A Bayesian Approach to Learning Scoring Systems

Şeyda Ertekin and Cynthia Rudin

Abstract

We present a Bayesian method for building *scoring systems*, which are linear models with coefficients that have very few significant digits. Usually the construction of scoring systems involve manual labor – humans invent the full scoring system or choose how logistic regression coefficients should be scaled and rounded. This obviously leads to suboptimal solutions. Our approach is different, in that humans need only specify the prior over what the coefficients should look like, and the rest is automatic. For this approach, we design Markov chain Monte Carlo (MCMC) steps that tend pull the coefficient values towards their "natural scale." The empirical evaluation on a wide variety of datasets demonstrates that the proposed method achieves a high degree of interpretability of the models while maintaining competitive generalization performances.

Note: Final Version of paper published in Big Data, volume 3, number 4, 2016.

Introduction

We consider a problem that is central to the practical use of statistical models for decision making, namely how to construct a useful scoring system. A scoring system is a linear model where the coefficients are simple, which usually means the coefficients have few significant digits. This has natural benefits: it allows a person to calculate the score without a calculator. It also allows a person to understand the joint relationship between variables, and helps to remind the user of the units of the variable. Students often make the mistake of trying to prune variables with small coefficients without considering whether the variables are on massively different scales; a coefficient of 0.0062519 on a variable that takes values in the thousands might easily be more important than coefficient of 0.6 taking values between 0 and 10. By expressing a variable (ideally) as a single significant digit and an order of magnitude (e.g., 6×10^{-4}) users can more easily see that the scale of variable is important.

The interpretability and general usability of a predictive model can make the difference between whether it is used or whether it forever sits on a shelf. In domains from medical diagnosis (Antman et al., 2000; Gage et al., 2001), bioinformatics (Freitas, Wieser, and Apweiler, 2010), credit scoring (Siddiqi, 2005; Martens et al., 2007; Martens, Baesens, and Van Gestel, 2009), churn analysis (Lima, Mues, and Baesens, 2009; Verbeke et al., 2011) to crime prediction (Ridgeway, 2013; Andrade, 2009), often a model can *only* be defensible if it is interpretable, and easy to use without a calculator (see Giraud-Carrier, 1998; Freitas, Wieser, and Apweiler, 2010).

In order to build a scoring system, we consider a prior that favors fewer significant digits, certain values for those coefficients, and sparsity in the number of coefficients. In order to make inference more efficient, we designed our Markov chain Monte Carlo (MCMC) proposal steps to encourage the scale of the coefficients towards their "natural scale." The natural scale of a coefficient is the inverse of the scale of its variable. For instance, for a variable taking values between 80 and 120, the natural scale would be 10^{-2}. If coefficients are actually on their natural scale, then the variables will contribute on the order of 10^{-1}, where the logistic curve is not at its (saturating) extreme points.

Related Work

Most machine learning methods are *black-box* models where there is a complicated relationship between the input variables and the predictions. Algorithms such as support vector machines (Vapnik and Vapnik, 1998), artificial neural networks (Hagan et al., 1996) and AdaBoost (Freund and Schapire, 1997) fall in this category. On the other end are transparent models, such as decision trees (Quinlan, 1986), decision lists (Rivest, 1987) and decision tables (Kohavi, 1995) (see also Freitas, 2014, for a review of the interpretability of various classification models). In between, there is the wide spectrum of techniques and methodologies to that aim to achieve, or improve, the interpretability of an existing black-box model. For example, instance explanation methods (Štrumbelj, Kononenko, and Robnik Šikonja, 2009; Štrumbelj and Kononenko, 2010; Baehrens et al., 2010) aim to explain the classifications of each instance by computing a score for each of the variables that indicates the extent to which that variable contributed to the classification of the instance. Other methods (Meinshausen, 2010; Van Assche and Blockeel, 2007; Fung, Sandilya, and Rao, 2005; Martens et al., 2007; Bien and Tibshirani, 2011) extract rules and prototype examples to illustrate the relationship between input variables and the predicted outcome. Our method is on the extreme end of the transparent modeling

spectrum, in that the models are required to be extremely sparse (unlike C4.5 trees or even lasso models), and have very few significant digits per coefficient.

