

Binarsity: Penalization for One-Hot Encoded Features

Mokhtar Z. Alaya



Journées MAS 2022
Université Rouen Normandie

Joint work with:



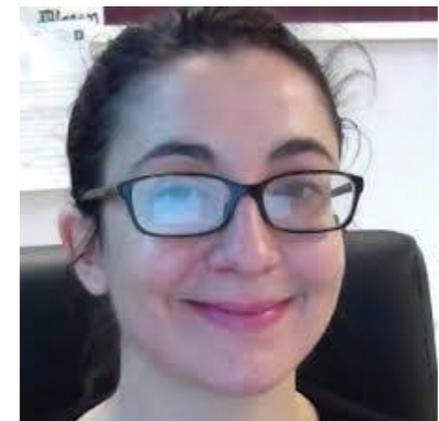
Simon Bussy

ML Lab of Califrais



Stéphane Gaïffas

LPSM, Univ. Paris Diderot



Agathe Guilloux

LaMME, Univ. Evry

Outline

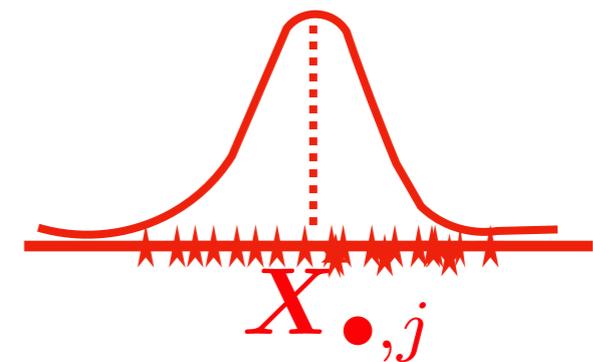
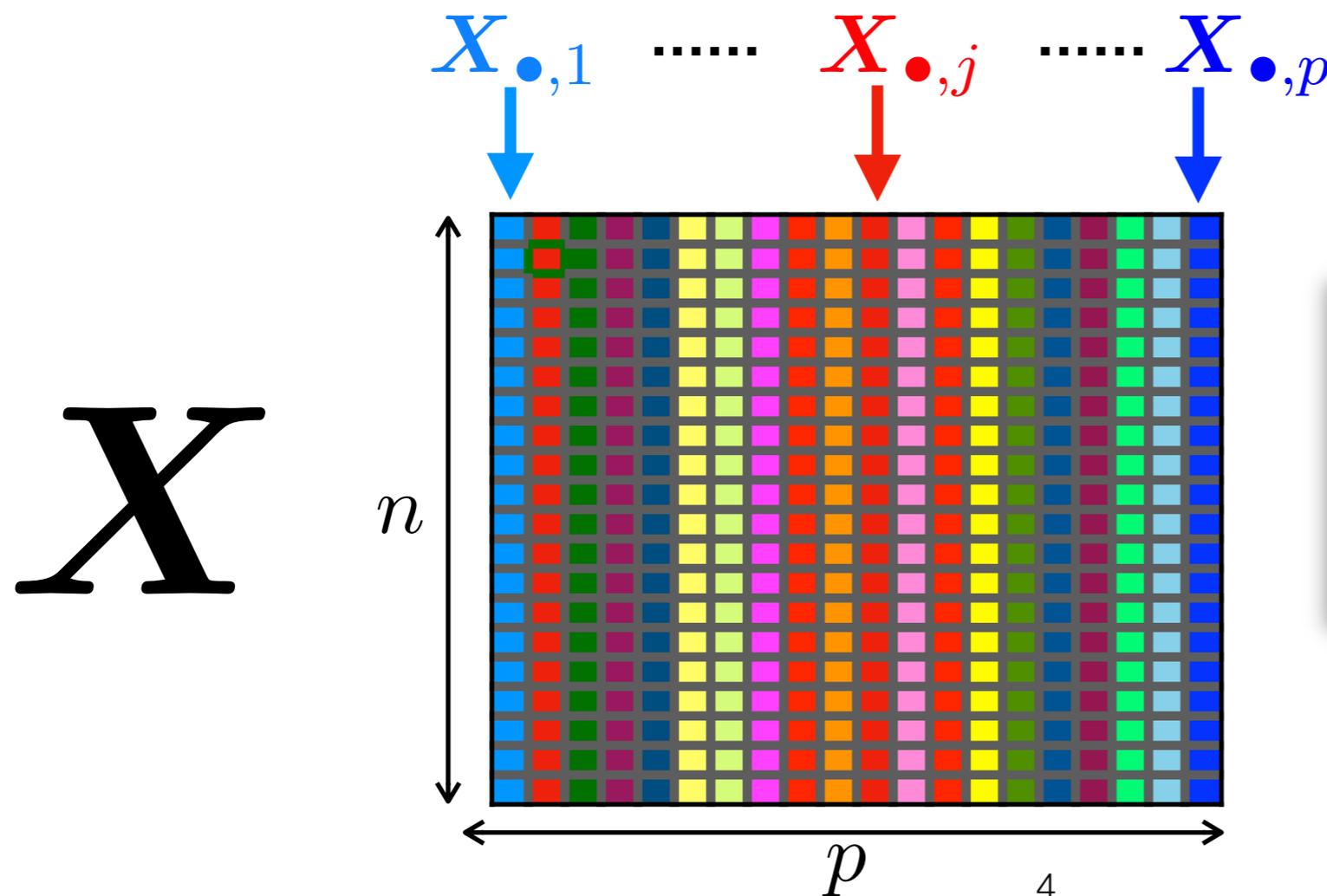
- 1. Features Binarization
- 2. Theoretical Gurantees for GLM + Binarsity
- 3. Numerical Experiments

I. Features Binarization

Supervised Learning: Setting

- Supervised training dataset $\mathcal{D}_n = \{(x_i, y_i) : i = 1, \dots, n\}$ with features $x_i = [x_{i,1}, \dots, x_{i,p}]^T \in \mathbb{R}^p$ and labels $y_i \in \mathcal{Y} \subset \mathbb{R}$

- Features matrix $\mathbf{X} = [x_{i,j}]_{1 \leq i \leq n; 1 \leq j \leq p} \in \mathbb{R}^{n \times p}$ with several continuous features



A well know-trick:
**One-Hot Encoding of
continuous features**

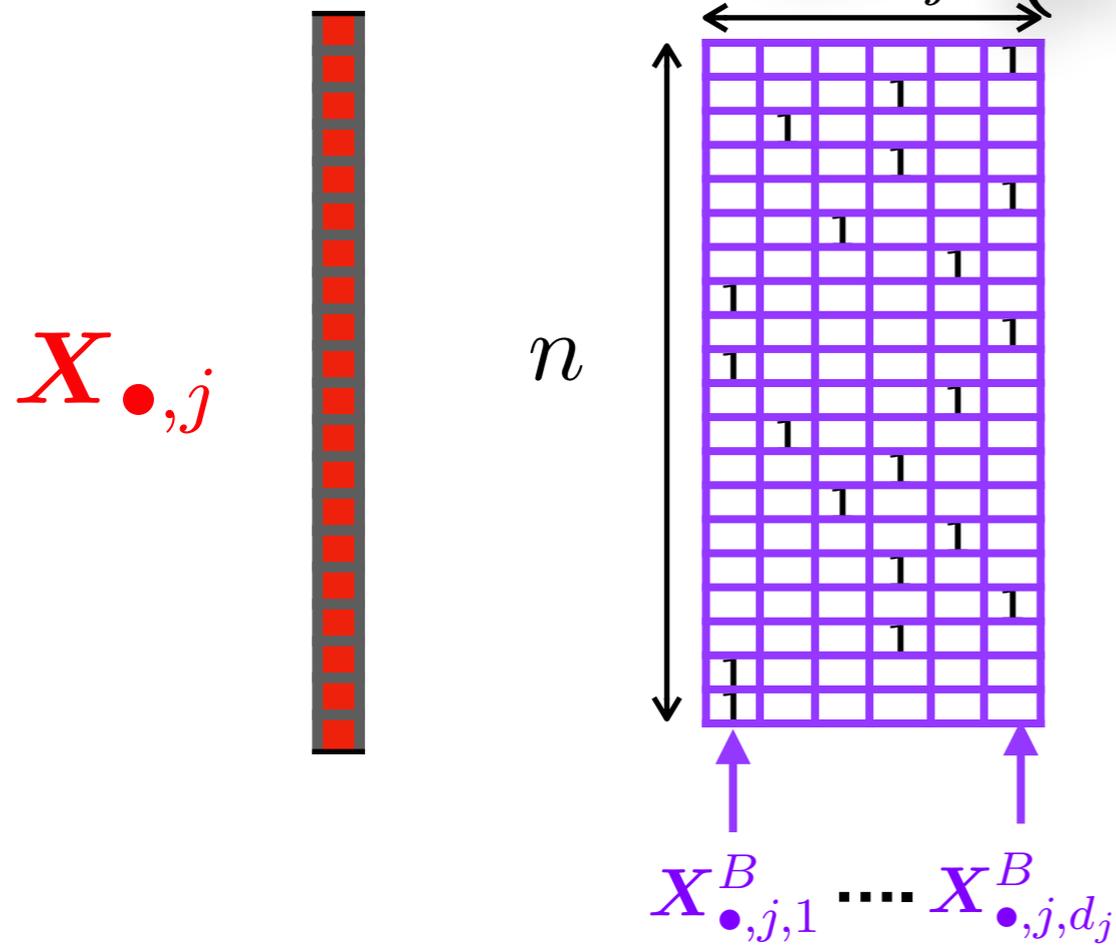
[Lieu et al.('02);
Wu and Coggeshall ('12)]

