What to Expect of Classifiers?

Reasoning about Logistic Regression with Missing Features

Pasha Khosravi **Yitao Liang** **YooJung Choi** **Guy Van den Broeck**
{pashak, yliang, yjchoi, guyvdb}@cs.ucla.edu

Motivation

Classifiers are generally not able to make predictions in presence of uncertainty over input features $\mathbf{X}$

$\Rightarrow$ *e.g., with missing values!*

The probabilistic way to deal with this, is to ***compute the expected predictions** of* a classifier given a feature distribution. That is, we want to classify a partial sample $\mathbf{y}$ as:

$$E_{\mathcal{F},P}(\mathbf{y}) = \mathop{\mathbb{E}}_{\mathbf{m} \sim P(\mathbf{M}|\mathbf{y})} [\mathcal{F}(\mathbf{ym})]$$

where $\mathcal{F}$ is a classifier, P a distribution over input features $\mathbf{X} = \mathbf{YM}$, and $\mathbf{M}$ denotes those that are missing.

How hard is computing expectations?

Surprisingly computing expectations is ***hard for even simple classifiers and distributions***:

- $\mathcal{F}$ is a *nontrivial classifier* and P is *uniform* $\Rightarrow$ ***#P-Hard** [1]*
- $\mathcal{F}$ is a *single-feature classifier* and P is an *arbitrary PGM* $\Rightarrow$ ***#P-Hard** [1]*
- $\mathcal{F}$ is a *logistic regressor* and P *Naive Bayes* $\Rightarrow$ ***we prove it to be NP-Hard!***

Conformant Learning

We say $P(\mathbf{X}, C)$ ***conforms*** with $\mathcal{F} : \mathcal{X} \to [0,1]$ if their classifications agree: $P(c \mid \mathbf{x}) = \mathcal{F}(\mathbf{x})$ for all $\mathbf{x}$.

Conformant learning finds the generative model $P_{\boldsymbol{\theta}}(\mathbf{X}, C)$ which conforms to a classifier $\mathcal{F}(\mathbf{x})$ and maximises the feature likelihood:

$$\operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{d=(\mathbf{x}) \in D} \sum_{c} P_{\boldsymbol{\theta}}(\mathbf{x}, c)$$
$$\text{s.t. } \forall \mathbf{x} : P_{\boldsymbol{\theta}}(c \mid \mathbf{x}) = \mathcal{F}(\mathbf{x})$$

Naive Conformant Learning (**NaCL**) employs a Naive Bayes model for P and a Logistic Regressor for $\mathcal{F}$

$\Rightarrow$ *efficiently solvable as geometric programming*

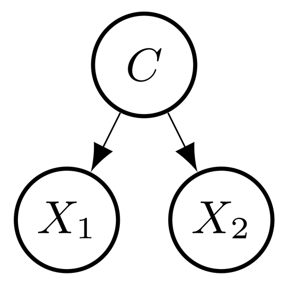
References

[1] Dan Roth. ***"On the hardness of approximate reasoning"***. In: Artificial Intelligence 82.1–2 (1996), pp. 273–302

