Discrétisation, regroupement de modalités et introduction d'interactions en régression logistique

Adrien Ehrhardt[1,2]

Christophe Biernacki[2]
Philippe Heinrich[3]
Vincent Vandewalle[2,4]

[1]Crédit Agricole Consumer Finance
[2]Inria Lille - Nord-Europe
[3]Université de Lille, Paul Painlevé
[4]Université de Lille, EA2694

03/10/2018

Table of Contents

Context and basic notations

Current practice

Job	Home	Time in job	Family status	Wages		Repayment
Craftsman	Owner	20	Widower	2000		0
?	Renter	10	Common-law	1700		0
Licensed profes-sional	Starter	5	Divorced	4000		1
Executive	By work	8	Single	2700		1
Office employee	Renter	12	Married	1400		0
Worker	By family	2	?	1200		0

Table: Dataset with outliers and missing values.

Current practice

Job	Home	Time in job	Family status	Wages		Repayment
Craftsman	Owner	20	Widower	2000		0
?	Renter	10	Common-law	1700		0
Licensed professional	Starter	5	Divorced	4000		1
Executive	By work	8	Single	2700		1
Office employee	Renter	12	Married	1400		0
Worker	By family	2	?	1200		0

Table: Dataset with outliers and missing values.

1. Feature selection
2. Discretization / grouping
3. Interaction screening
4. Logistic regression fitting

Current practice

Job			Family status	Wages		Repayment
Craftsman			Widower	2000		0
?			Common-law	1700		0
Licensed profes- sional			Divorced	4000		1
Executive			Single	2700		1
Office employee			Married	1400		0
Worker			?	1200		0

Table: Dataset with outliers and missing values.

1. **Feature selection**
2. Discretization / grouping
3. Interaction screening
4. Logistic regression fitting

Current practice

Job			Family status	Wages			Repayment
Craftsman			Widower	]1500;2000]			0
?			Common-law	]1500;2000]			0
Licensed professional			Divorced	$]2000;\infty[$			1
Executive			Single	$]2000;\infty[$			1
Office employee			Married	$]-\infty ; 1500]$			0
Worker			?	$]-\infty ; 1500]$			0

Table: Dataset with outliers and missing values.

1. Feature selection
2. Discretization / grouping
3. Interaction screening
4. Logistic regression fitting

Current practice

Job			Family status	Wages		Repayment
?+Low-qualified			?+Alone	]1500;2000]		0
?+Low-qualified			Union	]1500;2000]		0
High-qualified			?+Alone	]2000;∞[		1
High-qualified			?+Alone	]2000;∞[		1
?+Low-qualified			Union	]-∞ ; 1500]		0
?+Low-qualified			?+Alone	]-∞ ; 1500]		0

Table: Dataset with outliers and missing values.

1. Feature selection
2. Discretization / **grouping**
3. Interaction screening
4. Logistic regression fitting

Current practice

Job			Family status x Wages		Repayment
?+Low-qualified			?+Alone x]1500;2000]		0
?+Low-qualified			Union x]1500;2000]		0
High-qualified			?+Alone x]2000;∞[		1
High-qualified			?+Alone x]2000;∞[		1
?+Low-qualified			Union x]-∞ ; 1500]		0
?+Low-qualified			?+Alone x]-∞ ; 1500]		0

Table: Dataset with outliers and missing values.

1. Feature selection
2. Discretization / grouping
3. **Interaction screening**
4. Logistic regression fitting

Current practice

Job			Family status x Wages	Score	Repayment
?+Low-qualified			?+Alone x]1500;2000]	225	0
?+Low-qualified			Union x]1500;2000]	190	0
High-qualified			?+Alone x]2000;∞[	218	1
High-qualified			?+Alone x]2000;∞[	202	1
?+Low-qualified			Union x]-∞ ; 1500]	205	0
?+Low-qualified			?+Alone x]-∞ ; 1500]	192	0

Table: Dataset with outliers and missing values.

1. Feature selection
2. Discretization / grouping
3. Interaction screening
4. Logistic regression fitting

Current practice

Feature	Level	Points
Age	18-25	10
	25-45	20
	45-$+\infty$	30
Wages	$-\infty$-1000	15
	1000-2000	25
	2000-$+\infty$	35
...	...	...
Glucose level	...	...

Table: Final scorecard.

Mathematical reinterpretation

The whole process can be decomposed into two steps:

$$\begin{aligned} \mathcal{X} &\to \mathcal{E} && \to \mathcal{Y} \\ \boldsymbol{x} &\mapsto \boldsymbol{e} = \boldsymbol{f}(\boldsymbol{x}) && \mapsto y \end{aligned}$$

Mathematical reinterpretation

The whole process can be decomposed into two steps:

$$
\begin{aligned}
\mathcal{X} &\to \mathcal{E} &&\to \mathcal{Y} \\
\boldsymbol{x} &\mapsto \boldsymbol{e} = \boldsymbol{f}(\boldsymbol{x}) &&\mapsto y
\end{aligned}
$$

Selected features: $\boldsymbol{x} = (x_j)_1^d$ (continuous or categorical).

Mathematical reinterpretation

The whole process can be decomposed into two steps:

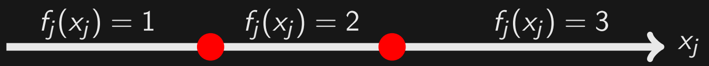
$$\begin{aligned} \mathcal{X} &\to \mathcal{E} && \to \mathcal{Y} \\ \boldsymbol{x} &\mapsto \boldsymbol{e} = \boldsymbol{f}(\boldsymbol{x}) && \mapsto y \end{aligned}$$

Selected features: $\boldsymbol{x} = (x_j)_1^d$ (continuous or categorical).

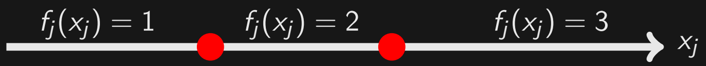
$\boldsymbol{f}$ is "component-wise", *i.e.* $\boldsymbol{f}(\boldsymbol{x}) = (f_j(x_j))_1^d$.

We restrict to discretization and grouping of factor levels.

