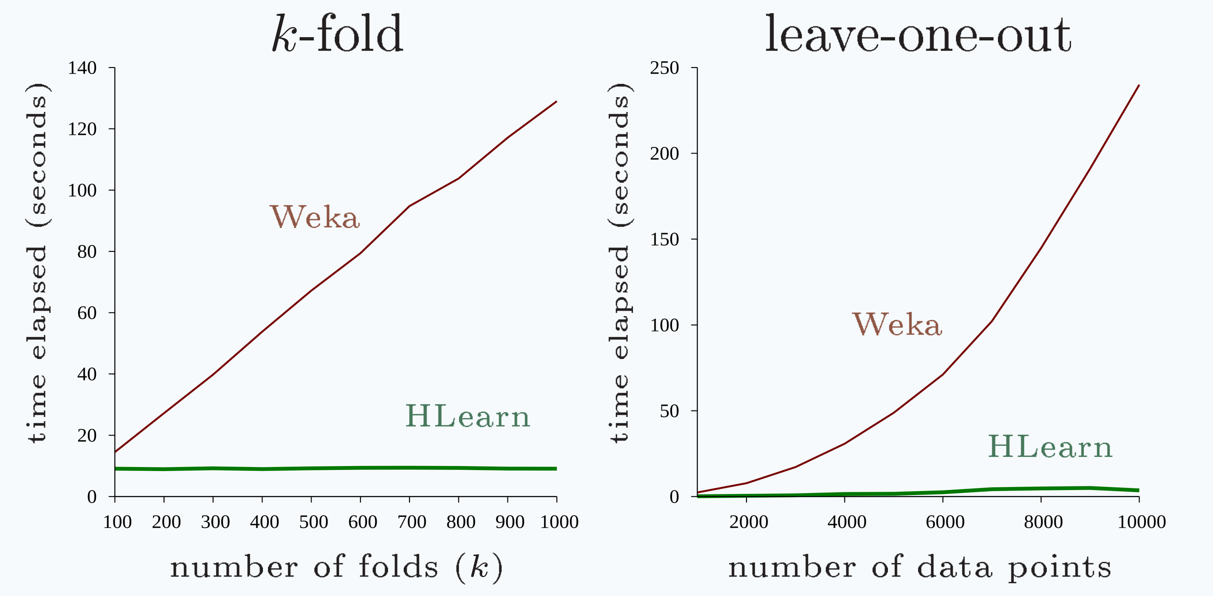
http://github.com/mikeizbicki/hlearn

Implemented algorithms include:

- **Univariate distributions**: exponential, log-normal, normal, kernel density estimator, binomial, categorical, geometric, poisson
- **Multivariate distributions**: normal, categorical, subset of markov networks
- **classifiers**: naive bayes, full bayes, decision stumps, decision trees, k-nearest neighbor (naive and kd-tree), perceptron, bagging, boosting (sort of)
- **other**: markov chains, k-centers, bin packing, multiprocessor scheduling

