Assorted Topics in Probabilistic ML (1)

CS772A: Probabilistic Machine Learning

Piyush Rai

Plan for today

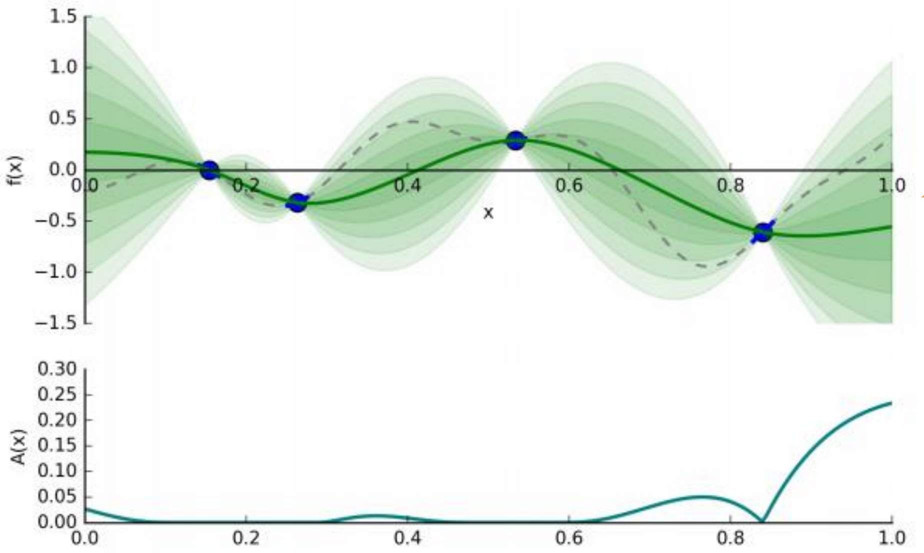
- Wrapping up BO (acquisition functions)
- Assorted Topics (1)
  - Frequentist vs Bayesian
  - Model Calibration to reduce overconfidence
  - Conformal Prediction (simple and fast way to get prediction uncertainty/set)

Bayesian Optimization

Note: Function values can be noisy too, e.g., $f(x_n) + \epsilon_n$

- BO requires two ingredients
  - A regression model to learn a surrogate of $f(x)$ given previous queries $\{(x_n, f(x_n))\}_{n=1}^{N}$
  - An acquisition function $A(x)$ to tell us where to query next