Mathematical reinterpretation: Feature Engineering



Mathematical reinterpretation: Feature Engineering



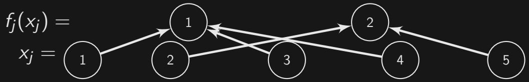
Discretization

Into m intervals with associated cutpoints
$\boldsymbol{c} = (c_0 = -\infty, c_1, \ldots, c_{m-1}, c_m = +\infty)$.

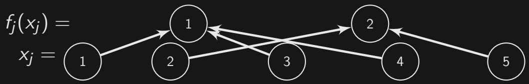
Discretization function

$$
\begin{aligned}
f_j(\cdot; \boldsymbol{c}, m) \colon \mathbb{R} &\to \{1, \ldots, m\} \\
x &\mapsto \sum_{k=1}^{m} k \, \mathbb{1}_{]c_{k-1}; c_k]}(x)
\end{aligned}
$$

Mathematical reinterpretation: Feature Engineering



Mathematical reinterpretation: Feature Engineering



Grouping

Grouping o values into m, $m \leq o$.

Grouping function

$f_j : \{1, \dots, o\} \to \{1, \dots, m\}$

f_j surjective: it defines a partition of $\{1, \dots, o\}$ in m elements.

Mathematical reinterpretation: Engineered Feature Space

Discretization

$$f_j \in \mathcal{M}_j = \{f_j(\cdot; \boldsymbol{c}_j, m_j) | m_j \in \mathbb{N}, c_{j,1} < \ldots < c_{j,m_j-1}\}$$

Mathematical reinterpretation: Engineered Feature Space

Discretization

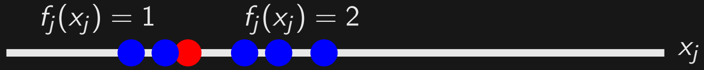
$$f_j \in \mathcal{M}_j = \{f_j(\cdot; \boldsymbol{c}_j, m_j) | m_j \in \mathbb{N}, c_{j,1} < \ldots < c_{j,m_j-1}\}$$

$\mathcal{M}_j$ is seemingly continuous but with a finite sample, a countable Feature Space can be recovered by remarking:

Discretization

$$f_j \in \mathcal{M}_j = \{f_j(\cdot; \boldsymbol{c}_j, m_j) | m_j \in \mathbb{N}, c_{j,1} < \ldots < c_{j,m_j-1}\}$$

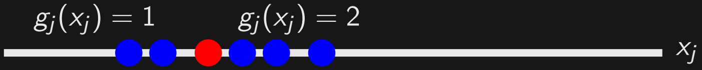
$\mathcal{M}_j$ is seemingly continuous but with a finite sample, a countable Feature Space can be recovered by remarking:



Discretization

$$f_j \in \mathcal{M}_j = \{f_j(\cdot; \boldsymbol{c}_j, m_j) | m_j \in \mathbb{N}, c_{j,1} < \ldots < c_{j,m_j-1}\}$$

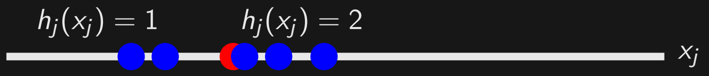
$\mathcal{M}_j$ is seemingly continuous but with a finite sample, a countable Feature Space can be recovered by remarking:



Discretization

$$f_j \in \mathcal{M}_j = \{f_j(\cdot; \boldsymbol{c}_j, m_j) | m_j \in \mathbb{N}, c_{j,1} < \ldots < c_{j,m_j - 1}\}$$

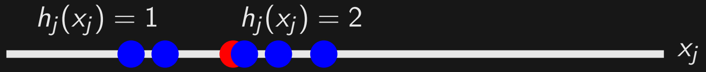
$\mathcal{M}_j$ is seemingly continuous but with a finite sample, a countable Feature Space can be recovered by remarking:



Discretization

$$f_j \in \mathcal{M}_j = \{f_j(\cdot; \boldsymbol{c}_j, m_j) | m_j \in \mathbb{N}, c_{j,1} < \ldots < c_{j,m_j-1}\}$$

$\mathcal{M}_j$ is seemingly continuous but with a finite sample, a countable Feature Space can be recovered by remarking:



Example ($n = 20$, $d = 10$): $\approx 10^{57}$ models in $\mathcal{M}_j^d$.

Grouping

$f_j \in \mathcal{M}_j = \{\text{Partitions from } \{1, \ldots, o_j\} \text{ to } \{1, \ldots, m_j\}; m_j \leq o_j\}.$

Mathematical reinterpretation: Engineered Feature Space

Grouping

$f_j \in \mathcal{M}_j = \{\text{Partitions from } \{1, \dots, o_j\} \text{ to } \{1, \dots, m_j\}; m_j \leq o_j\}$.

Its cardinality is given by the Stirling number of the second kind:
$|\mathcal{M}_j| = \sum_{m_j=1}^{o_j} \frac{1}{m_j!} \sum_{i=0}^{m_j} (-1)^{m_j - i} \binom{m_j}{i} i^{o_j}$.

Grouping

$f_j \in \mathcal{M}_j = \{\text{Partitions from } \{1, \ldots, o_j\} \text{ to } \{1, \ldots, m_j\}; m_j \leq o_j\}$.

Its cardinality is given by the Stirling number of the second kind: $|\mathcal{M}_j| = \sum_{m_j=1}^{o_j} \frac{1}{m_j!} \sum_{i=0}^{m_j} (-1)^{m_j-i} \binom{m_j}{i} i^{o_j}$.

Exhaustive search is untractable.

Mathematical reinterpretation: Objective

Target feature $y \in \{0, 1\}$ must be predicted given engineered features $\boldsymbol{f}(\boldsymbol{x}) = (f_j(x_j))_1^d$.

Mathematical reinterpretation: Objective

Target feature $y \in \{0, 1\}$ must be predicted given engineered features $\boldsymbol{f}(\boldsymbol{x}) = (f_j(x_j))_1^d$.

On "raw" data, logistic regression yields:

$$\text{logit}(p_{\boldsymbol{\theta}_{\text{raw}}}(1|\boldsymbol{x})) = \theta_0 + \sum_{j \text{ cont.}} \theta_j x_j + \sum_{j \text{ cat.}} \theta_j^{x_j}$$

Mathematical reinterpretation: Objective

Target feature $y \in \{0, 1\}$ must be predicted given engineered features $\boldsymbol{f}(\boldsymbol{x}) = (f_j(x_j))_1^d$.