Features Binarization: Setup of One-Hot Encoding

Binarization Setup:

- The j -th column $X_{\bullet,j}$ is replaced by a number $d_j \geq 2$ of binary columns (containing only zeros and ones) $X_{\bullet,j,1}^B, \dots, X_{\bullet,j,d_j}^B$
- Partition of the range of values of $X_{\bullet,j}$ into intervals $I_{j,1}, \dots, I_{j,d_j}$ and put:

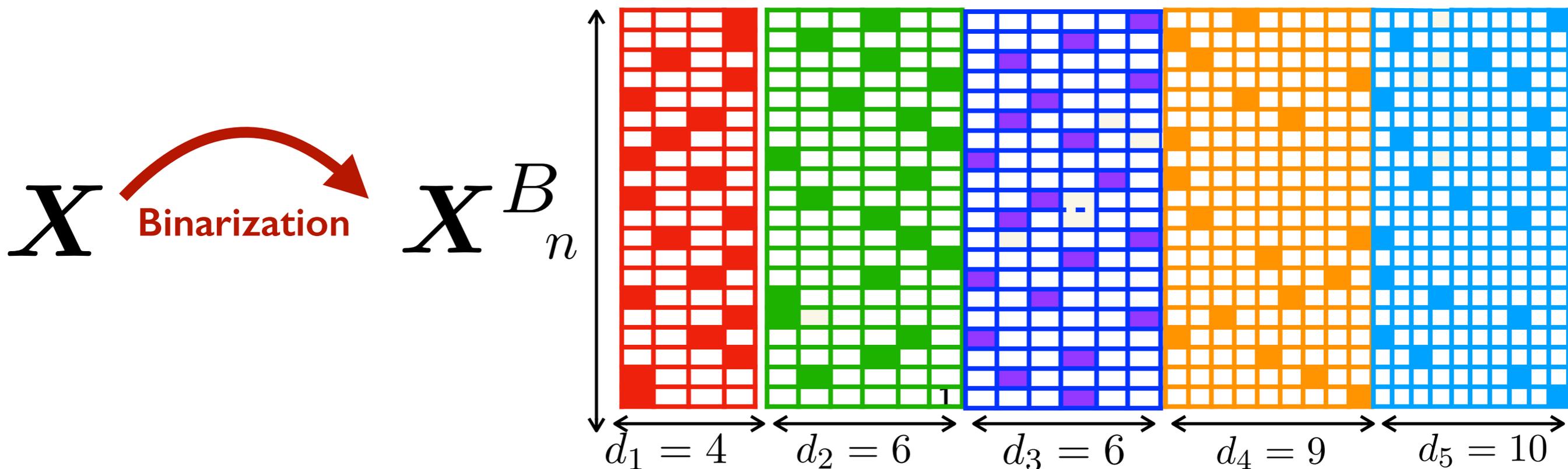
$$x_{i,j,k}^B = \begin{cases} 1, & \text{if } x_{i,j} \in I_{j,k}, \\ 0, & \text{otherwise} \end{cases}$$



Features Binarization: Setup of One-Hot Encoding

- If $X_{\bullet,j}$ takes values (modalities) in the set $\{1, \dots, M_j\}$ with cardinality M_j , we take $d_j = M_j$ and use one-hot coding of each modality by defining:

$$x_{i,j,k}^B = \begin{cases} 1, & \text{if } x_{i,j} = k, \\ 0, & \text{otherwise} \end{cases}$$



Binarized Features Matrix

Features Binarization: Inter-quantile Partition Intervals

- The i -th row of the binarized matrix \mathbf{X}^B reads as

$$x_i^B = [x_{i,1,1}^B, \dots, x_{i,1,d_1}^B, x_{i,2,1}^B, \dots, x_{i,2,d_2}^B, \dots, x_{i,p,1}^B, \dots, x_{i,p,d_p}^B]^\top \in \mathbb{R}^d$$

where $d = \sum_{j=1}^p d_j$

- Choice of the $I_{j,k}$ Intervals?  Natural Choice: Inter-quantile intervals

$$I_{j,1} = \left[q_j(0), q_j\left(\frac{1}{d_j}\right) \right] \text{ and } I_{j,k} = \left(q_j\left(\frac{k-1}{d_j}\right), q_j\left(\frac{k}{d_j}\right) \right]$$

for $k = 2, \dots, d_j$, and where $q_j(\alpha)$ = quantile of order $\alpha \in [0, 1]$ of $\mathbf{X}_{\bullet,j}$.

Features Binarization: Example

tick

ML /Python Package

(Bacry et al., 2018)

```
import numpy as np
import pandas as pd
pd.option_context('display.max_rows', None,
                  'display.max_columns', None)
import prettytable
import seaborn as sns
# tick
from tick.preprocessing import FeaturesBinarizer

# Origin matrix
features = np.array([[0.00902084, 0.46519565, 'z'],
                    [0.46599565, 3.46523565, 2.],
                    [0.82091721, -1.2650095, 2.],
                    [-0.17315496, 7.86545565, 1.],
                    [4.08180209, 6.26569565, 0.],
                    [1.6011727, 0.36548565, 0.],
                    [2.7347947, 1.46500565, 20.],
                    [-5.9890938, 4.55529565, 0.],
                    [6.3063761, 2.22548565, 1.],
                    [9.27110903, -3.46514565, 0.]])

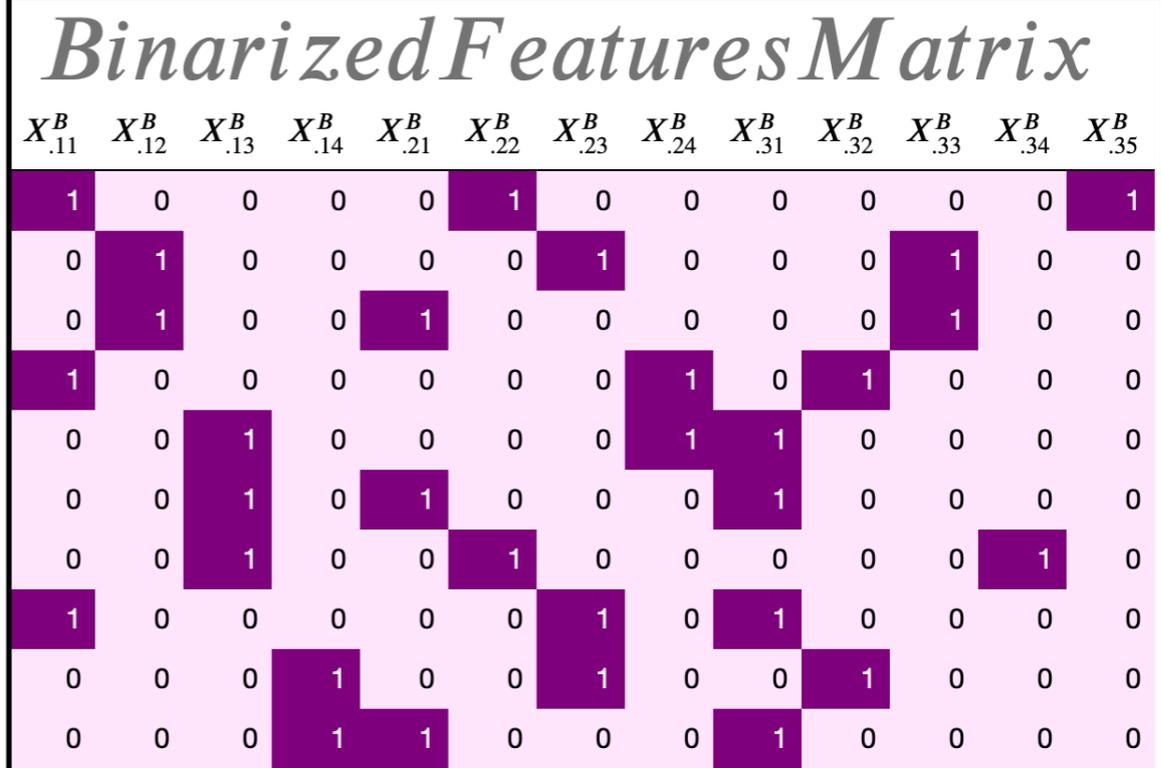
df = pd.DataFrame(data=features)

# Binarization preprocessing with $d_j=4$
# for continuous features
binarizer = FeaturesBinarizer(n_cuts=3)
binarized_features = binarizer.fit_transform(features)

# binarized matrix X^B
X_bin = binarized_features.toarray()
# print(X_bin.shape) # (10, 13)

columns_bin =
["$X^B_{.11}$", "$X^B_{.12}$", "$X^B_{.13}$", "$X^B_{.14}$",
"$X^B_{.21}$", "$X^B_{.22}$", "$X^B_{.23}$", "$X^B_{.24}$",
"$X^B_{.31}$", "$X^B_{.32}$", "$X^B_{.33}$", "$X^B_{.34}$",
"$X^B_{.35}$"]
df_bin = pd.DataFrame(data=X_bin, columns=columns_bin)
pd.options.display.float_format = '{:,.0f}'.format

# plot
cm = sns.light_palette("purple", as_cmap=True)
(df_bin.style
 .background_gradient(cmap=cm)
 # .highlight_max(subset=['total_amt_usd_diff', 'total_amt_usd_pct_diff'])
 .set_caption('${\qqquad \qqquad \qqquad \Huge{Binarized Features Matrix}}$')
 .format({'total_amt_usd_pct_diff': "{:.2%}"))
```



