LAMARR
INSTITUTE FOR MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Supervised methods for quantification

Mirko Bunse, Alejandro Moreo, and Fabrizio Sebastiani

LQ @ ECML-PKDD 2024 – September 13th

Partner institutions:

technische universität dortmund
Fraunhofer IAIS
UNIVERSITÄT BONN
Fraunhofer IML

Institutionally funded by:

Bundesministerium für Bildung und Forschung

Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen

Problem statement

Given: a labeled training set $\mathrm{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n} \sim \mathbb{P}^n$ where

- $\mathcal{X}$ is the feature space (e.g., $\mathcal{X} = \mathbb{R}^d$)
- $\mathcal{Y} = \{1, 2, \dots C\}$ is the set of class labels

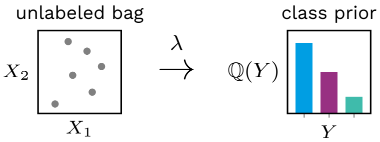
Problem statement

Given: a labeled training set $\mathrm{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n} \sim \mathbb{P}^n$ where

- $\mathcal{X}$ is the feature space (e.g., $\mathcal{X} = \mathbb{R}^d$)
- $\mathcal{Y} = \{1, 2, \ldots C\}$ is the set of class labels

Find: a quantifier $\lambda : \bigcup_{m=1}^{\infty} \mathcal{X}^m \rightarrow \Delta^{C-1}$ where

- $\bigcup_{m=1}^{\infty} \mathcal{X}^m$ is the space of unlabeled data bags of any size m
- $\Delta^{C-1} = \left\{ \mathbf{p} \in \mathbb{R}^C \,:\, \mathbf{p}_i \geq 0 \,\forall i,\ \sum_{i=1}^{C} \mathbf{p}_i = 1 \right\}$ is the space of class prevalences
- for any bag $\mathrm{B} \sim \mathbb{Q}^m$, we want to achieve that $\lambda(\mathrm{B}) = \mathbb{Q}(Y)$



Prior probability shift

We typically want to achieve $\lambda(\mathrm{B}) = \mathbb{Q}(Y)$ when otherwise unknown

Definitions:

- $\forall\, \mathbf{x} \in \mathrm{B} \;:\; \mathbf{x} \sim \mathbb{Q}(\mathbf{x})$ where $\mathbb{Q}(\mathbf{x}) \;=\; \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (law of total probability)
- $\forall\, (\mathbf{x}, y) \in \mathrm{D} \;:\; (\mathbf{x}, y) \;\sim\; \mathbb{P}(\mathbf{x}, y)$

[1] Kull and Flach, "Patterns of dataset shift", 2014.

Prior probability shift

We typically want to achieve $\lambda(\mathrm{B}) = \mathbb{Q}(Y)$ when otherwise unknown

Definitions:

- $\forall\, \mathbf{x} \in \mathrm{B} \,:\, \mathbf{x} \sim \mathbb{Q}(\mathbf{x})$ where $\mathbb{Q}(\mathbf{x}) = \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (law of total probability)
- $\forall\, (\mathbf{x}, y) \in \mathrm{D} \,:\, (\mathbf{x}, y) \sim \mathbb{P}(\mathbf{x}, y)$

Identically & independently distributed (IID) data:

- $\mathbb{Q}(X, Y) = \mathbb{P}(X, Y)$
- we could estimate $\mathbb{Q}(Y) = \mathbb{P}(Y)$

[1] Kull and Flach, “Patterns of dataset shift”, 2014.

Prior probability shift

We typically want to achieve $\lambda(\mathrm{B}) = \mathbb{Q}(Y)$ when otherwise unknown

Definitions:

- $\forall\, \mathbf{x} \in \mathrm{B} \;:\; \mathbf{x} \sim \mathbb{Q}(\mathbf{x})$ where $\mathbb{Q}(\mathbf{x}) = \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (law of total probability)
- $\forall\, (\mathbf{x}, y) \in \mathrm{D} \;:\; (\mathbf{x}, y) \sim \mathbb{P}(\mathbf{x}, y)$

Identically & independently distributed (IID) data:

- $\mathbb{Q}(X, Y) = \mathbb{P}(X, Y)$
- we could estimate $\mathbb{Q}(Y) = \mathbb{P}(Y)$

Prior probability shift (PPS):

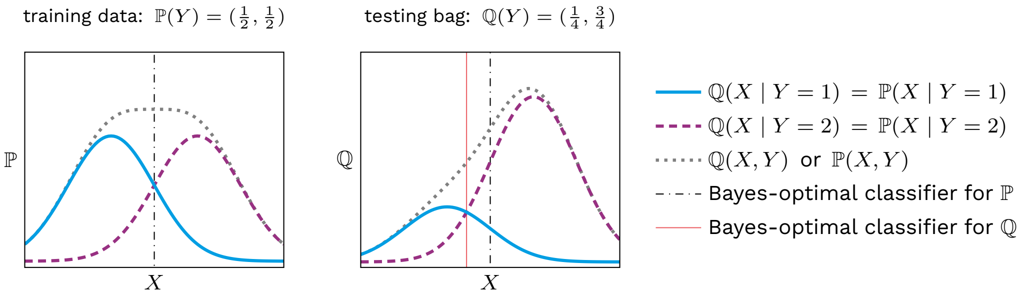
- $\mathbb{Q}(X \mid Y) = \mathbb{P}(X \mid Y)$
- $\mathbb{Q}(Y) \neq \mathbb{P}(Y)$

typical assumption in quantification

More types of data set shift exist.[1]

[1] Kull and Flach, "Patterns of dataset shift", 2014.