Assumption: $A(x)$ should be easier to optimize than $f(x)$



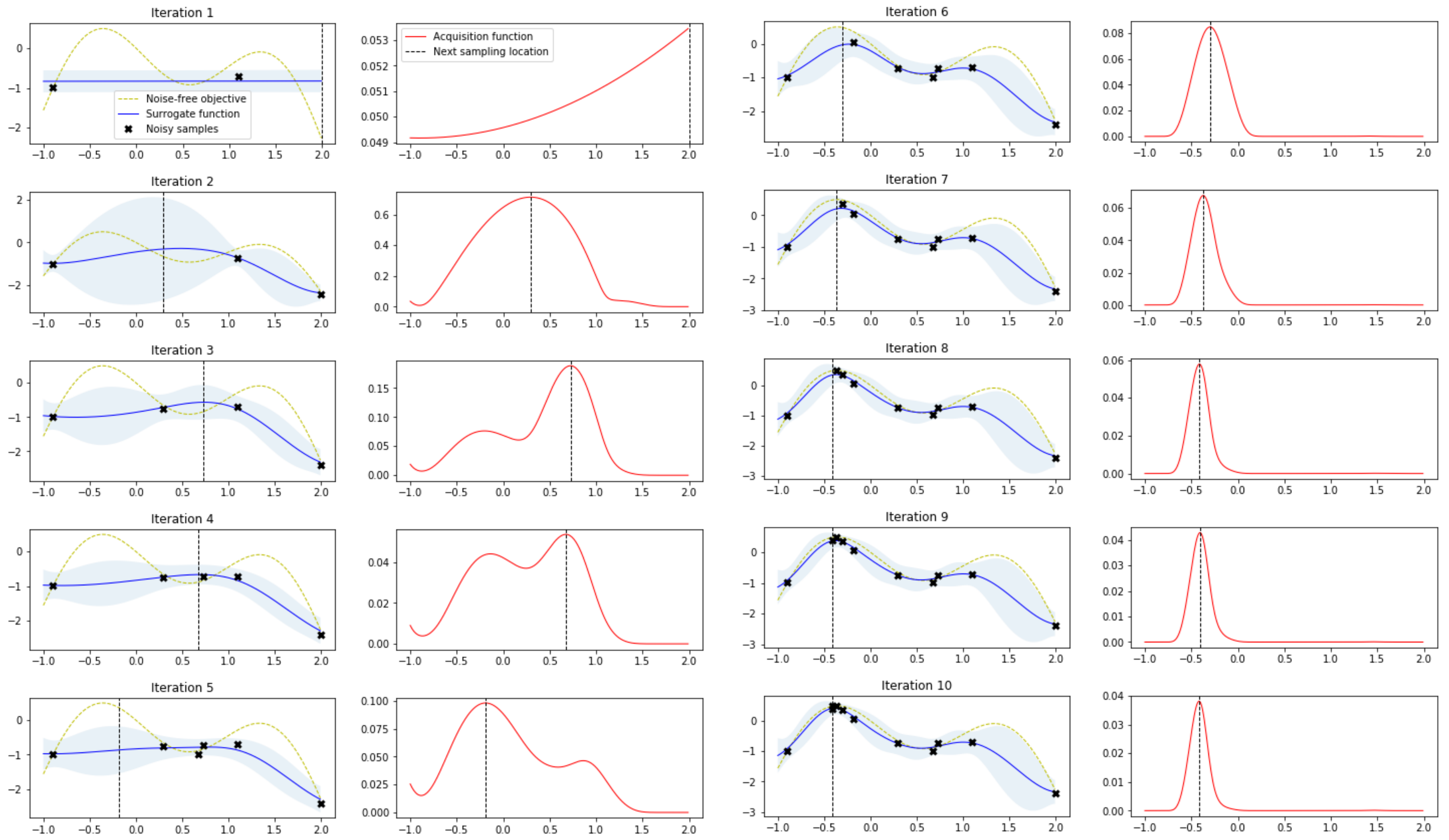
Dotted curve: True function
Green curve: Current surrogate of the function
Shaded region: Uncertainty in the function's estimate

A typical example of what $A(x)$ might look like, assuming that the goal is to find the maxima of $f(x)$

- Note: The regression model must also have estimate of function's uncertainty
  - Bayesian nonlinear regression, such as GP, Bayesian Neural network, etc would be ideal

Bayesian Optimization: An Illustration

- Suppose our goal is to find the <u>maxima</u> of $f(x)$ using BO



Pic source: http://krasserm.github.io/2018/03/21/bayesian-optimization/

Some Basic Acquisition Functions for BO

(assuming we are finding the minima)

Acquisition Functions: Probability of Improvement

- Assume past queries $\mathcal{D}_N = (\boldsymbol{X}, \boldsymbol{f}) = \left(x_n, f(x_n)\right)_{n=1}^{N}$ and suppose $\boldsymbol{f_{min}} = \min \boldsymbol{f}$
- Suppose f_{new} denotes the function's value at the next query point $\boldsymbol{x_{new}}$
- We have an improvement if $f_{new} < f_{min}$ (recall we are doing minimization)
- Assuming the function is real-valued, suppose the posterior predictive for $\boldsymbol{x_{new}}$ is

$$p(f_{new}|x_{new}, \mathcal{D}_N) = \mathcal{N}(f_{new}|\mu(x_{new}), \sigma^2(x_{new}))$$

