Bayesian Evaluation of Black-Box Classifiers

Disi Ji¹  Robert Logan¹  Padhraic Smyth¹  Mark Steyvers²
Departments of ¹Computer Science and ²Cognitive Science, University of California, Irvine

Introduction

Goal: Evaluate blackbox classifiers online in new environments after they have been trained.

- Neural network models are being widely deployed as blackbox classifiers.
- It has been recognized that deep neural networks can be miscalibrated.¹
- We propose a Bayesian framework for assessing performance characteristics of black-box classifiers, which enables third parties to infer on quantities such as accuracy and calibration bias, as well as measure uncertainty in their estimates.
- We use our framework to design efficient labeling methods which quickly identify weaknesses of blackbox classifiers.

Approach

- For a blackbox classifier $M$ and input $x$, denote normalized output as $p_k(x)$, $k = 1, 2, ..., K$
- Predicted label on $x$ made by $M$ is $\hat{k} = \arg \max_k p_k(x)$

  - Local score/confidence: for a given $x$, score the model assigned to the predicted label $S_M(x) = p_{\hat{k}}(x)$ Model's own assessment of accuracy at $x$

  - Local accuracy: for a given $x$, probability that predicted label is the same as true label $y$ True accuracy, need to be evaluated with true label $y$

  $A(x) = P(y = \hat{k} | x)$

  Empirical estimation of accuracy needs labeled data. When getting true label is expensive, estimating accuracy can be costly.

  - Accuracy over a region: expectation of local accuracy over region $R$:

    $A(R) = \int_{R} p(y = \hat{k} | x) dx$

  - Accuracy over region $R$: $A_R(x) \sim Beta(a, b)$
    - For $i = 1, 2, ..., N$:
      - $x_i$ = predicted by model and model makes prediction on it $\hat{i}$
      - Query true label: $(y_i = \hat{i}) \sim Bern(p_{\hat{i}}(x_i))$
    - Posterior of accuracy gets updated in closed form as more labels get revealed.

  - By partitioning the data space $D$ into disjoint subsets:
    - Finer grained estimation of model characteristics can be conducted on each subset to have more comprehensive assessment of model performance in environment $p(x, y)$:

    $A(R) = \int_{R} p(y = \hat{k} | x) dx$

    Denote each $A_R(x) = \beta$, model accuracy over each region independently with $\beta = Bern(a, b)$

    - Examples of different partitions:
      - For modeling reliability diagram: $R_i = \{ y_i = k \} \in \{ \frac{k}{10}, \frac{k+1}{10}, \ldots, \frac{10}{10} \}$
      - For modeling classwise accuracy: $R_k = \{ y_k = k \}$

    - Local calibration error: for a given $x$, $\epsilon$, the difference between local accuracy and confidence:

      $CE_M(x) = \Delta(S_M(x) - A(x))$

    - A model at $x$ is:
      - calibrated if $S_M(x) = A(x)$
      - overconfident if $S_M(x) > A(x)$
      - Calibration bias: $\Delta(x, y) = a - b$

Classwise Accuracy and Calibration Bias

Bayesian Reliability Diagram

Active Learning to Find Extreme Classes

Idea

Use Thompson sampling-based approach to efficiently determine most accurate/biased classes.

Algorithm

- Sample accuracies/biases from posterior.
- Determine least accurate/most biased class according to sample.
- Obtain label for a data point with least accurate/most biased class.
- Update posteriors.

Success rates of Thompson sampling vs. random selection strategy as a function of the number of queries submitted to the oracle. Averaged over 100 runs.

Conclusion

- Bayesian methods show promise for blackbox model assessment, allowing for uncertainty quantification in estimates of calibration and accuracy
- We also show how our framework can be used to quickly identify potential issues in a deployed model (e.g., least calibrated class predictions)

References


For any questions, email: disij@uci.edu