



university of
groningen

faculty of science
and engineering

Evaluation of Uncertainty Quantification

Dr. Matias Valdenegro
Department of Artificial Intelligence
University of Groningen

February 7, 2024

Today's Agenda

- ① Introduction
- ② Metrics for Regression and Classification
- ③ Evaluation of Calibration
- ④ Other Evaluation Plots

Outline

- 1 Introduction
- 2 Metrics for Regression and Classification
- 3 Evaluation of Calibration
- 4 Other Evaluation Plots

Concept

We need specific methods to evaluate the quality of Uncertainty produced by our models.

The standard evaluation formulations do not generally consider uncertainty at the output, so they cannot be used as they will ignore any output uncertainty.

What we need is losses, metrics, and evaluation methods to see if uncertainty is good or not, and assign scores to its quality.

Concept

The overall importance of Uncertainty and to have specific evaluation methods is:

Error and misclassification should be proportional to output uncertainty made by a model.

This way output uncertainty is useful for the end user of your model. This applies both to regression and classification with uncertainty.

Note that the relationship does not have to be perfectly linear.

Warning

Selection of loss functions and metrics is **the most important choice** that you make when developing a machine learning model.

Losses should be selected¹ according to the task (regression, classification, object detection, segmentation, etc).

Metrics (accuracy, mean absolute error, R^2 score, etc) should be selected based on what knowledge do you want to obtain from the model, and what kind of performance is required to be measured.

We will see some new losses and metrics in this lecture, specifically about evaluating models with output uncertainty.

¹Losses also need to be differentiable

Learning Performance

The first thing is to define how to measure the performance of a learning model.

Losses

Objective function that guides learning during the optimization process. It defines the task to be learned and the quality of solutions. Usually it has to be differentiable.

Metrics

Measurements of quality that let the ML developer evaluate the learning process' success. Usually non-differentiable. Losses can be used as Metrics as well.

Probabilistic Classifiers

Most classifiers output a probability vector p of length C . The class integer class index c can be recovered by:

$$c = \arg \max_i p_i \quad (1)$$

Note that for C classes, their indices go from 0 to $C - 1$. For binary classification, only a single probability is required:

$$f(x) = P(y = 1) = 1 - P(y = 0) \quad (2)$$

In this case, the classifier outputs the probability of class 1 (usually the positive class), while the probability for class 0 (the negative class) can be recovered by subtracting with one.

Obtaining Confidences

Classification

A confidence measure is the maximum probability:

$$\sigma = \max_i p_i \quad (3)$$

Another confidence measure is the entropy h :

$$h = - \sum_i p_i \log p_i \quad (4)$$

But entropy goes the other way, low entropy is high confidence, high entropy is low confidence.

Regression

For a regression model, the standard deviation σ (square root of variance) which should be an output head is a confidence measure.

Loss Functions - Classification

Categorical Cross-Entropy

For this loss, labels y^c should be one-hot encoded. Used for multi-class classification problems, where the model predictions are \hat{y}_i^c are class probabilities that sum to 1.

$$L(y, \hat{y}) = - \sum_i \sum_c y_i^c \log(\hat{y}_i^c)$$

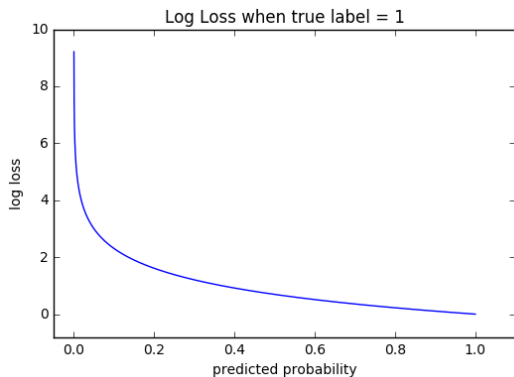
Binary Cross-Entropy

Used for binary classification problems with labels $y_i \in \{0, 1\}$

$$L(y, \hat{y}) = - \sum_i y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