- We can define a probability of improvement based acquisition function

Exercise: Verify

$$A_{PI}(x_{new}) = p(f_{new} \leq f_{min}) = \int_{-\infty}^{f_{min}} \mathcal{N}(f_{new}|\mu(x_{new}), \sigma^2(x_{new}))df_{new} = \Phi\left(\frac{f_{min} - \mu(x_{new})}{\sigma(x_{new})}\right)$$

$\Phi()$ denotes CDF of $\mathcal{N}(0,1)$

- The optimal query point will be one that maximizes $A_{PI}(x_{new})$

$$x_* = \operatorname{argmax}_{x_{new}} A_{PI}(x_{new})$$

Acquisition Functions: Expected Improvement

- PI doesn't take into account the amount of improvement

- Expected Improvement (EI) takes this into account and is defined as

$$A_{EI}(x_{new}) = \mathbb{E}[f_{min} - f_{new}] = \int_{-\infty}^{f_{min}} (f_{min} - f_{new})\mathcal{N}(f_{new}|\mu(x_{new}), \sigma^2(x_{new}))df_{new}$$

$$= (f_{min} - \mu(x_{new}))\Phi\left(\frac{f_{min} - \mu(x_{new})}{\sigma(x_{new})}\right) + \sigma(x_{new})\mathcal{N}\left(\frac{f_{min} - \mu(x_{new})}{\sigma(x_{new})}; 0,1\right)$$

Exercise: Prove this result

- The optimal query point will be one that maximizes $A_{EI}(x_{new})$

$$x_* = \operatorname{argmax}_{x_{new}} A_{EI}(x_{new})$$

Focus on points where the function has small values (since we are looking for its minima)

Focus on points where the function has high uncertainty (so that including them improves our estimate of the function)

- Note that the above acquisition function trades off exploitation vs exploration
  - Will prefer points with small predictive mean $\mu(x_{new})$: Exploitation
  - Will prefer points with large predictive variance $\sigma(x_{new})$: Exploration

Acquisition Functions: Lower Confidence Bound

- Lower Confidence Bound (LCB) also takes into account exploitation vs exploration
- Used when the regression model is a Gaussian Process (GP)

When using BO for maximization, we use Upper Confidence Bound (UCB) defined as $A_{UCB}(x_{new}) = \mu(x_{new}) + \kappa\, \sigma(x_{new})$ and $x_* = \text{argmax}_{x_{new}} A_{UCB}(x_{new})$

- Assume the posterior predictive for a new point to be

$$p(f_{new}|x_{new}, \mathcal{D}_N) = \mathcal{N}(f_{new}|\mu(x_{new}), \sigma^2(x_{new}))$$

- The LCB based acquisition function is defined as

$$A_{LCB}(x_{new}) = \mu(x_{new}) - \kappa\, \sigma(x_{new})$$

Thus prefer points at which the function has low mean but high variance

- Point with the smallest LCB is selected as the next query point

$$x_* = \text{argmin}_{x_{new}} A_{LCB}(x_{new})$$

- κ is a parameter to trade-off exploitation (low mean) and exploration (high variance)
- Under certain conditions, the iterative application of this acquisition function will converge to the true global optima of f (Srinivas et al. 2010)

Bayesian Optimization: The Overall Algo

- Initialize $\mathcal{D} = \{\}$
- For $n = 1,2, \dots, N$ (or until the budget doesn't exhaust)
  - Select the next query point x_n by optimizing the acquisition function

$$x_n = \operatorname{argopt}_x A(x)$$

  - Get function's value from the black-box oracle: $f_n = f(x_n)$
  - $\mathcal{D} = \{\mathcal{D} \cup (x_n, f_n)\}$