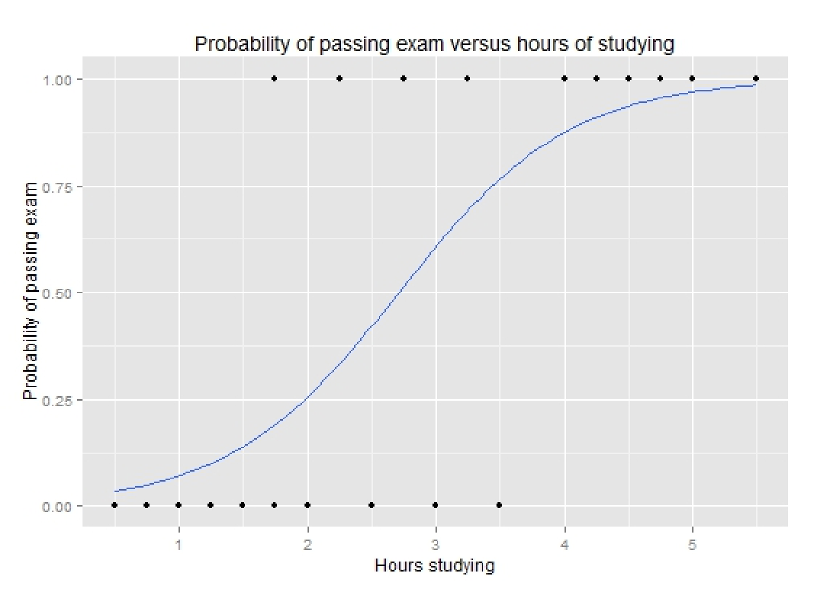
On "raw" data, logistic regression yields:

$$\text{logit}(p_{\boldsymbol{\theta}_{\text{raw}}}(1|\boldsymbol{x})) = \theta_0 + \sum_{j \text{ cont.}} \theta_j x_j + \sum_{j \text{ cat.}} \theta_j^{x_j}$$

On discretized / grouped data, logistic regression yields:

$$\text{logit}(p_{\boldsymbol{\theta}_{\boldsymbol{f}}}(1|\boldsymbol{f}(\boldsymbol{x}))) = \theta_0 + \sum_{j=1}^{d} \theta_j^{f_j(x_j)}$$

Mathematical reinterpretation: Objective



Example

True data

$$\text{logit}(p_{\text{true}}(1|\boldsymbol{x})) = \ln\left(\frac{p_{\text{true}}(1|\boldsymbol{x})}{1 - p_{\text{true}}(1|\boldsymbol{x})}\right) = \sin((x_1 - 0.7) \times 7)$$

Figure: True relationship between predictor and outcome

Example

Logistic regression on "raw" data:

$$\text{logit}(p_{\theta_{\text{raw}}}(1|\boldsymbol{x})) = \theta_0 + \theta_1 x_1$$

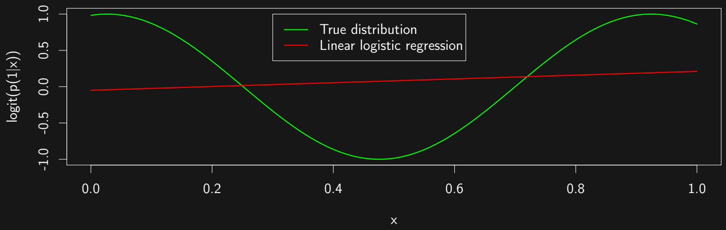


Figure: Linear logistic regression fit

Example

Logistic regression on discretized data:

If $\boldsymbol{f}$ is not carefully chosen ...

$$\text{logit}(p_{\theta_{\boldsymbol{f}}}(1|\boldsymbol{f}(\boldsymbol{x}))) = \theta_0 + \underbrace{\theta_1^{f_1(x_1)}}_{\theta_1^1,\ldots,\theta_1^{50}}$$

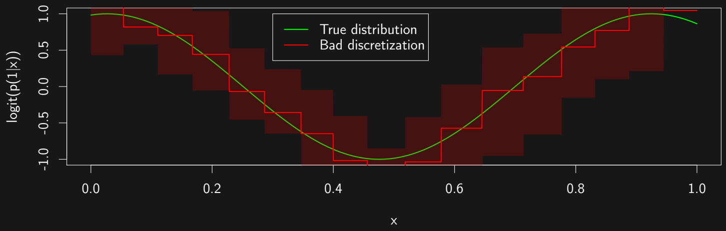


Figure: Bad (high variance) discretization

Example

Logistic regression on discretized data:

If $\boldsymbol{f}$ is carefully chosen ...

$$\operatorname{logit}(p_{\boldsymbol{\theta_f}}(1|\boldsymbol{f}(\boldsymbol{x}))) = \theta_0 + \underbrace{\theta_1^{f_1(x_1)}}_{\theta_1^1,\dots,\theta_1^3}$$

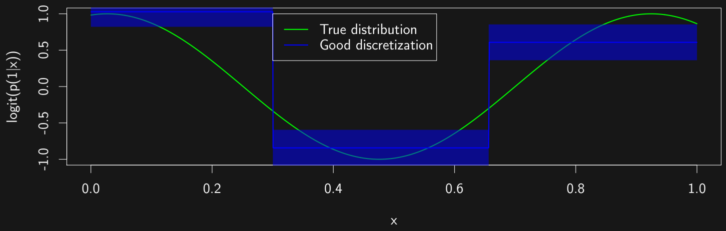


Figure: Good (bias/variance tradeoff) discretization

Criterion

$\boldsymbol{\theta}$ can be estimated for each discretization $\boldsymbol{f}$ and $\boldsymbol{f}^{\star}$ can be chosen through our favorite model choice criterion *e.g.* BIC.

Criterion

$\boldsymbol{\theta}$ can be estimated for each discretization $\boldsymbol{f}$ and $\boldsymbol{f}^\star$ can be chosen through our favorite model choice criterion *e.g.* BIC.

A model selection problem

$$(\boldsymbol{f}^\star, \boldsymbol{\theta}^\star) = \underset{\boldsymbol{f} \in \mathcal{F}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{f}}}{\operatorname{argmin}} -2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{f}(\boldsymbol{x}_i)) + |\boldsymbol{\theta}| \times \ln(n),$$

where $\boldsymbol{\theta}$ is classicaly estimated *via* MLE.