Scoring systems require users to add, subtract and multiply a few small numbers in order to make a prediction. Being primarily used in the medical community, these models are used to assess the risk of numerous serious medical conditions since they allow physicians to make quick predictions, without extensive training, and without the use of a computer. Some popular medical scoring systems include SAPS (Le Gall et al., 1984) and APACHE (Knaus et al., 1981) to assess ICU mortality risk, CHADS_2 (Gage et al., 2001) to assess the risk of stroke in patients with atrial fibrillation and TIMI (Antman et al., 2000) to assess the risk of death and ischemic events. Despite being very interpretable, many medical scoring systems that are currently in use were hand-crafted by physicians, whereby a panel of experts simply agreed on a model (Gage et al., 2001), but they were constructed without optimizing for predictive accuracy. In some cases, physicians built scoring systems by combining existing linear classification methods with heuristics. For instance a version of SAPS was constructed by rounding logistic regression coefficients, but this approach is at odds with the fact that rounding is known to produce suboptimal solutions. Rounding in high dimensions is purely a heuristic perturbation to all of the coefficients within a high dimensional space without regard to the objective function, and thus tends to produce poor results. The approach we present can be used directly to replace the rounding or manual model creation done by physicians.

An optimization-based (non-Bayesian) approach to learning scoring systems is presented by Ustun, Tracà, and Rudin (2013).

Model

Given a dataset $D = (\mathbf{x}_i, y_i)_{i=1\ldots N}$ where $\mathbf{x} \in \mathbb{R}^{\mathbf{P}}$ and $y \in \{+1, -1\}$, we are constructing a linear classifier

$$\hat{y} = \text{sign}(\boldsymbol{\lambda}^T \mathbf{x})$$

where $\boldsymbol{\lambda} \in \mathbb{R}^P$. We sample from a posterior distribution over $\boldsymbol{\lambda} \sim p(\boldsymbol{\lambda}|D)$. The likelihood is the standard likelihood for a linear model, leading to:

$$\log p(D|\boldsymbol{\lambda}) = -\sum_{i=1}^{N} \log(1 + \exp(-y_i \boldsymbol{\lambda}^T \mathbf{x}_i)).$$

The prior over $\boldsymbol{\lambda}$ encourages a user-specified notion of interpretability. Interpretability is not well-defined (Kodratoff, 1994; Pazzani, 2000) and is heavily context dependent. The notion of interpretability used in this work is a sum of interpretability penalties for each of the coefficients. Specifically, the user defines a table of penalties (an example is shown in Table 1). The interpretability score of coefficient λ_i is the smallest score of all of the interpretability penalties that apply to λ_i. The interpretability score of $\boldsymbol{\lambda}$ is the sum of the individual interpretability scores of all coefficients. Our prior belief is that the model will be as interpretable as possible (equivalently, that an interpretable model exists), thus we set the prior probability of $\boldsymbol{\lambda}$ to be:

$$\log p(\boldsymbol{\lambda}; \beta) = -\beta \log \left(1 + \sum_{j=1}^{P} \min_{k, \lambda_j \in L_k} s_i\right)$$

where β is a prior hyper parameter that determines the overall strength of the prior.

We proceed using a specialized Metropolis-Hastings sampling method, with a proposal distribution $Q(\boldsymbol{\lambda^*}|\boldsymbol{\lambda^t})$ that gives the probability of proposing $\boldsymbol{\lambda^*}$ when at $\boldsymbol{\lambda^t}$.

Standard MCMC techniques will not work, as we will discuss. To deal with this, we represent each coefficient value λ as $\lambda = v \times c \times 10^e$ where $v \in \{+1, -1\}$ denotes the sign, c is called the root and $e \in \mathbb{Z}$ is the exponent. This representation allows us to measure the interpretability of the root and exponent separately, and allows our MCMC not to get stuck far away from the high density region. For instance, in the example in Table 1, coefficient 5,000 and coefficient .05 are equally interpretable because they have the same value of c, which is 5. In Table 1, the score s_k is only a function of c, but in general could be a function of the exponent also.

MCMC

We define a (discrete) probability distribution γ over $1, .., K$, corresponding to the sets $\mathcal{L}_k$, with γ_k being the probability of choosing set $\mathcal{L}_k$ in the second step below.

We define $\bar{e}_j$ as the negative of the "natural exponent" of the median of feature j's values. The natural exponent of a number is the placement of the first significant digit. For instance, if the median value of feature j is 15, the natural exponent is 2, since 15 = $.15 \times 10^2$. The natural exponent of 19352 is 5, and the natural exponent of .0012 is -2. We need to use the negative of the natural exponent, because the coefficient should act to bring the feature values back to the range [0,1]. Thus, if the feature value is 19352, the coefficient for that feature should be on the order of 10^{-5}. The proposal steps, while trying to explore the space, favor directions leading to the natural exponent.