Features Binarization: Weights

- **Weights of one-hot encoded features**

To each binarized feature $X_{\bullet,j,k}^B$ corresponds a parameter $\theta_{j,k}$

- The parameters associated to the binarization of the j -th feature is denoted

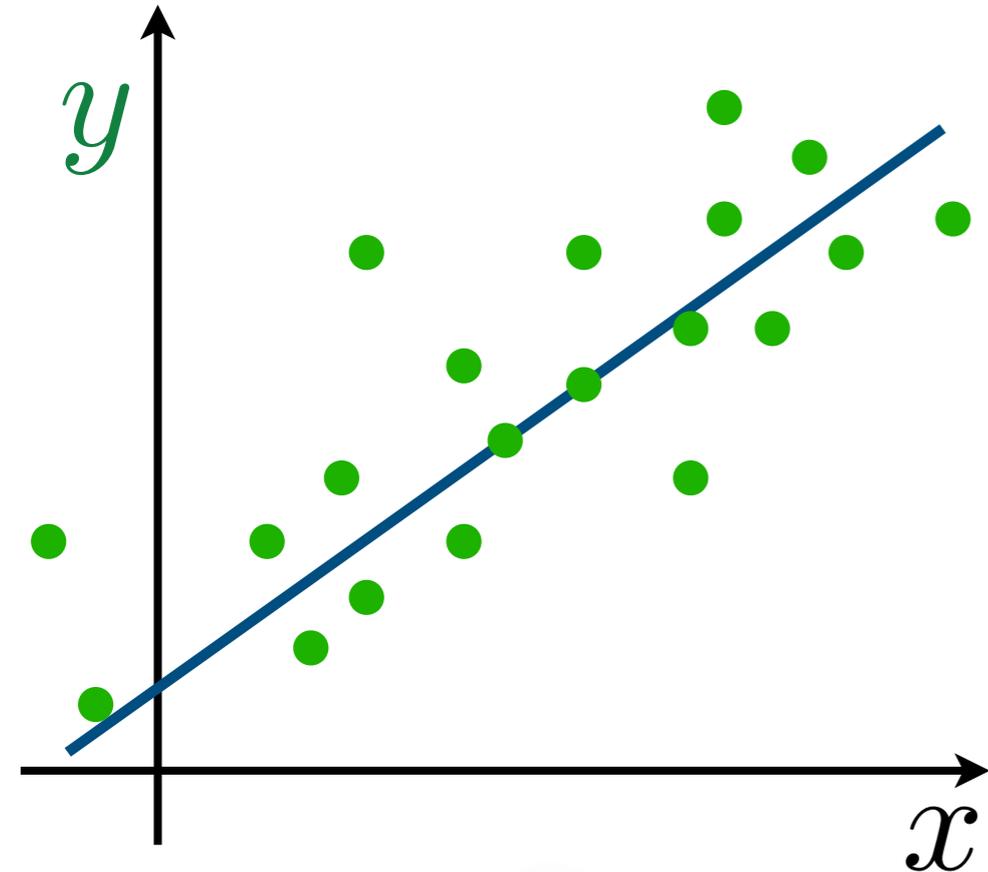
$$\theta_{j,\bullet} = [\theta_{j,1} \cdots \theta_{j,d_j}]^\top$$

- The full parameters vector of size $d = \sum_{j=1}^p d_j$, is simply

$$\theta = [\theta_{1,1} \cdots \theta_{1,d_1} \theta_{2,1} \cdots \theta_{2,d_2} \cdots \theta_{p,1} \cdots \theta_{p,d_p}]^\top \in \mathbb{R}^d$$

Features Binarization: Weights

Linear regression on raw features

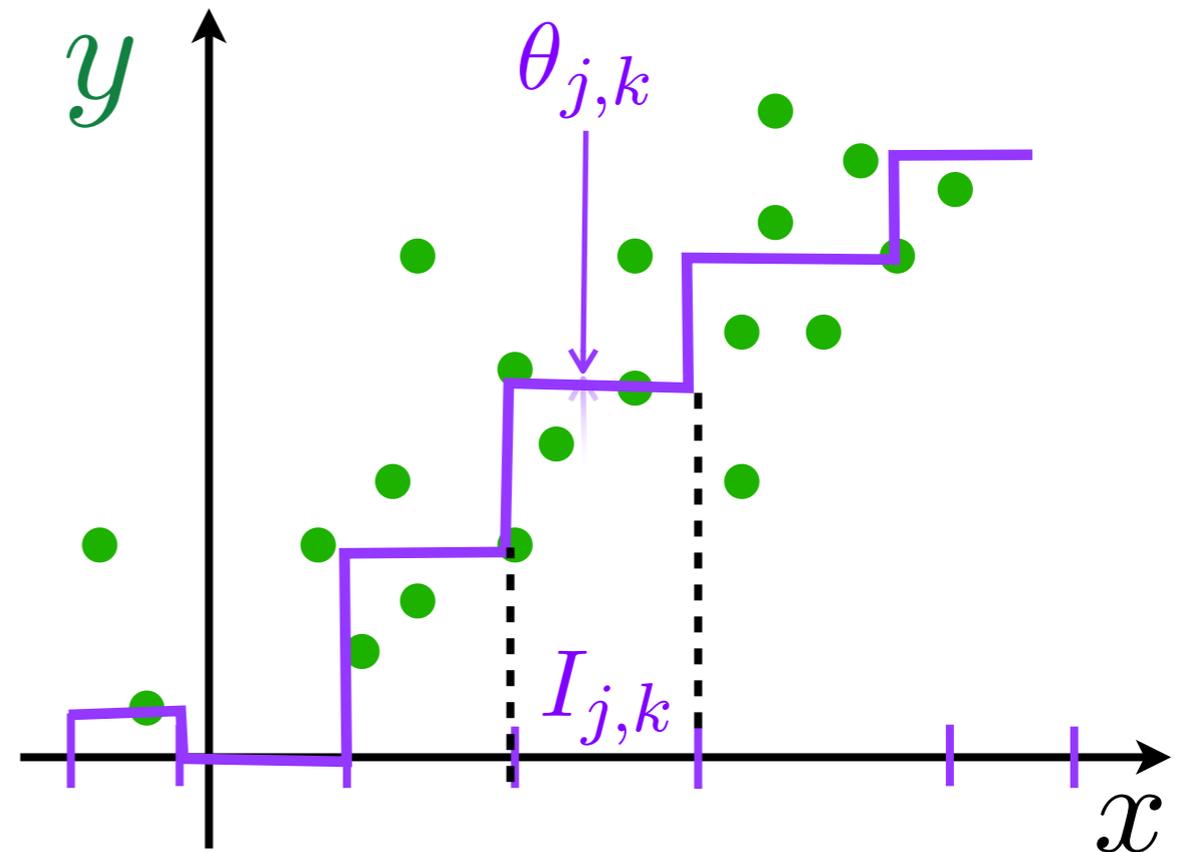


$$y_i = \omega^\top x_i + b = \sum_{j=1}^p \omega_j x_{i,j} + b$$

Impact of the j -th feature is linear and encoded by a single weight ω_j

$$x \mapsto \omega_j x$$

Linear regression on binarized features



$$y_i = \sum_{j=1}^p \sum_{k=1}^{d_j} \theta_{j,k} x_{i,j,k} + b = \sum_{j=1}^p \sum_{k=1}^{d_j} \theta_{j,k} \mathbb{1}(x_{i,j} \in I_{j,k}) + b$$

Impact of the j -th feature is **piecewise constant** and encoded by a **block**

$$\theta_{j,\bullet} = [\theta_{j,1} \cdots \theta_{j,d_j}]^\top$$

$$x \mapsto \sum_{k=1}^{d_j} \theta_{j,k} \mathbb{1}(x \in I_{j,k})$$

Features Binarization: Issues

(P1) Colinear binary features:

One-hot-encodings satisfy

$$\sum_{k=1}^{d_j} x_{i,j,k}^B = 1, \text{ for all } j = 1, \dots, p$$

→ X^B Not full rank

(P2) Overparametrization:

Increasing the number of bins d_j

→ Overfitting

(P3) Feature selection:

Some of the raw features $X_{\bullet,j}$ might be not relevant for the prediction task!