Criterion

$\boldsymbol{\theta}$ can be estimated for each discretization $\boldsymbol{f}$ and $\boldsymbol{f}^\star$ can be chosen through our favorite model choice criterion *e.g.* BIC.

A model selection problem

$$(\boldsymbol{f}^\star, \boldsymbol{\theta}^\star) = \underset{\boldsymbol{f} \in \mathcal{F}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{f}}}{\operatorname{argmin}} -2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{f}(\boldsymbol{x}_i)) + |\boldsymbol{\theta}| \times \ln(n),$$

where $\boldsymbol{\theta}$ is classicaly estimated *via* MLE.

Compromise between (over-)fitting the data and model complexity (and explainability in a sense!).

Criterion

$\boldsymbol{\theta}$ can be estimated for each discretization $\boldsymbol{f}$ and $\boldsymbol{f}^\star$ can be chosen through our favorite model choice criterion *e.g.* BIC.

A model selection problem

$$(\boldsymbol{f}^\star, \boldsymbol{\theta}^\star) = \underset{\boldsymbol{f} \in \mathcal{F}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{f}}}{\operatorname{argmin}} -2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{f}(\boldsymbol{x}_i)) + |\boldsymbol{\theta}| \times \ln(n),$$

where $\boldsymbol{\theta}$ is classicaly estimated *via* MLE.

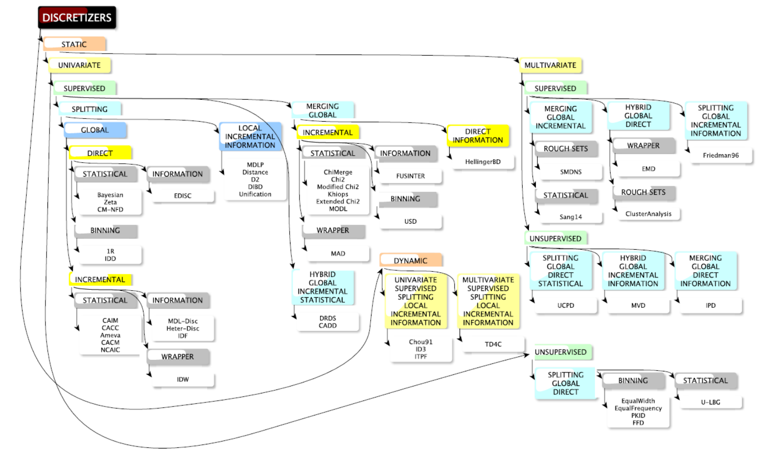
Compromise between (over-)fitting the data and model complexity (and explainability in a sense!).

$\mathcal{F}$ is discrete and combinatorial: how can we get around this problem?

State-of-the art

Current academic methods:

A lot of existing heuristics, see [Ramírez-Gallego et al., 2016]:



Quick example of χ^2:

Category	# samples	# cases	p-value
18-20	10	5	0.3
20-22	10	6	0.2
22-24	10	4	
...	...	...	...

Supervised multivariate discretization and factor levels grouping

Mathematical formalization

Discretized / grouped x_j denoted by e_j has been seen up to now as the result of a function of x_j:

$$e_j = f_j(x_j).$$

Mathematical formalization

Discretized / grouped x_j denoted by e_j has been seen up to now as the result of a function of x_j:

$$e_j = f_j(x_j).$$

Discretization / grouping e_j can be seen as a latent random variable for which

$$p(e_j|x_j) = \mathbb{1}_{e_j}(f_j(x_j)).$$

Mathematical formalization

Discretized / grouped x_j denoted by e_j has been seen up to now as the result of a function of x_j:

$$e_j = f_j(x_j).$$

Discretization / grouping e_j can be seen as a latent random variable for which

$$p(e_j|x_j) = \underbrace{\mathbb{1}_{e_j}(f_j(x_j))}_{\substack{\text{Heaviside-like function} \\ \text{difficult to optimize}}} .$$

Discretized / grouped x_j denoted by e_j has been seen up to now as the result of a function of x_j:

$$e_j = f_j(x_j).$$

Discretization / grouping e_j can be seen as a latent random variable for which

$$p(e_j|x_j) = \underbrace{\mathbb{1}_{e_j}(f_j(x_j))}_{\substack{\text{Heaviside-like function} \\ \text{difficult to optimize}}}.$$

Suppose for now that $\boldsymbol{m} = (m_j)_1^d$ is fixed.

Mathematical formalization

Discretized / grouped x_j denoted by e_j has been seen up to now as the result of a function of x_j:

$$e_j = f_j(x_j).$$

Discretization / grouping e_j can be seen as a latent random variable for which

$$p(e_j|x_j) = \underbrace{\mathbb{1}_{e_j}(f_j(x_j))}_{\substack{\text{Heaviside-like function} \\ \text{difficult to optimize}}}.$$

Suppose for now that $\boldsymbol{m} = (m_j)_1^d$ is fixed.

$$\boldsymbol{e} \in \boldsymbol{\mathcal{E}}_{\boldsymbol{m}} = \{1, \ldots, m_1\} \times \ldots \times \ldots \times \{1, \ldots, m_d\}.$$

First set of hypotheses

H1: implicit hypothesis of every discretization:

Predictive information about y in $\boldsymbol{x}$ is "squeezed" in $\boldsymbol{e}$, i.e. $p_{\text{true}}(y|\boldsymbol{x}, \boldsymbol{e}) = p_{\text{true}}(y|\boldsymbol{e})$.

First set of hypotheses

H1: implicit hypothesis of every discretization:

Predictive information about y in $\boldsymbol{x}$ is "squeezed" in $\boldsymbol{e}$, i.e. $p_{\text{true}}(y|\boldsymbol{x}, \boldsymbol{e}) = p_{\text{true}}(y|\boldsymbol{e})$.

H2: conditional independence:

Conditional independence of $e_j|x_j$ with other features $x_k, k \neq j$.

First set of hypotheses

H1: implicit hypothesis of every discretization:

Predictive information about y in $\boldsymbol{x}$ is "squeezed" in $\boldsymbol{e}$, i.e. $p_{\text{true}}(y|\boldsymbol{x}, \boldsymbol{e}) = p_{\text{true}}(y|\boldsymbol{e})$.

