Some thoughts about current Credit Scoring practices

Adrien Ehrhardt
AGOS Machine Learning Day

20/06/2019

Table of Contents

Context and notations

Context and notations: Industrial setting

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

Table: Dataset with outliers and missing values.

Context and notations: Industrial setting

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

Table: Dataset with outliers and missing values.

1. Discarding rejected applicants
2. Feature selection
3. Discretization / grouping
4. Interaction screening
5. Segmentation
6. Logistic regression fitting

Context and notations: Industrial setting

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

Table: Dataset with outliers and missing values.

1. **Discarding rejected applicants**
2. Feature selection
3. Discretization / grouping
4. Interaction screening
5. Segmentation
6. Logistic regression fitting

Context and notations: Industrial setting

Job			Family status	Wages		Repayment
Craftsman			Widower	2000		0
?			Common-law	1700		1
Licensed professional			Divorced	4000		0
Executive			Single	2700		1
~~Office employee~~	~~Renter~~	~~12~~	~~Married~~	~~1400~~		NA
~~Worker~~	~~By family~~	~~2~~	~~?~~	~~1200~~		NA

Table: Dataset with outliers and missing values.

1. Discarding rejected applicants
2. **Feature selection**
3. Discretization / grouping
4. Interaction screening
5. Segmentation
6. Logistic regression fitting

Context and notations: Industrial setting

Job			Family status	Wages		Repayment
Craftsman			Widower	]1500;2000]		0
?			Common-law	]1500;2000]		1
Licensed professional			Divorced	]2000;∞[		0
Executive			Single	]2000;∞[		1
~~Office employee~~	~~Renter~~	~~12~~	~~Married~~	~~1400~~		NA
~~Worker~~	~~By family~~	~~2~~	~~?~~	~~1200~~		NA

Table: Dataset with outliers and missing values.

1. Discarding rejected applicants
2. Feature selection
3. Discretization / grouping
4. Interaction screening
5. Segmentation
6. Logistic regression fitting

Context and notations: Industrial setting

Job			Family status	Wages		Repayment
?+Low-qualified			?+Alone	]1500;2000]		0
?+Low-qualified			Union	]1500;2000]		1
High-qualified			?+Alone	]2000;∞[		0
High-qualified			?+Alone	]2000;∞[		1
~~Office employee~~	~~Renter~~	~~12~~	~~Married~~	~~1400~~		NA
~~Worker~~	~~By family~~	~~2~~	~~?~~	~~1200~~		NA

Table: Dataset with outliers and missing values.

1. Discarding rejected applicants
2. Feature selection
3. Discretization / grouping
4. Interaction screening
5. Segmentation
6. Logistic regression fitting

Context and notations: Industrial setting

Job			Family status x Wages		Repayment
?+Low-qualified			?+Alone x]1500;2000]		0
?+Low-qualified			Union x]1500;2000]		1
High-qualified			?+Alone x]2000;∞[		0
High-qualified			?+Alone x]2000;∞[		1
~~Office employee~~	~~Renter~~	~~12~~	~~Married~~ ~~1400~~		NA
~~Worker~~	~~By family~~	~~2~~	~~?~~ ~~1200~~		NA

Table: Dataset with outliers and missing values.

1. Discarding rejected applicants
2. Feature selection
3. Discretization / grouping
4. **Interaction screening**
5. Segmentation
6. Logistic regression fitting

Context and notations: Industrial setting

Job			Family status x Wages		Repayment
?+Low-qualified			?+Alone x]1500;2000]		0
?+Low-qualified			Union x]1500;2000]		1
High-qualified			?+Alone x $]2000;\infty[$		0
High-qualified			?+Alone x $]2000;\infty[$		1
~~Office employee~~	~~Renter~~	~~12~~	~~Married~~ ~~1400~~		NA
~~Worker~~	~~By family~~	~~2~~	~~?~~ ~~1200~~		NA

Table: Dataset with outliers and missing values.

1. Discarding rejected applicants
2. Feature selection
3. Discretization / grouping
4. Interaction screening
5. **Segmentation**
6. Logistic regression fitting

Context and notations: Industrial setting

Job			Family status x Wages	Score	Repayment
?+Low-qualified			?+Alone x]1500;2000]	225	0
?+Low-qualified			Union x]1500;2000]	190	1
High-qualified			?+Alone x]2000;∞[	218	0
High-qualified			?+Alone x]2000;∞[	202	1
~~Office employee~~	~~Renter~~	~~12~~	~~Married~~ ~~1400~~	NA	NA
~~Worker~~	~~By family~~	~~2~~	~~?~~ ~~1200~~	NA	NA

Table: Dataset with outliers and missing values.

1. Discarding rejected applicants
2. Feature selection
3. Discretization / grouping
4. Interaction screening
5. Segmentation
6. Logistic regression fitting

Context and notations: Available data

Random variables: $\boldsymbol{X}, Y, Z$

Context and notations: Available data

Random variables: $\boldsymbol{X}, Y, Z$

Observations:
$\boldsymbol{x} = (x_1, \ldots, x_d)$: characteristics.
$x_j \in \mathbb{R}$ or $\{1, \ldots, l_j\}$: *e.g.* rent amount, job, ...
$y \in \{0, 1\}$: good or bad.
$z \in \{\mathrm{f}, \mathrm{nf}\}$: financed or not financed.

Context and notations: Available data

Random variables: $\boldsymbol{X}, Y, Z$

Observations:
$\boldsymbol{x} = (x_1, \dots, x_d)$: characteristics.
$x_j \in \mathbb{R}$ or $\{1, \dots, l_j\}$: *e.g.* rent amount, job, ...
$y \in \{0, 1\}$: good or bad.
$z \in \{\text{f}, \text{nf}\}$: financed or not financed.

True distribution of good and bad clients: $p(y|\boldsymbol{x})$

Context and notations: Available data

Need for a **computable model** that resembles p, often in the form of a **parametric** model $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$, which we can calculate for a new client.

Need for a **computable model** that resembles p, often in the form of a **parametric** model $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$, which we can calculate for a new client.

Example: logistic regression

$$\ln \frac{p_{\boldsymbol{\theta}}(1|\boldsymbol{x})}{(1 - p_{\boldsymbol{\theta}}(1|\boldsymbol{x}))} = \boldsymbol{x}'\boldsymbol{\theta}$$

Context and notations: Available data

Need for a **computable model** that resembles p, often in the form of a **parametric** model $p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$, which we can calculate for a new client.

Example: logistic regression

$$\ln \frac{p_{\boldsymbol{\theta}}(1|\boldsymbol{x})}{(1 - p_{\boldsymbol{\theta}}(1|\boldsymbol{x}))} = \boldsymbol{x}'\boldsymbol{\theta}$$

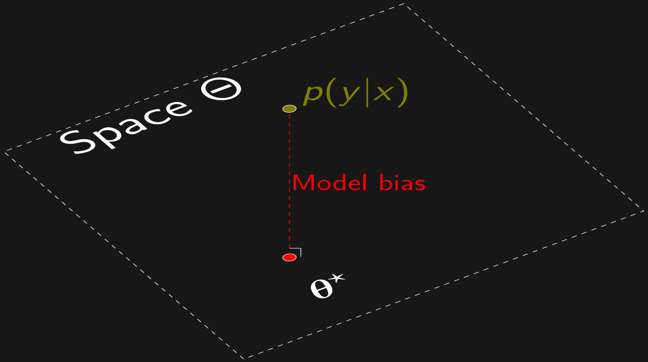
There is $\boldsymbol{\theta}^\star$ that makes $p_{\boldsymbol{\theta}^\star}$ "close" to p.

$$\boldsymbol{\theta}^\star = \operatorname{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{X}}[\mathrm{KL}(p||p_{\boldsymbol{\theta}})] = \int_{\mathcal{X}} \sum_{y\in\{0,1\}} p(y|\boldsymbol{x}) \ln \frac{p(y|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(y|\boldsymbol{x})}.$$

Well-specified model assumption

$\mathbb{E}_{\boldsymbol{X}}[\mathrm{KL}(p||p_{\boldsymbol{\theta}^\star})] = 0,$
$p_{\boldsymbol{\theta}^\star}(y|\boldsymbol{x}) = p(y|\boldsymbol{x}).$

Context and notations: Available data



Context and notations: Available data

p is unknown: access to an i.i.d. $n + n'$-sample
$\mathcal{T} = (\boldsymbol{x}_i, y_i, z_i)_1^{n+n'} \sim p$.

