Feature quantization for parsimonious and interpretable predictive models

Adrien Ehrhardt

Christophe Biernacki
Philippe Heinrich
Vincent Vandewalle

Modal's Days 2019

24/01/2019

Table of Contents

Context, basic notations, combinatorics of the problem

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.

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;∞[		1
Executive			Single	]2000;∞[		1
Office employee			Married	]-∞ ; 1500]		0
Worker			?	]-∞ ; 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
...	...	...

Table: Final scorecard.

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

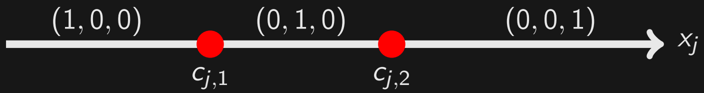
$$\boldsymbol{q}(\boldsymbol{x}) = (\boldsymbol{q}_1(x_1), \ldots, \boldsymbol{q}_d(x_d))$$
$$\boldsymbol{q}_j(x_j) = (q_{j,h}(x_j))_1^{m_j} \text{ (one-hot encoding)}$$
$$q_{j,h}(\cdot) = 1 \text{ if } x_j \in C_{j,h}, 0 \text{ otherwise, } 1 \leq h \leq m_j$$

Notations II

Discretization

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

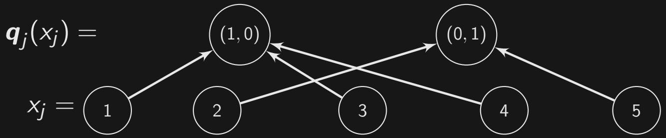
where $c_{j,1}, \ldots, c_{j,m_j-1}$ are increasing numbers called cutpoints, $c_{j,0} = -\infty$ and $c_{j,m_j} = +\infty$.



Notations III

Grouping

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



Notations IV

Embedding Feature Engineering in the predictive task

$$\begin{aligned} \mathcal{X} &\to \mathcal{Q} &&\to \mathcal{Y} \\ \boldsymbol{x} &\mapsto \boldsymbol{q}(\boldsymbol{x}) &&\mapsto y \end{aligned}$$

n - sample

$$(\mathrm{x}, \mathbf{y}) = (\boldsymbol{x}_i, y_i)_1^n$$

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:

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

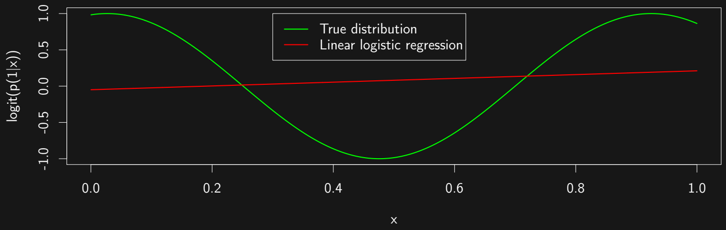


Figure: Linear logistic regression fit

Example

Logistic regression on discretized data:

$$\text{logit}(p_{\boldsymbol{\theta}_{\boldsymbol{q}}}(1|\boldsymbol{q}(\boldsymbol{x}))) = \theta_0 + \underbrace{\theta_1' \cdot \boldsymbol{q}_1(x_1)}_{\theta_1^1,\ldots,\theta_1^{50}}$$

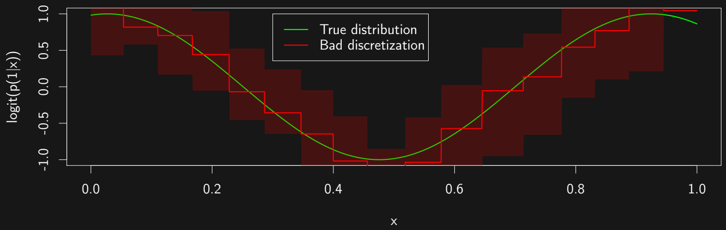


Figure: Bad (high variance) discretization

Example

Logistic regression on discretized data:

$$\text{logit}(p_{\boldsymbol{\theta}_{\boldsymbol{q}}}(1|\boldsymbol{q}(\boldsymbol{x}))) = \theta_0 + \underbrace{\theta_1' \cdot \boldsymbol{q}_1(x_1)}_{\theta_1^1,\ldots,\theta_1^3}$$