Here is the procedure for generating a proposal step:

1. Randomly select a feature j.
2. Sample an integer between 1 and K from the distribution γ (call k_j the index we select). This will choose one of the sets $\mathcal{L}_{k_j}$.
3. Select a coefficient c_j uniformly from the total number of values between 0 and 1 that have the number of significant digits specified in the $\mathcal{L}_{k_j}$. This is the root of the coefficient that will be used for the proposal $\boldsymbol{\lambda}_j^*$.
   - To give an example, if $\mathcal{L}_{k_j}$ contains numbers with 3 significant digits, we consider values such as [1.01,...,4.44,..,6.38,...,9.99].
   - If the $\mathcal{L}_{k_j}$ contains numbers with 1 significant digit whose value is 5, then we consider only the value 0.5.
4. Sample an exponent e_j^* by rounding a value from $\mathcal{N}\left(\frac{e_j^t + \bar{e}_j}{2}, \sigma^2\right)$ where e_j^t is the negative of exponent of $\boldsymbol{\lambda}_j^t$ and $\bar{e}_j$ is the negative of the natural exponent.

k	Score $s_k(c)$	Condition k
1	0	Coefficient is 0
2	1	Coefficient has 1 significant digit (= 1)
3	2	Coefficient has 1 significant digit (= 5)
4	5	Coefficient has 1 significant digit
5	10	Coefficient has 2 significant digits
6	25	Coefficient has 3 significant digits

Table 1: Interpretability buckets $\mathcal{L}_k$ and their associated scores.

5. Sample a sign s_j for $\boldsymbol{\lambda}_j^*$ from $2\times$ Bernoulli$(p_j)-1$, where p_j is the correlation coefficient of feature j with the class label y.

Now we have constructed λ_j^*.

Now, we define the transition probabilities. Define $|\mathcal{L}_{k_j}|$ to be the number of coefficients in $\mathcal{L}_{k_j}$ between 0 and 1 (that do not fall into earlier categories).

$$\begin{aligned} Q(\boldsymbol{\lambda}^*|\boldsymbol{\lambda}^t) &= \prod_{j=1}^{P} \gamma_{k_j} \frac{1}{|\mathcal{L}_{k_j}|} \mathbb{P}\left(e_j^*|e_j^t\right) \\ &= \prod_{j=1}^{P} \gamma_{k_j} \frac{1}{|\mathcal{L}_{k_j}|} \int_{e_j^*-1/2}^{e_j^*+1/2} \mathcal{N}\left(\tilde{e}; \frac{e_j^t+\bar{e}}{2}, \sigma_j^2\right) d\tilde{e} \\ &= \prod_{j=1}^{P} \gamma_{k_j} \frac{1}{|\mathcal{L}_{k_j}|} \left(\Phi(e_j^*+1/2; \frac{e_j^t+\bar{e}}{2}, \sigma_j^2) - \right. \\ &\qquad \left. \Phi(e_j^*-1/2; \frac{e_j^t+\bar{e}}{2}, \sigma_j^2)\right) \end{aligned}$$

where $\mathcal{N}(x;\mu,\sigma^2)$ and $\Phi(x;\mu,\sigma^2)$ are the normal density function and distribution function respectively with mean μ and variance σ^2, evaluated at x. So clearly,

$$\begin{aligned} \log Q(\boldsymbol{\lambda}^*|\boldsymbol{\lambda}^t) = \sum_{j=1}^{P} \log \left(\gamma_{k_j} \frac{1}{|\mathcal{L}_{k_j}|} \left(\Phi(e_j^*+1/2; \frac{e_j^t+\bar{e}}{2}, \sigma_j^2) - \right.\right. \\ \left.\left. \Phi(e_j^*-1/2; \frac{e_j^t+\bar{e}}{2}, \sigma_j^2)\right)\right). \end{aligned}$$

The Metropolis-Hastings acceptance ratio for the transition $\boldsymbol{\lambda}^t \to \boldsymbol{\lambda}^*$ is

$$a = \frac{p(\boldsymbol{\lambda}^*;\beta)p(D|\boldsymbol{\lambda}^*)Q(\boldsymbol{\lambda}^t|\boldsymbol{\lambda}^*)}{p(\boldsymbol{\lambda}^t;\beta)p(D|\boldsymbol{\lambda}^t)Q(\boldsymbol{\lambda}^*|\boldsymbol{\lambda}^t)}$$