Context and notations: Available data

p is unknown: access to an i.i.d. $n+n'$-sample $\mathcal{T} = (\boldsymbol{x}_i, y_i, z_i)_1^{n+n'} \sim p$.

We can deduce from the KL divergence the (log-)likelihood:

$$\ell(\boldsymbol{\theta}; \mathcal{T}) = \sum_{i=1}^{n+n'} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{x}_i).$$

Context and notations: Available data

p is unknown: access to an i.i.d. $n + n'$-sample
$\mathcal{T} = (\boldsymbol{x}_i, y_i, z_i)_1^{n+n'} \sim p$.

We can deduce from the KL divergence the (log-)likelihood:

$$\ell(\boldsymbol{\theta}; \mathcal{T}) = \sum_{i=1}^{n+n'} \ln p_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i).$$

The MLE $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}}\, \ell(\boldsymbol{\theta}; \mathcal{T})$ is a good approximation of $\boldsymbol{\theta}^\star$.

Context and notations: Available data

p is unknown: access to an i.i.d. $n+n'$-sample $\mathcal{T} = (\boldsymbol{x}_i, y_i, z_i)_1^{n+n'} \sim p$.

We can deduce from the KL divergence the (log-)likelihood:

$$\ell(\boldsymbol{\theta}; \mathcal{T}) = \sum_{i=1}^{n+n'} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{x}_i).$$

The MLE $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}}\, \ell(\boldsymbol{\theta}; \mathcal{T})$ is a good approximation of $\boldsymbol{\theta}^\star$.

Unfortunately, $\hat{\boldsymbol{\theta}}$ is not directly computable (no closed form solution).

Context and notations: Available data

p is unknown: access to an i.i.d. $n+n'$-sample $\mathcal{T} = (\boldsymbol{x}_i, y_i, z_i)_1^{n+n'} \sim p$.

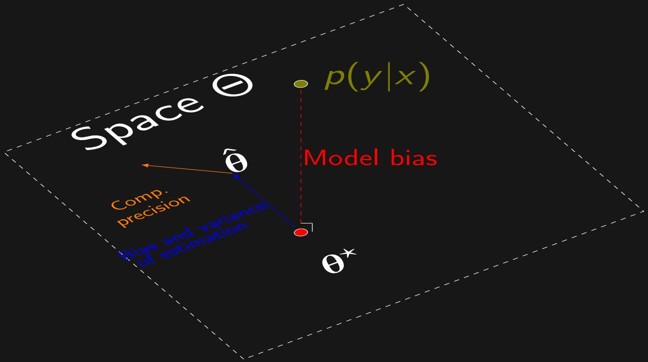
We can deduce from the KL divergence the (log-)likelihood:

$$\ell(\boldsymbol{\theta}; \mathcal{T}) = \sum_{i=1}^{n+n'} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{x}_i).$$

The MLE $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}}\, \ell(\boldsymbol{\theta}; \mathcal{T})$ is a good approximation of $\boldsymbol{\theta}^\star$.

Unfortunately, $\hat{\boldsymbol{\theta}}$ is not directly computable (no closed form solution).

$$\tilde{\boldsymbol{\theta}} = \text{Newton-Raphson}(\ell(\boldsymbol{\theta}; \mathcal{T})) \neq \hat{\boldsymbol{\theta}}.$$



All of this is “hidden” in your favourite statistical language / package / library but is essential to understanding Reject Inference.

Context and notations: Feature / model selection

Up to now, we assumed a parameter space Θ **fixed**.

Context and notations: Feature / model selection

Up to now, we assumed a parameter space Θ **fixed**.

Comparing models = different parameter spaces $\Theta^1, \Theta^2, \ldots$ Corresponding to feature subsets, different discretizations, interactions, . . . since **we don't know which parameter space is closest to the "truth"** p.

Context and notations: Feature / model selection

Up to now, we assumed a parameter space Θ **fixed**.

Comparing models = different parameter spaces $\Theta^1, \Theta^2, \ldots$
Corresponding to feature subsets, different discretizations, interactions, ... since **we don't know which parameter space is closest to the "truth"** p.

Model selection tools

$$\hat{\boldsymbol{\theta}}^{\text{best}} = \underset{\hat{\boldsymbol{\theta}}^k \in \Theta^k}{\operatorname{argmin}} \operatorname{BIC}(\hat{\boldsymbol{\theta}}^k) = -2\ell(\hat{\boldsymbol{\theta}}^k, \mathcal{T}) + \dim(\Theta^k) \ln n.$$

Context and notations: Feature / model selection

Up to now, we assumed a parameter space Θ **fixed**.

Comparing models = different parameter spaces $\Theta^1, \Theta^2, \dots$ Corresponding to feature subsets, different discretizations, interactions, ... since **we don't know which parameter space is closest to the "truth"** p.

Model selection tools

$$\hat{\boldsymbol{\theta}}^{\text{best}} = \underset{\hat{\boldsymbol{\theta}}^k \in \Theta^k}{\operatorname{argmin}} \operatorname{BIC}(\hat{\boldsymbol{\theta}}^k) = -2\ell(\hat{\boldsymbol{\theta}}^k, \mathcal{T}) + \dim(\Theta^k) \ln n.$$

BIC has nice statistical properties (consistency) but can be swapped in the entire presentation with your favourite model selection tool like Gini on $\mathcal{T}^{\text{test}}$.

Reject Inference

Reject Inference: Industrial setting

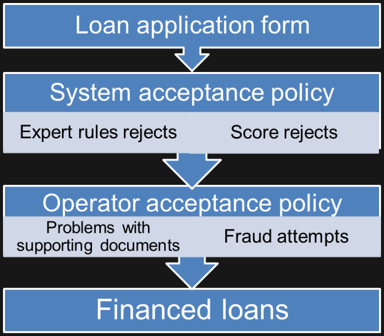


Figure: Simplified Acceptance mechanism in Crédit Agricole Consumer Finance

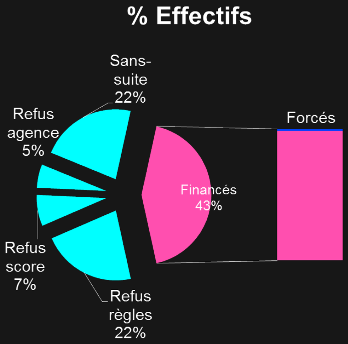


Figure: Proportion of “final” lending decisions for CACF France

Reject Inference: Industrial setting

The observed data are the following:

$$\mathcal{T} = \begin{array}{l} \mathcal{T}_{\text{f}} = \left(\mathbf{x}_{\text{f}} \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{pmatrix} \quad \mathbf{y}_{\text{f}} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \mathbf{z}_{\text{f}} \begin{pmatrix} \text{f} \\ \vdots \\ \text{f} \end{pmatrix} \right). \\ \cup \\ \mathcal{T}_{\text{nf}} = \left(\mathbf{x}_{\text{nf}} \begin{pmatrix} x_{n+1,1} & \cdots & x_{n+1,d} \\ \vdots & \vdots & \vdots \\ x_{n+n',1} & \cdots & x_{n+n',d} \end{pmatrix} \quad \mathbf{y}_{\text{nf}} \begin{pmatrix} \text{NA} \\ \vdots \\ \text{NA} \end{pmatrix} \quad \mathbf{z}_{\text{nf}} \begin{pmatrix} \text{nf} \\ \vdots \\ \text{nf} \end{pmatrix} \right). \end{array}$$

We traditionally build a logistic regression using only financed clients (**fixed parameter space** Θ):

$$\hat{\boldsymbol{\theta}}_{\text{f}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}; \mathcal{T}_{\text{f}}),$$

which asymptotically approximates:

$$\boldsymbol{\theta}^{\star}_{\text{f}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{X}}[\text{KL}(p||p_{\boldsymbol{\theta}})|Z = \text{f}].$$

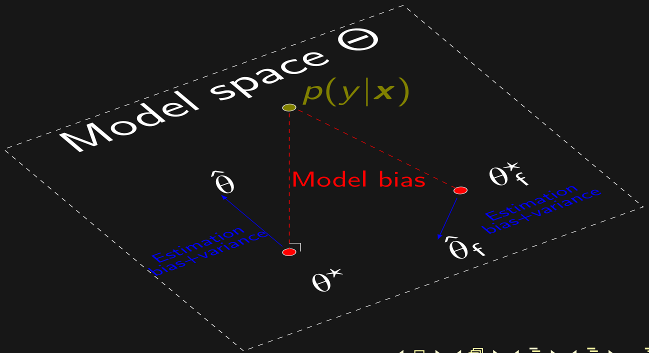
We wish we had:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}),$$

which asymptotically approximates:

$$\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}}[\mathrm{KL}(p||p_{\boldsymbol{\theta}})].$$

But we lack $\mathbf{y}_{\mathrm{nf}}$.