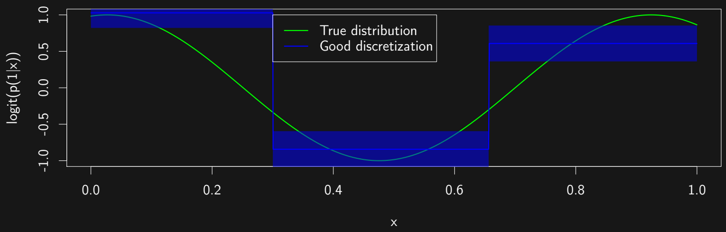


Figure: Good (bias/variance tradeoff) discretization

Model selection

Logistic regression coefficient $\hat{\theta}_{\boldsymbol{q}}$ given via MLE

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

Best quantization $\hat{\boldsymbol{q}}$ given by *e.g.* BIC

$$\hat{\boldsymbol{q}} = \operatorname*{argmin}_{\boldsymbol{q} \in \mathcal{Q}} \mathrm{BIC}(\hat{\theta}_{\boldsymbol{q}})$$

Obvious problem: $\mathcal{Q}$ is huge!

- $d = 10$ categorical features
- $l_j = 4$ levels each
- $|\mathcal{Q}| \approx 6 \cdot 10^{11}$

Supervised multivariate quantization: a relaxation

Smooth approximation of the quantization

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

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

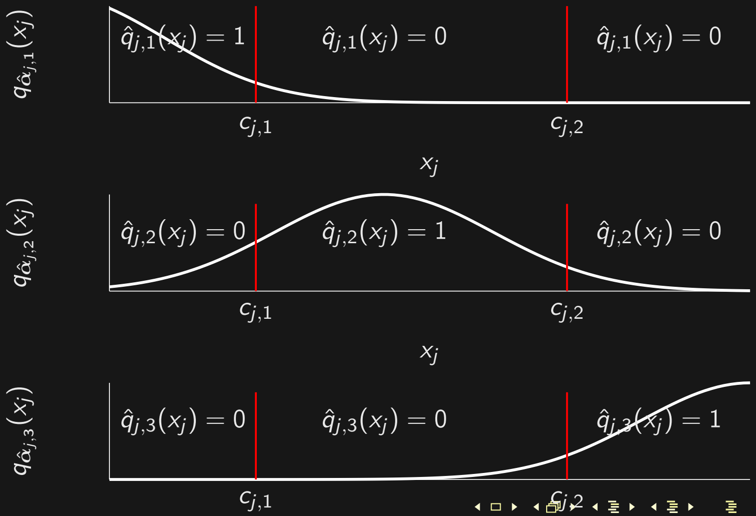
$$q_{\boldsymbol{\alpha}_{j,h}}(\cdot) = \frac{\exp(\alpha^0_{j,h} + \alpha^1_{j,h} \cdot)}{\sum_{g=1}^{m_j} \exp(\alpha^0_{j,g} + \alpha^1_{j,g} \cdot)}.$$

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

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

Go back to "hard" thresholding: MAP estimation

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



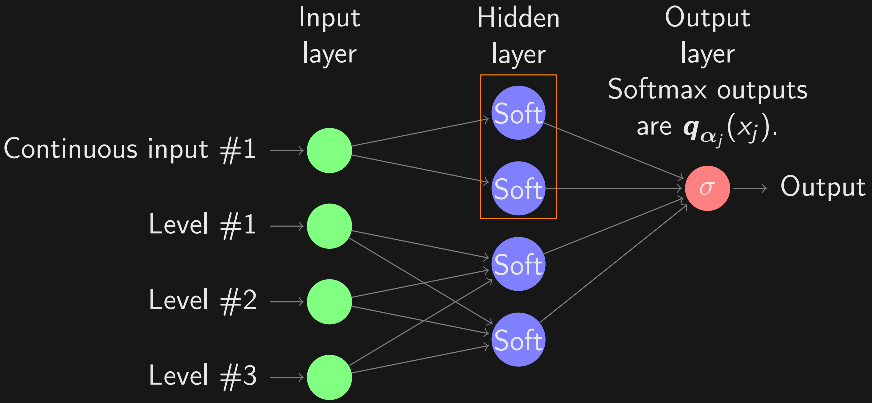
Validity of the approach

$$\ell_{\boldsymbol{q}_\alpha}(\boldsymbol{\theta}; (\mathbf{x}, \mathbf{y})) = \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(y_i | \boldsymbol{q}_{\boldsymbol{\alpha}}(\boldsymbol{x}_i))$$

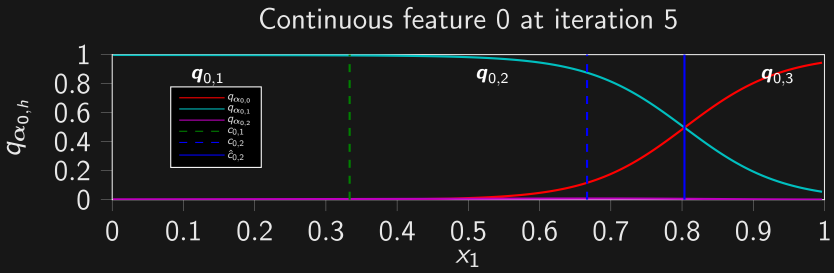
$$q_{j,h}^{\text{MAP}}(x_j) = 1 \text{ if } h = \underset{1 \leq h' \leq m_j}{\text{argmax}}\, q_{\hat{\alpha}_{j,h'}}, 0 \text{ otherwise.}$$