So we compute

$$\begin{aligned} \log a = \log p(\boldsymbol{\lambda}^*;\beta) + \log p(D|\boldsymbol{\lambda}^*) + \log Q(\boldsymbol{\lambda}^t|\boldsymbol{\lambda}^*) \\ - \log p(\boldsymbol{\lambda}^t;\beta) - \log p(D|\boldsymbol{\lambda}^t) - \log Q(\boldsymbol{\lambda}^*|\boldsymbol{\lambda}^t) \quad (1) \end{aligned}$$

and accept the transition if $\log a \geq 0$, and accept with probability $\exp(\log a)$ otherwise. In Equation (1), $p(\boldsymbol{\lambda};\beta)$ is the prior for the specified $\boldsymbol{\lambda}$, $Q(\boldsymbol{\lambda}^t|\boldsymbol{\lambda}^*)$ and $Q(\boldsymbol{\lambda}^*|\boldsymbol{\lambda}^t)$ are the transition probabilities for $\boldsymbol{\lambda}^* \to \boldsymbol{\lambda}^t$ and $\boldsymbol{\lambda}^t \to \boldsymbol{\lambda}^*$, and $p(D|\boldsymbol{\lambda})$ is the likelihood.

This MCMC algorithm can be used for sampling from the posterior, or if the user would like a MAP solution, that can be obtained as well. There are guarantees we can provide about the MAP solution that are useful for branch-and-bound type methods.

Bounds on the Optimal MAP Solution

Our first result allows us to exclude a large part of the search space for the MAP solution. If a current subset of coefficients are very bad, then no possible values for the other coefficients could compensate. Without loss of generality, we denote the bad subset of coefficients 1,...,θ for notational simplicity.

Theorem 1. *Define $\boldsymbol{\lambda}_{last}^t$ as the first part of our current solution, followed by the most optimistic possible other coefficients. Also define reference vector $\boldsymbol{\lambda}_{ref}$, which could be, for instance, the best solution found so far.*

$$\boldsymbol{\lambda}_{last}^t \in \operatorname*{argmin}_{[\widetilde{\lambda}_{\theta+1}\cdots\widetilde{\lambda}_p]\in\mathbb{R}^{p-\theta}} -\text{log likelihood}\left(\left[\lambda_1^t,\cdots,\lambda_\theta^t,\widetilde{\lambda}_{\theta+1},\cdots,\widetilde{\lambda}_p\right]\right).$$

Then define $\widetilde{\boldsymbol{\lambda}}^t = \left[\lambda_1^t,\cdots,\lambda_\theta^t,\widetilde{\boldsymbol{\lambda}}_{last}^t\right]$ as the best solution starting with $\lambda_1^t,\cdots,\lambda_\theta^t$. Let R_s denote negative log posterior and $V = R_s(\boldsymbol{\lambda}_{ref})$. If

$$-\text{log likelihood}(\widetilde{\boldsymbol{\lambda}}^t) + \beta \log\left(1 + \sum_{j=1}^{\theta} \min_{k:\lambda_j^k\in\mathcal{L}_k} s_k\right) \geq V$$

then for $\boldsymbol{\lambda}_{opt} \in \operatorname{argmin}_{\boldsymbol{\lambda}} R_s(\boldsymbol{\lambda})$ we know

$$[\lambda_{opt,1},\cdots,\lambda_{opt,\theta}] \neq [\lambda_1,\cdots,\lambda_\theta].$$

This result provides an easily checkable condition for whether coefficients $[\lambda_1,\cdots,\lambda_\theta]$ could be within an optimal MAP solution. Finding $\tilde{\boldsymbol{\lambda}}^t$ is of similar complexity to logistic regression, because it is precisely logistic regression with some of the coefficients being fixed:

$$\begin{aligned} &\sum_i \log\left(1 + e^{-\left(y_i \sum_j \lambda_j x_{ij}\right)}\right) \\ &= \sum_i \log\left(1 + e^{-\left(y_i \sum_{j=1}^{\theta} \lambda_j^T x_i + y_i \sum_{j=\theta+1}^{p} \lambda_j x_{ij}\right)}\right) \\ &= \sum_i \log\left(1 + \text{Const}_i \times e^{-y_i \sum_{j=\theta+1}^{p} \lambda_j x_{ij}}\right). \end{aligned}$$

Remember that it is not necessary to always test the first group of coefficients $\{1,...,\theta\}$. One could test any subset of $\{1,2,...,p\}$.

The second result provides a characterization of the score of an optimal solution. As a result, if the score of the coefficients for a model are too high, then it could not possibly be an optimal solution.

