Uncertainty Quantification 360

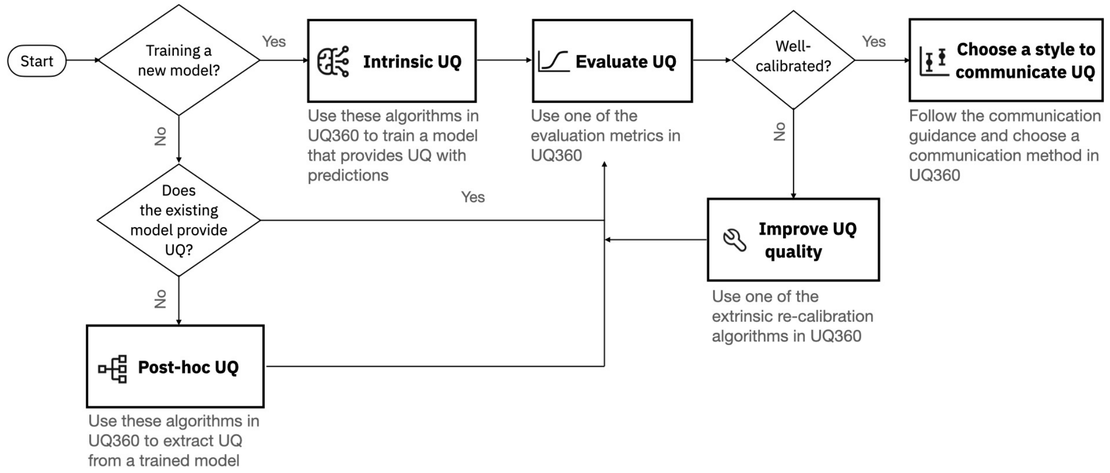
Please join the Slack community
http://aif360.mybluemix.net/community

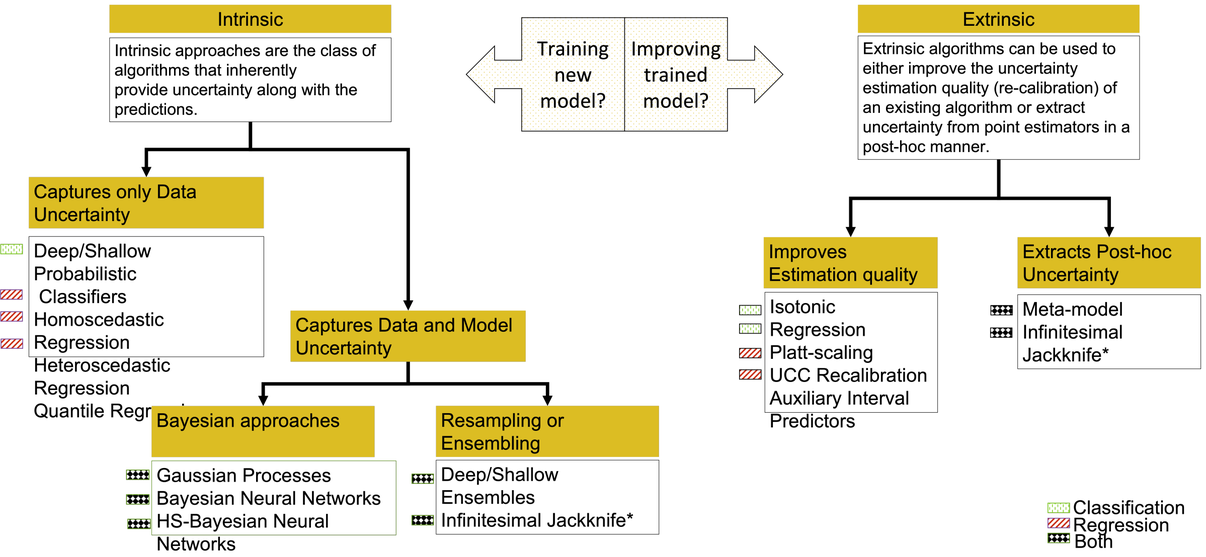
Channel: #uq360-users

Prasanna Sattigeri
Soumya Ghosh
Jiri Navratil
IBM Research

http://uq360.mybluemix.net
https://github.com/ibm/uq360
https://pypi.org/project/uq360

UQ360





Agenda

Introduction to UQ360
Interactive Web Experience
Package Installation
break
IJ
Meta-models + UCC
Connections to other Pillars of Trust

Please join the Slack community
http://aif360.mybluemix.net/community

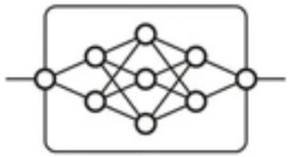
Channel: #uq360-users

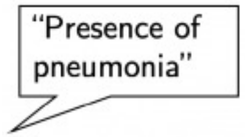
WHAT IS UNCERTAINTY QUANTIFICATION IN AI?

It is the ability for an AI model to say *I am UNSURE*

Patient X-ray

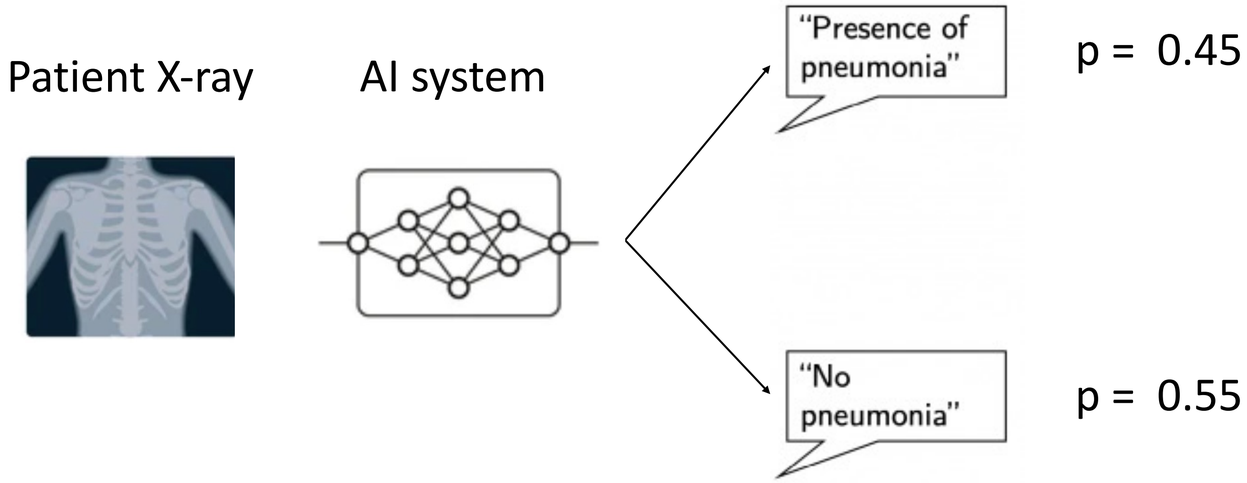
AI system





https://www.csail.mit.edu/news/ai-systems-work-wdoctors-and-know-when-step

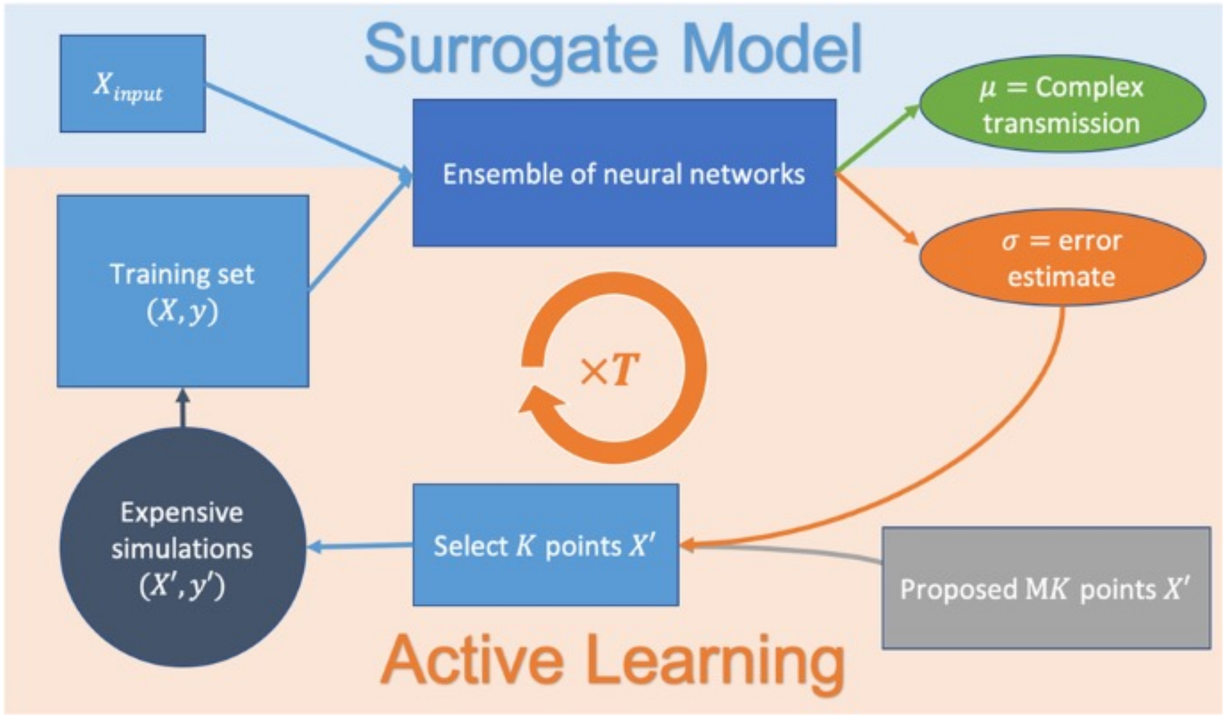
SAFETY AND HUMAN-AI COLLABORATION



https://www.csail.mit.edu/news/ai-systems-work-wdoctors-and-know-when-step

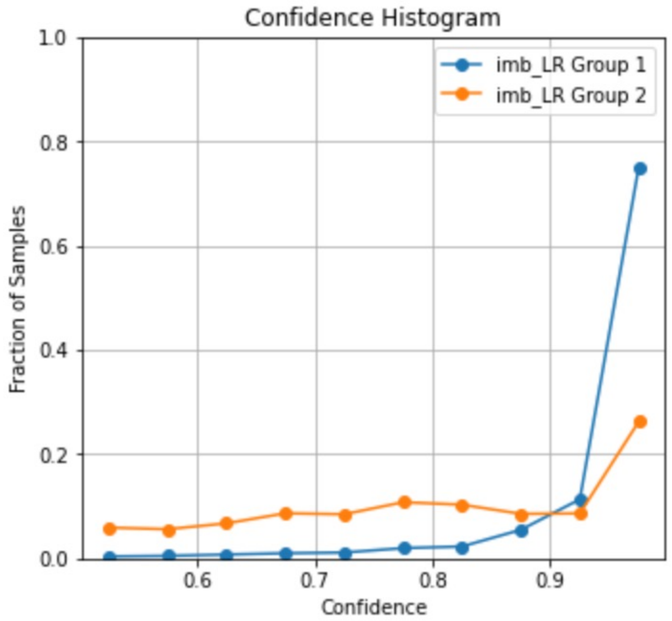
WHY CARE ABOUT UNCERTAINTY?

