Robustness quantification: a new method for assessing the reliability of the predictions of a classifier

Adrián.Detavernier@ugent.be

Jasper.De Bock@ugent.be

Let's say that you have a beloved bird called Flip who is feeling sick.

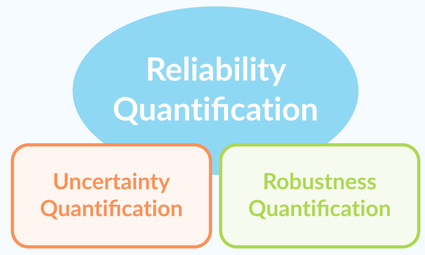
You decide to take him to the vet, who started using an ML **classifier**.

This classifier (accuracy 95%) is given a set of **features**: weight, blood results, etc.

Based on the features it now predicts the following **class**: severe illness.

How reliable is this particular prediction? Would you trust it and start up a risky treatment?

- **Severe illness**
- Mild illness
- Healthy



Generative classifiers

Generative classifiers are deterministic functions $h_P : \mathcal{F} \to C$ that map each feature vector $\mathbf{f} = (f_1, \ldots, f_N) \in \mathcal{F} := \bigtimes_{i=1}^{N} \mathcal{F}_i$ to a class $h(\mathbf{f}) \in C$ based on a joint probability distribution P on $C \times \mathcal{F}$. Since for us $\mathcal{F}$ and C are discrete, P is uniquely defined by its mass function p. The predicted class $\hat{c}$ is chosen to be (one of) the class(es) with maximum conditional probability:

$$h_P(\mathbf{f}) \in \arg\max_{c \in C} p(c|\mathbf{f}) = \arg\max_{c \in C} p(c, \mathbf{f}).$$

Naive Bayes Classifier (NBC)

An **NBC** assumes the features to be conditionally independent given the class. This implies that the joint probability mass function can be factorized as

$$p(c, \mathbf{f}) = p(c) \prod_{i=1}^{N} p(f_i|c).$$

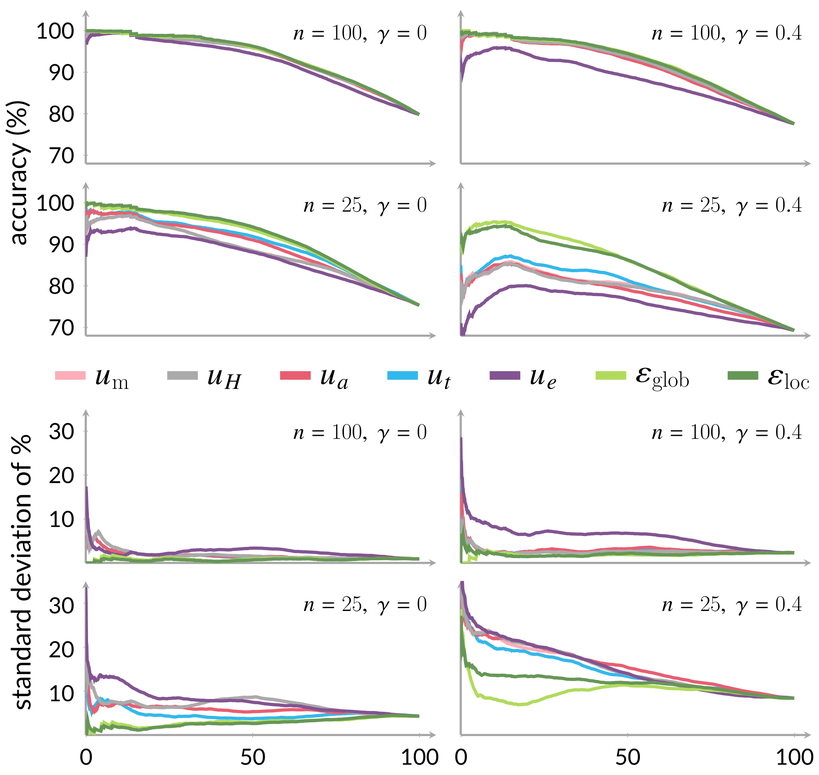
Uncertainty Quantification (UQ)

UQ aims to quantify the uncertainty associated with individual predictions, the idea being that uncertainty negatively influences **reliability**.