Loss Functions - Classification

Binary Cross-Entropy



Loss Functions - Negative Log-Likelihood

Log-likelihoods are a family of losses or objective functions. For regression with uncertainty, the following loss is commonly used.

$$-\log p(y_n|\mathbf{x}_n) = \frac{\log \sigma_i^2(\mathbf{x}_n)}{2} + \frac{(\mu_i(\mathbf{x}_n) - y_n)^2}{2\sigma_i^2(\mathbf{x}_n)} + C \quad (5)$$

Here the model outputs two variables. A mean $\mu(x)$ and variance $\sigma^2(x)$, and these are weighted. If the model is uncertain (large σ^2) then the squared error is ignored but the logarithm of variance counteracts this effect, and if the model is certain (small σ^2), then the opposite effect happens.

This loss assumes that the model outputs the parameters of a Gaussian distribution.

Loss Functions - Others

Kullback-Leibler Divergence

Distance measure between probability distributions p and q . The Cross-Entropy is a simplified version of this loss (with some assumptions).

$$L(p, q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \quad L(y, \hat{y}) = \sum_i y_i \log \left(\frac{y_i}{\hat{y}_i} \right)$$

Loss Functions with Uncertainty

Some special loss functions that I mentioned here do consider uncertainty.

- **Cross Entropy**. Special case of NLL for classification, also considers the probabilities/confidences of the correct class.
- **Gaussian NLL**. Special case of NLL for regression with Gaussian distributed output. Models uncertainty through the variance output.
- **KL Divergence**. General case of many losses, measures distance between probability distributions, which implicitly models uncertainty.

Reproducibility

Let's say we train the same neural network architecture, on the same dataset, 5 times.

Question. Are each of these models the same?

Question. Are predictions the same?

Question. Assuming UQ was applied, is Aleatoric Uncertainty the same?

Question. Assuming UQ was applied, is Epistemic Uncertainty the same?

Reproducibility

Let's say we train the same model, on the same dataset, 5 times.

A model is considered to be a specific instance where the following are unique:

1. The dataset/task it was trained on.
2. The weight values.
3. The model architecture (layers and equations).

Overall with different weights, predictions will be different.

Reproducibility

Aleatoric Uncertainty should be unchanged (to a degree of precision).

Epistemic Uncertainty will be different but to a varying degree (could be very similar or radically different).

This is of course making the assumption that the model is properly trained (loss decreased and converged to similar values).

Interaction between Aleatoric and Epistemic Uncertainty

An important detail we previously discussed, is that Aleatoric and Epistemic Uncertainty do have some degree of interaction, but this is **only** in the case where both sources of uncertainty are predicted by a model.

Aleatoric Uncertainty has a degree of its own Epistemic Uncertainty, since we are estimating Aleatoric Uncertainty with a model (the variance head trained with the Gaussian NLL).

Epistemic Uncertainty is directly related to the model so there is no additional interaction.

Outline

- 1 Introduction
- 2 Metrics for Regression and Classification
- 3 Evaluation of Calibration
- 4 Other Evaluation Plots

Proper Scoring Rules [DeGroot and Fienberg, 1983]

We consider scoring functions, which is the general concept for scoring probability distribution outputs.

A scoring rule is a function $S(p_\theta, (y, x))$ that evaluates the quality (higher is better) of a predicted probability distribution $p_\theta(y | x)$ relative to an event $y | x \sim q(y | x)$, with $q(y | x)$ being the true distribution that generates (y, x) values.

The expected scoring rule is given by:

$$S(p_\theta, q) = \int q(y, x) S(p_\theta, (y, x)) dy dx \quad (6)$$

Proper Scoring Rules - Definition

A proper scoring rule is one where:

$$S(p_\theta, q) \leq S(q, q) \quad (7)$$

With equality only holding if $p_\theta(y | x) = q(y | x)$ for all p_θ and q .

This is easy to interpret, a proper scoring rule obtains its best value only by predicting the true distribution q .