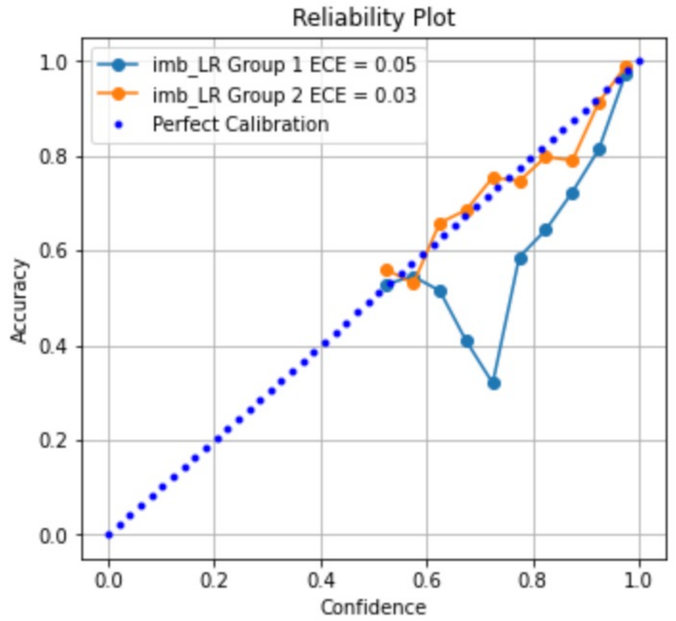
MODEL IMPROVEMENT



https://www.nature.com/articles/s41524-020-00431-2

ARE AI MODEL UNCERTAINTIES RELIABLE? - METRICS

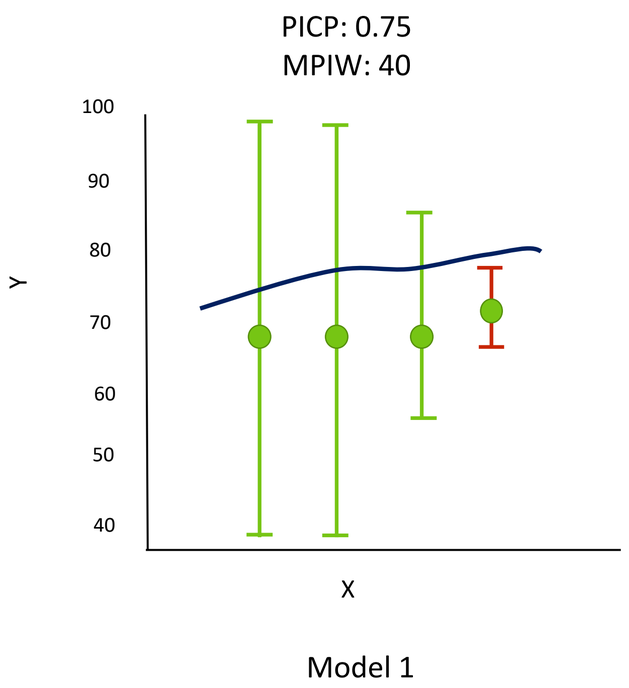




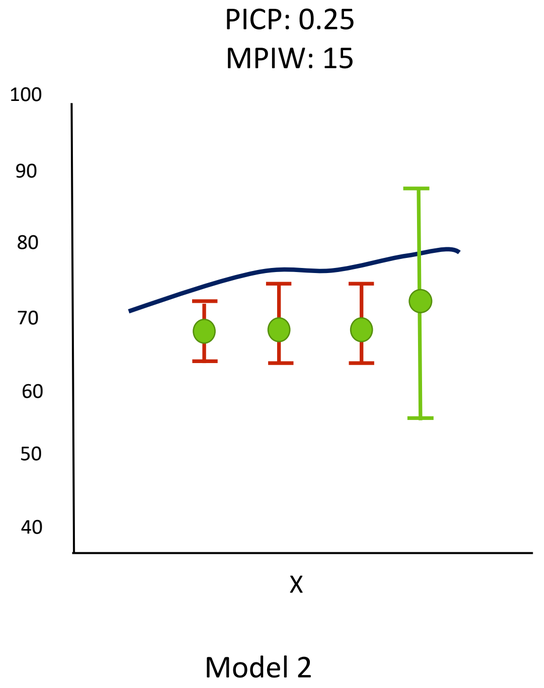
Guo, Chuan, et al. "On calibration of modern neural networks." *International Conference on Machine Learning*. PMLR, 2017.

ARE AI MODEL UNCERTAINTIES RELIABLE? - METRICS

- Prediction Interval Coverage Probability (PICP) or Miss-rate
- Mean Prediction Interval Width (MPIW) or Bandwidth

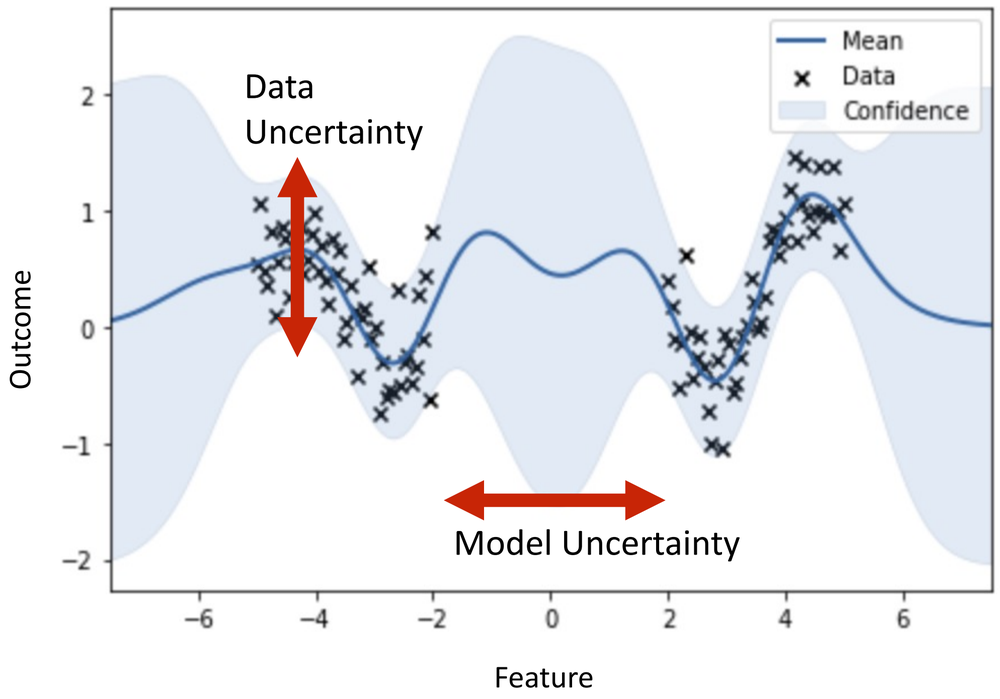


Model 1



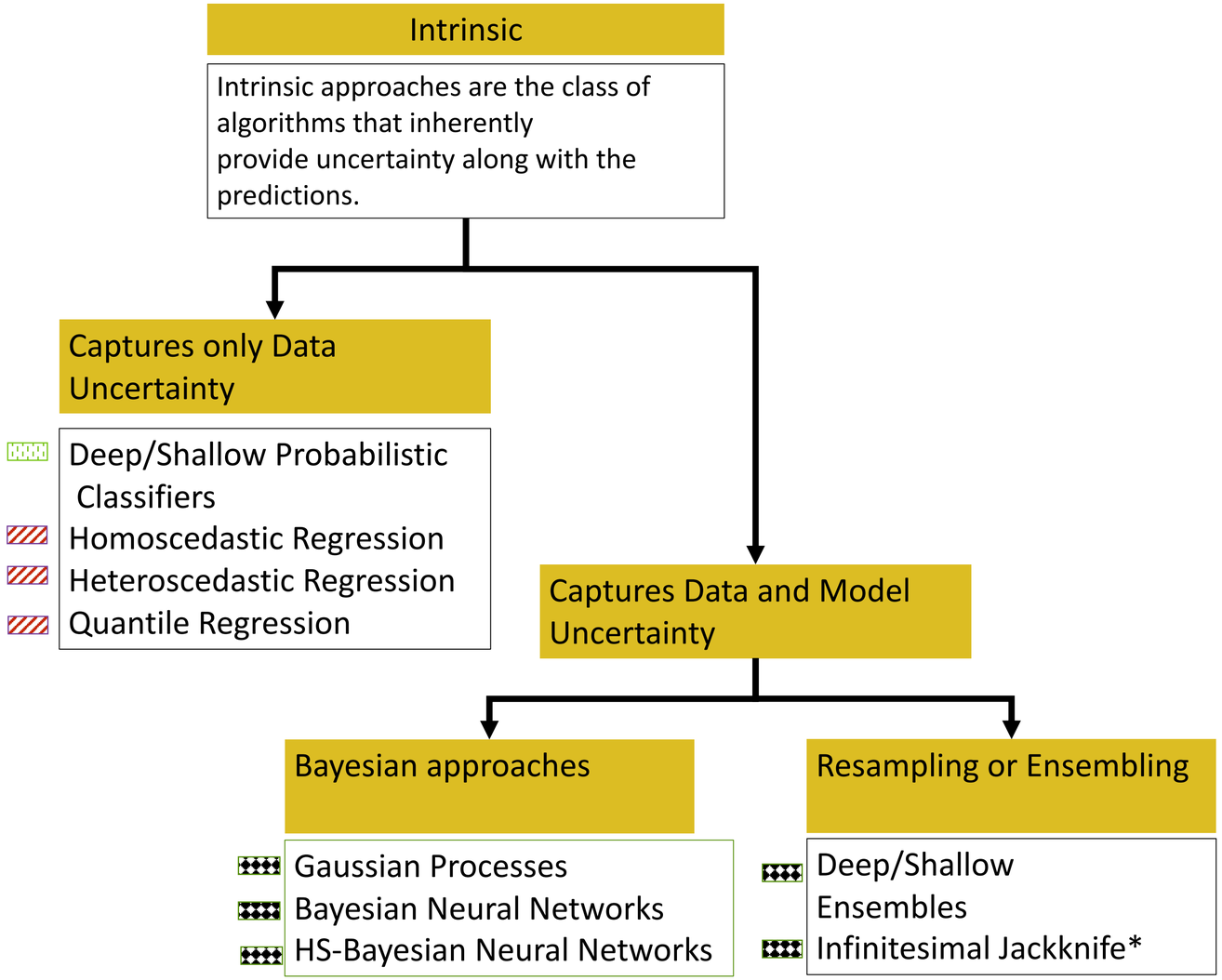
Model 2

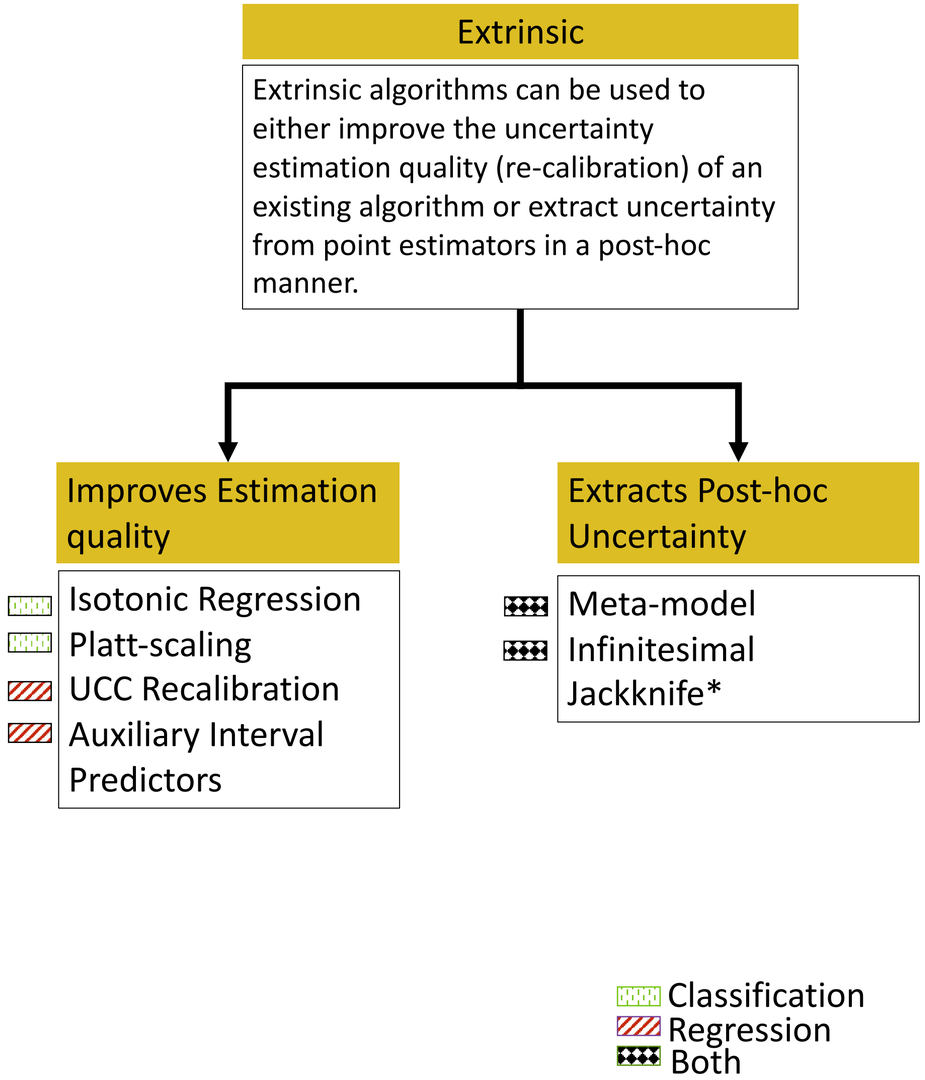
ARE AI MODEL UNCERTAINTIES RELIABLE?