The uncertainty metrics we consider are the following:

- $u_{\mathrm{m}}(\mathbf{f})$ *the probability of being wrong according to* $P_{D_{\mathrm{train}}}$
- $u_H(\mathbf{f})$ *the Shannon entropy of* $P_{D_{\mathrm{train}}}(\cdot|\mathbf{f})$
- $u_a(\mathbf{f})$ *the average Shannon entropy of all* $P_{i,D_{\mathrm{train}}}(\cdot|\mathbf{f})$ *learned by an ensemble*
- $u_t(\mathbf{f})$ *the Shannon entropy of* $P_{\mathrm{av},D_{\mathrm{train}}}(\cdot|\mathbf{f})$*, the average of all* $P_{i,D_{\mathrm{train}}}(\cdot|\mathbf{f})$ *learned by an ensemble*
- $u_e(\mathbf{f})$ $= u_a(\mathbf{f}) - u_t(\mathbf{f})$

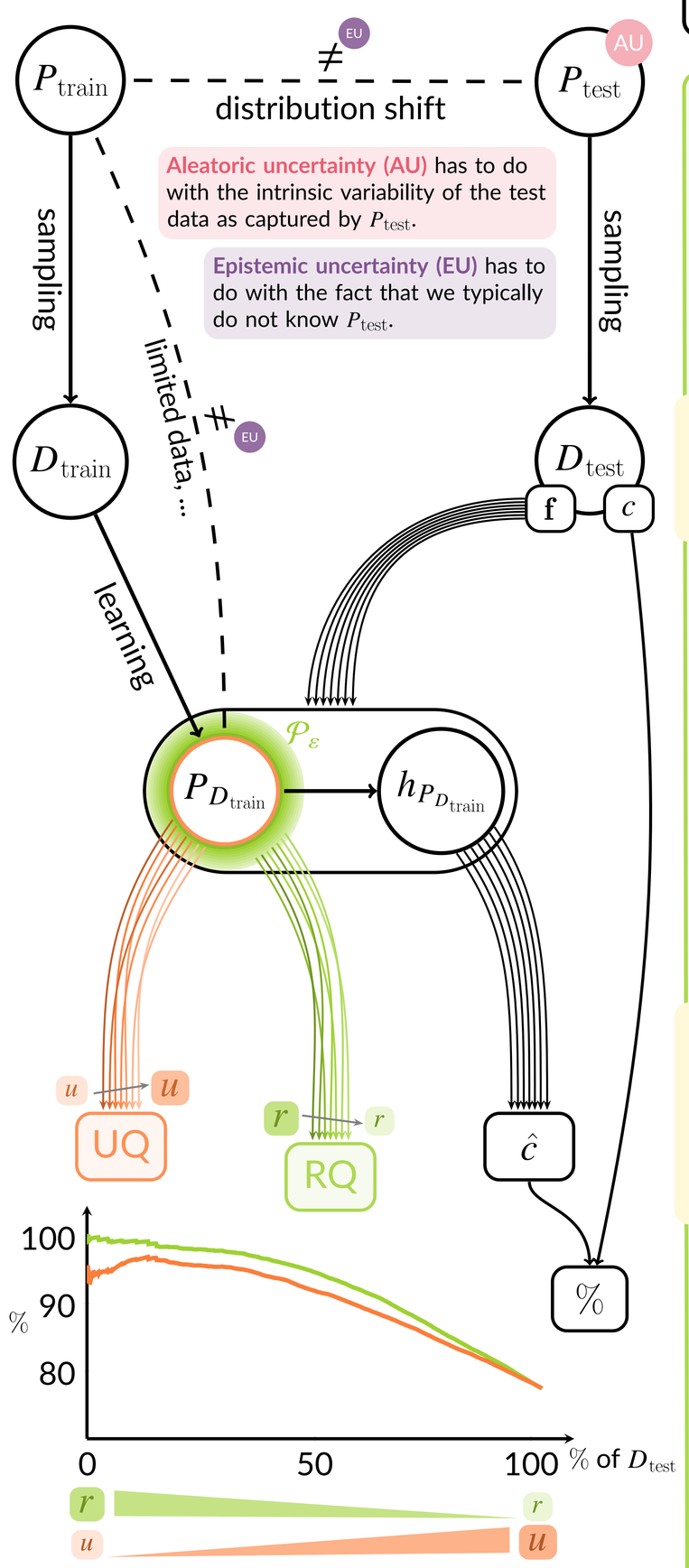
Experiments



In our experiments we studied the effect of **distribution shift** and **limited data**. To control this, we generated our own data sets. The test set D_{test} is sampled from the test distribution $P_{\mathrm{test}} = 0.7P_{\mathrm{fix}} + 0.3P_{\mathrm{rand}}$, with P_{fix} an NBC and P_{rand} a randomly generated distribution. The training set D_{train} is sampled from the training distribution $P_{\mathrm{train}} = (1-\gamma)P_{\mathrm{test}} + \gamma P_{\mathrm{shift}}$, with P_{shift} also random. The size n of D_{train} is 100 or 25 and the amount of distribution shift γ is 0 or 0.4. For each combination of n and γ we generate 10 different training distributions P_{train} (with 10 different P_{shift}) and use each of these to sample 10 different training sets D_{train} of size n. The graphs depict the average accuracy and its standard deviation for the resulting 100 training sets.