H2: conditional independence:

Conditional independence of $e_j|x_j$ with other features x_k, $k \neq j$.

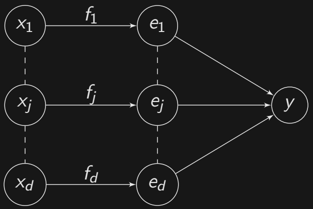


Figure: Dependance structure between x_j,e_j and y

Proposal: continuous relaxation

H3: link between x_j and e_j:

Proposal: continuous relaxation

H3: link between x_j and e_j:
Continuous relaxation of a discrete problem (cf neural nets)

Continuous features: relaxation of the "hard" discretization

Link between e_j and x_j is supposed to be polytomous logistic:

$$p_{\alpha_j}(e_j|x_j).$$

Proposal: continuous relaxation

H3: link between x_j and e_j:
Continuous relaxation of a discrete problem (cf neural nets)

Continuous features: relaxation of the "hard" discretization

Link between e_j and x_j is supposed to be polytomous logistic:

$$p_{\boldsymbol{\alpha}_j}(e_j|x_j).$$

Categorical features: relaxation of the grouping problem

A simple contingency table is used:

$$p_{\boldsymbol{\alpha}_j}(e_j = k|x_j = \ell) = \alpha_j^{k,\ell}.$$

Intuitions about how it works: model proposal

$$
\begin{aligned}
p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p(y|\boldsymbol{x}, \boldsymbol{e}) p(\boldsymbol{e}|\boldsymbol{x}) \\
&= \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p(y|\boldsymbol{e}) \prod_{j=1}^{d} p(e_j|x_j) \\
&= \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} \underbrace{p_{\boldsymbol{\theta_e}}(y|\boldsymbol{e})}_{\text{logistic}} \prod_{j=1}^{d} \underbrace{p_{\alpha_j}(e_j|x_j)}_{\text{logistic or table}} \\
&\approx p_{\boldsymbol{\theta}^\star}(y|\boldsymbol{e}^\star)
\end{aligned}
$$

Intuitions about how it works: model proposal

$$
\begin{aligned}
p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p(y|\boldsymbol{x}, \boldsymbol{e}) p(\boldsymbol{e}|\boldsymbol{x}) \\
&= \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p(y|\boldsymbol{e}) \prod_{j=1}^{d} p(e_j|x_j) \\
&= \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} \underbrace{p_{\boldsymbol{\theta}_{\boldsymbol{e}}}(y|\boldsymbol{e})}_{\text{logistic}} \prod_{j=1}^{d} \underbrace{p_{\boldsymbol{\alpha}_j}(e_j|x_j)}_{\text{logistic or table}} \\
&\approx p_{\boldsymbol{\theta}^\star}(y|\boldsymbol{e}^\star)
\end{aligned}
$$

Subsequently, it is equivalent to "optimize" $p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha})$.

Intuitions about how it works: model proposal

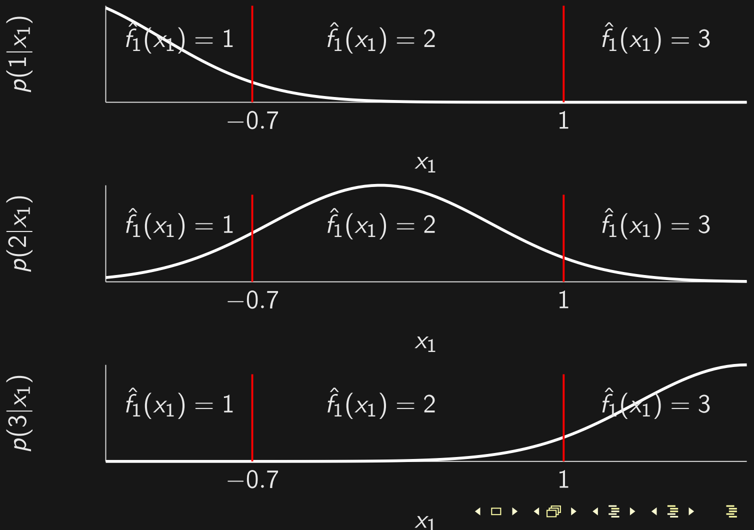
$$
\begin{aligned}
p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) &= \sum_{\boldsymbol{e} \in \mathcal{E}_{\boldsymbol{m}}} p(y|\boldsymbol{x}, \boldsymbol{e}) p(\boldsymbol{e}|\boldsymbol{x}) \\
&= \sum_{\boldsymbol{e} \in \mathcal{E}_{\boldsymbol{m}}} p(y|\boldsymbol{e}) \prod_{j=1}^{d} p(e_j|x_j) \\
&= \sum_{\boldsymbol{e} \in \mathcal{E}_{\boldsymbol{m}}} \underbrace{p_{\boldsymbol{\theta}_{\boldsymbol{e}}}(y|\boldsymbol{e})}_{\text{logistic}} \prod_{j=1}^{d} \underbrace{p_{\alpha_j}(e_j|x_j)}_{\text{logistic or table}} \\
&\approx p_{\boldsymbol{\theta}^\star}(y|\boldsymbol{e}^\star)
\end{aligned}
$$

Subsequently, it is equivalent to "optimize" $p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha})$.

$$
\max_{\boldsymbol{\theta}, \boldsymbol{e}} p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \simeq \max_{\boldsymbol{\theta}, \boldsymbol{\alpha}} p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha})
$$

Go back to “hard” thresholding: MAP estimation

$$\hat{f}_j(x_j) = \underset{1 \le k \le m_j}{\operatorname{argmax}} \, p_{\alpha_j}(k|x_j)$$



Estimation of the proposed model

Two very different estimation strategies

Two very different estimation strategies

1. In the statistics community: latent feature = EM-like algorithm.

We try to get $\max_{\boldsymbol{\theta},\boldsymbol{\alpha}} p(y|\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{\alpha})$ through SEM algorithm + Gibbs sampling step that explicity draws $\boldsymbol{e}$.

Estimation of the proposed model

Two very different estimation strategies

1. In the statistics community: latent feature = EM-like algorithm.

We try to get $\max_{\boldsymbol{\theta},\boldsymbol{\alpha}} p(y|\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{\alpha})$ through SEM algorithm + Gibbs sampling step that explicity draws $\boldsymbol{e}$.