Inability to capture all sources of uncertainty

UQ ALGORITHMS





UQ COMMUNICATION METHODS

Guidance on Communicating Uncertainty

Overview | Communicate for Regression | Communicate for Classification

Overview: What to consider when choosing communication methods

Communicating UQ means presenting the output of UQ estimates to stakeholders, assuming you have chosen the right UQ algorithm to generate the right type of UQ estimates (see our UQ algorithm selection guidance). This is a crucial step because even a well-calibrated UQ estimates could be misunderstood by people if they have difficulty or biases in interpreting the numbers or statistics. In this guide, we will introduce you to some key considerations for communicating UQ and example methods. In practice, it is necessary to conduct tests with your target users or stakeholders to make sure the chosen UQ communication method is understood.

Let's start with a few key questions that should guide your choice of UQ communication methods.

What is the form of the UQ output?

The first step is to identify the form of the UQ to be communicated, i.e. whether it is a single confidence score or a distribution of possible outcomes. In current ML tasks, the former is what UQ estimates of a classification model looks like, and the latter is the form of UQ estimates of a regression model.

Note here we focus on the form instead of the source of UQ. For example, for a regression model, UQ of different sources, whether it is data uncertainty, model uncertainty, or overall predictive uncertainty, are all distributions of possible outcomes. They could be communicated in the same way but it is possible that users would perceive them or act on them differently.

Communicating UQ of a single instance or a group of instances?

Then select a communication method below:

Range of interval (verbal)
Easy to read at a glance, but could miss the details of how possible values are distributed in the range

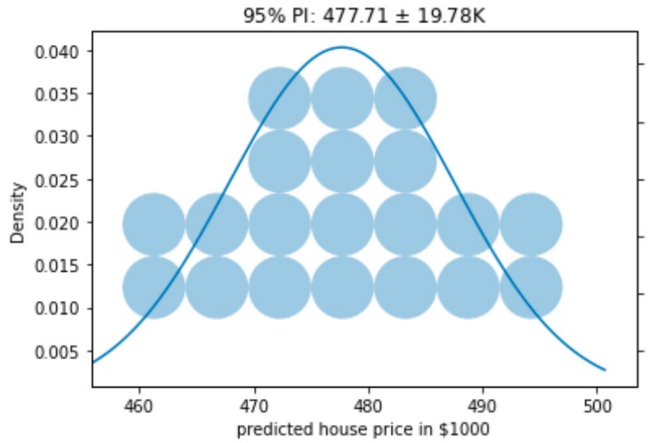
Probability density plot
Gives detailed information with a visualization about how possible values are distributed in the prediction interval

Quantile dot plot
Shows distribution with a visualization that makes it easier to judge the relative likelihood of where possible values can fall

Recommended price:

478K

The quantile plot below shows the probability of the right price.



Ella understands that while the model prediction output is 478K, the recommended range says the price can fall anywhere between 458K and 498K.

This is a somewhat large range. Ella has to also take into consideration what she knows about the housing market:

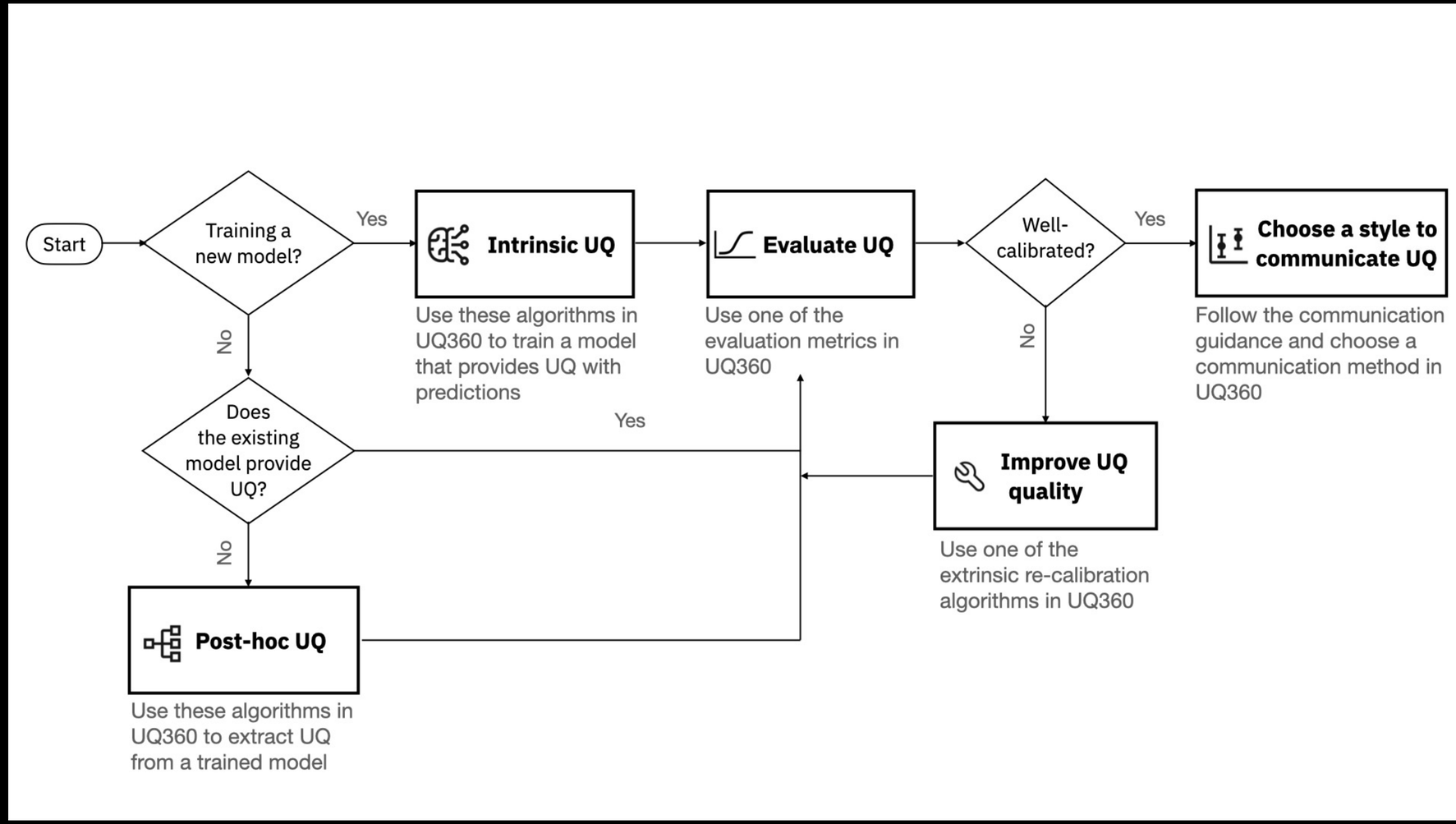
- The buying demand is on the rise in the neighborhood
- It is common to set the first price slightly higher

Many choices of UQ algorithms

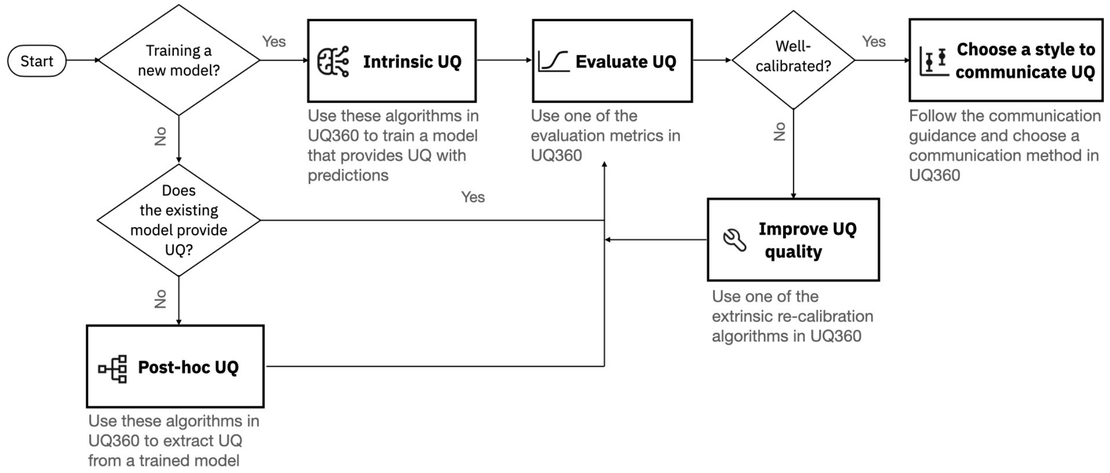
UQ needs to be evaluated, and if needed, improved

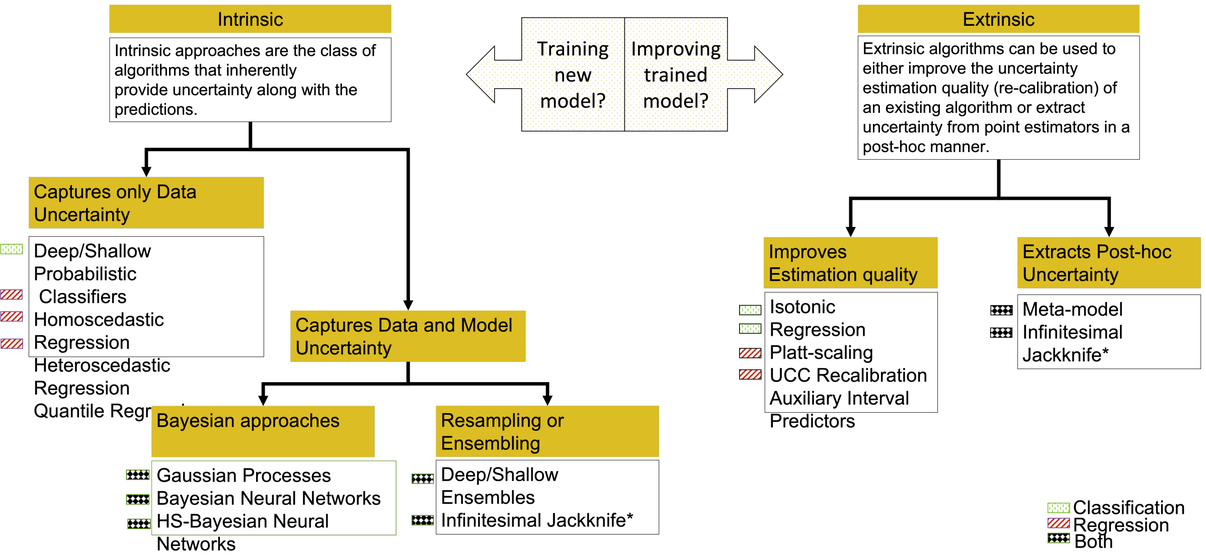
UQ needs to be communicated in the right way