Estimators :

1. "Oracle": $\sqrt{n+n'}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star) \xrightarrow[n,n' \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\boldsymbol{\theta}^\star})$
2. Current methodology: $\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{f}} - \boldsymbol{\theta}_{\mathrm{f}}^\star) \xrightarrow[n \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\mathrm{f},\boldsymbol{\theta}_{\mathrm{f}}^\star})$

Reject Inference: What is at stake?

Estimators :

1. "Oracle": $\sqrt{n+n'}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star) \xrightarrow[n,n' \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\boldsymbol{\theta}^\star})$
2. Current methodology: $\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{f}} - \boldsymbol{\theta}^\star_{\mathrm{f}}) \xrightarrow[n \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\mathrm{f},\boldsymbol{\theta}^\star_{\mathrm{f}}})$

Reject Inference: What is at stake?

Estimators :

1. "Oracle": $\sqrt{n+n'}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{\star}) \xrightarrow[n,n' \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\boldsymbol{\theta}^{\star}})$
2. Current methodology: $\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{f}} - \boldsymbol{\theta}_{\mathrm{f}}^{\star}) \xrightarrow[n \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\mathrm{f},\boldsymbol{\theta}_{\mathrm{f}}^{\star}})$

Estimators :

1. "Oracle": $\sqrt{n+n'}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star) \xrightarrow[n,n' \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\boldsymbol{\theta}^\star})$
2. Current methodology: $\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{f}} - \boldsymbol{\theta}_{\mathrm{f}}^\star) \xrightarrow[n \to \infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\mathrm{f},\boldsymbol{\theta}_{\mathrm{f}}^\star})$

Estimators :

1. "Oracle": $\sqrt{n+n'}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star) \xrightarrow[n,n'\to\infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\boldsymbol{\theta}^\star})$
2. Current methodology: $\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{f}} - \boldsymbol{\theta}^\star_{\mathrm{f}}) \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{N}_{d+1}(0, \Sigma_{\mathrm{f},\boldsymbol{\theta}^\star_{\mathrm{f}}})$

Question 1 : asymptotics of the estimators

$$\boxed{(\mathrm{Q1})\ \boldsymbol{\theta}^\star \overset{?}{=} \boldsymbol{\theta}^\star_{\mathrm{f}}}$$

$$\boxed{(\mathrm{Q2})\ \Sigma_{\boldsymbol{\theta}^\star} \overset{?}{=} \Sigma_{\mathrm{f},\boldsymbol{\theta}^\star_{\mathrm{f}}}}$$

Reject Inference: Missingness mechanism

- **MAR** : $\forall\, x, y, z,\ p(z|\boldsymbol{x}, y) = p(z|\boldsymbol{x})$
  $\rightarrow$ Acceptance is determined by an old score: $Z = \mathbb{1}_{\{\theta' X > \text{cut}\}}$.
- **MNAR** : $\exists\, x, y, z,\ p(z|\boldsymbol{x}, y) \neq p(z|\boldsymbol{x})$
  $\rightarrow$ Operators' "feeling" $\tilde{\boldsymbol{X}}$ influence the acceptance.
  $\rightarrow$ Expert rules based on features $\tilde{\boldsymbol{X}}$ not in $\boldsymbol{X}$.

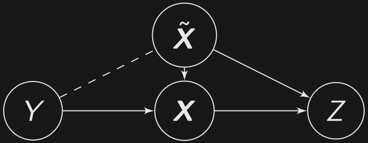


Figure: Dependencies between random variables Y, $\tilde{\boldsymbol{X}}$, $\boldsymbol{X}$ and Z

Reject Inference: Model specification

- **Well-specified model** : $p(y|\boldsymbol{x}) = p_{\boldsymbol{\theta}^\star}(y|\boldsymbol{x})$.
  $\rightarrow$ With real data $\Rightarrow$ hypothesis unlikely to be true.
- **Misspecified model** : $\boldsymbol{\theta}^\star$ is the "best" in the Θ family.
  $\rightarrow$ Logistic regression commonly used for its robustness to misspecification (no assumption about $p(\boldsymbol{x})$).

$p_{\boldsymbol{\theta}}(y\|\boldsymbol{x})$ \ $p(z\|\boldsymbol{x}, y)$	MAR	MNAR
Well specified	$\boldsymbol{\theta}_{\mathrm{f}}^\star = \boldsymbol{\theta}^\star$ $\Sigma_{\mathrm{f},\boldsymbol{\theta}_{\mathrm{f}}^\star} \neq \Sigma_{\theta^\star}$	$\boldsymbol{\theta}_{\mathrm{f}}^\star \neq \boldsymbol{\theta}^\star$ $\Sigma_{\mathrm{f},\boldsymbol{\theta}_{\mathrm{f}}^\star} \neq \Sigma_{\theta^\star}$
Misspecified	$\boldsymbol{\theta}_{\mathrm{f}}^\star \neq \boldsymbol{\theta}^\star$ $\Sigma_{\mathrm{f},\boldsymbol{\theta}_{\mathrm{f}}^\star} \neq \Sigma_{\theta^\star}$	

Table: (Q1) and (Q2) w.r.t. model specification and missingness mechanism

Reject Inference: How to use $\mathbf{x}_{\mathrm{nf}}$?

Question 2: How to construct a better estimator than $\hat{\theta}_{\mathrm{f}}$?

Question 2: How to construct a better estimator than $\hat{\theta}_{\mathrm{f}}$?

Scope for action:

- Change model space Θ,

Reject Inference: How to use $\mathbf{x}_{\mathrm{nf}}$?

Question 2: How to construct a better estimator than $\hat{\theta}_{\mathrm{f}}$?

Scope for action:

- Change model space Θ,
- Model acceptance/rejection process (i.e. $p_\gamma(z|\boldsymbol{x}, y)$),

Reject Inference: How to use $\mathbf{x}_{\text{nf}}$?

Question 2: How to construct a better estimator than $\hat{\theta}_{\text{f}}$?

Scope for action:

- Change model space Θ,
- Model acceptance/rejection process (i.e. $p_\gamma(z|\boldsymbol{x}, y)$),
- Use $\mathbf{x}_{\text{nf}}$.

Question 2: How to construct a better estimator than $\hat{\theta}_{\mathrm{f}}$?

Scope for action:

- Change model space Θ,
- Model acceptance/rejection process (i.e. $p_\gamma(z|\boldsymbol{x}, y)$),
- Use $\mathbf{x}_{\mathrm{nf}}$.

Natural way to achieve all three: generative approach

$$p_\alpha(\boldsymbol{x}, y, z) = p_{\beta_\alpha}(\boldsymbol{x}) p_{\theta_\alpha}(y|x) p_{\gamma_\alpha}(z|\boldsymbol{x}, y).$$

$$(\boxed{\hat{\theta}_\alpha}, \hat{\beta}_\alpha, \hat{\gamma}_\alpha) = \underset{\theta_\alpha, \beta_\alpha, \gamma_\alpha}{\operatorname{argmax}} \ell(\alpha; \boldsymbol{x}, \boldsymbol{y}_{\mathrm{f}}) = \underset{\theta_\alpha, \beta_\alpha, \gamma_\alpha}{\operatorname{argmax}} \sum_{i=1}^{n} \ln(p_{\theta_\alpha}(y_i|x_i)) + \sum_{i=1}^{n+n'} \ln(p_{\beta_\alpha}(\boldsymbol{x}_i)) \left(+ \sum_{i=1}^{n} \ln(p_{\gamma_\alpha}(z_i|x_i, y_i)) \right).$$

Reject Inference: How to use $\mathbf{x}_{\text{nf}}$?

Question 2: How to construct a better estimator than $\hat{\theta}_{\text{f}}$?