2. Machine Learning: neural networks natively learn representations of the data.

A 1-hidden layer neural network with softmax activation function that *via* Stochastic Gradient Descent tries to maximize the likelihood of $p_{\boldsymbol{\theta}}(y|\tilde{e} = (p_{\boldsymbol{\alpha}_j}(1|x_j), \ldots, p_{\boldsymbol{\alpha}_j}(m_j|x_j))_1^d)$.

Estimation *via* SEM

“Classical” estimation strategy with latent variables: EM algorithm.

Estimation *via* SEM

"Classical" estimation strategy with latent variables: EM algorithm.

There would still be a sum over $\boldsymbol{\mathcal{E}_m}$:
$p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\alpha_j}(e_j|x_j)$

Estimation *via* SEM

"Classical" estimation strategy with latent variables: EM algorithm.

There would still be a sum over $\boldsymbol{\mathcal{E}_m}$:
$p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\alpha_j}(e_j|x_j)$

Use a Stochastic-EM! Draw $\boldsymbol{e}$ knowing that:

Estimation *via* SEM

"Classical" estimation strategy with latent variables: EM algorithm.

There would still be a sum over $\boldsymbol{\mathcal{E}_m}$:
$p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\alpha_j}(e_j|x_j)$

Use a Stochastic-EM! Draw $\boldsymbol{e}$ knowing that:

$$p(\boldsymbol{e}|\boldsymbol{x}, y) = \frac{p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\alpha_j}(e_j|x_j)}{\underbrace{\sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\alpha_j}(e_j|x_j)}_{\text{still difficult to calculate}}}$$

Estimation *via* SEM

"Classical" estimation strategy with latent variables: EM algorithm.

There would still be a sum over $\boldsymbol{\mathcal{E}_m}$:
$p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\alpha_j}(e_j|x_j)$

Use a Stochastic-EM! Draw $\boldsymbol{e}$ knowing that:

$$p(\boldsymbol{e}|\boldsymbol{x}, y) = \frac{p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\boldsymbol{\alpha_j}}(e_j|x_j)}{\underbrace{\sum_{\boldsymbol{e} \in \boldsymbol{\mathcal{E}_m}} p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) \prod_{j=1}^{d} p_{\alpha_j}(e_j|x_j)}_{\text{still difficult to calculate}}}$$

Gibbs-sampling step:

$$p(e_j|\boldsymbol{x}, y, \boldsymbol{e}_{\{-j\}}) \propto p_{\boldsymbol{\theta}}(y|\boldsymbol{e}) p_{\boldsymbol{\alpha_j}}(e_j|x_j)$$

Algorithm

Initialization

$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{pmatrix} \overset{\text{at random}}{\Rightarrow} \begin{pmatrix} e_{1,1} & \cdots & e_{1,d} \\ \vdots & \vdots & \vdots \\ e_{n,1} & \cdots & e_{n,d} \end{pmatrix}$$

Loop

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \overset{\substack{\text{logistic} \\ \text{regression}}}{\Rightarrow} \begin{pmatrix} e_{1,1} & \cdots & e_{1,d} \\ \vdots & \vdots & \vdots \\ e_{n,1} & \cdots & e_{n,d} \end{pmatrix} \overset{\substack{\text{polytomous} \\ \text{regression}}}{\Rightarrow} \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{pmatrix}$$

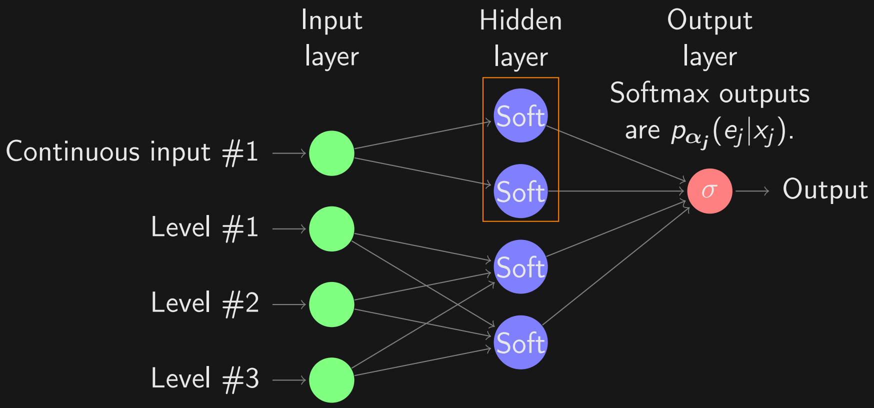
Updating e

$$\begin{pmatrix} p(y_1, e_{1,j} = k|\mathbf{x}_i) \\ \vdots \\ p(y_n, e_{n,j} = k|\mathbf{x}_i) \end{pmatrix} \overset{\substack{\text{random} \\ \text{sampling}}}{\Rightarrow} \begin{pmatrix} e_{1,j} \\ \vdots \\ e_{n,j} \end{pmatrix}$$

Calculating e_{MAP}

$$\begin{pmatrix} \hat{f}_j(x_{1,j}) \\ \vdots \\ \hat{f}_j(x_{n,j}) \end{pmatrix} \overset{\substack{\text{MAP} \\ \text{estimate}}}{=} \begin{pmatrix} \operatorname{argmax}_{e_j} p_{\boldsymbol{\alpha}_j}(e_j|x_{1,j}) \\ \vdots \\ \operatorname{argmax}_{e_j} p_{\boldsymbol{\alpha}_j}(e_j|x_{n,j}) \end{pmatrix}$$

Estimation *via* neural nets



In the end: the best discretization

New model selection criterion

We have drastically restricted the search space to provably clever candidates $\hat{\boldsymbol{f}}^{(1)}, \dots, \hat{\boldsymbol{f}}^{(\text{iter})}$ resulting either from the Gibbs sampling or the neural network and MAP estimation.

$$(\boldsymbol{f}^\star, \theta^\star) = \underset{\hat{\boldsymbol{f}} \in \{\hat{\boldsymbol{f}}^{(1)}, \dots, \hat{\boldsymbol{f}}^{(\text{iter})}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} -2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \hat{\boldsymbol{f}}(\boldsymbol{x}_i)) + (m_1 \times \cdots \times m_d - d) \times \ln(n)$$

In the end: the best discretization

New model selection criterion

We have drastically restricted the search space to provably clever candidates $\hat{\boldsymbol{f}}^{(1)}, \ldots, \hat{\boldsymbol{f}}^{(\text{iter})}$ resulting either from the Gibbs sampling or the neural network and MAP estimation.

$$(\boldsymbol{f}^\star, \boldsymbol{\theta}^\star) = \underset{\hat{\boldsymbol{f}} \in \{\hat{\boldsymbol{f}}^{(1)}, \ldots, \hat{\boldsymbol{f}}^{(\text{iter})}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} -2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \hat{\boldsymbol{f}}(\boldsymbol{x}_i)) + (m_1 \times \cdots \times m_d - d) \times \ln(n)$$

We would still need to loop over candidates $\boldsymbol{m}$!