UQ360 PROVIDES A WORKFLOW



UQ360





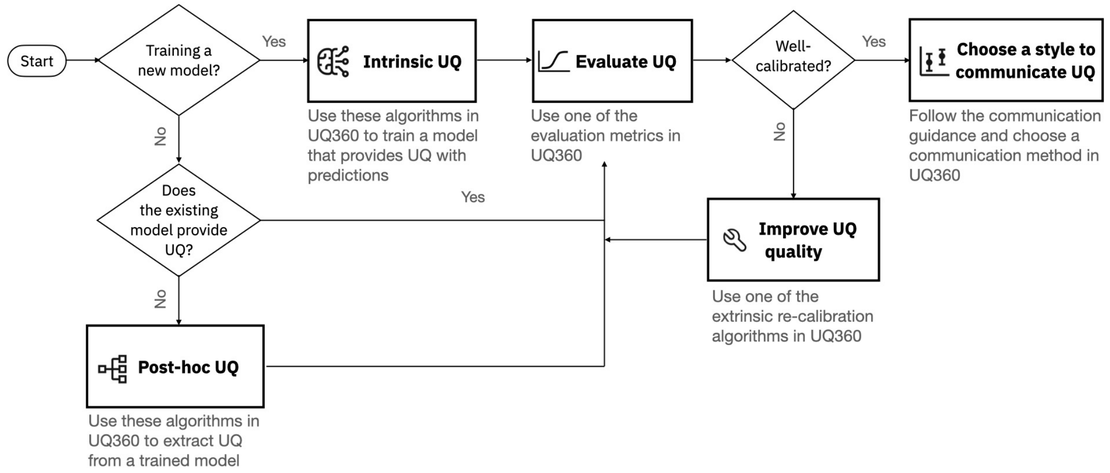
Agenda

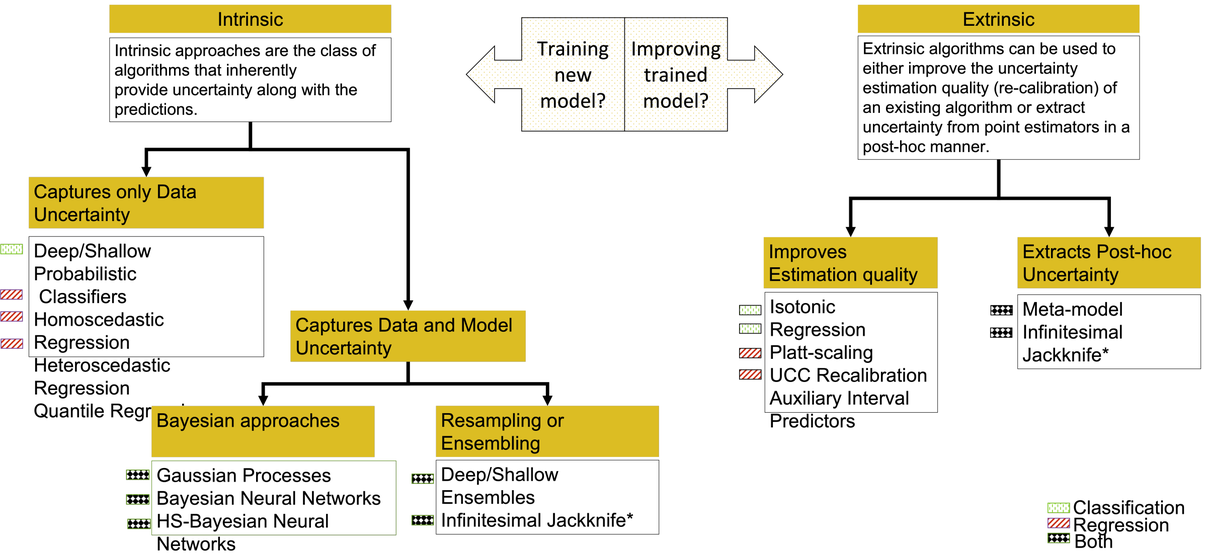
Introduction to UQ360
Interactive Web Experience
Package Installation
break
IJ
Meta-models + UCC
Connections to other Pillars of Trust

Please join the Slack community
http://aif360.mybluemix.net/community

Channel: #uq360-users

UQ360





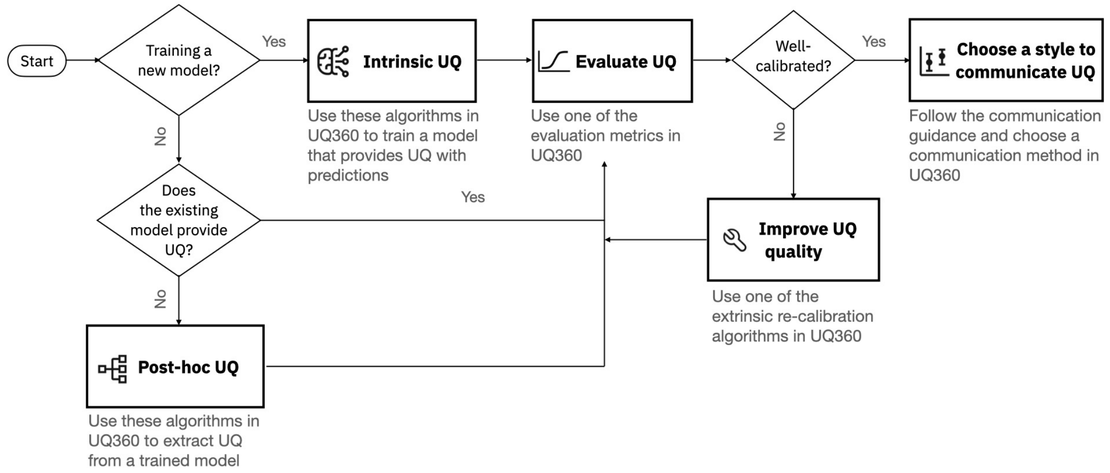
Agenda

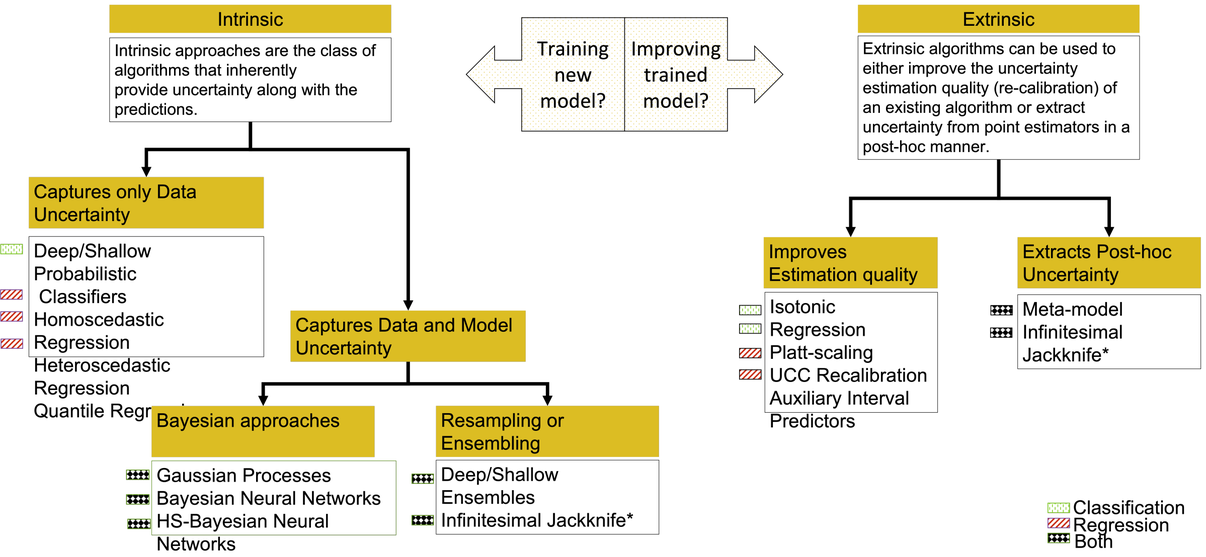
Introduction to UQ360
Interactive Web Experience
Package Installation
break
IJ
Meta-models + UCC
Connections to other Pillars of Trust

Please join the Slack community
http://aif360.mybluemix.net/community

Channel: #uq360-users

UQ360





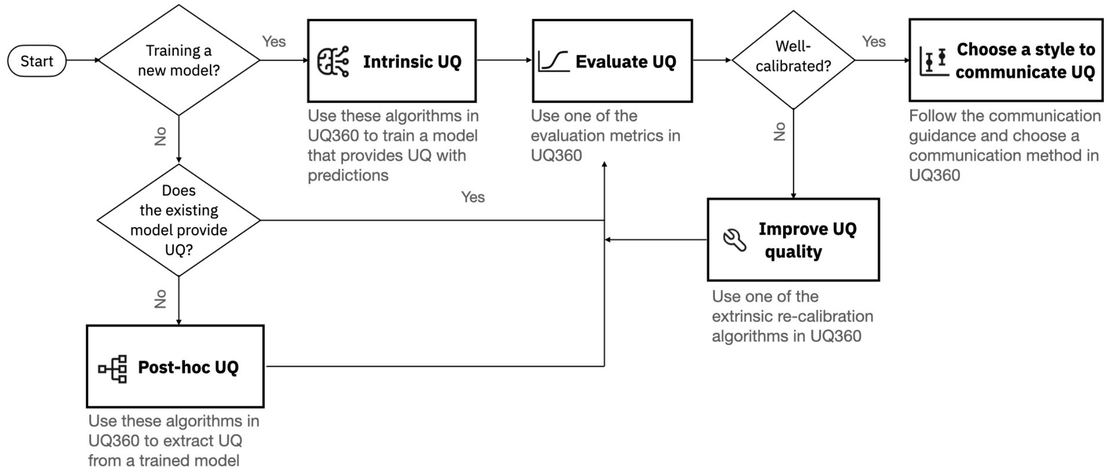
Agenda

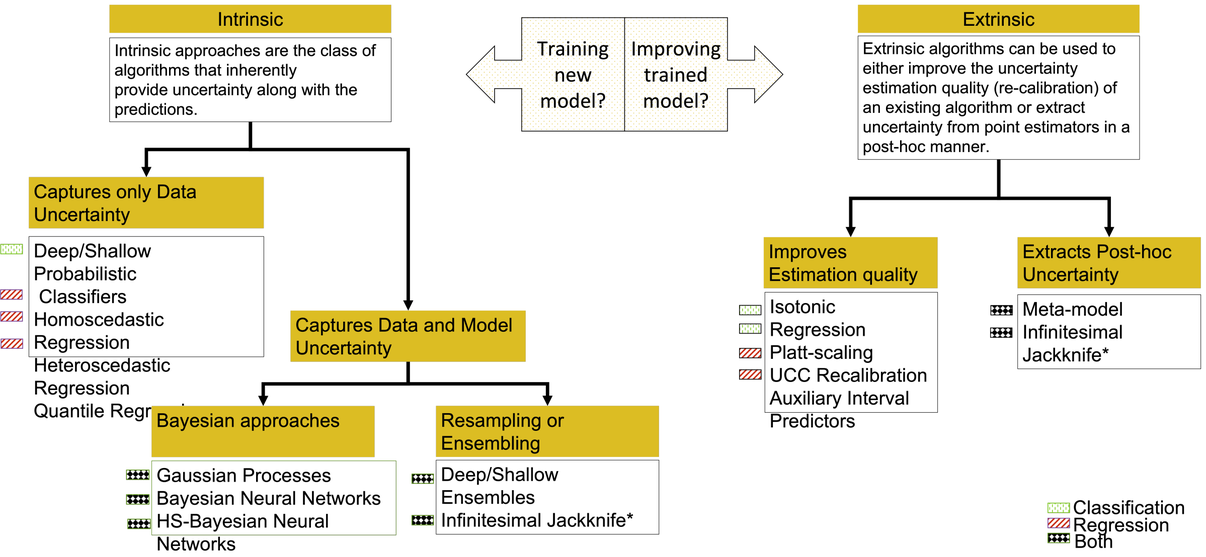
Introduction to UQ360
Interactive Web Experience
Package Installation
break
IJ
Meta-models + UCC
Connections to other Pillars of Trust