Theorem 2.

$$\textit{Score}(f_{opt}) \leq \exp\left[\frac{m\log 2 + \beta\log(1 + s_{\{0\}} \times p)}{\beta}\right] - 1.$$

	Logistic Regression		AIM				
	Train	Test	Train@MaxP.	Test@MaxP.	Train	Test	Accep. Rate
Haberman	75.34±1.49	72.64±3.87	75.24±1.90	72.05±4.00	74.96±1.58	72.47±3.50	2.66±0.45
B.Cancer	97.34±0.65	96.65±1.17	97.43±0.62	96.52±1.06	97.00±0.49	96.30±1.05	4.71±0.25
BankNote	99.09±0.13	98.59±0.38	99.20±0.25	98.77±0.25	99.12±0.16	98.69±0.28	1.04±0.26
Yeast	65.86±1.62	65.01±2.40	65.54±1.50	64.54±2.02	64.96±1.28	64.39±2.37	2.30±0.25
Abalone	84.05±0.31	84.41±0.77	84.00±0.35	84.41±0.95	83.94±0.31	84.41±0.74	1.75±0.36
BWH D.A.	83.34±1.16	78.86±1.84	83.32±1.07	78.98±1.71	82.56±0.98	79.60±1.70	6.82±0.46
BWH P.C.	73.35±0.88	72.21±1.85	73.21±0.94	72.25±1.87	72.11±0.65	71.29±1.35	5.78±0.26

Table 2: Classification accuracy on the datasets, averaged over 10 runs. BWH D.A. and BWH P.C. are the Domestic Abuse and Poor Condition partitions of the Brigham and Women's Hospital dataset, respectively. The last column is the acceptance rate of the proposals of AIM.

Theorems 1 and 2 provide screening tests that can be used as the algorithm is running to identify bounds for a branch and bound strategy. The screening test can determine that a subset of coefficients having particular values is no good, in that these coefficient values would provably never be included in a MAP solution.

To use Theorem 1 at any point during the optimization procedure, we start with a reference solution $\boldsymbol{\lambda}_{\text{ref}}$, which could be the best solution found so far. We then fix some of the coefficients and optimize likelihood over the remaining coefficients. If the resulting likelihood is too low (as in the statement of the theorem), we can provably exclude that combination of coefficient values, since it cannot provide a MAP solution. The problem of optimizing likelihood over a subset of variables is easy (smooth, convex) so this screening test is easy to perform. As iterations continue, we aim to screen sparser and sparser conditions (fewer coefficients being fixed) which excludes larger parts of the search space.

Theorem 2 is a screening test. It states that if the score of the coefficients for a model is too high, we can excluded this model, it can never be the MAP solution, Further, if any subset of coefficients exceed a particular score, any model with that combination can be excluded.

These two theorems all dramatically help reduce the search space or the MAP solution.

Experiments

We conducted experiments on various datasets that cover use cases in the fields of medicine, finance, bioinformatics and life sciences. Haberman's Survival, BankNote, Yeast and Abalone are publicly available datasets from the UCI Machine Learning Repository. BWH is a dataset donated by a nearby hospital that contains patients' socioeconomic and medical conditions and whether a patient was readmitted to the hospital after being released. The "Domestic Abuse" and "Poor Condition" partitions of the dataset include the subset of records of patients that have reported that those conditions apply to them.

In our experimental setup, we set the strength of the prior (β) to 1, the probability distributions of selecting the $\mathcal{L}$ sets (γ) equal at $1/6$ for each set and the standard deviation for sampling the exponents to $1/2$. Before starting the MCMC steps, we initialized $\boldsymbol{\lambda}_t$ with the coefficients produced by logistic regression that were rounded to three significant digits, so each coefficient was initialized at $\mathcal{L}_6$ (see Table 1). At each iteration, we randomly selected a feature to propose a new covariate while keeping the rest fixed at their latest accepted values. We ran AIM for 100,000 iterations on 10 random train/test splits for all datasets.

Table 2 presents the experimental results from the classification accuracy perspective. For AIM, we report the classification accuracy both at the maximum posterior point, as well as the average accuracy after a burn-in period of 10,000 iterations. Each metric is then averaged over 10 train/test splits. The results demonstrate that AIM yields competitive generalization performance compared to logistic regression, which is what we were aiming for. Since our interpretable coefficient sets are much more constrained than logistic regression's, we can hope only to achieve competitive performance.

An example of the coefficients obtained by running logistic regression and AIM is presented in Table 5. For logistic regression, we show the coefficients rounded to four decimal points whereas the coefficients for AIM are their exact values at the maximum a posteriori solution. The coefficients are completely different, between AIM and logistic regression, which indicates that the optimization procedure is doing substantially more work than simple rounding. It also illustrates the "Rashomon Effect" in that a large class of approximately equally good models exist for this problem (including ones that are interpretable).