A general overview

Our main goal is to provide *instance-based* **reliability** information for the predictions of a **generative classifier**. Our approach consists in accompanying each prediction with its own **robustness metrics**. We focus on the **Naive Bayes Classifier** and compare our metrics with **uncertainty metrics** in a series of **experiments**.



Applying **ε-contamination** to a mass function p yields the set of all mass functions of the type $(1-\varepsilon)p + \varepsilon p^*$, where p^* ranges over all possible mass functions whose domain is equal to that of p.

Learning local models

Since the joint of the **NBC** $P_{D_{\mathrm{train}}}$ factorizes, only the local mass functions need to be learned from D_{train}:

$$p(c) = \frac{n(c) + \alpha}{n + \alpha|C|} \quad \text{and} \quad p(f_i|c) = \frac{n(c, f_i) + \alpha}{n(c) + \alpha|\mathcal{F}_i|},$$

where n, $n(c)$ and $n(c, f_i)$ are the number of instances in total, of class c and of class c with feature value f_i in D_{train}, and where α is a smoothing parameter.

Robustness Quantification (RQ)

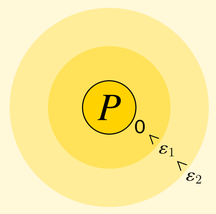
RQ aims to quantify how much **EU** the classifier *could* cope with before changing its prediction (how different P_{test} *can be* from $P_{D_{\mathrm{train}}}$), and this regardless of how much **EU** there actually is (how different P_{test} *is* from $P_{D_{\mathrm{train}}}$). To quantify this, we artificially **perturb** $P_{D_{\mathrm{train}}}$ until the prediction changes.

Consider a distribution P on $C \times \mathcal{F}$. Let $\mathcal{P}$ be a compact set of distributions on $C \times \mathcal{F}$ with $P \in \mathcal{P}$. Then we call $\mathcal{P}$ a **perturbation** of P.

We say that a prediction is **robust** w.r.t. a perturbation $\mathcal{P}$, if all distributions P in $\mathcal{P}$ yield the same prediction as $P_{D_{\mathrm{train}}}$. A prediction is then deemed **reliable** if it is **robust**, meaning that it would remain true even in the face of severe **EU**.

Usually, a **perturbation** $\mathcal{P}$ of $P_{D_{\mathrm{train}}}$ will be a 'neighborhood' around $P_{D_{\mathrm{train}}}$, consisting of distributions of a particular type whose distance from $P_{D_{\mathrm{train}}}$ is in some sense bounded. We increase the size of these perturbations using **parametrized perturbations**.

Consider a distribution P and, for all $\varepsilon \in [0, 1]$, a perturbation $\mathcal{P}_\varepsilon$ of P. Then the family $\mathcal{P}_\bullet := (\mathcal{P}_\varepsilon)_{\varepsilon \in [0,1]}$ is called a **parametrized perturbation** of P if $\mathcal{P}_0 = \{P\}$ and $\mathcal{P}_{\varepsilon_1} \subset \mathcal{P}_{\varepsilon_2}$ if $\varepsilon_1 < \varepsilon_2$.



For each **parametrized perturbation** of $P_{D_{\mathrm{train}}}$ the corresponding **robustness metric** is the minimal perturbation size ε for which the prediction is no longer **robust**.

If $\mathcal{P}_\varepsilon$ is the set of distributions we get by applying **ε-contamination** to the joint mass function of $P_{D_{\mathrm{train}}}$, we obtain the **global robustness metric**

$$\varepsilon_{\mathrm{glob}}(\mathbf{f}) = \frac{p(\hat{c}, \mathbf{f}) - \max_{c \in C \setminus \hat{c}} p(c, \mathbf{f})}{1 + p(\hat{c}, \mathbf{f}) - \max_{c \in C \setminus \hat{c}} p(c, \mathbf{f})}.$$

This metric requires no structural assumptions and **can** therefore **be applied to any generative classifier**.

Alternatively, we also consider an approach that is **specific for NBCs**. Here, $\mathcal{P}_\varepsilon$ is obtained by applying **ε-contamination** to the local mass functions for c and for f_i given c separately. The resulting **local robustness metric** $\varepsilon_{\mathrm{loc}}(\mathbf{f})$ is the unique value of ε for which:

$$p(\hat{c}, \mathbf{f}) = \max_{c \in C \setminus \{\hat{c}\}} \left(p(c) + \frac{\varepsilon}{1-\varepsilon}\right) \prod_{i=1}^{N} \left(p(f_i|c) + \frac{\varepsilon}{1-\varepsilon}\right),$$

where the right-hand side is an increasing function in ε.