Any scoring rule can be made into a loss function, by taking $L(\theta) = -S(p_\theta, q)$.

Proper Scoring Rules - Examples

- The general log-loss, with score function $S(p_\theta, (y, x)) = \log p_\theta(y | x)$.
- The cross-entropy loss.
- The KL Divergence.
- The Brier score.
- The Gaussian Negative Log-Likelihood.

Improper Scoring Rules

- Standard accuracy used for classification problems.

$$\text{Acc}(y, \hat{y}) = N^{-1} \sum 1[y_i = \hat{y}_i] \quad (8)$$

As it does not consider prediction confidence, and it is possible to obtain 100% accuracy while not predicting the right distribution.

- The mean squared error and mean absolute error, as they do not consider distribution outputs.
- Custom metrics that use geometric means, for example $M(y, \hat{y}) = \sqrt{y_i \hat{y}_i}$, since the metric can be minimized by predicting zeros.

What does this mean?

So the concept of proper scoring rules is slightly complex, but it can be summarized as:

- There are good scoring rules (proper), that when used, will lead to predicting the true distribution (with some limitations from model and data availability).
- There are bad scoring rules (improper), which can mislead the user, and do not always lead to predicting the true distribution.

And since we need distribution outputs for proper uncertainty quantification, we should focus on using proper scoring rules.

Coverage - Regression

This is a continuous version of a conceptual accuracy metric for regression with uncertainty.

Given a set of confidence intervals $[l, u]_i$ (with upper and lower bounds), the concept is that an interval can contain the correct label y or not, then coverage can be computed as:

$$\text{Cov}(y, \hat{y}) = N^{-1} \sum 1[\hat{y}_i^l \leq y \leq \hat{y}_i^h] \quad (9)$$

Where \hat{y}_i^l is the i -th predicted lower bound, and \hat{y}_i^h is the corresponding upper bound.

Differential Entropy - Regression

Entropy can also be extended to continuous distributions, measured as follows for a PDF $f(x)$:

$$h(X) = - \int f(x) \log f(x) dx \quad (10)$$

For example the entropy of a Gaussian distribution is $\frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2}$. As you can see, the entropy only depends on the variance σ^2 of the distribution.

Brier Score - Classification

This is basically the mean squared error, but applied to output probabilities vs the target probability distribution (could be one-hot encoding or per-class probabilities).

$$\text{Brier}(y, \hat{y}) = N^{-1} \sum (y_i - \hat{y}_i)^2 \quad (11)$$

The brier score measures how close are the predicted and true probabilities, in an interpretable and intuitive way. It is a proper scoring rule.

Entropy - Classification

Entropy is a measurement of the "information content" in a probability distribution. For classification, as we use a discrete probability distribution that is conditioned by each class $\mathbb{P}(x | c)$, we can define entropy as:

$$H = - \sum_c \mathbb{P}(x | c) \log \mathbb{P}(x | c) \quad (12)$$

Where c is each of the class indices $c \in [0, C - 1]$.

The entropy is important as it is directly related to uncertainty. The units of entropy are called bits if one uses the base-two logarithm.

The uniform distribution is the one that maximizes entropy, as its results are harder to predict. For a fixed mean and variance, the Gaussian distribution is the one that maximizes the entropy.

Outline

- 1 Introduction
- 2 Metrics for Regression and Classification
- 3 Evaluation of Calibration**
- 4 Other Evaluation Plots