$$\mathbf{w} = \begin{bmatrix} -1.16 \\ 2.23 \\ -0.20 \end{bmatrix}$$

X_1	X_2	$\mathcal{F}(x_1, x_2)$
1	1	0.70
1	0	0.74
0	1	0.20
0	0	0.24

A logistic regressor with weights $\mathbf{w}$ *and its predictions*



$P_1(c)$
0.5

C	$P_1(x_1 \mid C)$
1	0.8
0	0.3

C	$P_1(x_2 \mid C)$
1	0.45
0	0.5

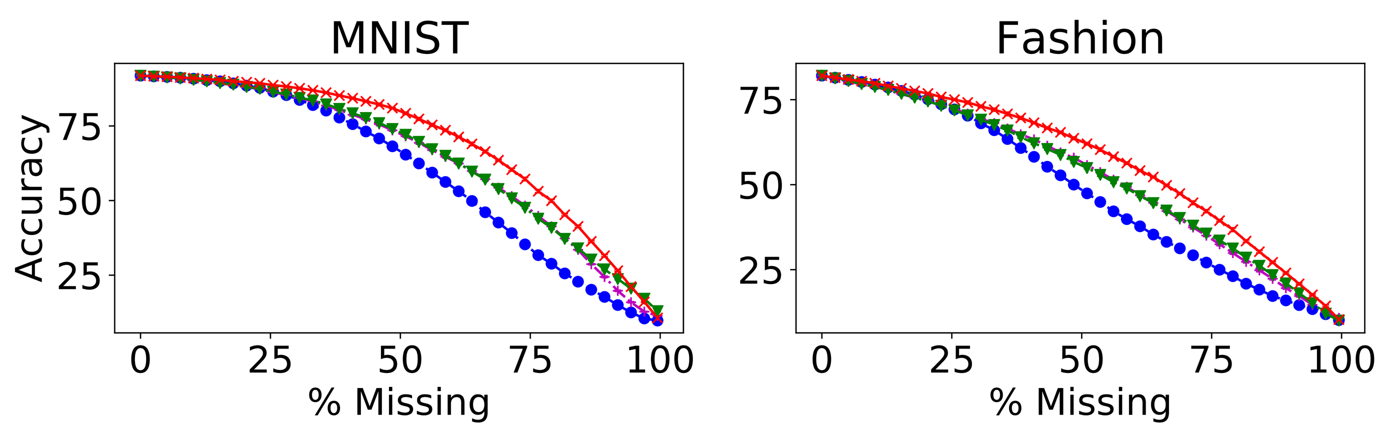
$P_2(c)$
0.36

C	$P_2(x_1 \mid C)$
1	0.6
0	0.14

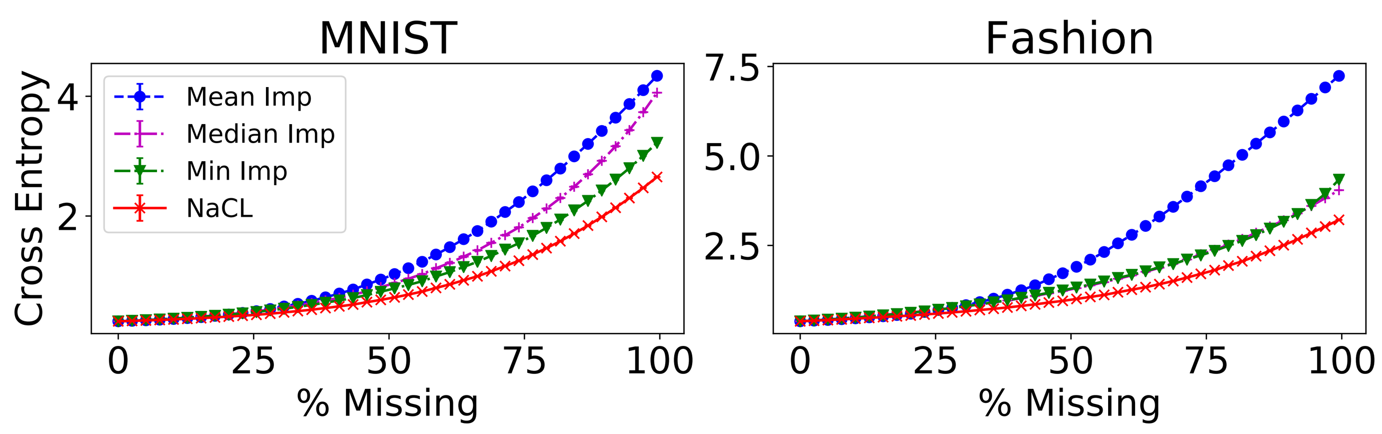
C	$P_2(x_2 \mid C)$
1	0.9
0	0.92

Two Naive Bayes models conforming to the above logistic regressor.