In Figure 1, we display the posterior values of accepted $\boldsymbol{\lambda}$'s for the Breast Cancer dataset, where we stratify by the number of nonzero coefficients in $\boldsymbol{\lambda}$. At $\beta = 1$, the accepted proposals include $\boldsymbol{\lambda}$'s that have all 10 coefficients being nonzero as well as $\boldsymbol{\lambda}$'s with one of the coefficients zero. The box plots indicate that on average, it is possible to achieve a posterior level with 9 features that is comparable to what can be achieved when all features are included in the model. At $\beta = 5$, the algorithm favors coefficients from $\mathcal{L}$ buckets with high interpretability since 0 is defined to be the most interpretable coefficient (according to $\mathcal{L}_1$). The set of accepted proposals now include $\boldsymbol{\lambda}$'s with two coefficients set to zero. In this setting, we similarly observe that the pos-

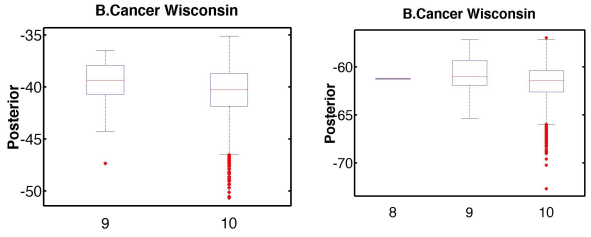
Feature Name	AIM	LR
Patient's Age	-1×10^{-4}	-0.0059
Operation Year	6×10^{-3}	-0.0154
Num. Nodes	-7.7×10^{-2}	-0.0924
Constant Term	9×10^{-1}	2.6671

Table 3: Haberman's Survival

Feature Name	AIM	LR	Feature Name	AIM	LR
Thickness	5.5×10^{-1}	0.5588	Bare Nuclei	3.2×10^{-1}	0.3865
Uniformity (Cell Size)	5×10^{-1}	0.4067	Bland Chromatin	5.7×10^{-1}	0.6029
Uniformity (Cell Shape)	6.89×10^{-1}	-0.18	Normal Nucleoli	9.88×10^{-2}	0.1624
Marginal Adhesion	2.2×10^{-3}	0.2836	Mitoses	-5.3×10^{-2}	0.419
Single Epithelial Cell Size	-8.5×10^{-3}	0.1223	*Constant Term*	-1.02×10^{1}	-10.12

Table 4: B. Cancer Wisconsin

Table 5: The coefficients of the features generated by AIM and logistic regression for Haberman's Survival and Breast Cancer Wisconsin datasets.



(a) $\beta = 1$ (b) $\beta = 5$

Figure 1: Breakdown of posterior distributions according to the number of nonzero coefficients in $\boldsymbol{\lambda}^t$ at prior weights $\beta = 1$ and $\beta = 5$.

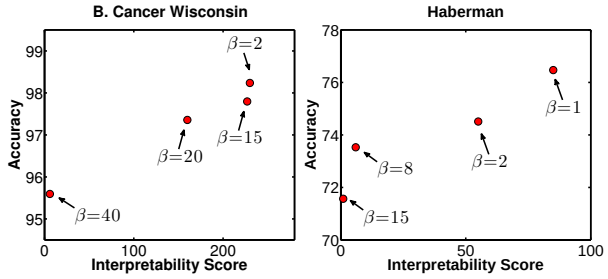


Figure 2: Accuracy and interpretability values obtained at different β settings for Breast Cancer and Haberman's Survival datasets.

teriors on average change very little across the number of non-zero features in the model.

In Figure 2, we show the effect of the weight of the prior on the accuracy vs. interpretability tradeoff. As shown, increasing the weight β improves the overall interpretability score of the model, and decreases accuracy accordingly.

The case for using data exponents

As we described earlier, our MCMC steps incorporate the "natural" scale of coefficients within the proposal distribution. We define the natural scale to be the value that brings the median point of the feature in the dataset to the $[0, 1]$ range. Scaling each coefficient in a way that is correlated with the features' distributions not only makes the coefficients more intuitive, but as we discuss next, helps AIM to stay in higher density exponent regions as well.

Figure 3 presents the proposed exponents for one of the features of the Haberman's Survival dataset. When the natural exponent from data is incorporated in the proposal condition, we see that the proposed exponents tend to mix well, by taking into account the scale/characteristics of the data. In particular, the trace tends not to drift into regions where the overall magnitude of coefficients becomes highly negative.