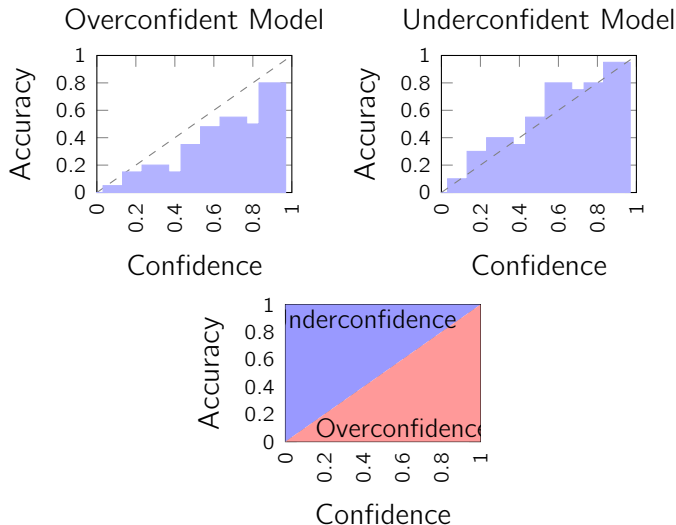
Calibration

- We talked about a concept that indicates how much we can trust the confidence of a model.
- This can be formalized by comparing task performance (such as accuracy) as the confidence of predictions change.
- For example, if a prediction is made with 10% confidence, then we expect that such predictions will be correct 10% of the time.
- And correspondingly, if a prediction is made with 90% confidence, then only 10% of such predictions will be incorrect.

Calibration - Reliability Plots

- Calibration can be observed by making a Reliability plot.
- We take the predictions of a model over a dataset, divide the predictions by confidence values $\text{conf}(B_i)$ into bins B_i , for each bin the accuracy $\text{acc}(B_i)$ is computed, and then the values $(\text{conf}(B_i), \text{acc}(B_i))$ are plotted.
- Regions where $\text{conf}(B_i) < \text{acc}(B_i)$ indicate that the model is underconfident, while regions $\text{conf}(B_i) > \text{acc}(B_i)$ indicate overconfidence.
- The line $\text{conf}(B_i) = \text{acc}(B_i)$ indicates perfect calibration.

Calibration - Reliability Plots



Over and Under Confidence

Overconfidence

Regions where overall confidence is higher than accuracy indicate an overconfident model, so confidences should be lower than they are. This is the worst case as a high confidence gives a false sense of security.

Underconfidence

Regions where overall confidence is lower than accuracy indicate that the model is underconfident, which means that the confidence should actually be higher than it is, in order to match accuracy.

Note that a model can be both under and over confident at the same time, but in different regions of the confidence space.

Calibration - Metrics

Calibration Error

This is the standard metric to measure miscalibration. It is affected by variations in the number of samples in each bin.

$$CE = \sum_i |\text{acc}(B_i) - \text{conf}(B_i)|$$

Calibration - Metrics [Guo et al., 2017]

Expected Calibration Error

In order to compensate for the varying number of elements in each bin B_i , ECE weights the error produced by each bin by the proportion of samples in that bin with respect of the total of samples.

$$\text{ECE} = \sum_i N^{-1} |B_i| |\text{acc}(B_i) - \text{conf}(B_i)|$$

This produces a metric that is much more stable and less prone to outliers. This is usually the metric that is commonly reported in scientific publications.

Calibration - Metrics [Guo et al., 2017]

Maximum Calibration Error

For risk averse applications, the MCE computes the maximum level of miscalibration, so appropriate risk can be estimated and addressed by the application designed. Variations in miscalibration in each bin can hide in the overall mean or expectation.

$$\text{MCE} = \max_i |\text{acc}(B_i) - \text{conf}(B_i)|$$

For example, in Autonomous Driving, the maximum miscalibration should be close to zero.

Calibration - Reliability Plots with MC-Dropout on MNIST

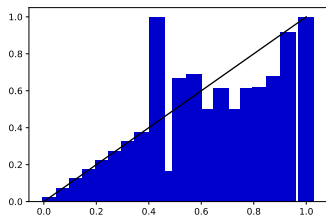


Figure: Classical NN, Calibration error is 0.18

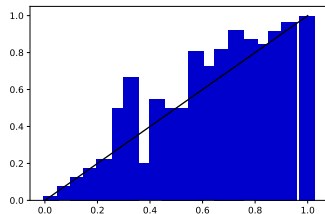


Figure: Bayesian NN with MC-Dropout, Calibration error is 0.11

Calibration in Regression

In a regression setting, we have confidence intervals $[l, u]$ that have a certain confidence α associated with them, that is, for a prediction, the following should hold:

$$\mathbb{P}(l \leq y \leq u) = \alpha \quad (13)$$

This means for a predictive interval $[l, u]$, the probability that it contains the true value is α . This value is called the confidence of the interval.