Please join the Slack community
http://aif360.mybluemix.net/community

Channel: #uq360-users

UQ360





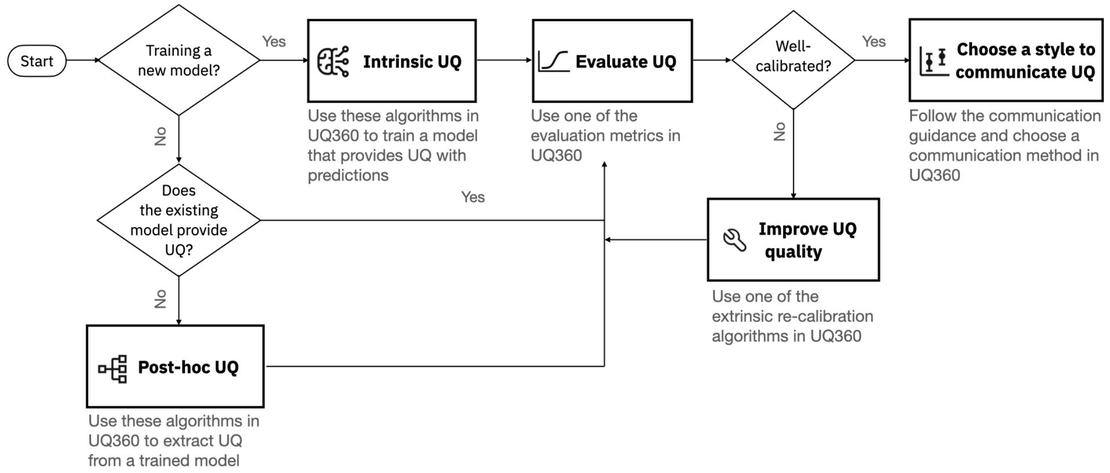
Agenda

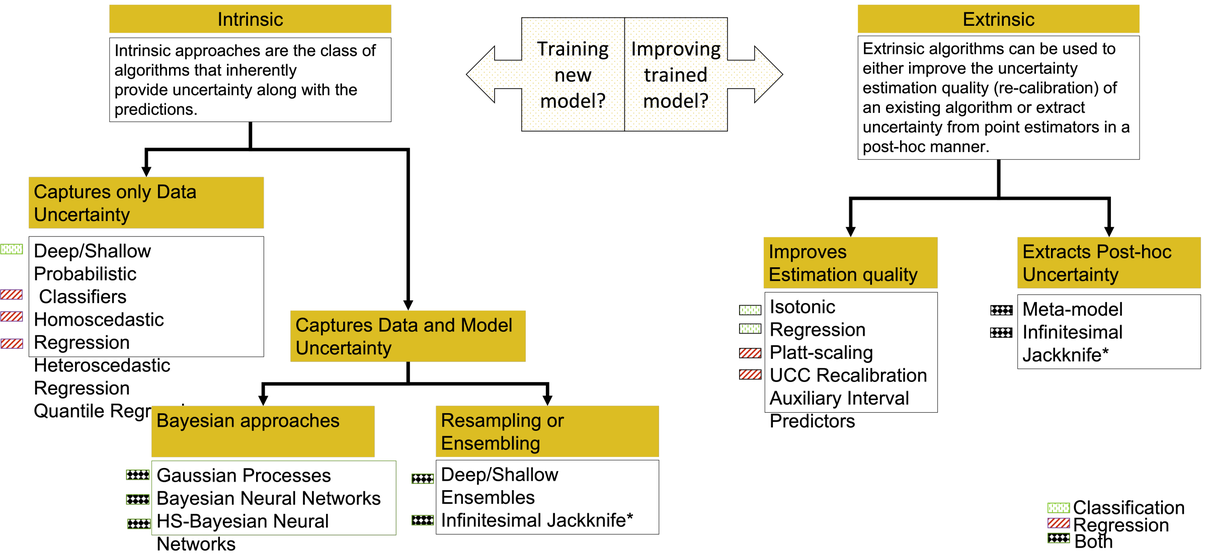
Introduction to UQ360
Interactive Web Experience
Package Installation
break
IJ
Meta-models + UCC
Connections to other Pillars of Trust

Please join the Slack community
http://aif360.mybluemix.net/community

Channel: #uq360-users

UQ360





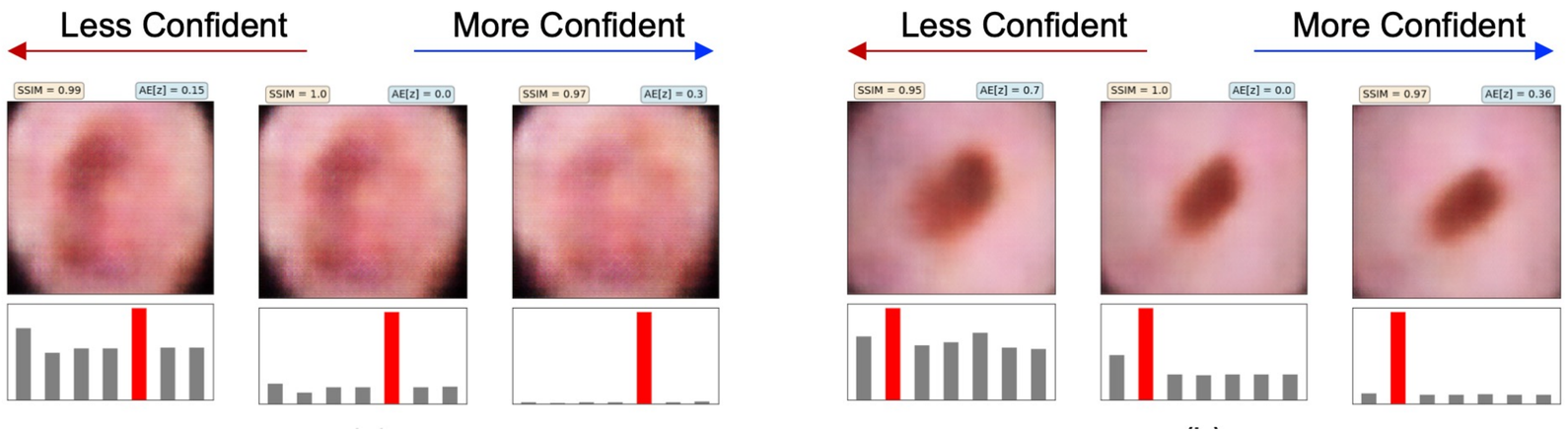
Agenda

Introduction to UQ360
Interactive Web Experience
Package Installation
break
IJ
Meta-models + UCC
Connections to other Pillars of Trust

Please join the Slack community
http://aif360.mybluemix.net/community

Channel: #uq360-users

INTERSECTION WITH OTHER PILLARS OF TRUST- TRANSPARENCY AND EXPLAINABILITY

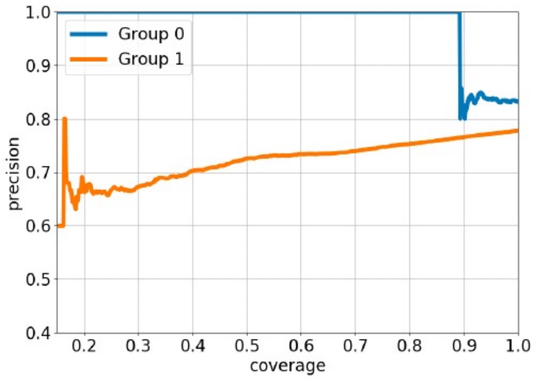


Improving Reliability of Clinical Models using Prediction Calibration, UNSURE (Uncertainty for Safe Utilization of Machine Learning in Medical Imaging), MICCAI 2020.

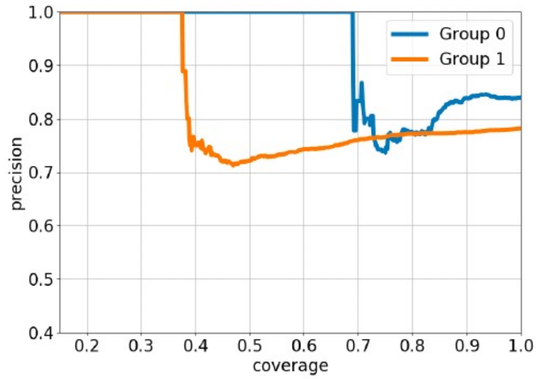
Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty, Submitted AIES 2021.

INTERSECTION WITH OTHER PILLARS OF TRUST-

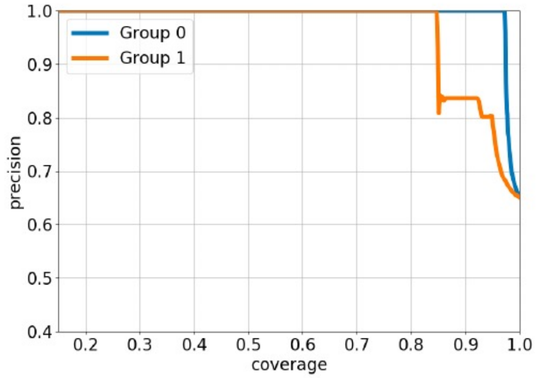
FAIRNESS IN SELECTIVE CLASSIFICATION



(a) Baseline



(b) DRO



(c) Sufficiency-regularized

Fair Selective Classification via Sufficiency, ICML 2021.

Fast uncertainty quantification via the Infinitesimal Jackknife

Soumya Ghosh

Supervised Learning

Given a dataset $\mathcal{D} = \{x_1, y_1, x_2, y_2, \ldots, x_N, y_N\}$ estimate $y = f_\theta(x)$

Empirical (regularized) risk minimization

$$\hat{\theta}(\mathcal{D}) := \hat{\theta} := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} l_n(\theta) + \lambda R(\theta)$$

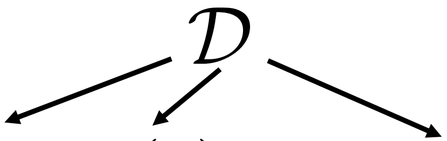
regularizer

$$l_n(\theta) := l(y_n, f_\theta(x_n))$$

loss function

Perturb and Refit

Classical Approach: Perturb dataset and refit to perturbed variants



$$\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(S)}$$

Perturbations of D

$$\mathcal{D}^{(1)} = \{x_2, y_2, x_3, y_3, \ldots, x_N, y_N\}$$

$$\mathcal{D}^{(2)} = \{x_1, y_1, x_3, y_3, \ldots, x_N, y_N\}$$

Jackknife / LOOCV

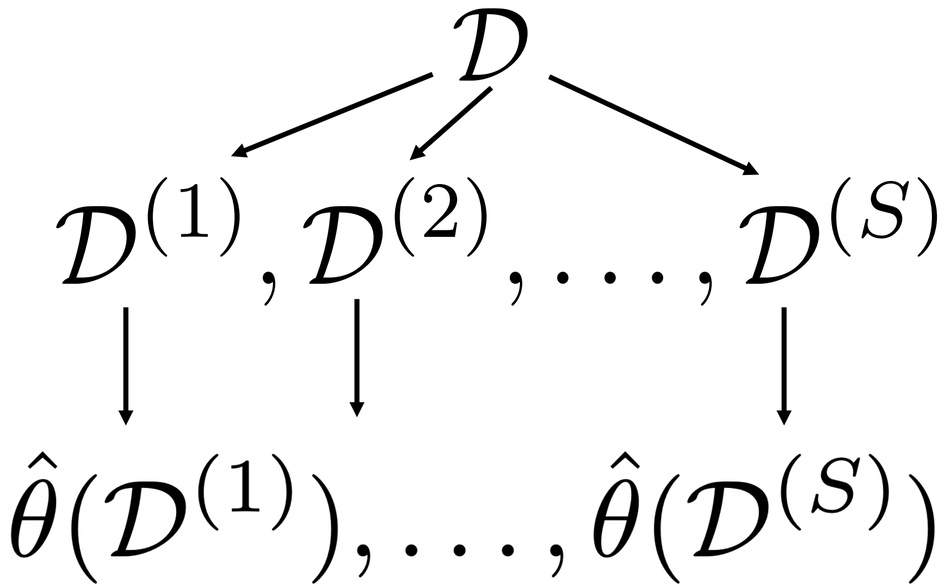
$$\vdots$$

$$\mathcal{D}^{(N)} = \{x_1, y_1, x_2, y_2, \ldots, x_{N-1}, y_{N-1}\}$$

Refit to each dataset variant $\hat{\theta}(\mathcal{D}^{(1)}), \ldots, \hat{\theta}(\mathcal{D}^{(N)})$

Perturb and Refit

Classical Approach: Perturb dataset and refit to perturbed variants



Perturb

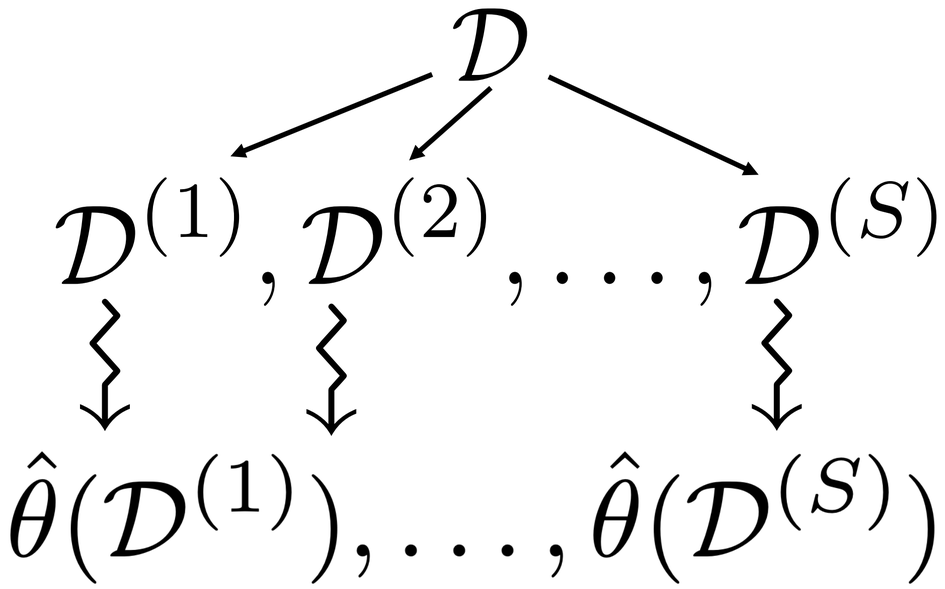
Refit

Widely used in practice: jackknife, bootstrap, and cross-validation

Procedurally trivial but computationally intensive (requires S refits)