→ Block-Sparsity

$$\theta_{j,1} = 0, \dots, \theta_{j,d_j} = 0$$

Features Binarization: Solutions

- To deal with **(P1)**, we impose a **linear constraint** in each block (Agresti, 2015)

$$\text{(S1)} \quad n_j^\top \theta_{j,\bullet} = \sum_{k=1}^{d_j} n_{j,k} \theta_{j,k} = 0 \text{ for all } j = 1, \dots, p$$

$$n_j = [n_{j,1}, \dots, n_{j,d_j}]^\top \in \mathbb{N}^{d_j} \quad \text{where } n_{j,k} = |\{i : x_{i,j} \in I_{j,k}\}|$$

- To tackle **(P2)**, we keep the number of different values taken by $\theta_{j,\bullet}$ to minimal level by using a within **block weighted total-variation penalization**

$$\text{(S2)} \quad \sum_{j=1}^p \|\theta_{j,\bullet}\|_{\text{TV}, \hat{\omega}_{j,\bullet}} = \sum_{k=2}^{d_j} \hat{\omega}_{j,k} |\theta_{j,k} - \theta_{j,k-1}|$$

(S1) + (S2) solve (P3)

Binarsity

$$\text{bina}(\theta) = \sum_{j=1}^p \left(\sum_{k=2}^{d_j} \hat{\omega}_{j,k} |\theta_{j,k} - \theta_{j,k-1}| + \delta_j(\theta_{j,\bullet}) \right)$$

where

$$\delta_j(u) = \begin{cases} 0 & \text{if } n_j^\top u = 0, \\ \infty & \text{otherwise} \end{cases}$$

and $\hat{\omega}_{j,k} = \mathcal{O} \left(\sqrt{\frac{\log d}{n}} \hat{\pi}_{j,k} \right)$ with $\hat{\pi}_{j,k} = \frac{|\{i = 1, \dots, n : x_{i,j} \in \cup_{k'=k}^{d_j} I_{j,k'}\}|}{n}$

$\hat{\pi}_{j,k}$ **==**

Proportion of 1's in the sub-matrix obtained by deleting the first k columns in the j -th binarized block matrix

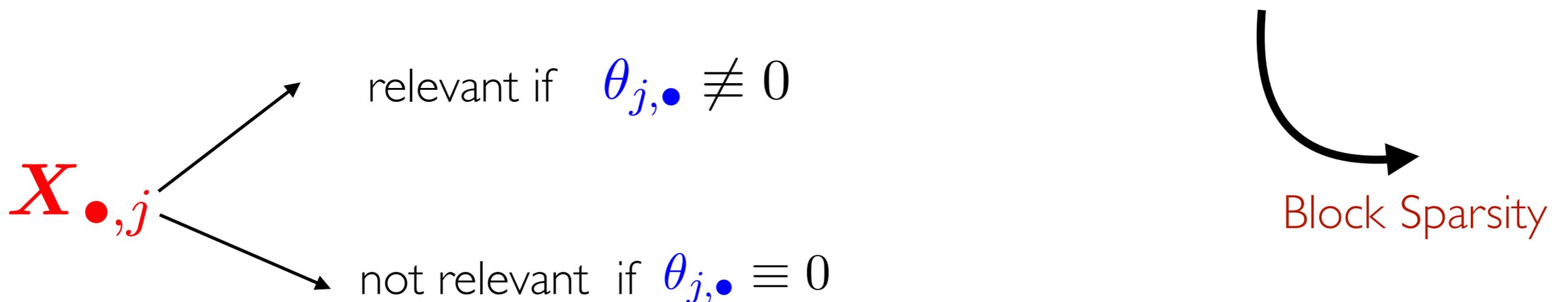
Binarsity: Interpretation

Large d_j \rightarrow $\theta_{j,k}$ and $\theta_{j,k-1}$ to be close

Penalization ℓ_1 \rightarrow Sparsity

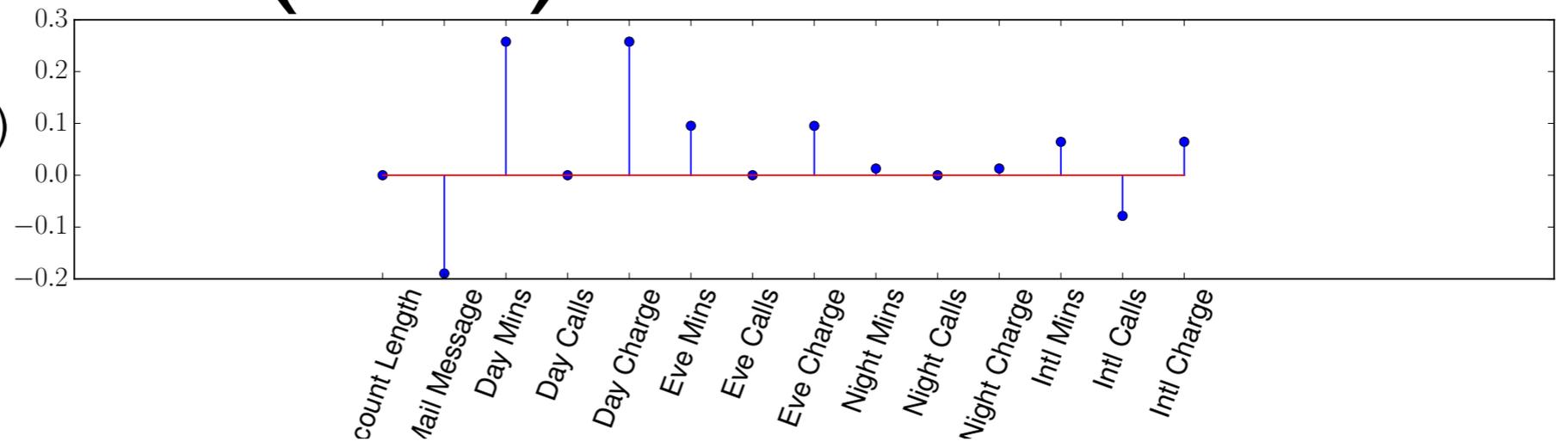
Penalization **Binarsity** \rightarrow Piecewise constant $\theta_{j,\bullet}$

- If $\theta_{j,\bullet}$ is constant then the linear constraint $n_j^\top \theta_{j,\bullet} = 0$ entails $\theta_{j,\bullet} \equiv 0$

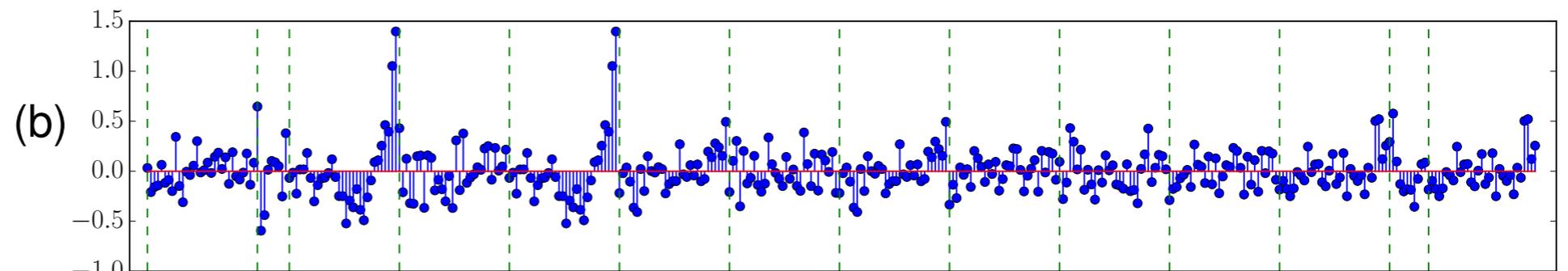


Weights of a Logistic Regression on Churn Dataset (UCI) $n = 3333, p = 14$

Raw continuous features (a)