Scope for action:

- ~~Change model space Θ~~ logistic regression,
- Model acceptance/rejection process (i.e. $p_\gamma(z|\boldsymbol{x}, y)$),
- Use $\mathbf{x}_{\text{nf}}$.

~~Natural way to achieve all three: generative approach~~

$$p_{\alpha}(\boldsymbol{x}, y, z) = p_{\beta_\alpha}(\boldsymbol{x}) p_{\theta_\alpha}(y|\boldsymbol{x}) p_{\gamma_\alpha}(z|\boldsymbol{x}, y).$$

$$\left(\boxed{\hat{\theta}_\alpha}, \hat{\beta}_\alpha, \hat{\gamma}_\alpha\right) = \underset{\alpha}{\operatorname{argmax}}\, \ell(\alpha; \boldsymbol{x}, \boldsymbol{y}_{\text{f}}) = \underset{\theta_\alpha, \beta_\alpha, \gamma_\alpha}{\operatorname{argmax}} \sum_{i=1}^{n} \ln(p_{\theta_\alpha}(y_i|x_i))$$

$$+ \sum_{i=1}^{n+n'} \ln(p_{\beta_\alpha}(\boldsymbol{x}_i)) \left(+ \sum_{i=1}^{n} \ln(p_{\gamma_\alpha}(z_i|x_i, y_i)) \right).$$

Reject Inference: How to use $\mathbf{x}_{\text{nf}}$?

Question 2: How to construct a better estimator than $\hat{\theta}_{\text{f}}$?

Scope for action:

- ~~Change model space Θ~~ logistic regression,
- ~~Model acceptance/rejection process (i.e. $p_\gamma(z|\boldsymbol{x}, y)$)~~
  γ cannot be estimated,
- Use $\mathbf{x}_{\text{nf}}$.

Reject Inference: How to use $\mathbf{x}_{\mathrm{nf}}$?

Question 2: How to construct a better estimator than $\hat{\theta}_{\mathrm{f}}$?

Scope for action:

- ~~Change model space Θ~~ logistic regression,
- ~~Model acceptance/rejection process (i.e. $p_\gamma(z|\boldsymbol{x}, y)$)~~ γ cannot be estimated,
- Use $\mathbf{x}_{\mathrm{nf}}$.

Remember that $\mathcal{T}^{\mathrm{OOT}}$ also comes from $p(y|\boldsymbol{x}, \mathrm{f})$ such that applying a *Reject Inference* method and getting a higher Gini is no guarantee that it would on the Through-the-Door population (on the contrary!).

Reject Inference: How to use $\mathbf{x}_{\mathrm{nf}}$?

For logistic regression, *Reject Inference* methods amount to:

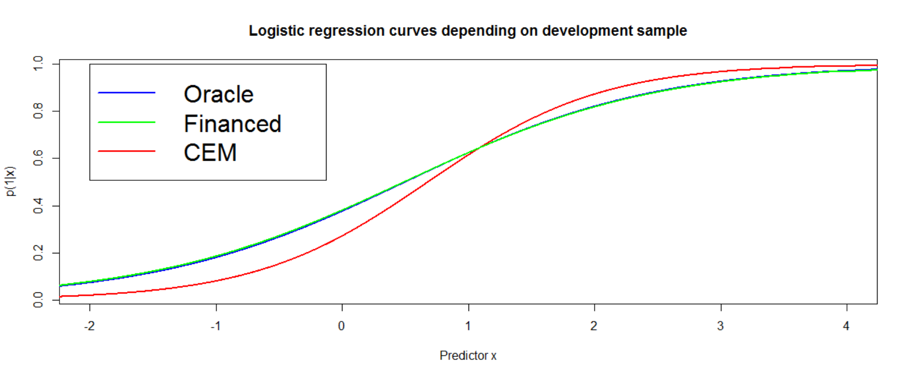
$$\mathcal{T}_{\mathrm{c}}^{(1)} = \begin{array}{c} \\ \mathbf{x}_{\mathrm{f}} \\ \\ \\ \mathbf{x}_{\mathrm{nf}} \\ \\ \end{array} \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \\ x_{n+1,1} & \cdots & x_{n+1,d} \\ \vdots & \vdots & \vdots \\ x_{n+n',1} & \cdots & x_{n+n',d} \end{pmatrix}, \begin{array}{c} \\ \mathbf{y}_{\mathrm{f}} \\ \\ \\ \mathbf{y}_{\mathrm{nf}} \\ \\ \end{array} \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ \hat{y}_{n+1}^{(1)} \\ \vdots \\ \hat{y}_{n+n'}^{(1)} \end{pmatrix}, \begin{array}{c} \\ \mathbf{z}_{\mathrm{f}} \\ \\ \\ \mathbf{z}_{\mathrm{nf}} \\ \\ \end{array} \begin{pmatrix} \mathrm{f} \\ \vdots \\ \mathrm{f} \\ \mathrm{nf} \\ \vdots \\ \mathrm{nf} \end{pmatrix}.$$

Reject Inference: How to use $\mathbf{x}_{nf}$?

Reclassification[1] :

$$(\hat{\boldsymbol{\theta}}^{\text{CEM}}, \hat{\mathbf{y}}^{\text{nf}}) = \underset{\boldsymbol{\theta}, \mathbf{y}^{\text{nf}}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}; \mathcal{T}_c^{(1)}) \text{ where } \hat{y}_i = \underset{y_i}{\operatorname{argmax}}\, p_{\hat{\boldsymbol{\theta}}_{\text{f}}}(y_i | \boldsymbol{x}_i).$$

Problem: inconsistent estimator.



[1] [4, 1, 2]

Reject Inference: How to use $\mathbf{x}_{\text{nf}}$?

Augmentation[2]: MAR / misspecified model.

$$\ell_{\text{Aug}}(\boldsymbol{\theta}; \mathcal{T}_{\text{f}}) = \sum_{i=1}^{n} \frac{1}{p(\text{f}|\boldsymbol{x}_i)} \ln(p_{\boldsymbol{\theta}}(y_i|\boldsymbol{x}_i)).$$

Problem: estimation of $p(\text{f}|x_i)$ + assumes $p(\text{f}|x_i) > 0$ (clearly not true).

Parcelling [3]:

$$\ell(\theta; \boldsymbol{x}, \mathbf{y}_{\text{f}}, \hat{\mathbf{y}}_{\text{nf}}) \text{ where } \hat{y}_i = \begin{cases} 1 \text{ w.p. } \alpha_i p_{\hat{\boldsymbol{\theta}}_{\text{f}}}(1|\boldsymbol{x}_i, \text{f}) \\ 0 \text{ w.p. } 1 - \alpha_i p_{\hat{\boldsymbol{\theta}}_{\text{f}}}(1|\boldsymbol{x}_i, \text{f}) \end{cases}.$$

Problem: MNAR assumptions hidden in $\hat{\mathbf{y}}_{\text{nf}}$ (α_i) impossible to test.

---

[2] [4, 1, 2, 3]
[3] [4, 1, 2]

Reject Inference: Additional remarks

WARNING

All this stands for logistic regression and all "local" methods [5].

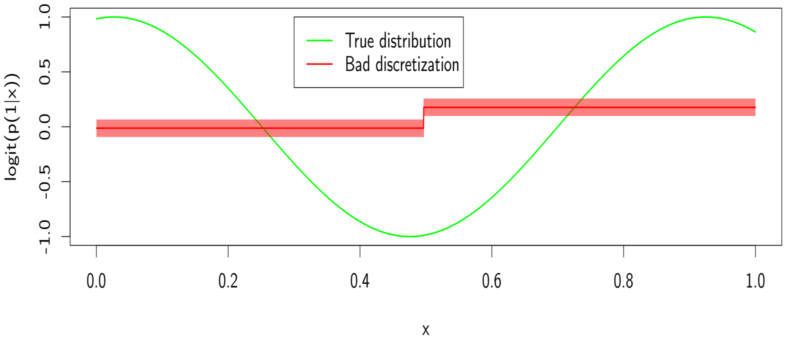
All "global" methods (explicit or implicit modelling of $p(\boldsymbol{x})$) will produce biased estimates under MAR.