Calibration in Regression

The interval bounds u and l are a function of the confidence α , so let's start from our previous definition:

$$\mathbb{P}(l \leq y \leq u) = \alpha \quad (14)$$

Given a fixed value of α , how can we compute u and l ?

Calibration in Regression

The interval bounds u and l are a function of the confidence α , so let's start from our previous definition:

$$\mathbb{P}(l \leq y \leq u) = \alpha \quad (14)$$

Given a fixed value of α , how can we compute u and l ?

For this we need to make some assumptions, for example, assume that y is Gaussian distributed, or uniformly distributed. By using the following equation:

$$\mathbb{P}(a \leq X \leq b) = F(b) - F(a) \quad (15)$$

What is F in the above formula?

Gaussian Confidence Intervals [Spiegel and Stephens, 2018]

So let's assume that F is the CDF of a Gaussian distribution. Then the computation of a α confidence interval has a pseudo-closed form:

$$l = \mu - |z_{\frac{\beta}{2}}| \sigma \quad (16)$$

$$u = \mu + |z_{\frac{\beta}{2}}| \sigma \quad (17)$$

Where $\beta = 1 + \alpha$ and $z_{\frac{\beta}{2}}$ is the z score² corresponding to the $\frac{\beta}{2}$ quantile. This can be computed with the quantile function (inverse of the CDF, $z = F^{-1}(\frac{\beta}{2})$), but for Gaussian distributions there are standard tables that you can use.

² $z = \frac{x - \mu}{\sigma}$

Gaussian Confidence Intervals

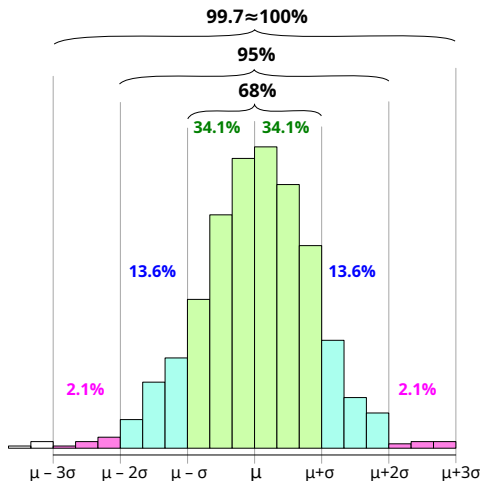
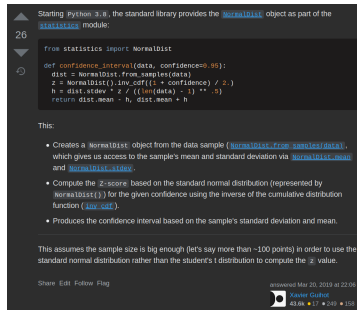


Figure: Empirical rules to compute specific confidence intervals for a Gaussian Distribution. Source: https://en.wikipedia.org/wiki/68-95-99.7_rule

Practical Computation of Gaussian Confidence Intervals

In practice there are already made functions that can compute a confidence interval, for example

`scipy.stats.norm.interval`, and even the python standard library includes the inverse CDF of the Gaussian.



```
Starting Python 3.8, the standard library provides the NormalDist object as part of the statistics module.
```

```
26 from statistics import NormalDist
```

```
def confidence_interval(data, confidence=0.95):  
    dist = NormalDist.from_samples(data)  
    z = NormalDist().inv_cdf((1 + confidence) / 2.)  
    h = dist.stdev * z / ((len(data) - 1) ** .5)  
    return dist.mean - h, dist.mean + h
```

This:

- Creates a `NormalDist` object from the data sample (`NormalDist.from_samples(data)`), which gives us access to the sample's mean and standard deviation via `NormalDist.mean` and `NormalDist.stdev`.
- Compute the `z`-score based on the standard normal distribution (represented by `NormalDist()`) for the given confidence using the inverse of the cumulative distribution function (`inv_cdf`).
- Produces the confidence interval based on the sample's standard deviation and mean.