Can get the function's minima from this set of function's values

  - Update the regression model for f using data $\mathcal{D}$

BO: Some Challenges/Open Problems

- Learning the regression model for the function
  - GPs are flexible but can be expensive as N grows
  - Bayesian neural networks can be an more efficient alternative to GPs (Snoek et al, 2015)
  - Hyperparams of the regression model itself (e.g., GP cov. function, Bayesian NN hyperparam)
- High-dimensional Bayesian Optimization (optimizing functions of many variables)
  - Most existing methods work well only for a moderate-dimensional x
  - Number of function evaluations required would be quite large in high dimensions
  - Lot of recent work on this (e.g., based on dimensionality reduction)
- Multitask Bayesian Optimization (joint BO for several related functions)
  - Basic idea: If two functions are similar their optima would also be nearby

BO: Some Further Resources

- Some survey papers:
  - A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning (Brochu et al., 2010)
  - Taking the Human Out of the Loop: A Review of Bayesian Optimization (Shahriari et al., 2015)

- Some open source software libraries
  - BoTorch: Bayesian Optimization in PyTorch
  - GPflowOpt: Bayesian Optimization in Tensorflow (and using GP for modeling the function)
  - Also available in scikit-optimize

Frequentist Statistics
(vs Bayesian Statistics)

Frequentist Statistics

- The Bayesian approach treats parameters/model unknowns as random variables
- In the Bayesian approach, the posterior over these r.v.'s help capture the uncertainty
- The Frequentist approach is a different way to capture uncertainty
  - Don't treat parameters as r.v. but as fixed unknowns
  - Treat parameters as a function of the dataset, e.g., $\hat{\theta}(\mathcal{D}) = \pi(\mathcal{D})$
  - Variations in param estimates over different datasets represents their uncertainty

This can be some point estimate, e.g., MLE, MAP, method of moments, etc.

A random dataset drawn from the true data distribution

True unknown value of the parameter

$$\tilde{\mathcal{D}}^{(s)} = \{\boldsymbol{x}_n \sim p(\boldsymbol{x}_n|\boldsymbol{\theta}^*) : n = 1 : N\} \qquad (s = 1, 2, \dots, S)$$

The estimated distribution of the parameters given any randomly drawn dataset from the true data distribution

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S}\delta(\boldsymbol{\theta} = \pi(\tilde{\mathcal{D}}^{(s)}))$$

Param estimate using the s-th sampled dataset

As $S \to \infty$, this is known as the "sampling distribution" of the estimator

Note that sampling distribution is different from a posterior distribution we infer in Bayesian learning (there, we condition on a fixed training set)

But if the estimator is MLE and Bayesian method's prior is uniform, then both distributions are very similar (sampling distribution is often called "poor man's posterior"

Approximating the sampling distribution

- Since the true $\boldsymbol{\theta}^*$ is not known, we can't compute the sampling distribution exactly

$$\tilde{\mathcal{D}}^{(s)} = \{\boldsymbol{x}_n \sim p(\boldsymbol{x}_n|\boldsymbol{\theta}^*) : n = 1:N\} \qquad (s = 1, 2, \ldots, S)$$

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S}\delta(\boldsymbol{\theta} = \pi(\tilde{\mathcal{D}}^{(s)}))$$

- Bootstrap is a popular method to approximate the sampling distribution
- Two types of bootstrap methods: parametric and nonparametric bootstrap

Parametric Bootstrap

- Get a point est. of $\boldsymbol{\theta}$ using training data

$$\hat{\boldsymbol{\theta}} = \pi(\mathcal{D})$$

- Generate multiple datasets using $\hat{\boldsymbol{\theta}}$ as

$$\tilde{\mathcal{D}}^{(s)} = \{\boldsymbol{x}_n \sim p(\boldsymbol{x}_n|\hat{\boldsymbol{\theta}}) : n = 1:N\} \quad (s = 1, 2, \ldots, S)$$

- Now compute the approximation as

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S}\delta(\boldsymbol{\theta} = \pi(\tilde{\mathcal{D}}^{(s)}))$$