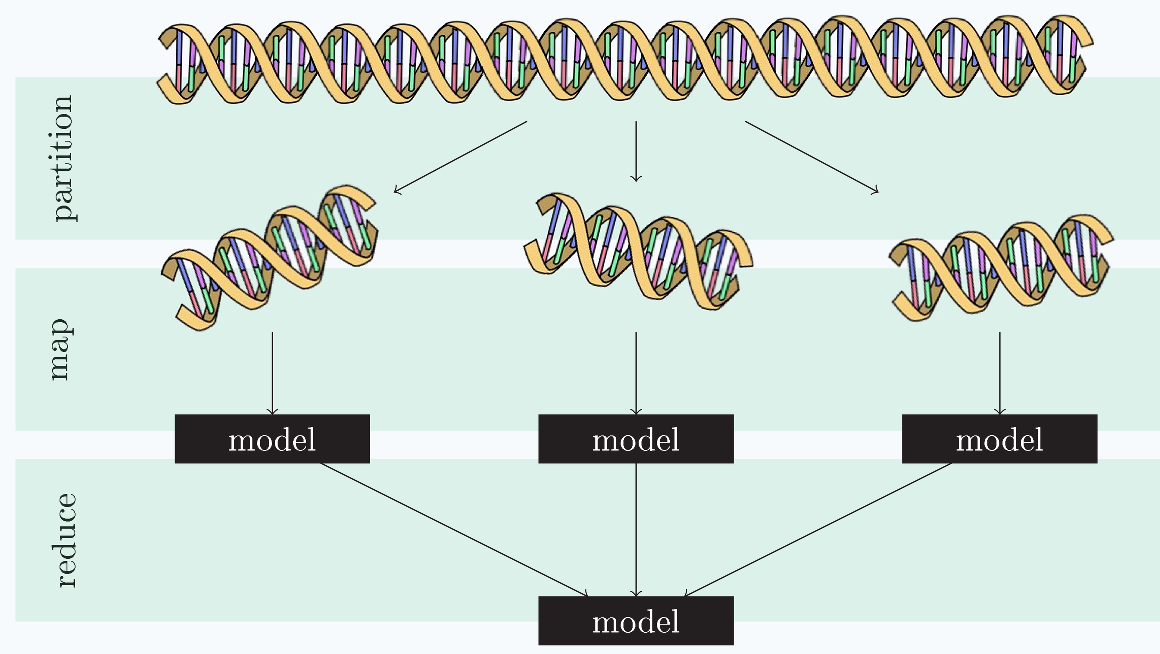
WHAT WE GET

FAST CROSS-VALIDATION



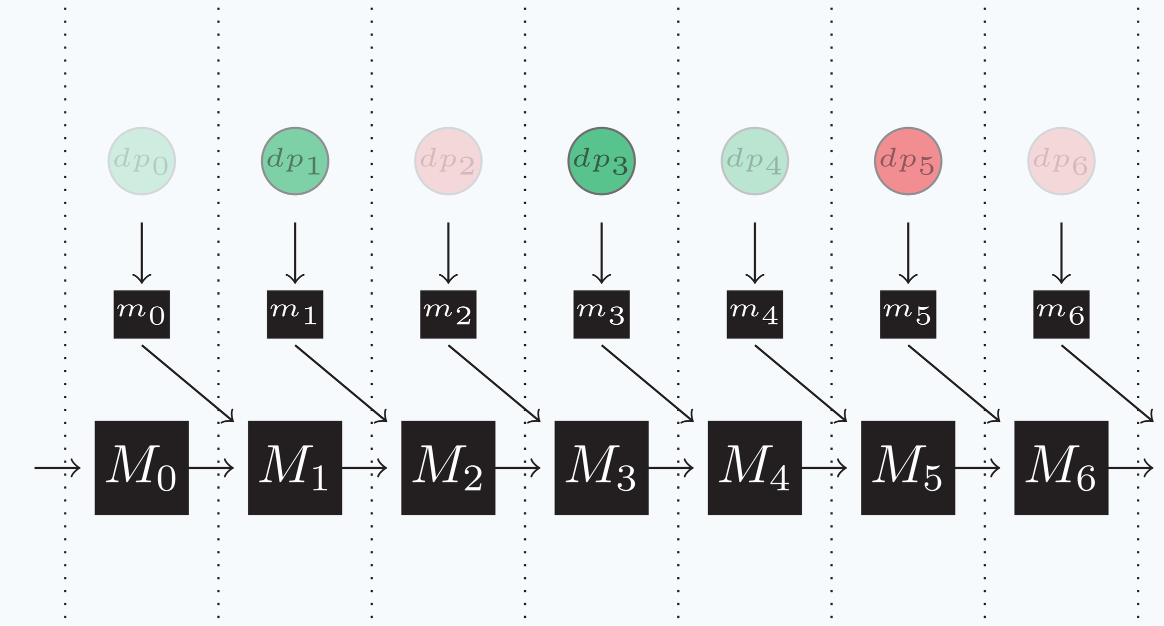
Standard k-fold cross-validation runs in time $\Theta(kn)$, but monoid cross-validation runs in time $\Theta(n)$. Both algorithms give *exactly* the same results. These graphs plot actual run times for the naive bayes classifier.

PARALLEL TRAINING



If our model is a monoid, then we can do exact MapReduce; and if we can do exact MapReduce then our model is a monoid.