This assumes the sample size is big enough (let's say more than ~100 points) in order to use the standard normal distribution rather than the student's `t` distribution to compute the `z` value.

Show Edit Follow Flag answered Mar 20, 2019 at 22:06
Kater Golub
43.6k • 17 • 559 • 158

Source:

<https://stackoverflow.com/questions/15033511/compute-a-confidence-interval-from-sample-data>

Calibration in Regression

1. Define a set of confidences $S_\alpha = [0.0, 0.1, 0.2, \dots, 0.9, 1.0]$ by dividing the $[0, 1]$ range into N equally spaced values.
2. For each confidence $\alpha \in S_\alpha$:
 - 2.1 Define an α confidence interval for the per-sample distribution with parameters μ_i and σ_i^2 (Note³). These are one confidence interval per sample in your dataset, defined by the distribution and confidence level α .
 - 2.2 Compute the coverage (confidence interval accuracy) acc using the previously computed per-sample confidence intervals.
 - 2.3 Add (α, acc) to the plot.
3. Display the plot.

³ i being the sample index

Calibration in Regression

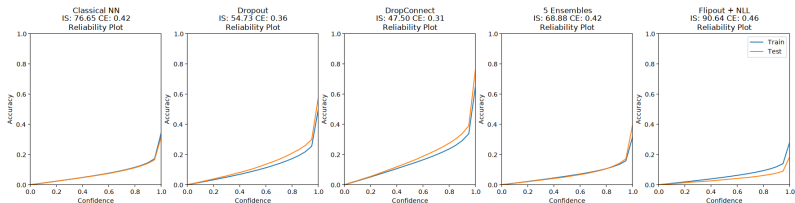


Figure: Reliability plot for toy regression, across different UQ methods in the test set. It is clear their uncertainty is not well calibrated, as there is a large gap to the diagonal.

Outline

- 1 Introduction
- 2 Metrics for Regression and Classification
- 3 Evaluation of Calibration
- 4 Other Evaluation Plots

Error vs Confidence Plots

The idea of uncertainty is that it should be proportional to error or misclassification. One way to evaluate this is to plot the relationship between confidence and error.

Conceptually what we want to do is look at error in individual predictions, as a function of uncertainty/confidence.

What should happen is:

- Predictions with low confidence, should have a large error or be misclassified.
- Predictions with high confidence, should have a small error or be correctly classified.

This is similar to calibration but not the same.

Error vs Confidence Plots

1. Take predictions with uncertainty over a dataset, take the min and max uncertainty/confidence σ , divide the range into steps U .
2. For each step $u \in U$, perform:
 - 2.1 Threshold all predictions with $\sigma \geq u$, discarding predictions that do not meet the threshold.
 - 2.2 Compute some error metric e for the remaining predictions, using μ and ground truth labels y (for example accuracy/error or mean squared error).
 - 2.3 Add (u, e) to the plot.
3. Display the plot.

Error vs Confidence Plots - Classification

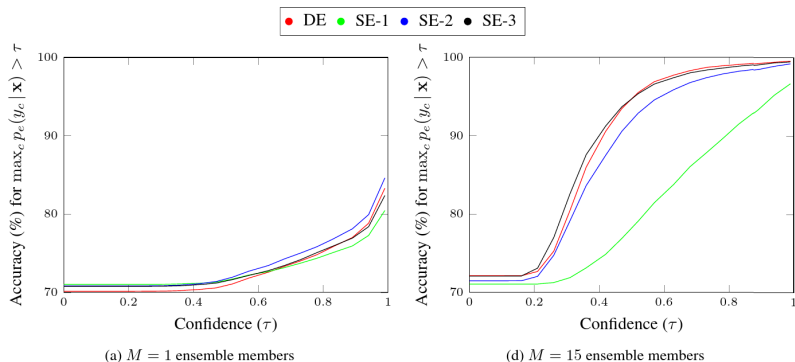


Figure: Comparison of Ensembles and their confidence-accuracy plot, Model trained on SVHN and tested on CIFAR10 (This is Out of Distribution)