Perturb and Refit

Classical Approach: Perturb dataset and refit to perturbed variants



Perturb

Approximate

Widely used in practice: jackknife, bootstrap, and cross-validation

Procedurally trivial but computationally intensive (requires S refits)

Weighted risk minimization

Original learning problem

$$\hat{\theta} := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} l_n(\theta) + \lambda R(\theta)$$

Weighted learning problem

$$\hat{\theta}(\mathbf{w}) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n l_n(\theta) + \lambda R(\theta)$$

Weights for data perturbations

$$\hat{\theta}(\mathbf{w}) := \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} w_n l_n(\theta) + \lambda R(\theta)$$

Original problem

$$\mathbf{w_1} := [1, 1, 1, 1, 1, 1, \ldots, 1] \in \mathbb{R}^N \qquad \hat{\theta}(\mathcal{D}) = \hat{\theta}(\mathbf{w}_1)$$

LOO cross-validation
Jackknife

$$\mathbf{w}_{/n} := [1, 1, 0, 1, 1, 1, \ldots, 1] \qquad \hat{\theta}(\mathcal{D}^{(n)}) = \hat{\theta}(\mathbf{w}_{/n})$$

k-Fold cross-validation

$$\mathbf{w}_{\text{k-cv}} := [1, 0, 0, 0, 0, 1, \ldots, 1]$$

Bootstrap

$$\mathbf{w}_b \sim \text{Multinomial}(N, N^{-1})$$

Fast approximations through the Infinitesimal Jackknife

Key Idea: Approximate $\hat{\theta}(\mathbf{w})$ using a Taylor series approximation about $\mathbf{w_1}$

$$\hat{\theta}(\mathbf{w_1} + \Delta\mathbf{w}) = \hat{\theta}(\mathbf{w_1}) + \frac{d\hat{\theta}(\mathbf{w})}{d\mathbf{w}}\Delta\mathbf{w} + \text{higher order terms}$$

$\hat{\theta}_{\mathrm{IJ}}(\mathbf{w})$

Accurate for large N and small Δw

$$\hat{\theta}_{\mathrm{IJ}}(\mathbf{w_1} + \Delta\mathbf{w}) = \hat{\theta}(\mathbf{w_1}) + \sum_{n=1}^{N} \mathrm{H}^{-1}(\theta)\nabla_\theta l_n(\theta)\Delta\mathbf{w}_n$$

Hessian

Gradient

Approximate don’t refit

$$\hat{\theta}_{\mathrm{IJ}}(\mathbf{w_1} + \Delta\mathbf{w}) = \hat{\theta}(\mathbf{w_1}) + \sum_{n=1}^{N} \mathrm{H}^{-1}(\theta) \nabla_\theta l_n(\theta) \Delta\mathbf{w}_n \qquad (1)$$

O(p^3)

1) Fit model once to the entire dataset $\mathcal{D}$
2) Compute and invert the Hessian

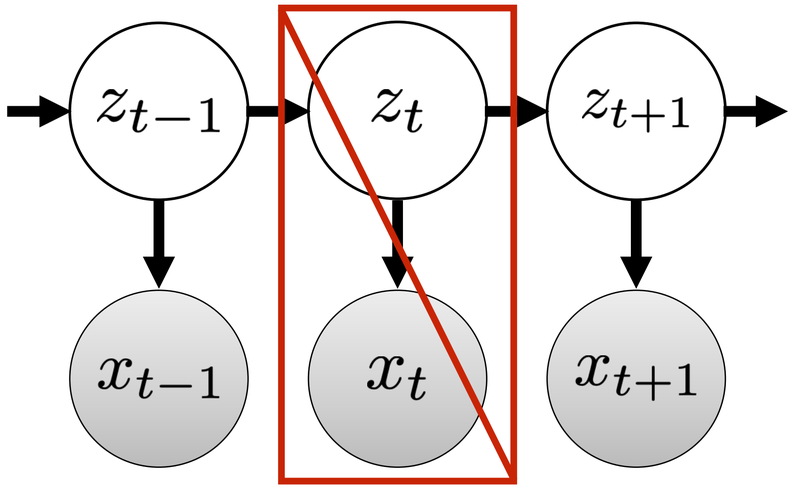
Needs to be done only once

3) For a perturbed dataset $\mathcal{D}^{(s)}$, figure out the weights w
4) Use equation 1 to approximate $\hat{\theta}(\mathcal{D}^{(s)})$

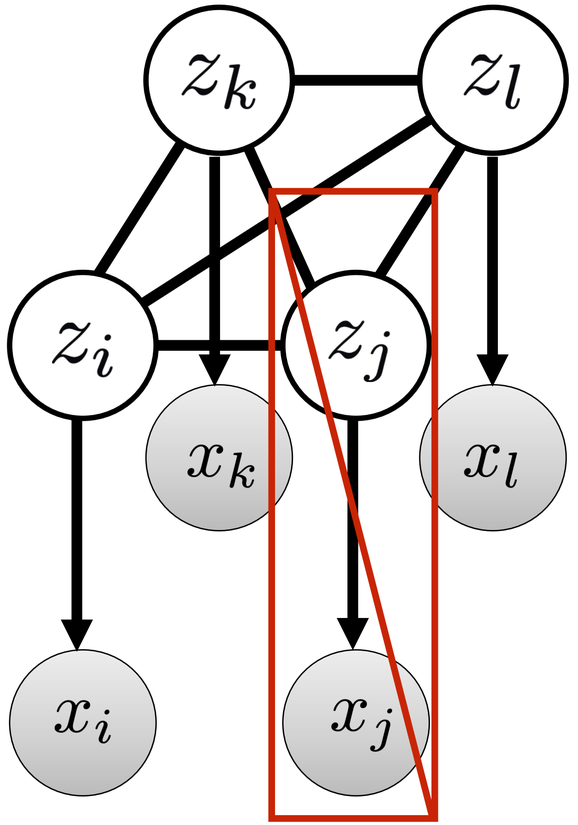
Cheap.
Once for each perturbed dataset

5) Repeat steps 3 and 4 for different perturbations of $\mathcal{D}$

Extensions to structured models



Hidden Markov Models



Markov Random fields

Extensions to structured models through weighted belief propagation (dynamic programming)

On to the demo

- *A swiss army infinitesimal jackknife*. *AISTATS 2019.*
  Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick.

- *Approximate cross-validation for structured models.* *NeurIPS 2020.*
  Soumya Ghosh, William T. Stephenson, Tin D. Nguyen, Sameer K. Deshpande, and Tamara Broderick.

- Demo: UQ360/examples/infinitesimal_jackknife/demo_infinitesimal_jackknife.ipynb

UQ360: Examples (Part 2)

Outline

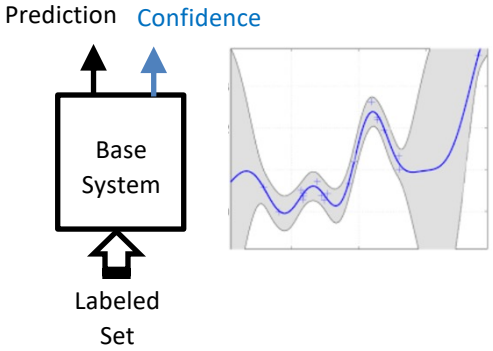
- Method: Meta Modeling for UQ
- Metric: Uncertainty Characteristics Curve (UCC)

Meta-Modeling for Uncertainty Quantification

Introduction

Intrinsic Approach

Choose appropriate model to generate uncertainty estimates along with actual predictions – from within the model

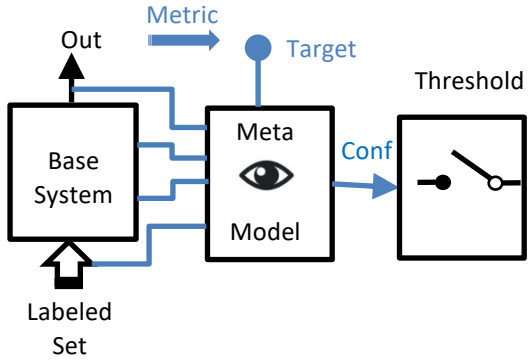


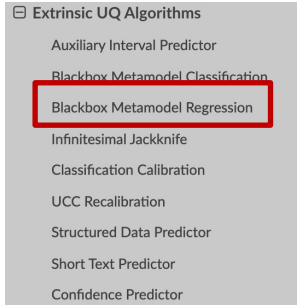
Examples: (1) Bayesian models, (2) GP models, (3) Runtime drop-out techniques, (4) Ensembles

Extrinsic Approach

Train an “observer” meta-model to predict outcome metrics

Metrics relate to success/failure of the base: Accuracy, NDCG (ranking), regression error, etc.





Why Meta-Modeling

- It is practical
  - Works with any base model, any task
  - Easy to include out-of-domain data
- It is effective
  - Can model both the epistemic and the aleatoric uncertainty
  - Shown to (out)perform state-of-the art

White-Box Meta-Model for image recognition:

- T. Chen, J. Navrátil, V. Iyengar, K. Shanmugam, "Confidence Scoring Using Whitebox Meta-models with Linear Classifier Probes," AISTATS-2019

Meta-Modeling for structured data tasks under drift:

- B. Elder, M. Arnold, A. Murthi, J. Navratil, "Learning Prediction Intervals for Model Performance," AAAI, 2021

White-Box and Joint Meta-Modeling for sequential regression tasks:

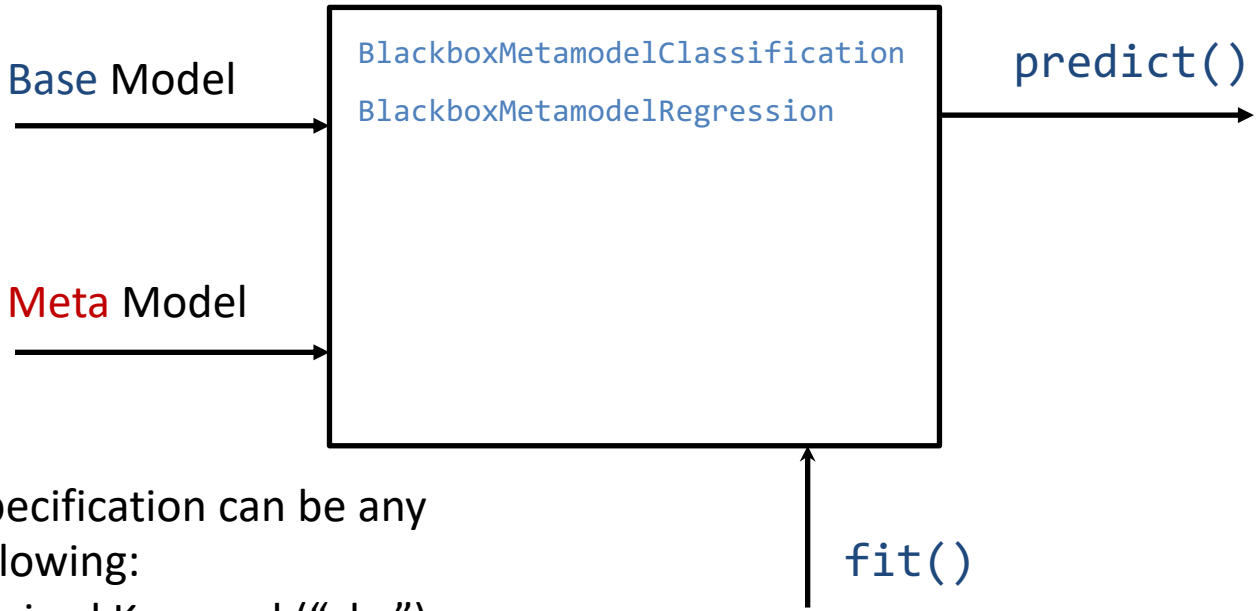
- J. Navratil, M. Arnold, B. Elder, "Uncertainty Prediction for Deep Sequential Regression Using Meta Models," arXiv, 2020