We might have:

Gini	Logistic regression	Decision trees
Financed	40	45
Through-the-door	40	35

Feature quantization

Feature quantization: By an example



Some more notations I

Raw data

$$\boldsymbol{x} = (x_1, \ldots, x_d)$$
$$x_j \in \mathbb{R} \text{ (continuous case)}$$
$$x_j \in \{1, \ldots, l_j\} \text{ (categorical case)}$$
$$y \in \{0, 1\} \text{ (target)}$$

Quantized data

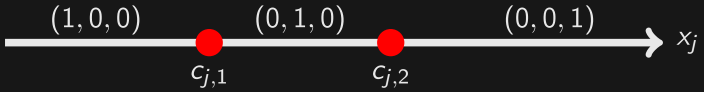
$$\boldsymbol{q}(\boldsymbol{x}) = (\boldsymbol{q}_1(x_1), \ldots, \boldsymbol{q}_d(x_d))$$
$$\boldsymbol{q}_j(x_j) = (q_{j,h}(x_j))_1^{m_j} \text{ (one-hot encoding)}$$
$$q_{j,h}(\cdot) = 1 \text{ if } x_j \in C_{j,h}, 0 \text{ otherwise, } 1 \leq h \leq m_j$$

Some more notations II

Discretization

$$C_{j,h} = (c_{j,h-1}, c_{j,h}]$$

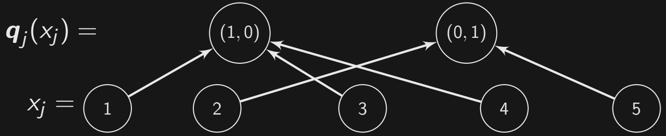
where $c_{j,1}, \dots, c_{j,m_j-1}$ are increasing numbers called cutpoints, $c_{j,0} = -\infty$ and $c_{j,m_j} = +\infty$.



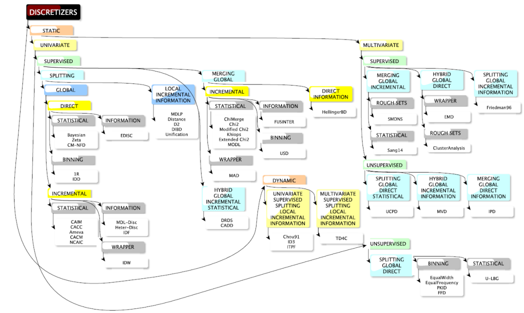
Some more notations III

Grouping

$$\bigsqcup_{h=1}^{m_j} C_{j,h} = \{1, \ldots, l_j\}.$$



Feature quantization: Existing approaches



You maximize an *ad hoc* criterion:

$$\hat{\boldsymbol{q}} = \underset{\boldsymbol{q}}{\operatorname{argmax}} \operatorname{CRIT}(\mathcal{T}_{\mathrm{f}}),$$

and **hope** that it's aligned with your original goal:

$$\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{q}}} = \underset{\boldsymbol{\theta}_{\hat{\boldsymbol{q}}}}{\operatorname{argmax}} \ell(\boldsymbol{\theta}_{\hat{\boldsymbol{q}}}; \mathcal{T}_{\mathrm{f}}).$$

Feature quantization: Approximation

$$\boldsymbol{q}_{\boldsymbol{\alpha}_j}(\cdot) = \left(q_{\boldsymbol{\alpha}_{j,h}}(\cdot)\right)_{h=1}^{m_j} \text{ with } \begin{cases} \sum_{h=1}^{m_j} q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = 1, \\ 0 \leq q_{\boldsymbol{\alpha}_{j,h}}(\cdot) \leq 1, \end{cases}$$

Feature quantization: Approximation

$$\boldsymbol{q}_{\boldsymbol{\alpha}_j}(\cdot) = \left(q_{\boldsymbol{\alpha}_{j,h}}(\cdot)\right)_{h=1}^{m_j} \text{ with } \begin{cases} \sum_{h=1}^{m_j} q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = 1, \\ 0 \leq q_{\boldsymbol{\alpha}_{j,h}}(\cdot) \leq 1, \end{cases}$$

For continuous features, we set for $\boldsymbol{\alpha}_{j,h} = (\alpha^0_{j,h}, \alpha^1_{j,h}) \in \mathbb{R}^2$

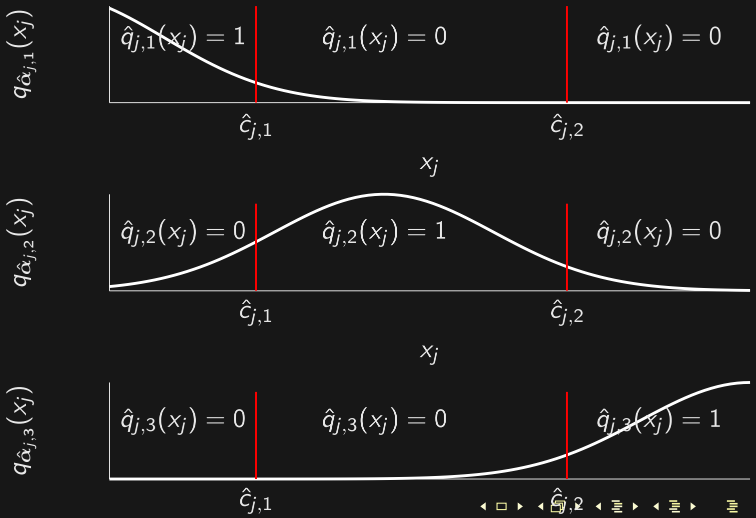
$$q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = \frac{\exp(\alpha^0_{j,h} + \alpha^1_{j,h} \cdot)}{\sum_{g=1}^{m_j} \exp(\alpha^0_{j,g} + \alpha^1_{j,g} \cdot)}.$$

For categorical features, we set for
$\boldsymbol{\alpha}_{j,h} = (\alpha_{j,h}(1), \ldots, \alpha_{j,h}(l_j)) \in \mathbb{R}^{l_j}$

$$q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = \frac{\exp\left(\alpha_{j,h}(\cdot)\right)}{\sum_{g=1}^{m_j} \exp\left(\alpha_{j,g}(\cdot)\right)}.$$

Feature quantization: Estimation MAP

$$q_{j,h}^{\mathrm{MAP}}(x_j) = 1 \text{ if } h = \underset{1 \leq h' \leq m_j}{\operatorname{argmax}} \, q_{\hat{\alpha}_{j,h'}}, 0 \text{ otherwise.}$$



We wish to maximize the following likelihood:

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\theta},\boldsymbol{\alpha}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} p_{\boldsymbol{\theta}}(y_i | \boldsymbol{q}_{\alpha}(\boldsymbol{x}_i)).$$

If there is a true quantization $\boldsymbol{q}^\star$, then $\boldsymbol{\alpha}^\star = \lim_{n\to\infty} \hat{\boldsymbol{\alpha}}$ is such that $\boldsymbol{q}_{\boldsymbol{\alpha}^\star} = \boldsymbol{q}^\star$.

Feature quantization: Neural networks

We wish to maximize the following likelihood:

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\theta}, \boldsymbol{\alpha}}{\operatorname{argmax}} \, \ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} p_{\boldsymbol{\theta}}(y_i | \boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i)).$$

If there is a true quantization $\boldsymbol{q}^\star$, then $\boldsymbol{\alpha}^\star = \lim_{n \to \infty} \hat{\boldsymbol{\alpha}}$ is such that $\boldsymbol{q}_{\boldsymbol{\alpha}^\star} = \boldsymbol{q}^\star$.

If not, $\boldsymbol{q}^{\text{MAP}}$ is "guaranteed" to be a good candidate quantization.

Feature quantization: Neural networks

We wish to maximize the following likelihood:

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\theta},\boldsymbol{\alpha}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} p_{\boldsymbol{\theta}}(y_i | \boldsymbol{q}_{\alpha}(\boldsymbol{x}_i)).$$

If there is a true quantization $\boldsymbol{q}^{\star}$, then $\alpha^{\star} = \lim_{n\to\infty} \hat{\alpha}$ is such that $\boldsymbol{q}_{\alpha^{\star}} = \boldsymbol{q}^{\star}$.

If not, $\boldsymbol{q}^{\mathrm{MAP}}$ is "guaranteed" to be a good candidate quantization.

Problem: $\ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \boldsymbol{x}, \boldsymbol{y})$ cannot be directly maximized (it's not even convex).