Nonparametric Bootstrap

- Use sampling with replacement on original training set to generate $\boldsymbol{S}$ datasets with $\boldsymbol{N}$ datapoints in each

Each dataset will contain roughly 63% unique datapoints from original training set

- Now compute the approximation as

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S}\delta(\boldsymbol{\theta} = \pi(\tilde{\mathcal{D}}^{(s)}))$$

Model Calibration

Evaluation of Predictive Models

- Assume a predictive distribution $p_\theta(y|x)$
- Define score of p_θ on an example $(x,y) \sim p^*(x,y) = p^*(x)p^*(y|x)$ as $s(p_\theta,(x,y))$
- The expected score of p_θ will be

$$s(p_\theta, p^*) = \int p^*(x)p^*(y|x)s\big(p_\theta,(x,y)\big)dydx$$

- A scoring rule is said to be a "proper scoring rule" if $s(p_\theta, p^*) \leq s(p^*, p^*)$
- The log-likelihood $s\big(p_\theta,(x,y)\big) = \log p_\theta(y|x)$ is a proper scoring rule because

$$S(p_{\boldsymbol{\theta}}, p^*) = \mathbb{E}_{p^*(\boldsymbol{x})p^*(y|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(y|\boldsymbol{x})\right] \leq \mathbb{E}_{p^*(\boldsymbol{x})p^*(y|\boldsymbol{x})}\left[\log p^*(y|\boldsymbol{x})\right]$$

Holds because of Gibbs inequality

- Another proper scoring rule is the Brier score (lower is better)

Squared error of predictive distribution as compared to one-hot vector

$$S(p_{\boldsymbol{\theta}}, (y, \boldsymbol{x})) \triangleq \frac{1}{C}\sum_{c=1}^{C}(p_{\boldsymbol{\theta}}(y=c|\boldsymbol{x}) - \mathbb{I}\,(y=c))^2$$

Calibration

- A model called calibrated if predicted class probabilities match empirical frequencies
- Example: For binary classification, if for all test examples for which the model predicts $p(y = 1|x) = 0.8$, about 80% have true label = 1, then this model is well-calibrated
- Expected Calib. Error (ECE) often used a measure of model calib. (so is Brier Score)
- Suppose $f(x)_c = p(y = c|x)$, $\hat{y}_n = \text{argmax}_{c=\{1,2,\dots,C\}} f(x_n)_c$, $\hat{p}_n = \max_{c=\{1,2,\dots,C\}} f(x_n)_c$
- Suppose predicted probabilities are divided into B bins
- Assume $\mathcal{B}_b$ as set of samples whose predicted probabilities fall in $I_b = (\frac{b-1}{B}, \frac{b}{B}]$

$$\text{acc}(\mathcal{B}_b) = \frac{1}{|\mathcal{B}_b|} \sum_{n \in \mathcal{B}_b} \mathbb{I}(\hat{y}_n = y_n) \quad \text{conf}(\mathcal{B}_b) = \frac{1}{|\mathcal{B}_b|} \sum_{n \in \mathcal{B}_b} \hat{p}_n \quad \text{ECE}(f) = \sum_{b=1}^{B} \frac{|\mathcal{B}_b|}{B} |\text{acc}(\mathcal{B}_b) - \text{conf}(\mathcal{B}_b)|$$