Predictions with missing values



$\Rightarrow$ *Competitive w.r.t. test set predictions (accuracy)*



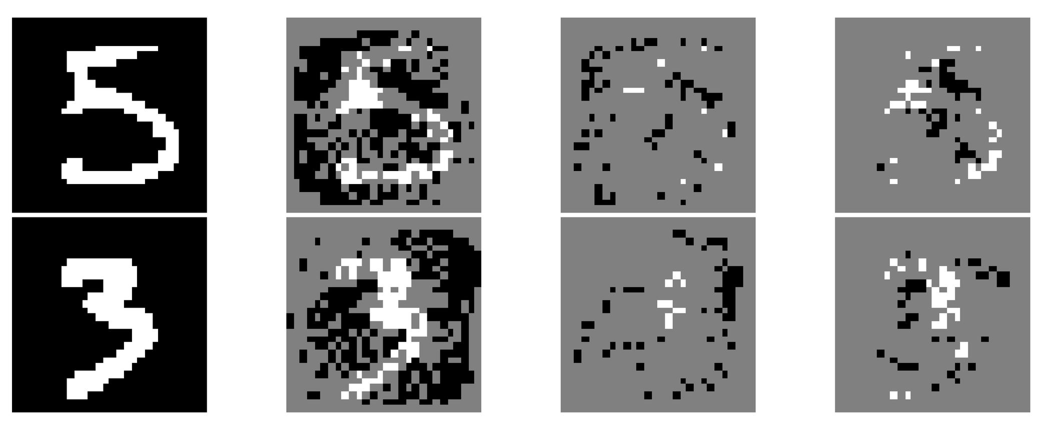
$\Rightarrow$ *Preserving logistic regression predictions (cross-entropy)*

Generating local explanations

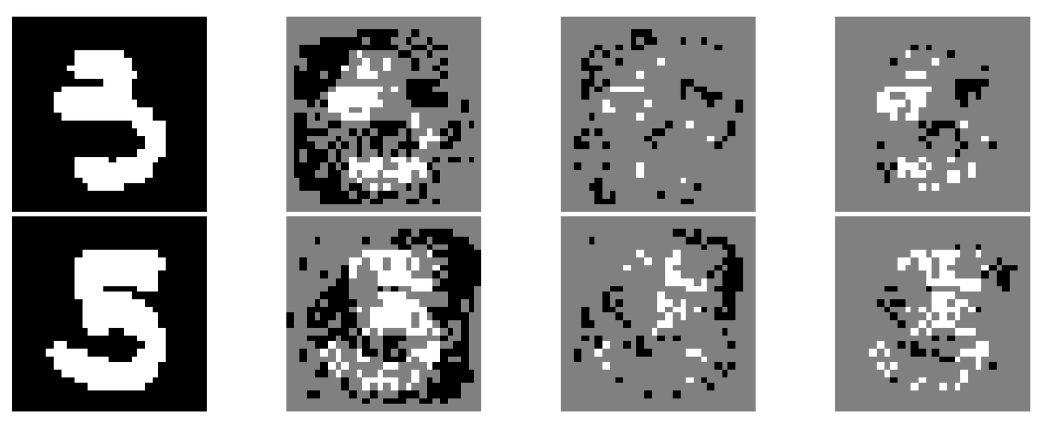
We look for the ***sufficient explanation*** of $\mathcal{F}(\mathbf{x})$ w.r.t. P as:

$$\operatorname*{argmin}_{\mathbf{e} \subseteq \mathbf{x}_+} |\, \mathbf{e} \,|$$
$$\text{s.t.} \quad \text{sign}(E_{\mathcal{F},P}(\mathbf{e}\mathbf{x}_-) - 0.5) = \text{sign}(\mathcal{F}(\mathbf{x}) - 0.5)$$

with $\mathbf{x}_+$ as the ***supporting features***, and $\mathbf{x}_-$ the ***opposing*** ones.



Correctly classified samples



Misclassified samples