MM in UQ360

- Classes derive from `PostHocUQ()`
  - `MetamodelClassification(PostHocUQ)`
  - `MetamodelRegression(PostHocUQ)`

- Demos
  - `UQ360/examples/blackbox_metamodel/demo_blackbox_metamodel_classification.ipynb`
  - `UQ360/examples/blackbox_metamodel/demo_blackbox_metamodel_regression.ipynb`

Class Interface



Model Specification can be any of the following:

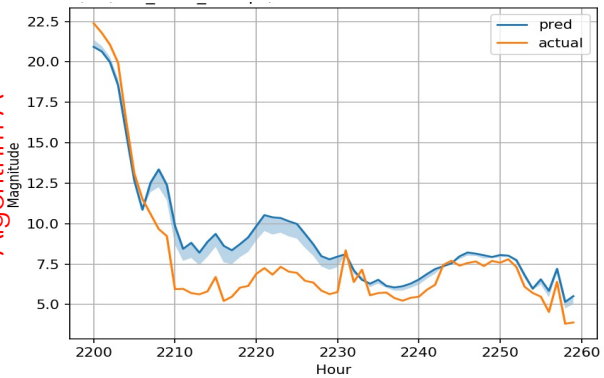
- Recognized Keyword (“gbr”)
- Class Declaration
  (“sklearn.ensemble.GradientBoostingRegressor”)
- Existing Instance

Uncertainty Characteristics Curve: An Assessment Tool for Prediction Intervals

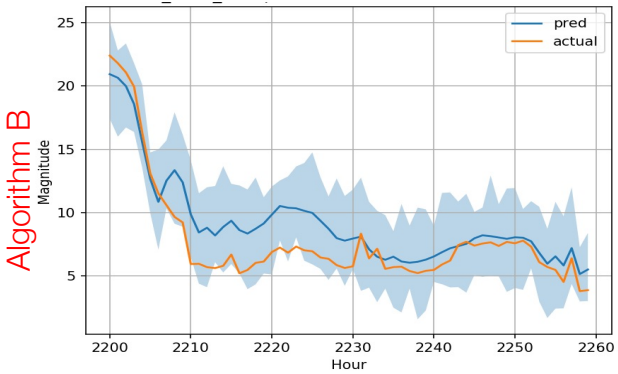
Motivating Example

Prediction and Actual are same in all plots but uncertainty bands are produced by different algorithms

Algorithm A

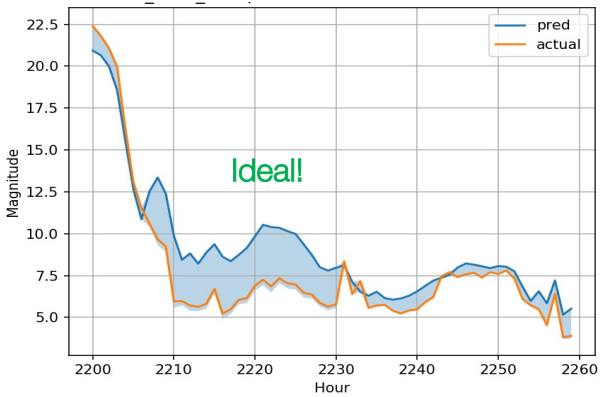


Algorithm B

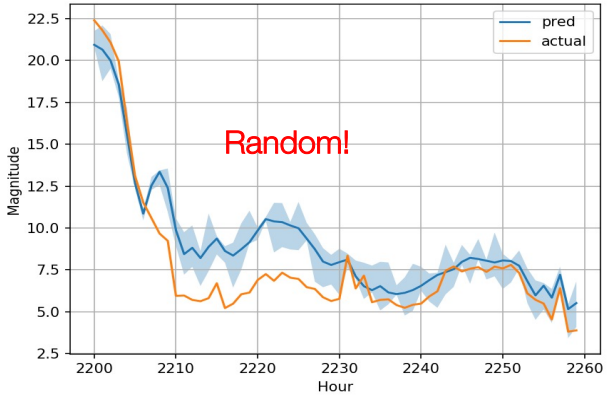


Q: Which is better?

Scale up



Squeeze

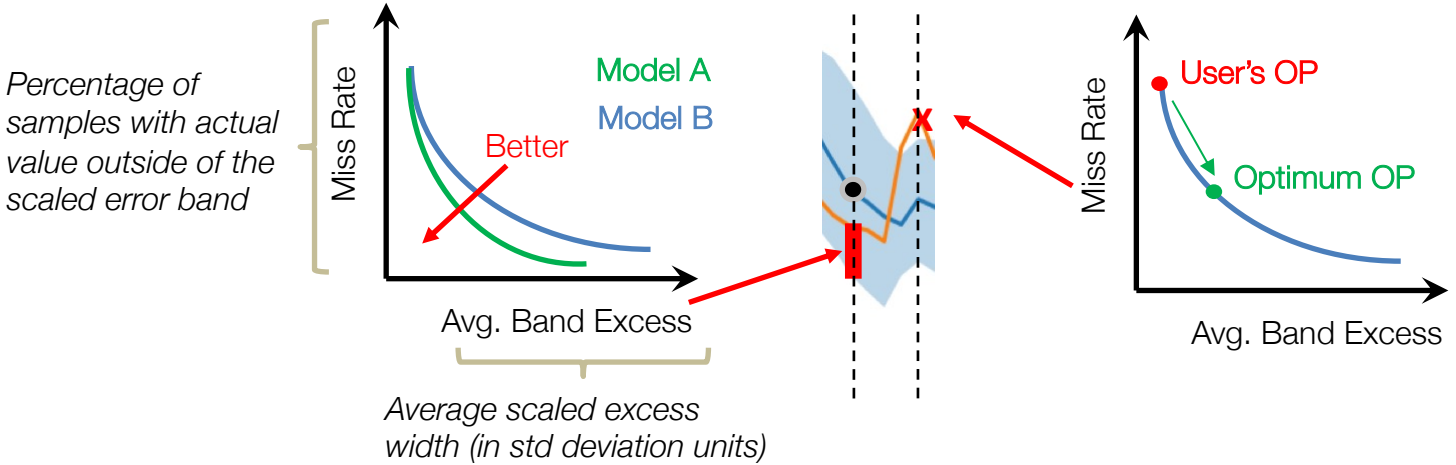


Uncertainty Characteristics Curve (UCC)

What is it: A diagnostic tool providing several metrics to assess error bar (prediction interval) quality
Input: set of data samples {actual, predicted, error interval}
Output: Trade-off curve between the prediction interval bandwidth and Miss Rate

Construction: Apply linear scaling of bands and observe how Miss Rate changes

Note on implementation:
User can designate any two of {Miss Rate, Bandwidth, Excess, Deficit} as axes

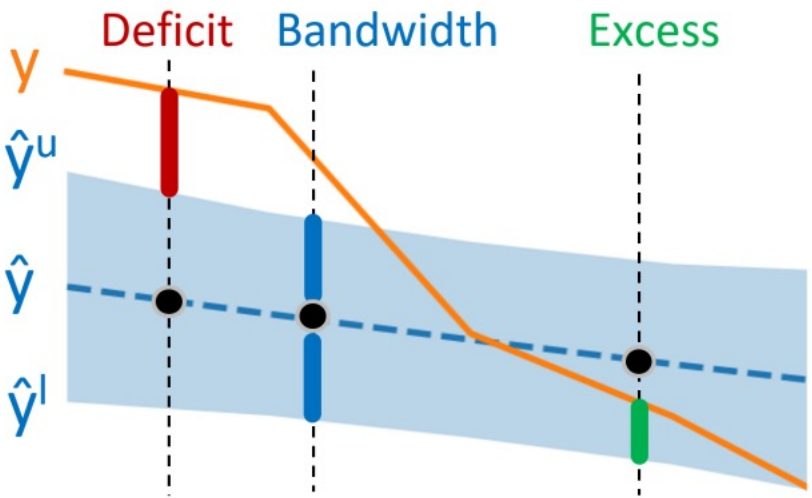


Cost-agnostic metric:
The area under the curve reflects an operating point agnostic measure of the uncertainty band quality. Thus, error models can be compared!

Cost-based metric
If you can specify the relative costs of Miss Rate vs excessive bands, the optimum operating point (OP) can be determined!

UCC Axes Metrics

- Missrate, Bandwidth, Excess, Deficit



What does UCC analysis provide?

- OP-agnostic comparison of different algorithms producing uncertainty predictions
- Comparison of actual and optimal operating point (OP) of your uncertainty bars → returns “recipe” how to achieve optimum OP

UCC in UQ360

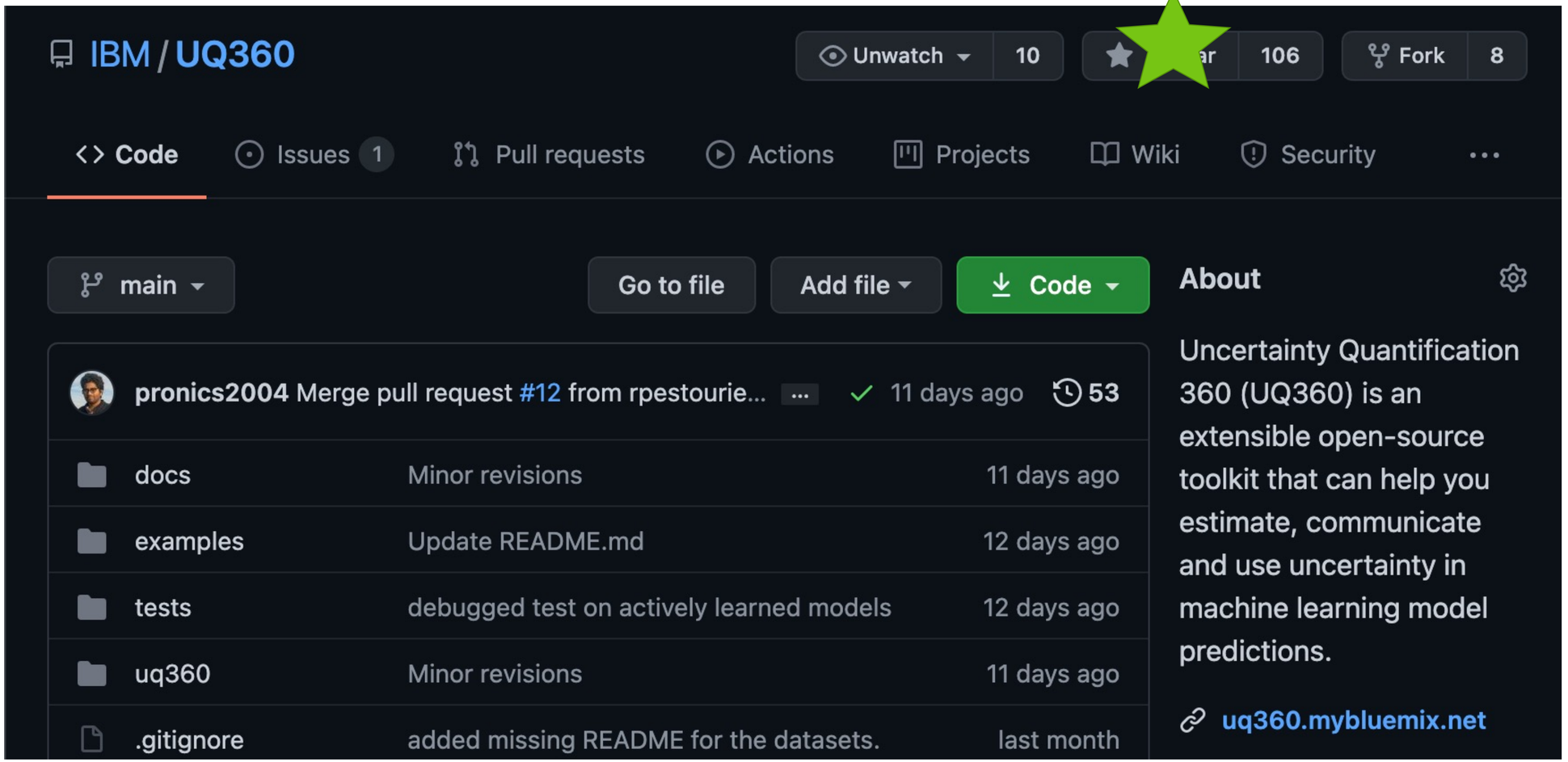
- Class: `UQ360/uq360/metrics/uncertainty_characteristics_curve/uncertainty_characteristics_curve.py`
- Used in: `UQ360/uq360/metrics/regression_metrics.py`
- Demo: `UQ360/examples/ucc_metric/demo_ucc_class.ipynb`

THANK YOU FOR LISTENING!