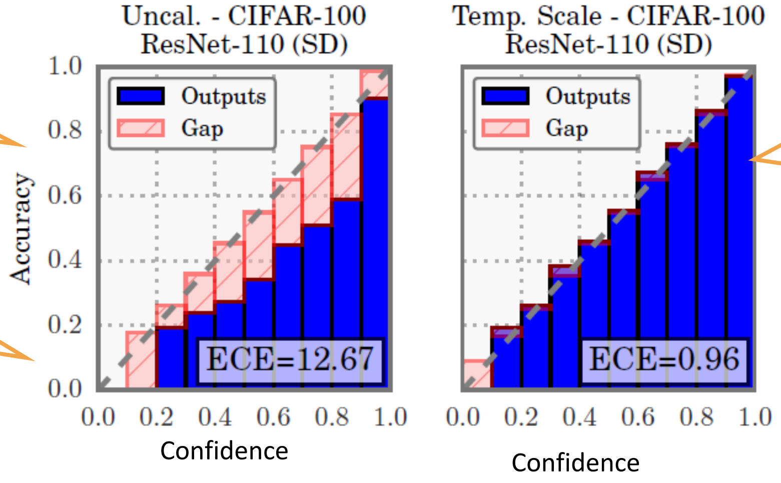
Difference between confidence and accuracy

Calibration

- A reliability diagram is often used as a visual indicator of calibration

ECE is the average "gap" area in the reliability diagram

Reliability diagram of an uncalibrated model



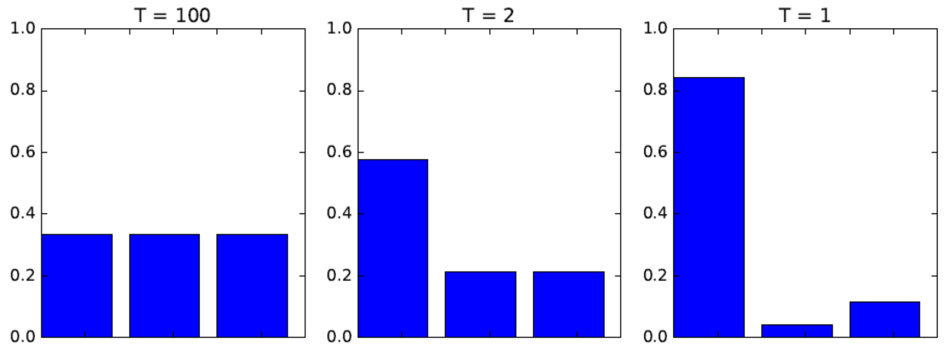
Reliability diagram of the same model after applying calibration post-processing via temperature scaling method

Many other approaches: Platt Scaling, Histogram Binning, Label Smoothing, etc are also popular, and can be applied as post-processing step to the outputs of Bayesian/non-Bayesian methoods to improve calibration

- Several approaches to improve a model's calibration
- In general, we want to reduce the model's overconfidence

Bayesian methods are usually better calibrated but can still have poor calibration if test data is from a different distribution

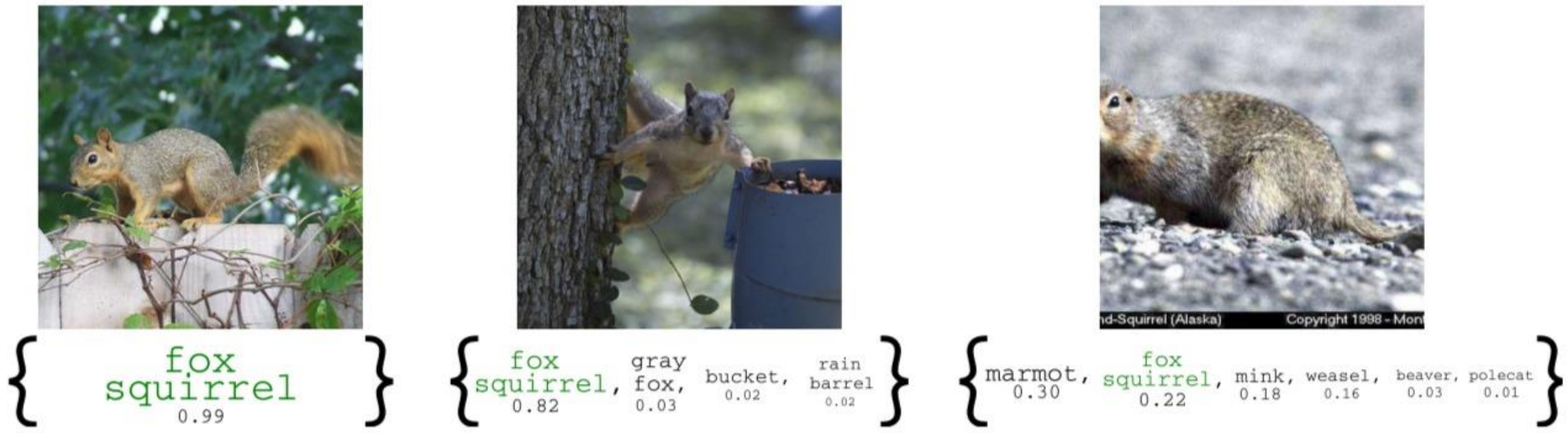
"Temperature scaling" of softmax outputs as $softmax(a/T)$ is a popular and simple approach to reduce overconfidence (for figure on right, $a = [3,0,1]$)