Feature quantization: Neural networks

We wish to maximize the following likelihood:

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\theta},\boldsymbol{\alpha}}{\operatorname{argmax}} \, \ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} p_{\boldsymbol{\theta}}(y_i | \boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i)).$$

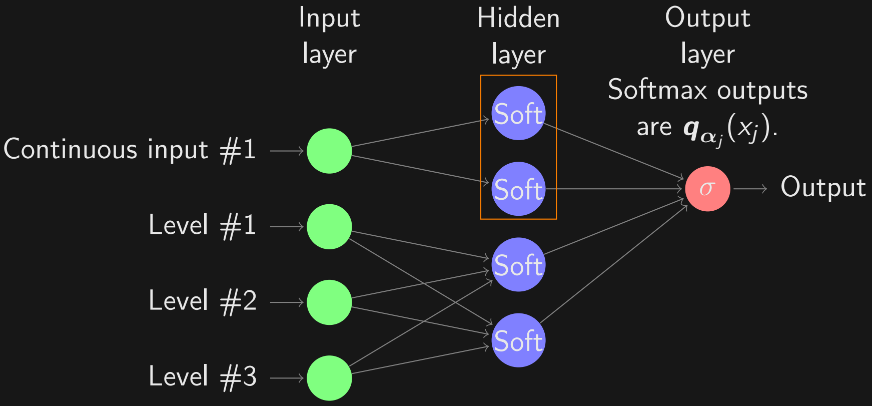
If there is a true quantization $\boldsymbol{q}^\star$, then $\boldsymbol{\alpha}^\star = \lim_{n\to\infty} \hat{\boldsymbol{\alpha}}$ is such that $\boldsymbol{q}_{\boldsymbol{\alpha}^\star} = \boldsymbol{q}^\star$.

If not, $\boldsymbol{q}^{\text{MAP}}$ is "guaranteed" to be a good candidate quantization.

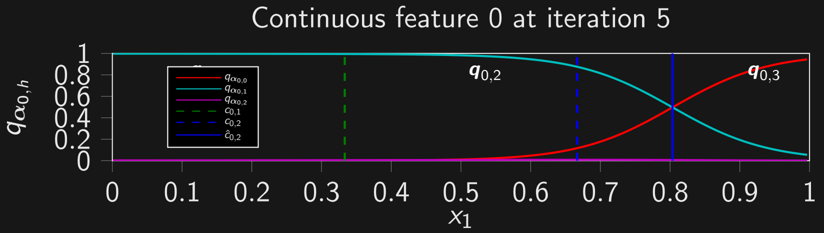
Problem: $\ell(\boldsymbol{\theta}, \boldsymbol{\alpha}; \boldsymbol{x}, \boldsymbol{y})$ cannot be directly maximized (it's not even convex).

Solution: Resort to gradient descent (not guaranteed to converge to a global maximum!).

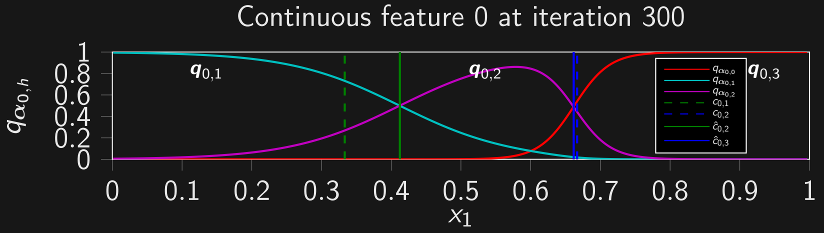
Feature quantization: Neural networks



Estimation *via* neural networks



(a) Quantization $\hat{\boldsymbol{q}}_1^{(s)}(x_1)$ resulting from the MAP at iter $t = 5$ and $m_{\max} = 3$.



(b) Quantizations $\hat{\boldsymbol{q}}_1^{(s)}(x_1)$ resulting from the MAP at iter $t = 300$ and $m_{\max} = 3$.

Feature quantization: Model = quantization selection

New model selection criterion

We have drastically restricted the search space to clever candidates $\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP}(\text{iter})}$ resulting from the the gradient descent steps.

$$(\boldsymbol{q}^\star, \boldsymbol{\theta}^\star) = \underset{\hat{\boldsymbol{q}} \in \{\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP}(\text{iter})}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} \text{BIC}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{q}}})$$

Feature quantization: Model = quantization selection

New model selection criterion

We have drastically restricted the search space to clever candidates $\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP(iter)}}$ resulting from the the gradient descent steps.

$$(\boldsymbol{q}^\star, \boldsymbol{\theta}^\star) = \underset{\hat{\boldsymbol{q}} \in \{\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP(iter)}}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} \text{BIC}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{q}}})$$

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

Feature quantization: Model = quantization selection

New model selection criterion

We have drastically restricted the search space to clever candidates $\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP(iter)}}$ resulting from the the gradient descent steps.

$$(\boldsymbol{q}^\star, \boldsymbol{\theta}^\star) = \underset{\hat{\boldsymbol{q}} \in \{\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP(iter)}}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} \text{BIC}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{q}}})$$

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

In practice if $\forall i,\ q_{\alpha_{j,h}}(x_j) \ll 1$, then level h disappears while performing the argmax.

Feature quantization: Model = quantization selection

New model selection criterion

We have drastically restricted the search space to clever candidates $\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP(iter)}}$ resulting from the the gradient descent steps.

$$(\boldsymbol{q}^\star, \boldsymbol{\theta}^\star) = \underset{\hat{\boldsymbol{q}} \in \{\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP(iter)}}\}, \boldsymbol{\theta} \in \Theta_{\boldsymbol{m}}}{\operatorname{argmin}} \text{BIC}(\hat{\boldsymbol{\theta}}_{\hat{\boldsymbol{q}}})$$

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

In practice if $\forall i,\ q_{\alpha_{j,h}}(x_j) \ll 1$, then level h disappears while performing the argmax.

Start with $\boldsymbol{m} = (m_{\max})_1^d$ and "wait" …

Feature quantization: SEM-Gibbs

Originally (and as implemented in the R package glmdisc), the optimization was a bit different:

Feature quantization: SEM-Gibbs

Originally (and as implemented in the R package glmdisc), the optimization was a bit different:

- $\boldsymbol{q}$ is considered a latent (unobserved) feature;

Feature quantization: SEM-Gibbs

Originally (and as implemented in the R package glmdisc), the optimization was a bit different:

- $\boldsymbol{q}$ is considered a latent (unobserved) feature;
- A classical EM algorithm is intractable since it requires an Expectation step over all possible quantizations;

Feature quantization: SEM-Gibbs

Originally (and as implemented in the R package glmdisc), the optimization was a bit different:

- $\boldsymbol{q}$ is considered a latent (unobserved) feature;
- A classical EM algorithm is intractable since it requires an Expectation step over all possible quantizations;
- Solution: random draw $\approx$ Bayesian statistics;

SEM-Gibbs: estimation

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

SEM-Gibbs: estimation

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

There would still be a sum over $\mathcal{Q}_{\boldsymbol{m}}$:
$p(y|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \sum_{\boldsymbol{q} \in \mathcal{Q}_{\boldsymbol{m}}} p_{\boldsymbol{\theta}}(y|\boldsymbol{q}) \prod_{j=1}^{d} p_{\alpha_j}(\boldsymbol{q}_j|x_j)$

SEM-Gibbs: estimation

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

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

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

SEM-Gibbs: estimation

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

There would still be a sum over $\mathcal{Q}_{\boldsymbol{m}}$:
$p(y|\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\alpha}) = \sum_{\boldsymbol{q}\in\mathcal{Q}_{\boldsymbol{m}}} p_{\boldsymbol{\theta}}(y|\boldsymbol{q}) \prod_{j=1}^{d} p_{\alpha_j}(\boldsymbol{q}_j|x_j)$

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

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

SEM-Gibbs: estimation

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

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

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

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

Gibbs-sampling step:

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

Initialization

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

Loop

$$\begin{pmatrix} y_{\mathbf{1}} \\ \vdots \\ y_n \end{pmatrix} \overset{\substack{\text{logistic} \\ \text{regression}}}{\Rightarrow} \begin{pmatrix} \boldsymbol{q}_{\mathbf{1},\mathbf{1}} & \cdots & \boldsymbol{q}_{\mathbf{1},d} \\ \vdots & \vdots & \vdots \\ \boldsymbol{q}_{n,\mathbf{1}} & \cdots & \boldsymbol{q}_{n,d} \end{pmatrix} \overset{\substack{\text{polytomous} \\ \text{regression}}}{\Rightarrow} \begin{pmatrix} x_{\mathbf{1},\mathbf{1}} & \cdots & x_{\mathbf{1},d} \\ \vdots & \vdots & \vdots \\ x_{n,\mathbf{1}} & \cdots & x_{n,d} \end{pmatrix}$$