Error vs Confidence Plots - Regression

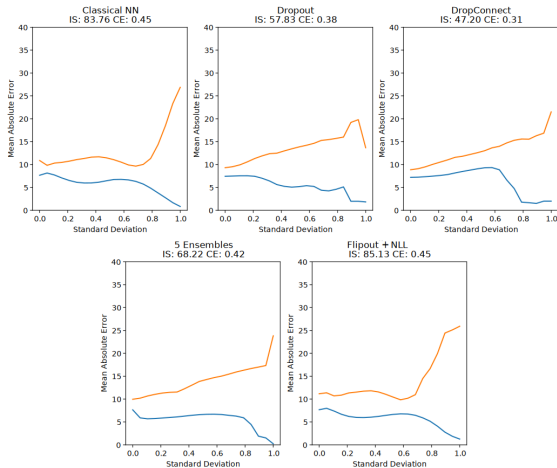


Figure: Comparison of different UQ methods and their confidence vs error plot, on a toy regression problem. The standard deviation is normalized for comparison purposes. Blue is train, and orange is test.

Error vs Confidence Plots - Error Metrics

Classification

Loss, accuracy, error (1 - accuracy), brier score, etc.

Regression

Mean squared error, mean absolute error, R^2 score.

Selecting different error metrics will make a different plot, revealing varying properties of your model and its quality of uncertainty.

Error vs Confidence Plots - Interpretation

The interpretation of a error vs confidence plot is less clear than the reliability plot.

- The plot does not need to be compared to a diagonal line.
- Increasing confidence should lead to a lower error/higher accuracy.
- Decreasing confidence should lead to a higher error/lower accuracy.
- This plot can be used to compare different models or uncertainty quantification methods.

Note that error and accuracy have inverse relationships.

Alternative Error vs Confidence

An alternate plot that can be made is plotting each sample in a dataset, with some measure of error in the Y axis, and confidence/uncertainty in the X axis. There is no thresholding in this case, but a direct comparison between error and confidence.

The plot should be a scatterplot, as we are plotting relationship of individual data points, not particularly a function.

One important detail is that there should be a monotonic (increasing) relationship between error and confidence, but this relationship does not have to be linear.

Alternative Error vs Confidence

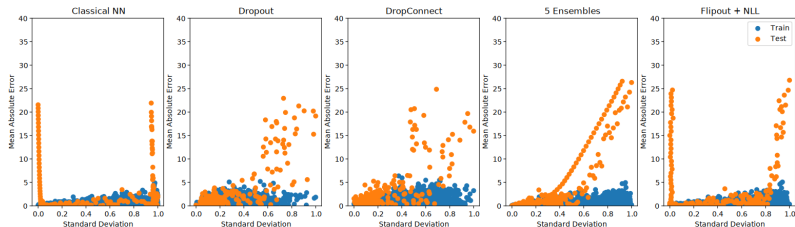


Figure: Comparison of different UQ methods and their confidence vs error scatterplot, on a toy regression problem. The standard deviation is normalized for comparison purposes. Blue is train, and orange is test.

Questions?

Questions to Think About

- What particular loss functions consider uncertainty and which ones do not?
- How is a calibration plot / reliability plot made?
- How is a confidence interval built?
- What to consider when selecting metrics/losses for your problem?

Bibliography I

- Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.
- Murray R Spiegel and Larry J Stephens. *Schaum's outline of statistics*. McGraw-Hill Education, 2018.