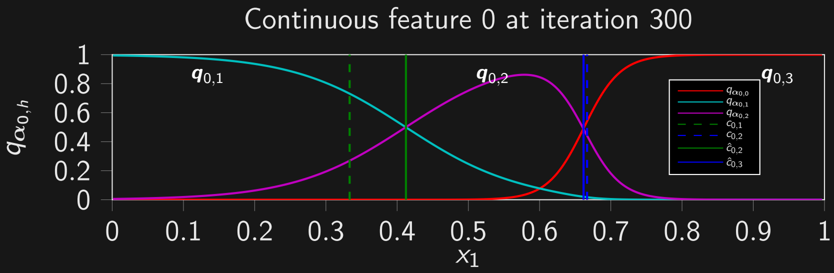
1. MAP procedure yields contiguous intervals [Samé et al., 2011].
2. The $\boldsymbol{\alpha}$ parameters can be written explicitly w.r.t. cutpoints [Chamroukhi et al., 2009].
3. Under classical regularity conditions and if the model is well-specified, maximizing $\ell_{\boldsymbol{q}_\alpha}(\boldsymbol{\theta}; (\mathbf{x}, \mathbf{y}))$ w.r.t. $(\boldsymbol{\alpha}, \boldsymbol{\theta})$ is equivalent to maximizing $\ell_{\boldsymbol{q}}(\boldsymbol{\theta}; (\mathbf{x}, \mathbf{y}))$ over $(\boldsymbol{q}, \boldsymbol{\theta})$ which was untractable.

Estimation *via* neural networks



Estimation *via* neural networks





In the end: the best discretization

New model selection criterion

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

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

In the end: the best discretization

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.

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

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 clever candidates $\boldsymbol{q}^{\text{MAP}(1)}, \ldots, \boldsymbol{q}^{\text{MAP(iter)}}$ resulting from the the gradient descent steps.

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

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.

In the end: the best discretization

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.

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

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" ...

Results

Results: NN + 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	60
1,000	$[0.656, 0.666]$	90	32
		9	8
		0	88
10,000	$[0.666, 0.666]$	100	12
		0	0

Results: NN + 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)

Results: NN + Credit Scoring 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)	43.8 (3.2)
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)	61.4 (4.6)
Electronics	52.9 (11.9)	55.8 (10.8)	56.3 (10.2)	72.6 (7.4)

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...),
- Big gain for statisticians relying on logistic regression.

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- Big gain for statisticians relying on logistic regression.
- Implementation in Python/TensorFlow/Keras to be released.

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- Big gain for statisticians relying on logistic regression.
- Implementation in Python/TensorFlow/Keras to be released.

Perspectives

- Tested for logistic regression: adaptable to other models p_θ!

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- Big gain for statisticians relying on logistic regression.
- Implementation in Python/TensorFlow/Keras to be released.

Perspectives

- Tested for logistic regression: adaptable to other models p_θ!
- To be compared with SEM approach:

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- Big gain for statisticians relying on logistic regression.
- Implementation in Python/TensorFlow/Keras to be released.

Perspectives

- Tested for logistic regression: adaptable to other models p_θ!
- To be compared with SEM approach:
  - R implementation of glmdisc available on Github, to be submitted to CRAN.

Take-aways

Conclusion

- Interpretability + good empirical results and statistical guarantees (to some extent...),
- Big gain for statisticians relying on logistic regression.
- Implementation in Python/TensorFlow/Keras to be released.

Perspectives

- Tested for logistic regression: adaptable to other models p_θ!
- To be compared with SEM approach:
  - R implementation of glmdisc available on Github, to be submitted to CRAN.
  - Python implementation of glmdisc available on Github and PyPi.

Thanks!

References

Chamroukhi, F., Samé, A., Govaert, G., and Aknin, P. (2009).
A regression model with a hidden logistic process for feature extraction from time series.
In Neural Networks, 2009. IJCNN 2009. International Joint Conference on, pages 489–496. IEEE.

Samé, A., Chamroukhi, F., Govaert, G., and Aknin, P. (2011).
Model-based clustering and segmentation of time series with changes in regime.
Advances in Data Analysis and Classification, 5(4):301–321.