Conformal Prediction

Conformal Prediction

- A simple technique to easily obtain confidence intervals
  - In classification, such an interval may refer to the set of highly likely classes for a test input


{ fox squirrel 0.99 }
{ fox squirrel, 0.82 gray fox, 0.03 bucket, 0.02 rain barrel 0.02 }
{ marmot, 0.30 fox squirrel, 0.22 mink, 0.18 weasel, 0.16 beaver, 0.03 polecat 0.01 }
d-Squirrel (Alaska)
Copyright 1998 - Mon

  - For more difficult test inputs, the set would typically be larger
- In a way, conformal prediction gives predictive uncertainty
  - However, unlike Bayesian ML, we don't get model uncertainty
  - Only one model is learned in the standard way and we construct the set of likely classes
  - It's like a black-box method; no change to training procedure for the model

Conformal Prediction

Assume it's a classification model which produces softmax scores

Conformal prediction can be used for regression problems too*

- Assume we already have a trained model $\hat{f}$ using some labelled data
- Idea: Use a calibration set of n examples to generate a prediction set $\mathcal{C}(X_{test})$ s.t.

α is a user chosen error rate

Its true label

Another fresh test input

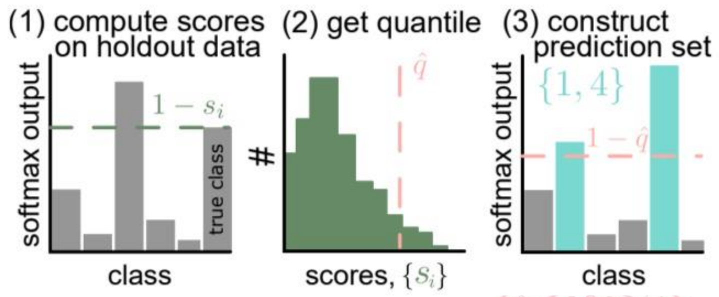
$$1 - \alpha \leq p\big(Y_{test} \in \mathcal{C}(X_{test})\big) \leq 1 - \alpha + \frac{1}{n+1}$$

- The approach* to construct the prediction set $\mathcal{C}(X_{test})$ is as follows:
  - Assuming classification task, for each example in the calibration set, compute

high means bad prediction by the model

Conformal score: one minus the softmax score of the correct class

$$s_i = 1 - \hat{f}(x_i)_{y_i}$$



  - Compute the $1 - \alpha$ quantile of $s_1, s_2, \ldots, s_n$. Call it $\hat{q}$
  - Now the calibration set for a new test input X_{test} can be defined as

Set of all classes whose predicted softmax values are "high enough"

$$\mathcal{C}(X_{test}) = \{y : \hat{f}(X_{test})_y \geq 1 - \hat{q}\}$$

*A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification (Angelopoulos and Bates, 2022)