In the same figure, we see the behavior of the exponent search when the natural exponents are not included in the MCMC steps. As the proposal is free to move without any constraints, they tend to drift into the region that makes the coefficients' exponents highly negative. In this region, there are numerical problems, because the difference of the likelihoods $\log p(D|\boldsymbol{\lambda}^*) - \log p(D|\boldsymbol{\lambda}^t)$ or the transition probabilities $\log Q(\boldsymbol{\lambda}^t|\boldsymbol{\lambda}^*) - \log Q(\boldsymbol{\lambda}^*|\boldsymbol{\lambda}^t)$ (see Eq. 1) due to the transition $\boldsymbol{\lambda}^t \rightarrow \boldsymbol{\lambda}^*$ numerically become zero, leading to more *missteps* being accepted with probability $\exp(\log a)$ or as a result of very small changes in the prior. Consequently, if the proposals have no notion of a "realistic" scale of the coefficients, most of the iterations can be spent on regions that have no practical use towards achieving high accuracy and/or interpretability.

Figure 3: Proposed exponent values with and without using data exponents for one of the features of the Haberman's Survival dataset over the course of iterations.

Conclusions

Our goal in this paper is to design *scoring systems* that can be directly used, without rounding of coefficients or other manual manipulations that are typically done by physicians. In AIM, practitioners define and directly control the level of interpretability of the resulting models, by defining sets

of interpretability conditions and associating penalties for coefficients from each set, and then searching through the space of coefficients that jointly maximize the classification accuracy *and* minimize the penalties of the coefficients. The MCMC steps for this search leads AIM towards a set of coefficients that achieve a balance between accuracy and interpretability. Our empirical results demonstrate that AIM significantly improves the interpretability of the coefficients while maintaining competitive generalization performance on all datasets we tried.

Proofs

Proof. (Of Theorem 1)

$$
\begin{aligned}
&R_s\left(\boldsymbol{\lambda}_{\text{ref}}\right) \\
&< -\text{log likelihood}(\widetilde{\boldsymbol{\lambda}}^t) + \beta \log\left(1 + \sum_{j=1}^{\theta} \min_{k:\lambda_j^t \in \mathcal{L}_k} s_k\right) \\
&= \min_{\widetilde{\boldsymbol{\lambda}}_{\text{last}}^t \in \mathbb{R}^{P-Q}} -\text{log likelihood}([\lambda_1^t, \cdots, \lambda_\theta^t, \widetilde{\boldsymbol{\lambda}}_{\text{last}}^t]) \\
&\qquad + \beta \log\left(1 + \sum_{j=1}^{\theta} \min_{k:\tilde{\lambda}_{\text{last},j}^t \in \mathcal{L}_k} s_k + \sum_{\theta+1}^{P} 0\right) \\
&\leq -\text{log likelihood}([\lambda_1^t, \cdots, \lambda_\theta^t, \boldsymbol{\lambda}_{\text{last}}^{'}]) \\
&\qquad + \beta \log\left(1 + \sum_{j=1}^{\theta} \min_{k:\tilde{\lambda}_{\text{last},j}^t \in \mathcal{L}_k} s_k + \sum_{\theta+1}^{P} \min_{k,\lambda_{\text{last},j}^{'} \in \mathcal{L}_k} s_k\right) \\
&\qquad \text{for any } \boldsymbol{\lambda}_{\text{last}}^{'} \in \mathbb{R}^{P-Q} \\
&= R_s([\lambda_1^t, \cdots, \lambda_\theta^t, \boldsymbol{\lambda}_{\text{last}}^{'}]) \text{ for any } \boldsymbol{\lambda}_{\text{last}}^{'} \in \mathbb{R}^{P-Q}
\end{aligned}
$$

□

Proof. (Of Theorem 2)

$$
\begin{aligned}
R_s(f_{\text{opt}}) &= -\text{log likelihood} - \text{log prior} \\
&= \sum_i \log\left(1 + e^{-y_i \boldsymbol{\lambda}_{\text{opt}}^T \mathbf{x}_i}\right) + \beta \log\left(1 + \text{Score}(f_{\text{opt}})\right) \\
&\geq \beta \log(1 + \text{Score}(f)).
\end{aligned}
$$

Separately, we can use the empty model as a reference function.

$$
R_s(f_{\text{opt}}) \leq R_s(0) = \sum_i \left(\log(2)\right) + \beta \log(1 + s_{\{0\}} \times p).
$$

$$
\text{Score}(f_{\text{opt}}) \leq \exp\left[\frac{m \log 2 + \beta \log(1 + s_{\{0\}} \cdot p)}{\beta}\right] - 1.
$$

□

References

Andrade, J. T. 2009. Handbook of violence risk assessment and treatment: New approaches for mental health professionals.