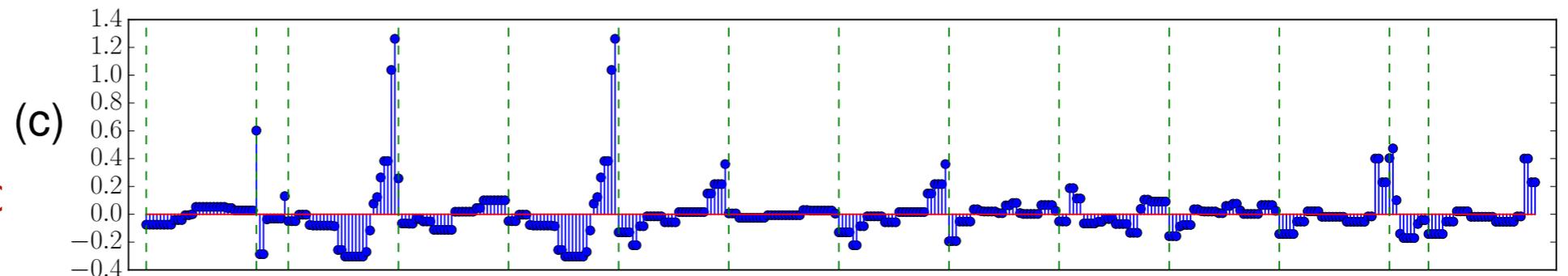


Binarized features
No-penalization



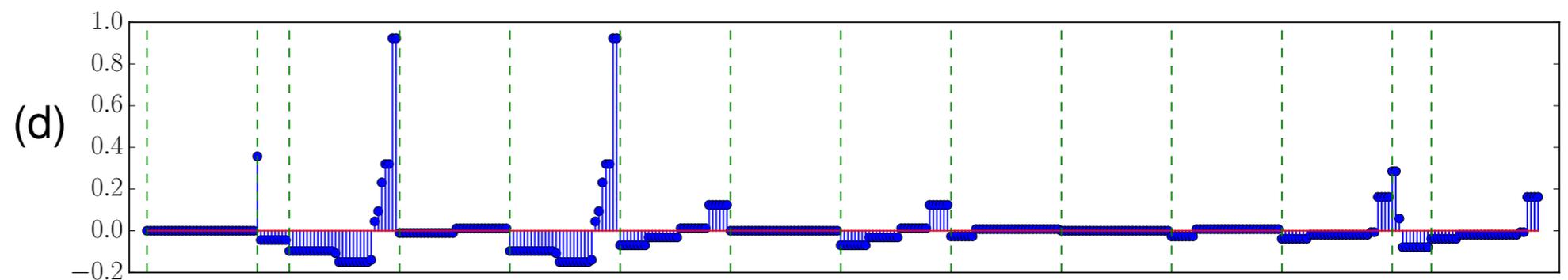
Binarized features

Low binarsity penalizatic



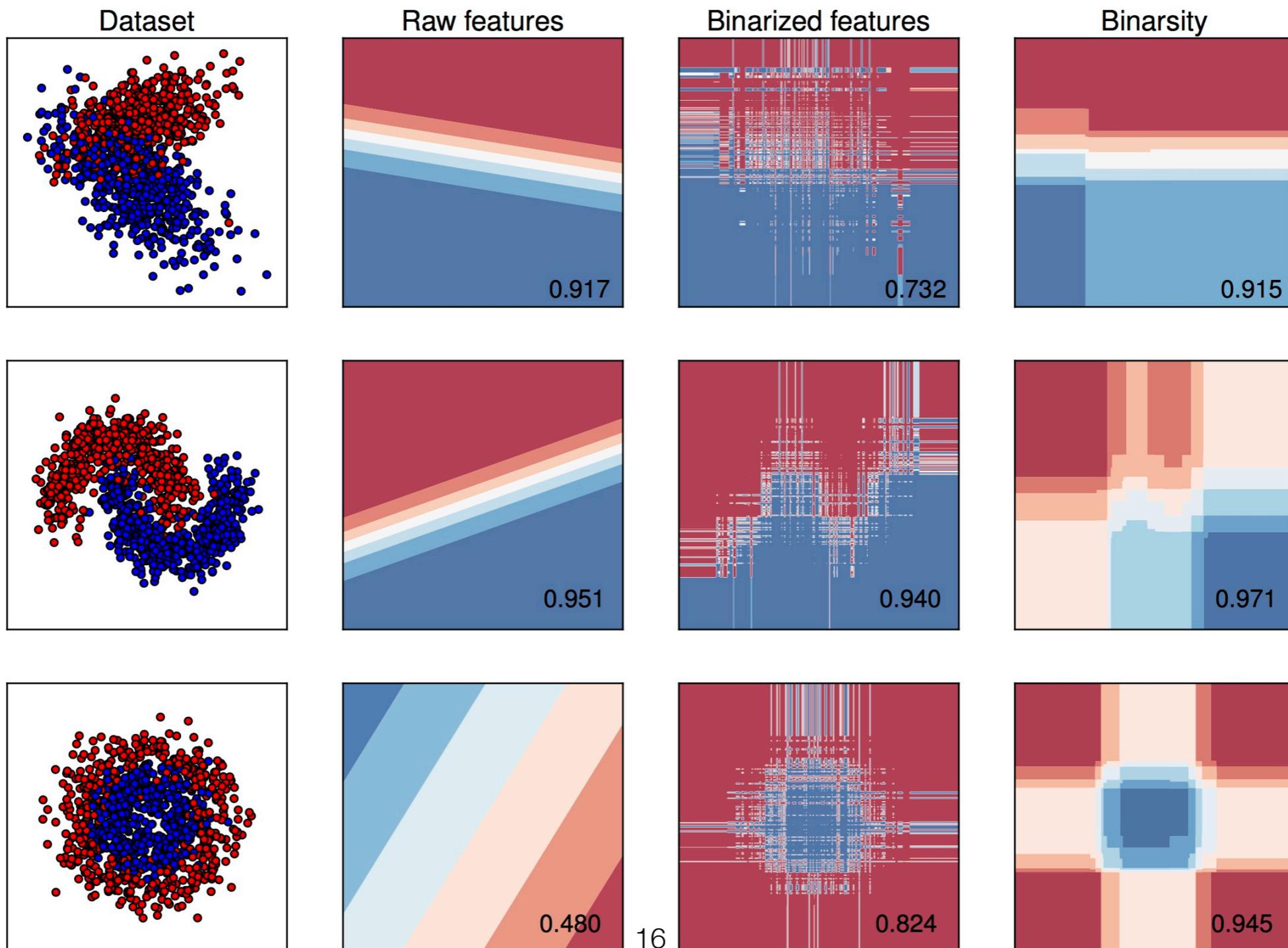
Binarized features

Strong binarsity
penalization



Decision Boundaries of a Logistic Regression

- Toy datasets with $n = 1000$, $p = 2$ and $d_1 = d_2 = 100$



2. Theoretical Guarantees (GLM + Binarity)

Generalized Linear Models

$$\mathbb{P}(y|x) = \exp \left(\frac{ym^0(x) - b(m^0(x))}{\phi} + c(y, \phi) \right)$$

- The functions $b(\cdot)$ and $c(\cdot)$, and the dispersion parameter ϕ are known.
- The natural parameter $m^0(\cdot)$ is *unknown* with

$$m^0(x) = g(\mathbb{E}[y|x]), \quad \text{where } b' = g^{-1}$$

Examples:

- Logistic and probit regression for binary data or multinomial regression for categorical data, Poisson regression for count data, etc ...

GLM: Goodness-of-fit

- Empirical risk

$$R_n(m_\theta) = \frac{1}{n} \sum_{i=1}^n \ell(y_i, m_\theta(x_i)) \quad \text{where} \quad m_\theta(x) = \theta^\top x^B$$

GLM loss function $\ell(y, y') = -yy' + b(y')$

- We estimate m^0 by:

$$\hat{m} = m_{\hat{\theta}}$$

$$\hat{\theta} \in \operatorname{argmin}_{\theta \in \mathbb{R}^d} \{ R_n(m_\theta) + \text{bina}(\theta) \}$$

Fast Oracle Inequality for: GLM + Binarsity

Assumptions on GLM:

$$\mathbb{P}(y|x) = \exp\left(\frac{ym^0(x) - b(m^0(x))}{\phi} + c(y, \phi)\right)$$

A1. $b(\cdot)$ is three times continuously **A2.** $|b'''(z)| \leq C_b |b''(z)|$ for some $C_b > 0$

A3. $C_n = \max_{i=1, \dots, n} |m^0(x_i)| < \infty$ **A4.** $L_n \leq \max_{i=1, \dots, n} b''(m^0(x_i)) \leq U_n$

- Satisfied for the following standard GLM :

Model	ϕ	$b(z)$	$b'(z)$	$b''(z)$	$b'''(z)$	C_b	L_n	U_n
Normal	σ^2	$\frac{z^2}{2}$	z	1	0	0	1	1
Logistic	1	$\log(1 + e^z)$	$\frac{e^z}{1 + e^z}$	$\frac{e^z}{(1 + e^z)^2}$	$\frac{1 - e^z}{1 + e^z} b''(z)$	2	$\frac{e^{C_n}}{(1 + e^{C_n})^2}$	$1/4$
Poisson	1	e^z	e^z	e^z	e^z	1	e^{-C_n}	e^{C_n}

Fast Oracle Inequality for: GLM + Binarsity

- Non asymptotic oracle inequality in terms of **excess risk**:

$$R(\hat{m}) - R(m_0) = \mathbb{E}_{\mathbb{P}(y|x)} [R_n(\hat{m}) - R_n(m_0)]$$

- **A new measures of sparsity: binarsity**

For $\theta \in \mathbb{R}^d$, let $J(\theta) = [J_1(\theta), \dots, J_p(\theta)]$ be the concatenation of the support sets relative to the total-variation penalization, that is

$$J_j(\theta) = \{k : \theta_{j,k} \neq \theta_{j,k-1}, \text{ for } k = 2, \dots, d_j\}.$$

$$\mathit{binarsity}(\theta) = |J(\theta)| = \sum_{j=1}^p |J_j(\theta)| = \sum_{j=1}^p |\{k : \theta_{j,k} \neq \theta_{j,k-1}, \text{ for } k = 2, \dots, d_j\}|$$



binarsity(θ) counts the number of non-equal consecutive values of θ

Fast Oracle Inequality for: GLM + Binarsity

Restricted Eigenvalues Assumption

Let $K = [K_1, \dots, K_p]$ be a concatenation of index sets such that $\sum_{j=1}^p |K_j| \leq J^*$. Assume

$$\kappa(K) \in \inf_{u \in \mathcal{C}_{\text{TV}, \hat{\omega}}(K) \setminus \{\mathbf{0}_d\}} \left\{ \frac{\|\mathbf{X}^B u\|_2}{\sqrt{n} \|u_K\|_2} \right\} > 0$$

with $\mathcal{C}_{\text{TV}, \hat{\omega}}(K) = \left\{ u \in \mathbb{R}^d : \sum_{j=1}^p \|(u_{j, \bullet})_{K_j}\|_{\text{TV}, \hat{\omega}_{j, \bullet}} \leq 2 \sum_{j=1}^p \|(u_{j, \bullet})_{K_j}\|_{\text{TV}, \hat{\omega}_{j, \bullet}} \right\}$.