ONLINE TRAINING

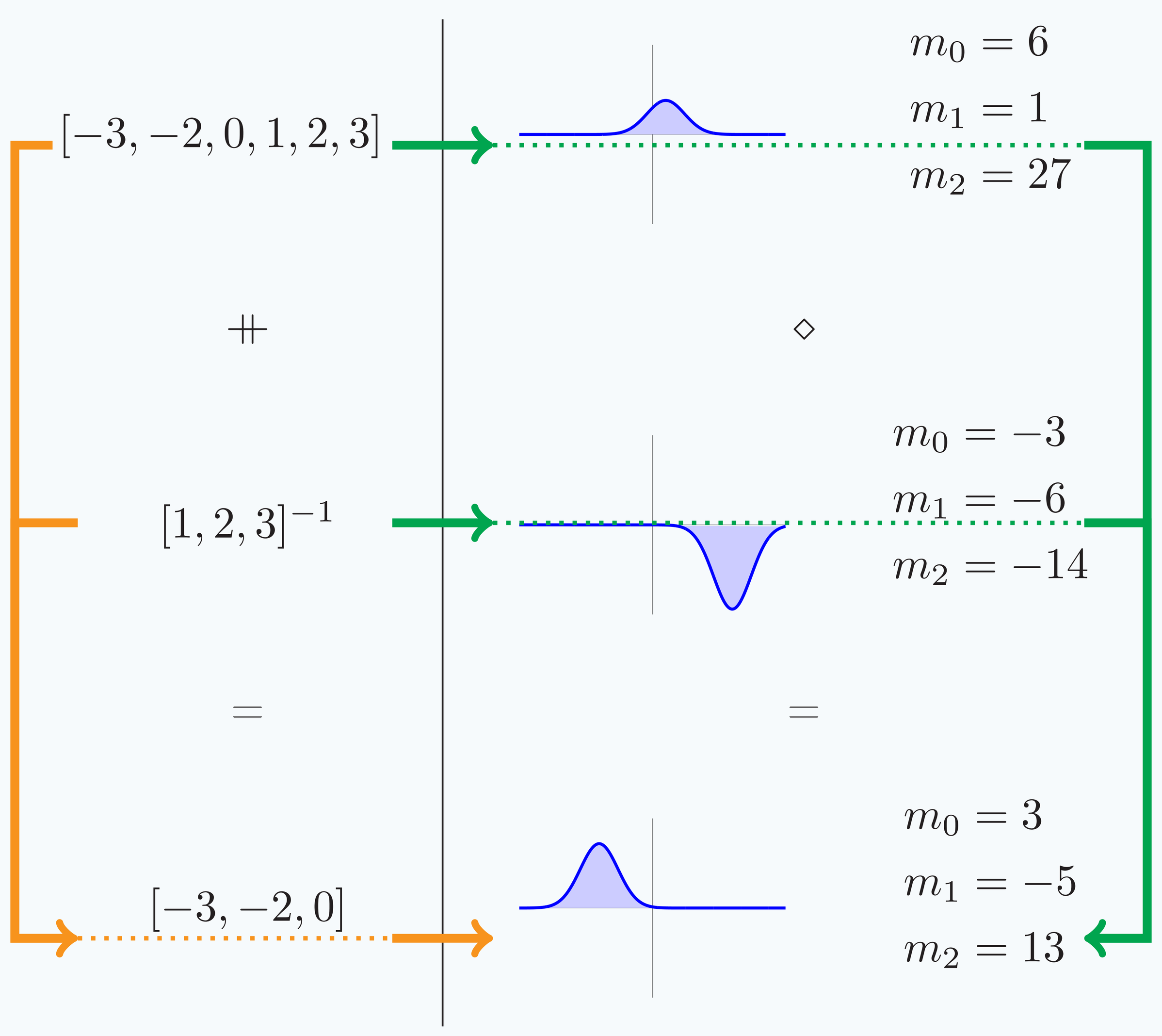


Any monoid can easily be trained online using the formula:

$$M_{t+1} = M_t \diamond m_t$$

EXAMPLES

NORMAL DISTRIBUTION



The normal distribution is a monoid homomorphism because this diagram commutes.

BAYESIAN CLASSIFIERS

Bayesian models classify using the formula

$$C_{Bayes}(a) = \arg\max_{l \in \mathcal{L}} P(L = l) P(A = a | L = l)$$

In order to do this calculation, we need to know the probability distributions $P(L)$ and $P(A|L)$. The model for the Bayesian classifier is therefore a tuple containing these distributions.

$$\mathcal{M}_{Bayes} = (P(L), P(A|L))$$

If these base distributions have monoid structure, then the Bayesian classifier will as well. The binary operation is defined as:

$$(P_1(L), P_1(A|L)) \diamond (P_2(L), P_2(A|L)) = (P_1(L) \diamond P_2(L), P_1(A|L) \diamond P_2(A|L))$$

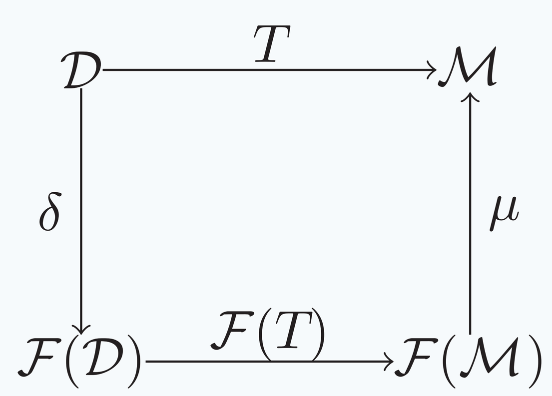
The identity element is:

$$\epsilon_{Bayes} = (\epsilon_{P(L)}, \epsilon_{P(A|L)})$$

FREE HOMTRAINER

Category theory gives us a way to imbue any learning model with algebraic structure. This is called the free object $\mathcal{F}(\mathcal{M})$ generated by model $\mathcal{M}$. For monoids, this process is a generalization of bagging.

If $\mathcal{D}$ is our data set, $T : \mathcal{D} \to \mathcal{M}$ is the training function, and $\mathcal{M}$ is the model space, then our goal is to select functions δ and μ such that the following diagram commutes:



In the paper, we use this technique to derive new boosting inspired parallel algorithms.