WE WELCOME CONTRIBUTIONS AND FEEDBACK

https://github.com/IBM/UQ360

Please give us a star and spread the word ☺



BACKUP

API AND IMPLEMENTATION

SKLEARN COMPATIBLE ALGORITHMS

```
from uq360.algorithms.quantile_regression import QuantileRegression
```

Train Quantile Regression

```
config = {
    "alpha":0.95,
    "n_estimators":20,
    "max_depth":3,
    "learning_rate":0.1,
    "min_samples_leaf":20,
    "min_samples_split":20
}

uq_model = QuantileRegression(model_type='gbr', config=config)
```

```
uq_model = uq_model.fit(X_train, y_train.squeeze())
```

```
y_mean, y_lower, y_upper = uq_model.predict(X_test)
y_mean, y_lower, y_upper = scaler_y.inverse_transform(y_mean), scaler_y.inverse_transform(y_lower)
, scaler_y.inverse_transform(y_upper)
```

uq360 models with sklearn's `GridsearchCV`

```
sklearn_picp = make_sklearn_compatible_scorer(task_type="regression", metric="picp", greater_is_be
tter=True)
```

```
clf = GridSearchCV(QuantileRegression(config=base_config), configs, scoring=sklearn_picp)
```

```
clf.fit(X_train, y_train)
```


SKLEARN COMPATIBLE METRICS

```
sklearn_aurrrc = make_sklearn_compatible_scorer(task_type="classification", metric="aurrrc", great
er_is_better=False)
sklearn_ece = make_sklearn_compatible_scorer(task_type="classification", metric="ece", greater_is_
better=False)
```

```
import lale
from lale.lib.lale import Hyperopt
lale.wrap_imported_operators()
```

```
from sklearn import datasets
X, y = datasets.load_breast_cancer(return_X_y=True)
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

```
from lale.lib.sklearn import AdaBoostClassifier as Model
```

```
clf_ece = Hyperopt(estimator=Model, cv=3, max_evals=20, scoring=sklearn_ece, verbose=True)
clf_aurrrc = Hyperopt(estimator=Model, cv=3, max_evals=20, scoring=sklearn_aurrrc, verbose=True)
```

```
trained_with_ece = clf_ece.fit(X_train, y_train)
```

```
100%|██████████| 20/20 [00:29<00:00,  1.46s/trial, best loss: 0.259819517887197]
```

```
trained_with_aurrrc = clf_aurrrc.fit(X_train, y_train)
```

```
100%|██████████| 20/20 [00:30<00:00,  1.51s/trial, best loss: 0.004077380952380953]
```


DATA UNCERTAINTY: HETEROSCEDASTIC NEURAL NETWORKS

Captures only Data Uncertainty

Assumption: The target behaves according a parametric probability distribution and its parameters are assumed to depend on the inputs.

Models can be optimized to predict distributional parameters.

For instance, the **Heteroscedastic Neural Networks**,

- Target is assumed to be distributed normally

$$\mathrm{L_{HNN}}(\mathrm{y}_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = \frac{(\mathrm{y}_i - \boldsymbol{\mu}_i)^2}{2\boldsymbol{\sigma}_i^2} + \frac{1}{2}\log(\boldsymbol{\sigma}_i^2)$$

MODEL UNCERTAINTY: ENSEMBLES

Captures Data and Model Uncertainty

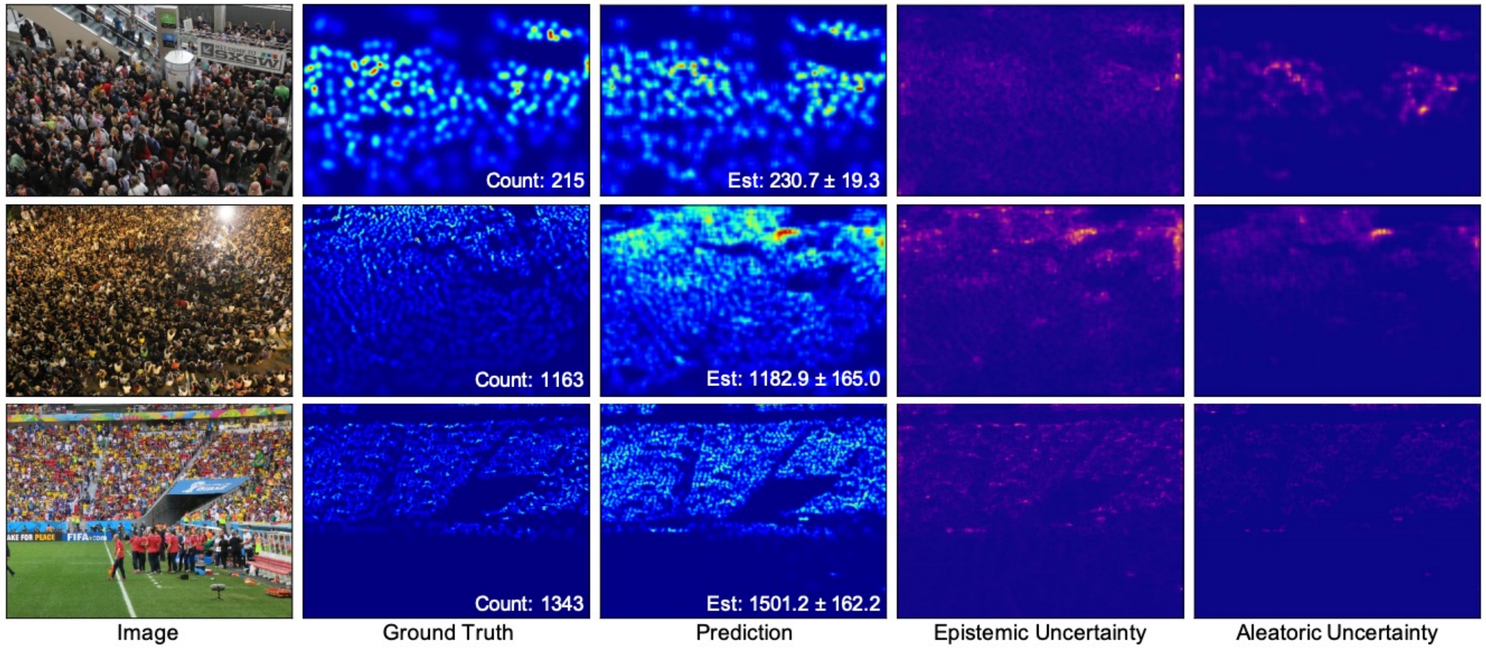
Not all models make the same mistake.

Train an *ensemble* of models and study statistical heuristics.

How do we construct the ensemble?

- Training data (k-fold cross-validation, bootstrap, random training subsets)
- Model design (multiple training runs, initializations)
- Aggregation strategy (model averaging, weighted averaging, boosting)

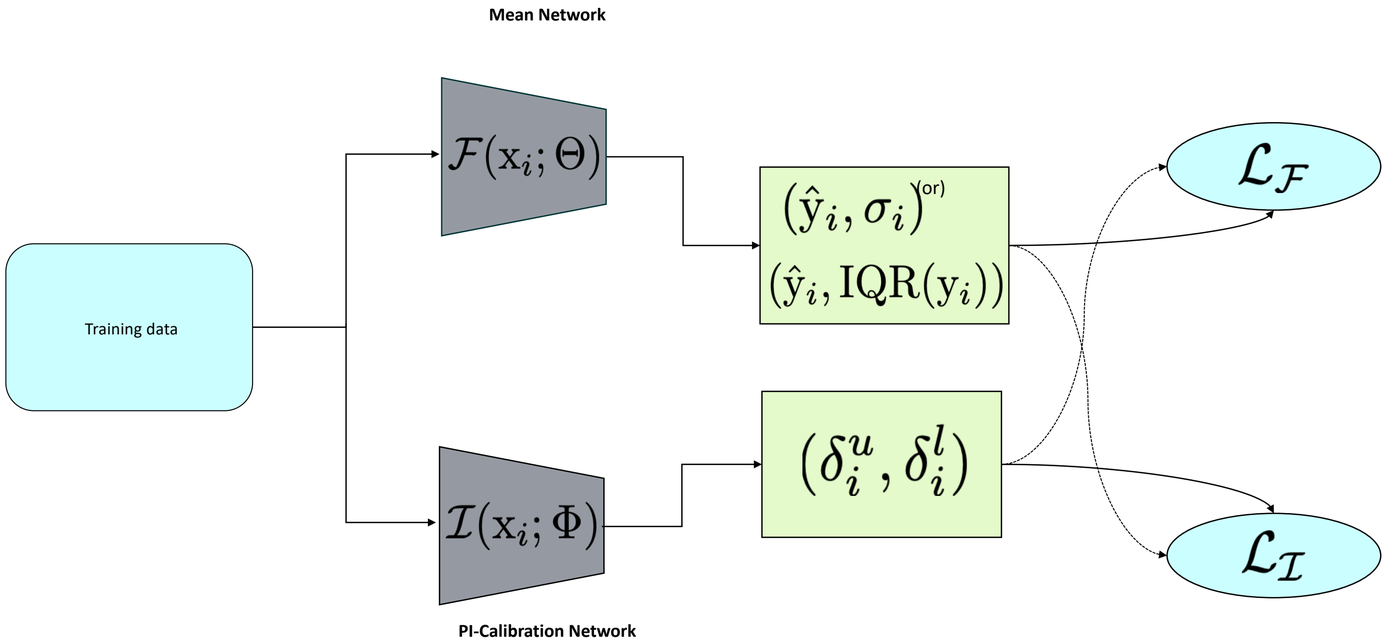
Model uncertainty → Variance estimate across the ensemble



Crowd Counting with Decomposed Uncertainty, AAAI 2020.

META-MODELS FOR IMPROVING CALIBRATION

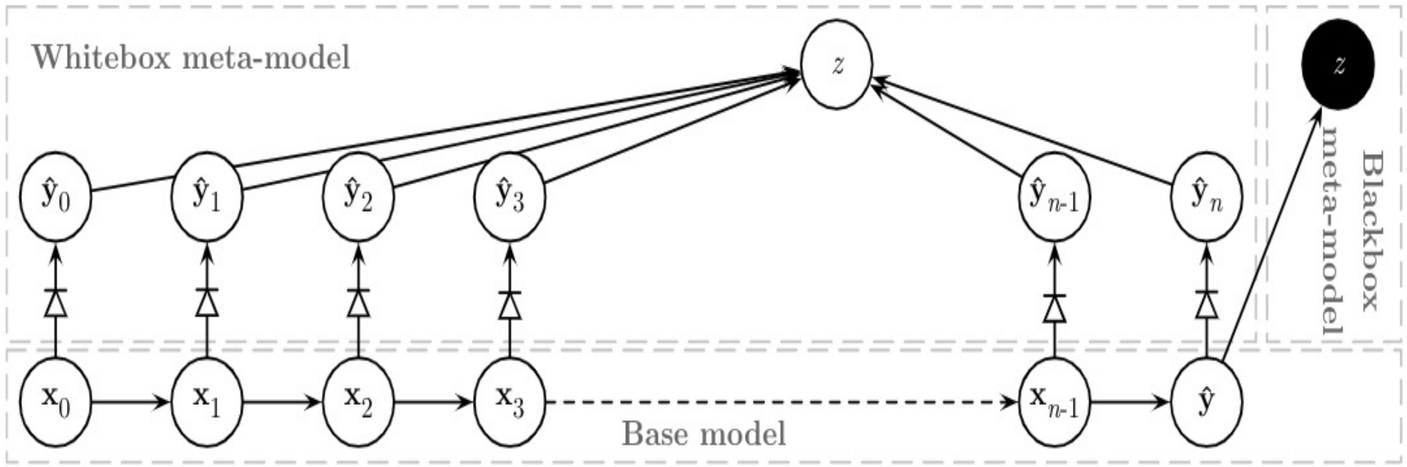
Improves Estimation quality



Building Calibrated Deep Models via Uncertainty Matching with Auxiliary Interval Predictors, AAAI 2020.

UNCERTAINTY WRAPPERS FOR POINT ESTIMATORS

Extracts Post-hoc Uncertainty



Confidence scoring using whitebox meta-models with linear classifier probes, AISTATS 2019.