Theorem: Fast Rate $\approx (\text{binarsity}(\theta) \times \log(d)) / n$

$$R(m_{\hat{\theta}}) - R(m^0) \leq \inf_{\substack{\theta \in B_d(\rho) \\ \forall j \mathbf{1}^\top \theta_{j, \bullet} = 0 \\ |J(\theta)| \leq J^*}} \left\{ 3(R(m_\theta) - R(m^0)) + \frac{\xi |J(\theta)|}{\kappa^2(J(\theta))} \max_{j=1, \dots, p} \|(\hat{\omega}_{j, \bullet})_{J_j(\theta)}\|_\infty^2 \right\},$$

where $B_d(\rho) = \{\theta \in \mathbb{R}^d : \sum_{j=1}^p \|\theta_{j, \bullet}\|_\infty \leq \rho\}$, and $\xi = \text{Cst}(C_n, \rho, p, L_n, U_n)$.

3. Numerical Experiments

Binary Classification

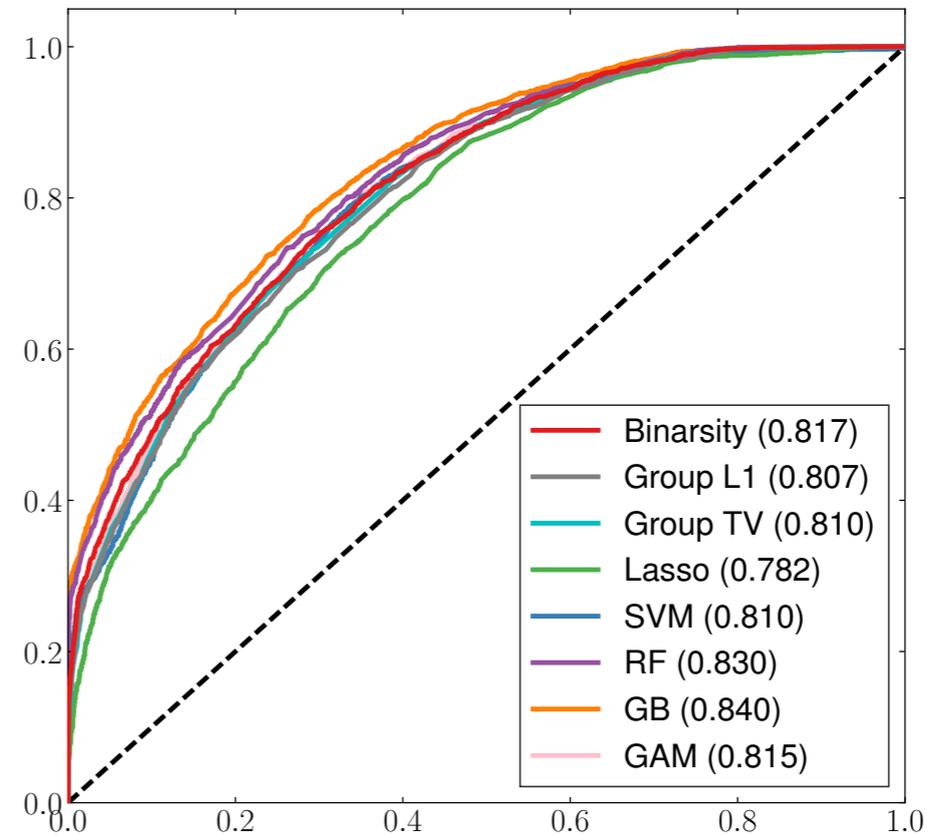
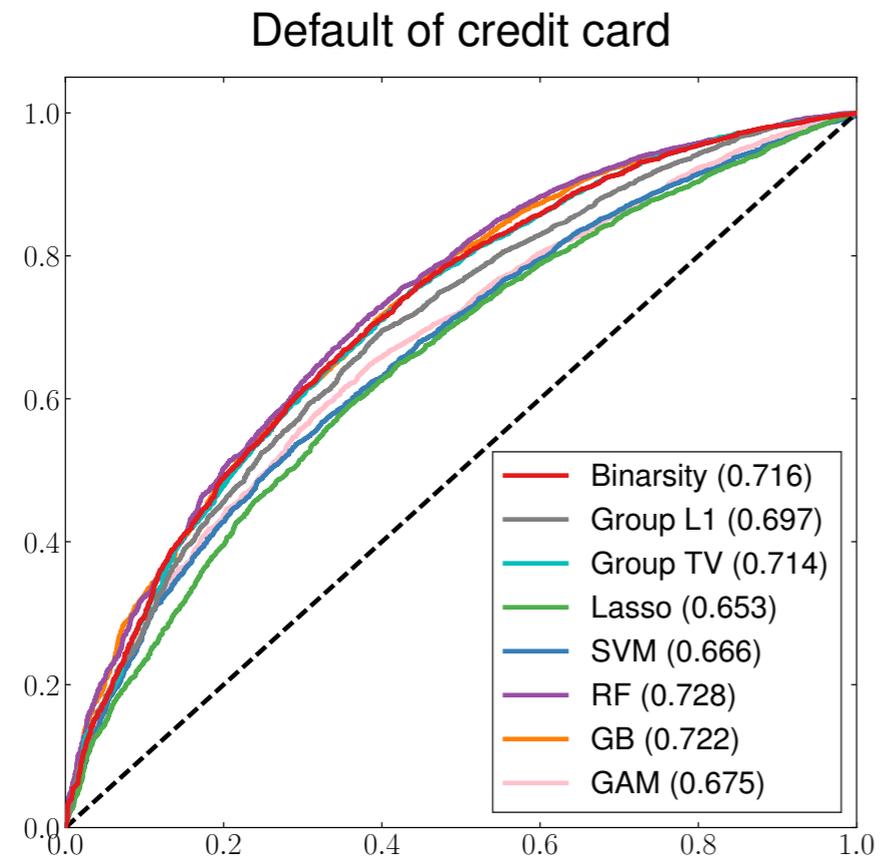
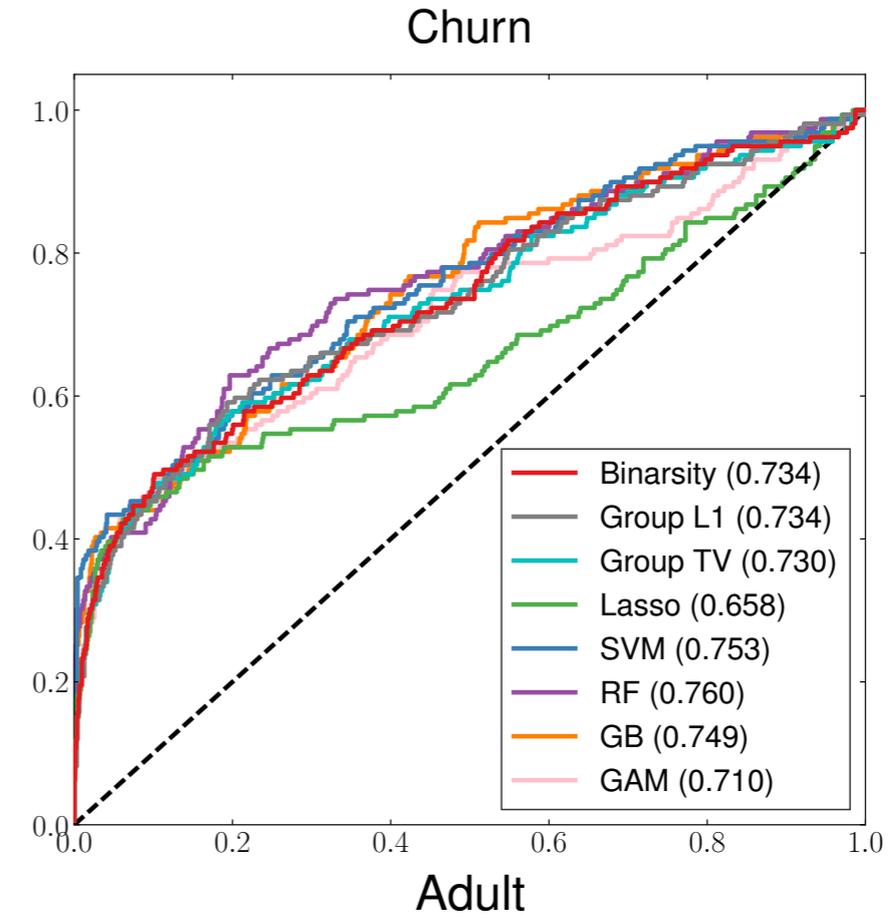
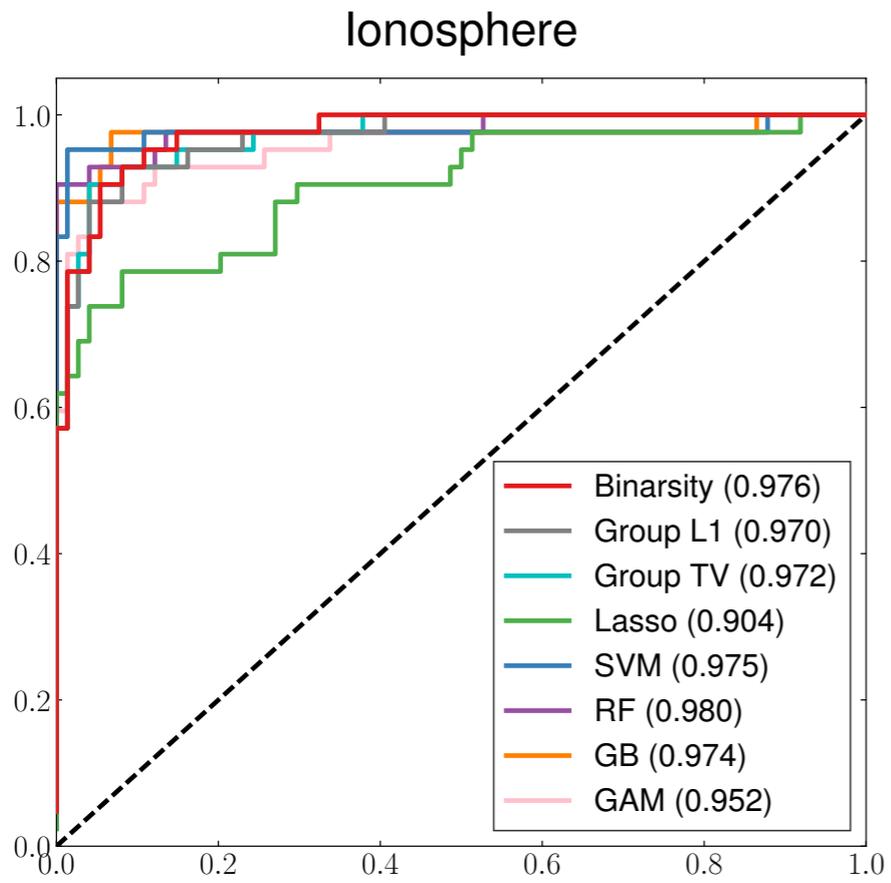
Source: UCI Machine Learning Repository