In the end: the best discretization

New model selection criterion

We have drastically restricted the search space to provably clever candidates $\hat{\boldsymbol{f}}^{(1)}, \ldots, \hat{\boldsymbol{f}}^{(\text{iter})}$ resulting either from the Gibbs sampling or the neural network and MAP estimation.

$$(\boldsymbol{f}^\star, \boldsymbol{\theta}^\star) = \underset{\hat{\boldsymbol{f}} \in \{\hat{\boldsymbol{f}}^{(1)}, \ldots, \hat{\boldsymbol{f}}^{(\text{iter})}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} -2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \hat{\boldsymbol{f}}(\boldsymbol{x}_i)) + (m_1 \times \cdots \times m_d - d) \times \ln(n)$$

We would still need to loop over candidates $\boldsymbol{m}$!

In practice if $\forall i, \; p(e_{i,j} = 1 | x_{i,j}, y_i) \ll 1$, then $e_j = 1$ disappears. . .

In the end: the best discretization

New model selection criterion

We have drastically restricted the search space to provably clever candidates $\hat{\boldsymbol{f}}^{(1)}, \ldots, \hat{\boldsymbol{f}}^{(\text{iter})}$ resulting either from the Gibbs sampling or the neural network and MAP estimation.

$$(\boldsymbol{f}^\star, \boldsymbol{\theta}^\star) = \underset{\hat{\boldsymbol{f}} \in \{\hat{\boldsymbol{f}}^{(1)}, \ldots, \hat{\boldsymbol{f}}^{(\text{iter})}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} -2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \hat{\boldsymbol{f}}(\boldsymbol{x}_i)) + (m_1 \times \cdots \times m_d - d) \times \ln(n)$$

We would still need to loop over candidates $\boldsymbol{m}$!

In practice if $\forall i,\ p(e_{i,j} = 1 | x_{i,j}, y_i) \ll 1$, then $e_j = 1$ disappears...

Start with $\boldsymbol{m} = (m_{\max})_1^d$ and "wait" ... eventually until $\boldsymbol{m} = \boldsymbol{1}$.

Selecting interactions in logistic regression

Notations

Upper triangular matrix with $\delta_{k,\ell} = 1$ if $k < \ell$ and features p and q "interact" in the logistic regression.

$$\text{logit}(p_{\boldsymbol{\theta_f}}(1|\boldsymbol{f}(\boldsymbol{x}))) = \theta_0 + \sum_{j=1}^{d} \theta_j^{f_j(x_j)} + \sum_{1 \leq k < \ell \leq d} \delta_{k,\ell} \theta_{k,\ell}^{f_k(x_k) f_\ell(x_\ell)}$$

Notations

Upper triangular matrix with $\delta_{k,\ell} = 1$ if $k < \ell$ and features p and q "interact" in the logistic regression.

$$\text{logit}(p_{\boldsymbol{\theta_f}}(1|\boldsymbol{f}(\boldsymbol{x}))) = \theta_0 + \sum_{j=1}^{d} \theta_j^{f_j(x_j)} + \sum_{1 \leq k < \ell \leq d} \delta_{k,\ell} \theta_{k,\ell}^{f_k(x_k) f_\ell(x_\ell)}$$

Imagine for now that the discretization $\boldsymbol{f}(\boldsymbol{x})$ is fixed. The criterion becomes:

$$(\boldsymbol{\theta}^\star, \boldsymbol{\delta}^\star) = \underset{\boldsymbol{\theta}, \boldsymbol{\delta} \in \{0,1\}^{\frac{d(d-1)}{2}}}{\text{argmin}} \underbrace{-2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{\delta}) + |\boldsymbol{\theta}| \ln(n)}_{\text{BIC}[\boldsymbol{\delta}]}$$

Notations

Upper triangular matrix with $\delta_{k,\ell} = 1$ if $k < \ell$ and features p and q "interact" in the logistic regression.

$$\text{logit}(p_{\boldsymbol{\theta}_f}(1|\boldsymbol{f}(\boldsymbol{x}))) = \theta_0 + \sum_{j=1}^{d} \theta_j^{f_j(x_j)} + \sum_{1 \leq k < \ell \leq d} \delta_{k,\ell} \theta_{k,\ell}^{f_k(x_k) f_\ell(x_\ell)}$$

Imagine for now that the discretization $\boldsymbol{f}(\boldsymbol{x})$ is fixed. The criterion becomes:

$$(\boldsymbol{\theta}^\star, \boldsymbol{\delta}^\star) = \underset{\boldsymbol{\theta}, \boldsymbol{\delta} \in \{0,1\}^{\frac{d(d-1)}{2}}}{\text{argmin}} \underbrace{-2 \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{\delta}) + |\boldsymbol{\theta}| \ln(n)}_{\text{BIC}[\boldsymbol{\delta}]}$$

Analogous to previous problem: $2^{\frac{d(d-1)}{2}}$ models.

Model proposal

δ is latent and hard to optimize over: use a stochastic algorithm!

Model proposal

δ is latent and hard to optimize over: use a stochastic algorithm!

Strategy used here: Metropolis-Hastings algorithm.

Model proposal

δ is latent and hard to optimize over: use a stochastic algorithm!

Strategy used here: Metropolis-Hastings algorithm.

$$
\begin{aligned}
p(y|\boldsymbol{e}) &= \sum_{\boldsymbol{\delta} \in \{0,1\}^{\frac{d(d-1)}{2}}} p(y|\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\delta})p(\boldsymbol{\delta}) \\
p(\boldsymbol{\delta}|\boldsymbol{f}(\boldsymbol{x}), y) &\propto p(y|\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\delta})p(\boldsymbol{\delta}) \\
&\approx \exp(-\mathrm{BIC}[\boldsymbol{\delta}]/2)p(\boldsymbol{\delta})
\end{aligned}
$$

Model proposal

δ is latent and hard to optimize over: use a stochastic algorithm!