Updating $\boldsymbol{q}$

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

Calculating $\boldsymbol{q}^{\text{MAP}}$

$$\begin{pmatrix} \boldsymbol{q}^{\text{MAP},\mathbf{1},j} \\ \vdots \\ \boldsymbol{q}^{\text{MAP},n,j} \end{pmatrix} \overset{\substack{\text{MAP} \\ \text{estimate}}}{=} \begin{pmatrix} \operatorname{argmax}_{\boldsymbol{q}_j} p_{\alpha_{\boldsymbol{j}}}(\boldsymbol{q}_j|x_{\mathbf{1},j}) \\ \vdots \\ \operatorname{argmax}_{\boldsymbol{q}_j} p_{\alpha_{\boldsymbol{j}}}(\boldsymbol{q}_j|x_{n,j}) \end{pmatrix}$$

Feature quantization: Results

Simulated data

Table: For different sample sizes n, (A) CI of $\hat{c}_{j,2}$ for $c_{j,2} = 2/3$. (B) CI of $\hat{m}$ for $m_1 = 3$. (C) CI of $\hat{m}_3$ for $m_3 = 1$.

n	(A) $\hat{c}_{j,2}$	(B)	$\hat{m}_1$	(C)	$\hat{m}_3$
1,000	$[0.656, 0.666]$	1 90 9		60 32 8	
10,000	$[0.666, 0.666]$	0 100 0		88 12 0	

Feature quantization: Results

UCI data

Table: Gini indices (the greater the value, the better the performance) of our proposed quantization algorithm *glmdisc* and two baselines: ALLR and MDLP / χ^2 tests obtained on several benchmark datasets from the UCI library.

Dataset	ALLR	MDLP/χ^2	*glmdisc*
Adult	81.4 (1.0)	85.3 (0.9)	80.4 (1.0)
Australian	72.1 (10.4)	84.1 (7.5)	92.5 (4.5)
Bands	48.3 (17.8)	47.3 (17.6)	58.5 (12.0)
Credit	81.3 (9.6)	88.7 (6.4)	92.0 (4.7)
German	52.0 (11.3)	54.6 (11.2)	69.2 (9.1)
Heart	80.3 (12.1)	78.7 (13.1)	86.3 (10.6)

Feature quantization: Results

CACF data

Table: Gini indices (the greater the value, the better the performance) of our proposed quantization algorithm *glmdisc*, the two baselines of Table 4 and the current scorecard (manual / expert representation) obtained on several portfolios of Crédit Agricole Consumer Finance.

Portfolio	ALLR	Current	MDLP/χ^2	*glmdisc*
Automobile	59.3 (3.1)	55.6 (3.4)	59.3 (3.0)	58.9 (2.6)
Renovation	52.3 (5.5)	50.9 (5.6)	54.0 (5.1)	56.7 (4.8)
Standard	39.7 (3.3)	37.1 (3.8)	45.3 (3.1)	44.0 (3.1)
Revolving	62.7 (2.8)	58.5 (3.2)	63.2 (2.8)	62.3 (2.8)
Mass retail	52.8 (5.3)	48.7 (6.0)	61.4 (4.7)	61.8 (4.6)
Electronics	52.9 (11.9)	55.8 (10.8)	56.3 (10.2)	72.6 (7.4)

See this gist for χ^2 automated grouping tests.

Bivariate interactions

Bivariate interactions: Notations

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

$$\mathrm{logit}(p_{\boldsymbol{\theta}_{\boldsymbol{f}}}(1|\boldsymbol{q}(\boldsymbol{x}))) = \theta_0 + \sum_{j=1}^{d} \theta_j^{\boldsymbol{q}_j(x_j)} + \sum_{1 \leq k < \ell \leq d} \delta_{k,\ell} \theta_{k,\ell}^{\boldsymbol{q}_k(x_k) f_\ell(x_\ell)}$$

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

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

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

$$(\boldsymbol{\theta}^\star, \boldsymbol{\delta}^\star) = \underset{\boldsymbol{\theta}, \boldsymbol{\delta} \in \{0,1\}^{\frac{d(d-1)}{2}}}{\text{argmax}} \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{q}(\boldsymbol{x}_i), \boldsymbol{\delta}) - \text{penalty}(n; \boldsymbol{\theta})$$

Bivariate interactions: Notations

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

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

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

$$(\boldsymbol{\theta}^\star, \boldsymbol{\delta}^\star) = \underset{\boldsymbol{\theta}, {\color{red}\delta \in \{0,1\}^{\frac{d(d-1)}{2}}}}{\text{argmax}} \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{q}(\boldsymbol{x}_i), \boldsymbol{\delta}) - \text{penalty}(n; \boldsymbol{\theta})$$

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

Bivariate interactions: Model proposal

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

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

Strategy used here: Metropolis-Hastings sampling algorithm.
Idea: Propose "clever" interactions and accept / reject them based on the BIC criterion of the resulting logistic regression.

Bivariate interactions: Model proposal

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

Strategy used here: Metropolis-Hastings sampling algorithm.
Idea: Propose “clever” interactions and accept / reject them based on the BIC criterion of the resulting logistic regression.

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

$$p(\boldsymbol{\delta}|\boldsymbol{q}, y) \propto \exp(-\mathrm{BIC}[\delta]/2)p(\boldsymbol{\delta})$$

Bivariate interactions: Model proposal

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

Strategy used here: Metropolis-Hastings sampling algorithm.
Idea: Propose "clever" interactions and accept / reject them based on the BIC criterion of the resulting logistic regression.

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

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

Bivariate interactions: Model proposal

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

Strategy used here: Metropolis-Hastings sampling algorithm.
Idea: Propose "clever" interactions and accept / reject them based on the BIC criterion of the resulting logistic regression.

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

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

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

Bivariate interactions: Model proposal

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

Bivariate interactions: Model proposal

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

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

Bivariate interactions: 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.

Bivariate interactions: 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.

If the interaction between two features is meaningful when only these two features are considered, there is a (provably) good chance that it will be in the full multivariate model.

Bivariate interactions: 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.

If the interaction between two features is meaningful when only these two features are considered, there is a (provably) good chance that it will be in the full multivariate model.

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

Données UCI

Table: Gini indices (the greater the value, the better the performance) of our proposed quantization algorithm *glmdisc* and two baselines: ALLR and MDLP / χ^2 tests obtained on several benchmark datasets from the UCI library.

Dataset	ALLR	*ad hoc* methods	Our proposal: *glmdisc*-NN	Our proposal: *glmdisc*-SEM	*glmdisc*-SEM w. interactions
Adult	81.4 (1.0)	**85.3** (0.9)	80.4 (1.0)	81.5 (1.0)	81.5 (1.0 - no interaction)
Australian	72.1 (10.4)	84.1 (7.5)	92.5 (4.5)	**100** (0)	**100** (0 - no interaction)
Bands	48.3 (17.8)	47.3 (17.6)	58.5 (12.0)	**58.7** (12.0)	**58.8** (13.0)
Credit	81.3 (9.6)	88.7 (6.4)	**92.0** (4.7)	87.7 (6.4)	87.7 (6.4 - no interaction)
German	52.0 (11.3)	54.6 (11.2)	**69.2** (9.1)	54.5 (10)	
Heart	80.3 (12.1)	78.7 (13.1)	**86.3** (10.6)	82.2 (11.2)	84.5 (10.8)

Bivariate interactions: Results

Medicine data

Table: Gini indices of our proposed quantization algorithm *glmdisc*-SEM and two baselines: ALLR and ALLR with all pairwise interactions on several medicine-related benchmark datasets.

	Pima	Breast	Birthwt
ALLR	**73.0**	94.0	34.0
ALLR LR w. interactions	60.0	51.0	15.0
glmdisc	57.0	93.0	18.0
glmdisc w. interactions	62.0	**95.0**	**54.0**

CACF data

Table: Gini indices (the greater the value, the better the performance) of our proposed quantization algorithm *glmdisc*, the two baselines of Table 4 and the current scorecard (manual / expert representation) obtained on several portfolios of Crédit Agricole Consumer Finance.