Dataset	#Samples	#Features
Ionosphere	351	34
Churn	3333	21
Default of Credit card	30000	24
Adult	32561	14
Bank Marketing	45211	17
Covertypes	550088	10
SUSY	5000000	18
HEPMASS	10500000	28
HIGGS	11000000	24

Baselines

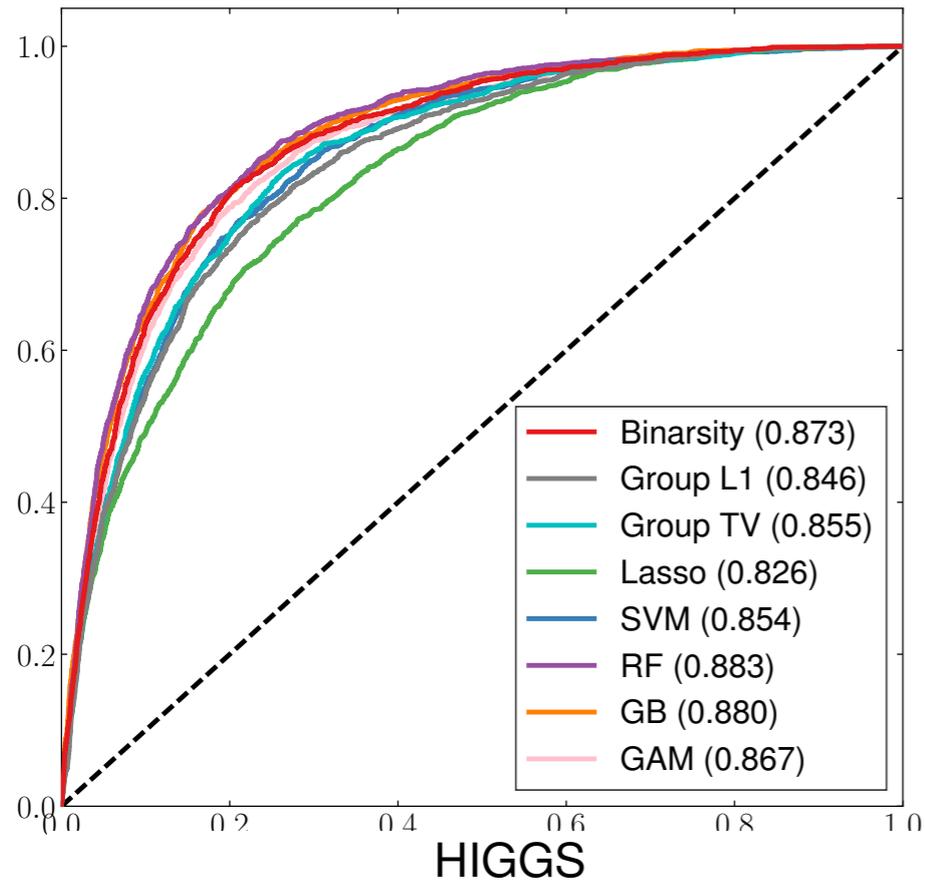
Method	Description
Lasso	Logistic regression with ℓ_1 penalization
Group Lasso	Logistic regression with group ℓ_1 penalization
Group TV	Logistic regression with Group Total-Variation penalization
SVM	Support Vector Machine with Gaussian kernel
GAM	Generalized Additive Model
RF	Random Forest
GB	Gradient Boosting