Antman, E. M.; Cohen, M.; Bernink, P. J.; McCabe, C. H.; Horacek, T.; Papuchis, G.; Mautner, B.; Corbalan, R.; Radley, D.; and Braunwald, E. 2000. The timi risk score for unstable angina/non–st elevation mi: a method for prognostication and therapeutic decision making. *Jama* 284(7):835–842.

Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and Müller, K.-R. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research* 11:1803–1831.

Bien, J., and Tibshirani, R. 2011. Prototype selection for interpretable classification. *The Annals of Applied Statistics* 2403–2424.

Freitas, A. A.; Wieser, D. C.; and Apweiler, R. 2010. On the importance of comprehensible classification models for protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 7(1):172–182.

Freitas, A. A. 2014. Comprehensible classification models: A position paper. *SIGKDD Explorations Newsletter* 15(1):1–10.

Freund, Y., and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55(1):119–139.

Fung, G.; Sandilya, S.; and Rao, R. B. 2005. Rule extraction from linear support vector machines. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 32–40. ACM.

Gage, B. F.; Waterman, A. D.; Shannon, W.; Boechler, M.; Rich, M. W.; and Radford, M. J. 2001. Validation of clinical classification schemes for predicting stroke: results from the national registry of atrial fibrillation. *Jama* 285(22):2864–2870.

Giraud-Carrier, C. 1998. Beyond predictive accuracy: what? In *Proceedings of the ECML-98 Workshop on Upgrading Learning to Meta-Level: Model Selection and Data Transformation*, 78–85.

Hagan, M. T.; Demuth, H. B.; Beale, M. H.; et al. 1996. *Neural network design*. Pws Pub. Boston.

Knaus, W. A.; Zimmerman, J. E.; Wagner, D. P.; Draper, E. A.; and Lawrence, D. E. 1981. Apache-acute physiology and chronic health evaluation: a physiologically based classification system. *Critical care medicine* 9(8):591–597.

Kodratoff, Y. 1994. The comprehensibility manifesto. *KDD Nugget Newsletter* 94(9).

Kohavi, R. 1995. The power of decision tables. In *Machine Learning: ECML-95*. Springer. 174–189.

Le Gall, J.-R.; Loirat, P.; Alperovitch, A.; Glaser, P.; Granthil, C.; Mathieu, D.; Mercier, P.; Thomas, R.; and Villers, D. 1984. A simplified acute physiology score for icu patients. *Critical care medicine* 12(11):975–977.

Lima, E.; Mues, C.; and Baesens, B. 2009. Domain knowledge integration in data mining using decision tables: case studies in churn prediction. *Journal of the Operational Research Society* 60(8):1096–1106.

Martens, D.; Baesens, B.; and Van Gestel, T. 2009. Decompositional rule extraction from support vector machines by active learning. *IEEE Trans. on Knowl. and Data Eng.* 21(2):178–191.

Martens, D.; Baesens, B.; Gestel, T. V.; and Vanthienen, J. 2007. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research* 183(3):1466 – 1476.

Meinshausen, N. 2010. Node harvest. *The Annals of Applied Statistics* 2049–2072.

Pazzani, M. J. 2000. Knowledge discovery from data? *Intelligent systems and their applications, IEEE* 15(2):10–12.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning* 1(1):81–106.

Ridgeway, G. 2013. The pitfalls of prediction. *NIJ Journal* 271:34–40.

Rivest, R. L. 1987. Learning decision lists. *Machine learning* 2(3):229–246.

Siddiqi, N. 2005. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Wiley and SAS Business Series. Wiley.

Štrumbelj, E.; Kononenko, I.; and Robnik Šikonja, M. 2009. Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering* 68(10):886–904.

Ustun, B.; Tracà, S.; and Rudin, C. 2013. Supersparse linear integer models for predictive scoring systems. In *AAAI Late-Breaking Developments*.

Van Assche, A., and Blockeel, H. 2007. Seeing the forest through the trees: Learning a comprehensible model from an ensemble. In *Machine Learning: ECML 2007*. Springer. 418–429.

Vapnik, V. N., and Vapnik, V. 1998. *Statistical learning theory*, volume 2. Wiley New York.

Verbeke, W.; Martens, D.; Mues, C.; and Baesens, B. 2011. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications* 38(3):2354–2364.

Štrumbelj, E., and Kononenko, I. 2010. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research* 11:1–18.