What to Expect of Classifiers?
Reasoning about Logistic Regression with Missing Features

Pasha Khosravi, Yitao Liang, YooJung Choi and Guy Van den Broeck.
Computer Science Department, UCLA
Motivation

Train Classifier (ex. Logistic Regression)
Common Approaches

• Common approach is to fill out the missing features, i.e. doing imputation.

• They make unrealistic assumptions (mean, median, etc).

• More sophisticated methods such as MICE don’t scale to bigger problems (also have assumptions).

• We want a more principled way of dealing with this while staying efficient.
## Generative vs Discriminative Models

<table>
<thead>
<tr>
<th>Discriminative Models (ex. Logistic Regression)</th>
<th>Generative Models (ex. Naïve Bayes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(C \mid X)$</td>
<td>$P(C, X)$</td>
</tr>
</tbody>
</table>

### Missing Features
- **Discriminative Models**: 😞
- **Generative Models**: 😊

### Classification Accuracy
- **Discriminative Models**: 😊
- **Generative Models**: 😞
Expected Predication

- How can we leverage both discriminative and generative models?

- “Expected Prediction” is a principled way to reason about outcome of a classifier, $F(X)$, w.r.t. a feature distribution $P(X)$.

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

- $M$: Missing features
- $y$: Observed Features
Expected Predication Intuition

- **Imputation Techniques**: Replace the missing-ness uncertainty with *one* or *multiple* possible inputs, and evaluate the models.

- **Expected Prediction**: Considers *all possible inputs* and reason about expected behavior of the classifier.

\[
E_{\mathcal{F},P}(y) = \sum_{m} P(m \mid y) \cdot \mathcal{F}(ym) = \mathbb{E}_{m \sim P(M \mid y)} [\mathcal{F}(ym)]
\]
Hardness of Taking Expectations

• How can we compute the expected prediction?

• In general, it is intractable for arbitrary pairs of discriminative and generative models.

• Even when F is Logistic Regression and P is Naïve Bayes, the task is NP-Hard.
Conformant learning

Given a discriminative classifier and a dataset, learn a generative model that

1. *Conforms* to the classifier.

2. Maximizes the likelihood of joint feature distribution $P(X)$

No missing features $\rightarrow$ Same quality of classification ☺

Has missing features $\rightarrow$ No problem, do inference ☻
Naïve Conformant Learning (NaCL)

We focus on of Conformant Learning involving Logistic Regression and Naïve Bayes

- Given a NB model there is unique LR model that conform to it
- Given a LR model there is many NB models that conform to it
Naïve Conformant Learning (NaCL)

- We showed that we can write the Naïve Conformant Learning Optimization task as a Geometric Program.

- **Geometric Programs** are a special type of constraint optimization problems that have an exact and efficient algorithm to optimize, and modern GP solvers can handle large problems.

- For NaCL, we have $O(nk)$ number of constraints. $n$ is the number of features, and $k$ is the number of classes.
Naïve Conformant Learning (NaCL)

Logistic Regression Weights → NaCL → “Best” Conforming Naïve Bayes

GitHub: [github.com/UCLA-StarAI/NaCL](https://github.com/UCLA-StarAI/NaCL)
Experiments: Fidelity to Original Classifier

Using Cross Entropy to compare
- probabilities of the original classifier vs probabilities of NaCL's learned model
Experiments: Classification Accuracy

![Graphs showing classification accuracy for MNIST and Fashion datasets with different missing features percentages. The graph indicates that as the percentage of missing features increases, the classification accuracy decreases. Different methods such as Mean Imp, Median Imp, Min Imp, and NaCL are compared across different percentages of missing features.](image-url)
Other Applications

We saw *Expected Prediction* is very effective with handling missing features.

What else can we do?

• Explanations
• Feature Selection
• Fairness
Local Explanations using Missing-ness

**Goal**: To explain an instance of classification

- **Support Features**: Making them missing $\rightarrow$ probability goes **down**
- **Opposing Features**: Making them missing $\rightarrow$ probability goes **up**

**Sufficient Explanations**
Remove maximum number of supporting features until expected classification is about to change, then show the remaining support features.
Conclusion

• Expected Prediction is an effective tool for several applications such as missing data, generating explanations

• We introduced NaCL, an efficient algorithm, to convert a Logistic Regression model to a conforming Naïve Bayes model.

• Future work would be looking at more expressive pair of models, and potentially choose models that make the expected prediction tractable.
Thank You

What to Expect of Classifiers? Reasoning about Logistic Regression with Missing Features

GitHub: github.com/UCLA-StarAI/NaCL