Results

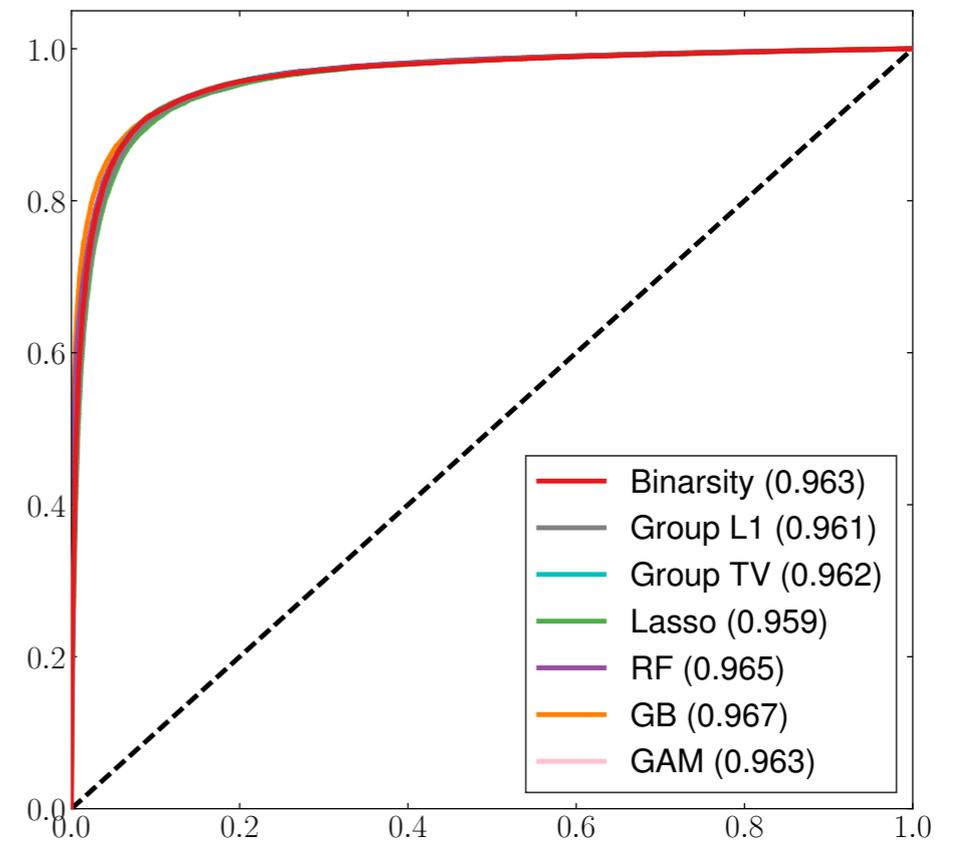
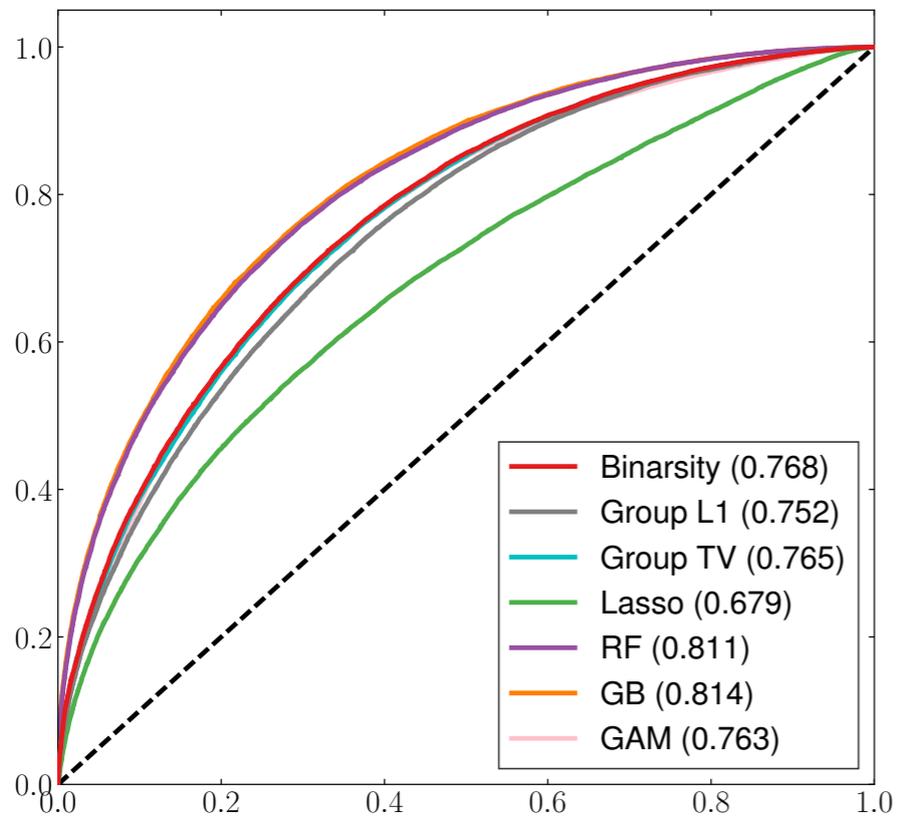
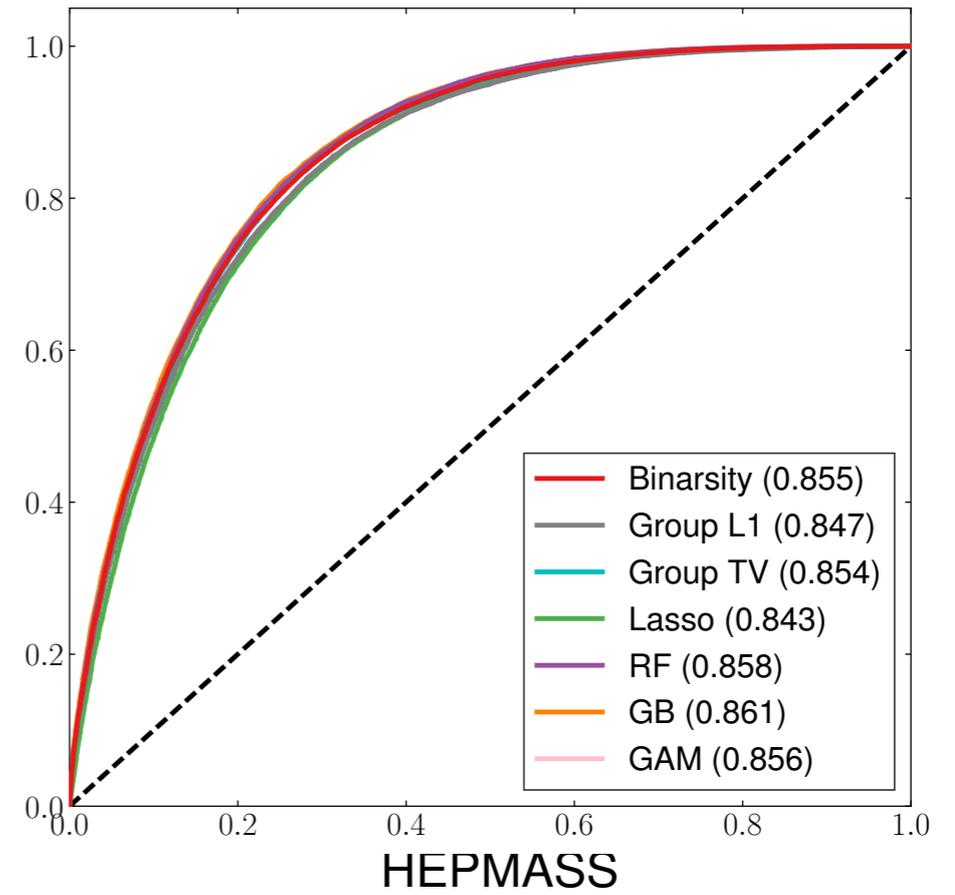


Results

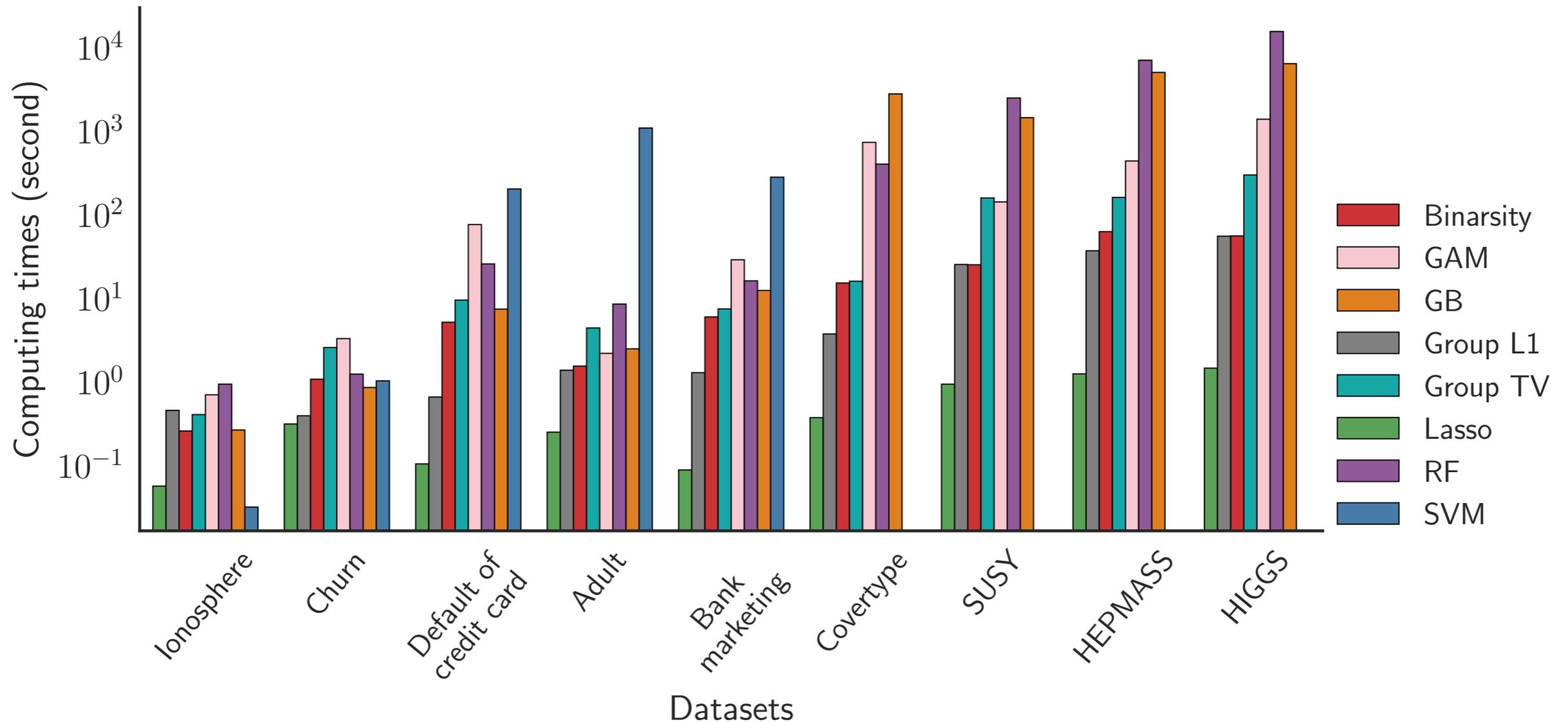
Bank marketing



SUSY



Computing Time Comparisons



- Log-scaled computing time comparisons between the methods on the considered datasets. Binarsity is between 2 and 5 times slower than Lasso but more than 100 times faster than RF and GB on large datasets like HIGGS.

Take Home Message

- We introduced the binarsity penalization for one-hot encodings of continuous features
- We illustrated the good statistical properties of binarsity for generalized linear models by proving non-asymptotic oracle inequalities.
- We conducted extensive comparisons of binarsity with state-of-the-art algorithms for binary classification on several standard datasets.

Thank you!