Portfolio	ALLR	Current performance	*ad hoc* methods	Our proposal: *glmdisc*-NN	Our proposal: *glmdisc*-SEM	*glmdisc*-SEM w. interactions
Automobile	59.3 (3.1)	55.6 (3.4)	59.3 (3.0)	58.9 (2.6)	57.8 (2.9)	**64.8** (2.0)
Renovation	52.3 (5.5)	50.9 (5.6)	54.0 (5.1)	**56.7** (4.8)	55.5 (5.2)	55.5 (5.2)
Standard	39.7 (3.3)	37.1 (3.8)	45.3 (3.1)	43.8 (3.2)	36.7 (3.7)	**47.2** (2.8)
Revolving	62.7 (2.8)	58.5 (3.2)	63.2 (2.8)	62.3 (2.8)	60.7 (2.8)	**67.2** (2.5)
Mass retail	52.8 (5.3)	48.7 (6.0)	61.4 (4.7)	**61.8** (4.6)	61.0 (4.7)	60.3 (4.8)
Electronics	52.9 (11.9)	55.8 (10.8)	56.3 (10.2)	**72.6** (7.4)	62.0 (9.5)	63.7 (9.0)

Bivariate interactions: Results

Older results

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

Segmentation: logistic regression trees

Segmentation: logistic regression trees

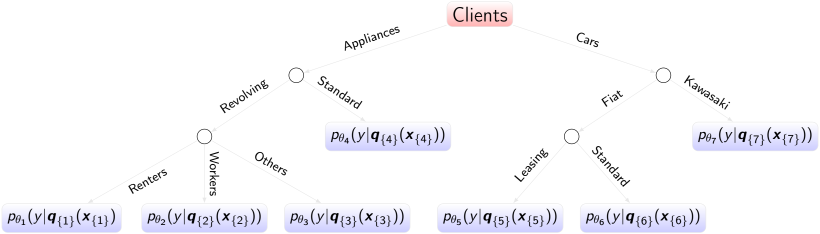


Figure: Scorecards tree structure in acceptance system.

Segmentation: logistic regression trees: Notations

K segments.

Segmentation: logistic regression trees: Notations

K segments.

$c \in \{1, \ldots, K\}$: latent feature of the client's segment.

Segmentation: logistic regression trees: Notations

K segments.

$c \in \{1, \dots, K\}$: latent feature of the client's segment.

We suppose there is a true segmentation $c^\star, K^\star$ and logistic regressions $\boldsymbol{\theta}^{\star, c^\star}$ at its leaves.

Segmentation: logistic regression trees: Notations

K segments.

$c \in \{1, \ldots, K\}$: latent feature of the client's segment.

We suppose there is a true segmentation $c^\star, K^\star$ and logistic regressions $\boldsymbol{\theta}^{\star, c^\star}$ at its leaves.

If we could evaluate all segmentations, the true one would be selected by

$$\underset{c,K}{\operatorname{argmax}} \sum_{c=1}^{K} \mathrm{BIC}(\hat{\boldsymbol{\theta}}^c),$$

where $\hat{\boldsymbol{\theta}}^c$ is the MLE of the logistic regression on .

Segmentation: logistic regression trees: Model proposal

Similarly to the quantization proposal: ability to be in several segments at a time.

Segmentation: logistic regression trees: Model proposal

Similarly to the quantization proposal: ability to be in several segments at a time.

$$p(y|\boldsymbol{x}) = \sum_{c=1}^{K} p_{\boldsymbol{\theta}}(y|\boldsymbol{x}; c) p_{\beta}(c|\boldsymbol{x}).$$

Segmentation: logistic regression trees: Model proposal

Similarly to the quantization proposal: ability to be in several segments at a time.

$$p(y|\boldsymbol{x}) = \sum_{c=1}^{K} p_{\boldsymbol{\theta}}(y|\boldsymbol{x};c)p_{\beta}(c|\boldsymbol{x}).$$

$$c_i^{(s+1)} \sim p_{\boldsymbol{\theta}^{\cdot(s)}}(y_i|\boldsymbol{x}_i)p_{\beta^{(s)}}(\cdot|\boldsymbol{x}_i).$$

Segmentation: logistic regression trees: Model proposal

Similarly to the quantization proposal: ability to be in several segments at a time.

$$p(y|\boldsymbol{x}) = \sum_{c=1}^{K} p_{\boldsymbol{\theta}}(y|\boldsymbol{x}; c) p_{\beta}(c|\boldsymbol{x}).$$

$$c_i^{(s+1)} \sim p_{\boldsymbol{\theta}^{\cdot(s)}}(y_i|\boldsymbol{x}_i) p_{\beta^{(s)}}(\cdot|\boldsymbol{x}_i).$$

$$\boldsymbol{\theta}^{c(s+1)} = \underset{\boldsymbol{\theta}^c}{\operatorname{argmax}} \sum_{i=1}^{n} \mathbb{1}_c(c_i^{s+1)}) \ln p_{\boldsymbol{\theta}^c}(y_i|\boldsymbol{x}_i, c).$$

Segmentation: logistic regression trees: Model proposal

Similarly to the quantization proposal: ability to be in several segments at a time.

$$p(y|\boldsymbol{x}) = \sum_{c=1}^{K} p_{\boldsymbol{\theta}}(y|\boldsymbol{x}; c) p_{\beta}(c|\boldsymbol{x}).$$

$$c_i^{(s+1)} \sim p_{\boldsymbol{\theta}^{\cdot(s)}}(y_i|\boldsymbol{x}_i) p_{\beta^{(s)}}(\cdot|\boldsymbol{x}_i).$$

$$\boldsymbol{\theta}^{c(s+1)} = \underset{\boldsymbol{\theta}^c}{\operatorname{argmax}} \sum_{i=1}^{n} \mathbb{1}_c(c_i^{s+1)}) \ln p_{\boldsymbol{\theta}^c}(y_i|\boldsymbol{x}_i, c).$$

$$\beta^{(s+1)} = \text{C4.5}(\mathbf{c}^{(s+1)}, \mathrm{x}).$$

Segmentation: logistic regression trees: Some results

	Oracle = ALLR	*glmtree*-SEM	FAMD	PLS	LMT	MOB
Gini	69.7	**69.7**	65.3	47.0	**69.7**	64.8

	Oracle	ALLR	*glmtree*-SEM	FAMD	PLS	LMT	MOB
Gini	69.7	25.8	**69.7**	17.7	48.4	65.8	**69.7**

Bonus

Bonus I

Big “unstructured” data
Some theoretical results about an ever bigger d (not the one you think about though).

Online logistic regression
What if we dynamically adjusted logistic regression coefficients of a given scorecard (still learnt on a cold database) on new data as they come in?

Bonus II

Profitability
Good / bad label is merely a proxy of the true performance measure: profitability.
Already done: weighting observations by the amount of the loan gives rise to roughly the same logistic regression coefficients.

Predicting IR3 in 2 months based on the month's applications
Current process: finance people, wait 3 months, if risk $\neq$ budget then adjust acceptance policy, wait 3 months again and repeat.
Couldn't we anticipate by looking at the quality (*e.g.* through the score) of the applications?

Thanks!

References I

John Banasik and Jonathan Crook. "Reject inference, augmentation, and sample selection". In: European Journal of Operational Research 183.3 (2007), pp. 1582–1594. url: http://www.sciencedirect.com/science/article/pii/S0377221706011969 (visited on 08/25/2016).

Asma Guizani et al. "Une Comparaison de quatre Techniques d'Inférence des Refusés dans le Processus d'Octroi de Crédit". In: 45 èmes Journées de statistique. 2013. url: http://cedric.cnam.fr/fichiers/art_2753.pdf (visited on 08/25/2016).

Ha Thu Nguyen. Reject inference in application scorecards: evidence from France. Tech. rep. University of Paris West-Nanterre la Défense, EconomiX, 2016. url: http://economix.fr/pdf/dt/2016/WP_EcoX_2016-10.pdf (visited on 08/25/2016).

References II

Françoise Fogelman Soulié and Emmanuel Viennet. “Le Traitement des Refusés dans le Risque Crédit”. In: Revue des Nouvelles Technologies de l’Information Data Mining et Apprentissage Statistique : application en assurance, banque et marketing, RNTI-A-1 (2007), pp. 22–44.

Bianca Zadrozny. “Learning and evaluating classifiers under sample selection bias”. In: Proceedings of the twenty-first international conference on Machine learnin ACM. 2004, p. 114.