Strategy used here: Metropolis-Hastings algorithm.

$$p(y|\boldsymbol{e}) = \sum_{\boldsymbol{\delta} \in \{0,1\}^{\frac{d(d-1)}{2}}} p(y|\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\delta}) p(\boldsymbol{\delta})$$

$$p(\boldsymbol{\delta}|\boldsymbol{f}(\boldsymbol{x}), y) \propto p(y|\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\delta}) \cancel{p(\boldsymbol{\delta})}$$

$$\approx \exp(-\mathrm{BIC}[\delta]/2) \cancel{p(\boldsymbol{\delta})} \qquad p(\delta_{p,q}) = \frac{1}{2}$$

Model proposal

δ is latent and hard to optimize over: use a stochastic algorithm!

Strategy used here: Metropolis-Hastings algorithm.

$$p(y|\boldsymbol{e}) = \sum_{\boldsymbol{\delta} \in \{0,1\}^{\frac{d(d-1)}{2}}} p(y|\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\delta}) p(\boldsymbol{\delta})$$

$$p(\boldsymbol{\delta}|\boldsymbol{f}(\boldsymbol{x}), y) \propto p(y|\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\delta}) \cancel{p(\boldsymbol{\delta})}$$

$$\approx \exp(-\mathrm{BIC}[\delta]/2) \cancel{p(\boldsymbol{\delta})} \qquad p(\delta_{p,q}) = \frac{1}{2}$$

Which transition proposal $q : (\{0,1\}^{\frac{d(d-1)}{2}}, \{0,1\}^{\frac{d(d-1)}{2}}) \mapsto [0;1]$?

Model proposal

$2^{d(d-1)}$ probabilities to calculate...

Model proposal

$2^{d(d-1)}$ probabilities to calculate...

We restrict changes to only one entry $\delta_{k,\ell}$.

Model proposal

$2^{d(d-1)}$ probabilities to calculate...

We restrict changes to only one entry $\delta_{k,\ell}$.

Proposal: gain/loss in BIC between bivariate models with / without the interaction.

Model proposal

$2^{d(d-1)}$ probabilities to calculate...

We restrict changes to only one entry $\delta_{k,\ell}$.

Proposal: gain/loss in BIC between bivariate models with / without the interaction.

Trick: alternate one discretization / grouping step and one "interaction" step.

Performance asserted on simulated data.
Good performance on real data:

Gini	Current performance	**glmdisc**	Basic **glm**
Auto (n=50,000 ; d=15)	57.9	64.84	58
Revolving (n=48,000 ; d=9)	58.57	67.15	53.5
Prospects (n=5,000 ; d=25)	35.6	47.18	32.7
Electronics (n=140,000 ; d=8)	57.5	58	-10
Young (n=5,000 ; d=25)	$\approx$ 15	30	12.2
Basel II (n=70,000 ; d=13)	70	71.3	19

Relatively fast computing time: between 2 hours and a day on a laptop according to number of observations, features, . . .

"Inexisting" human time.

Results: medicine datasets

	Pima	Breast	Heart	Birthwt
Naïve LR	**0.73**	0.94	0.78	0.34
Naïve LR w. interactions	0.60	0.51	0.47	0.15
glmdisc	0.57	0.93	**0.82**	0.18
glmdisc w. interactions	0.62	**0.95**	0.67	**0.54**

Conclusion and future work

Take-aways

Conclusion

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- R implementation of glmdisc available on Github, to be submitted to CRAN,

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- R implementation of glmdisc available on Github, to be submitted to CRAN,
- Python implementation of glmdisc available on Github and PyPi,

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- R implementation of glmdisc available on Github, to be submitted to CRAN,
- Python implementation of glmdisc available on Github and PyPi,
- Big gain for statisticians relying on logistic regression.

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- R implementation of glmdisc available on Github, to be submitted to CRAN,
- Python implementation of glmdisc available on Github and PyPi,
- Big gain for statisticians relying on logistic regression.

Perspectives

Tested for logistic regression and polytomous logistic links: can be adapted to other models p_θ and p_α!

Thanks!

References

Ramírez-Gallego, S., García, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Alonso-Betanzos, A., Benítez, J. M., and Herrera, F. (2016).
Data discretization: taxonomy and big data challenge.
Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 6(1):5–21.

Interaction discovery: proposal

$$p(\delta_{k,\ell} = 1 | e_k, e_\ell, y) = g(\mathrm{BIC}[\delta_{k,\ell} = 1] - \mathrm{BIC}[\delta_{k,\ell} = 0])$$
$$\approx \exp\left(\tfrac{1}{2}(\mathrm{BIC}[p_{\boldsymbol{\theta}}(y|e_k, e_\ell, \delta_{k,\ell} = 0)] - \mathrm{BIC}[p_{\boldsymbol{\theta}}(y|e_k, e_\ell, \delta_{k,\ell} = 1)])\right)$$

$q(\boldsymbol{\delta}, \boldsymbol{\delta}') = |\delta_{k,\ell} - p_{k,\ell}|$ for the unique couple (k, ℓ) s.t. $\delta^{(s)}_{k,\ell} \neq \delta'_{k,\ell}$

$$\alpha = \min\left(1, \frac{p(\boldsymbol{\delta}'|\boldsymbol{e},y)}{p(\boldsymbol{\delta}|\boldsymbol{e},y)} \frac{1-q(\boldsymbol{\delta},\boldsymbol{\delta}')}{q(\boldsymbol{\delta},\boldsymbol{\delta}')}\right)$$
$$\approx \min\left(1, \exp\left(\tfrac{1}{2}(\mathrm{BIC}[p_{\boldsymbol{\theta}}(y|\boldsymbol{e}, \boldsymbol{\delta})] - \mathrm{BIC}[p_{\boldsymbol{\theta}}(y|\boldsymbol{e}, \boldsymbol{\delta}')])\right) \frac{1-q(\boldsymbol{\delta},\boldsymbol{\delta}')}{q(\boldsymbol{\delta},\boldsymbol{\delta}')}\right)$$