Prior probability shift



We cannot learn a classifier (solely) from $\mathbb{P}$ that is also optimal for $\mathbb{Q}$.

Classification versus quantification

For both tasks, we are given $\mathrm{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$

Classification:

- find $h : \mathcal{X} \rightarrow \mathcal{Y}$
- reason about individual data items
- (typically) assume IID data

Quantification:

- find $\lambda : \bigcup_{m=1}^{\infty} \mathcal{X}^m \rightarrow \Delta^{C-1}$
- reason about bags of data
- (typically) assume PPS

PPS requires quantifiers that are more sophisticated than Classify & Count.[2]

[2] Forman, "Quantifying counts and costs via classification", 2008.

1. Problem statement

2. Desirable properties of quantifiers

3. Binary quantifiers
4. Multi-class quantifiers
5. Numerical optimization
6. Loss functions & data representations
7. Beyond linear systems of equations

Fisher consistency

Definition (Fisher consistency for PPS):

If a quantifier had access to the entire population, it would return the correct class prevalences:

$$\underbrace{\lambda'\big(\mathbb{Q}(X)\big)}_{\substack{\text{population}\\ \text{analogue}\\ \text{of } \lambda(\mathrm{B})}} = \mathbb{Q}(Y) \quad \underbrace{\forall\ \mathbb{Q} : \mathbb{Q}(X \mid Y) = \mathbb{P}(X \mid Y)}_{\text{for any } \mathbb{Q} \text{ with PPS}}$$

Fisher consistency

Definition (Fisher consistency for PPS):

If a quantifier had access to the entire population, it would return the correct class prevalences:

$$\underbrace{\lambda'\big(\mathbb{Q}(X)\big)}_{\substack{\text{population} \\ \text{analogue} \\ \text{of } \lambda(\mathrm{B})}} = \mathbb{Q}(Y) \quad \underbrace{\forall\ \mathbb{Q} : \mathbb{Q}(X \mid Y) = \mathbb{P}(X \mid Y)}_{\text{for any } \mathbb{Q} \text{ with PPS}}$$

Notes:

- can also be defined for other types of data set shift
- is different from unbiasedness and different from asymptotical consistency
- does not indicate good performance on finite samples
- hence, not a sufficient but certainly **a necessary criterion** for quantifier selection

Tip: write down this definition; there might be a small assignment!

Estimation error

Since data is limited, we cannot solely rely on Fisher consistency.

Empirical evaluation: test quantifiers on data

- employ suitable protocols (as discussed the previous part of this tutorial)
- employ a representative collection of data sets

Estimation error

Since data is limited, we cannot solely rely on Fisher consistency.

Empirical evaluation: test quantifiers on data

- employ suitable protocols (as discussed the previous part of this tutorial)
- employ a representative collection of data sets

Asymptotical consistency: look for desirable asymptotical behaviour; with any bound of the type

$$\|\lambda(\mathrm{B}) - \mathbf{p}^*\| \;\leq\; f\Big(\lambda, |\mathrm{D}|, |\mathrm{B}|, \delta\Big)$$

prefer those quantifiers λ that achieve a small upper bound with a high probability $1 - \delta$

Resource efficiency

User perspective:

- little waiting times for predictions
- without requiring excessive hardware

[3] Google, *Environmental Report*, 2024.

Resource efficiency

User perspective:

- little waiting times for predictions
- without requiring excessive hardware

Environmental perspective:

- greenhouse gas emissions: use little computation and green energy
- Google:[3] "reducing emissions may be challenging due to increasing energy demands from the greater intensity of AI compute."
  (their emissions increased by 48%, as compared to 2019, despite their goal of reducing emissions by 50% in 2030)

[3] Google, *Environmental Report*, 2024.

Resource efficiency

User perspective:

- little waiting times for predictions
- without requiring excessive hardware

Environmental perspective:

- greenhouse gas emissions: use little computation and green energy
- Google:[3] "reducing emissions may be challenging due to increasing energy demands from the greater intensity of AI compute."
  (their emissions increased by 48%, as compared to 2019, despite their goal of reducing emissions by 50% in 2030)

Implications on quantification research:

- reduce resource consumption
- report on resource consumption (prediction times, memory consumption, GHG emissions, ...)

[3] Google, *Environmental Report*, 2024.

3. Binary quantifiers

Binary Adjusted Classify & Count

Preliminaries:

- let $\mathcal{Y} = \{1, 2\}$ (binary quantification)
- assume a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$

Binary Adjusted Classify & Count

Preliminaries:

- let $\mathcal{Y} = \{1, 2\}$ (binary quantification)
- assume a classifier $h : \mathcal{X} \to \mathcal{Y}$

$$\Rightarrow \quad \mathbb{Q}(h(X) = 1) = \sum_{i \in \mathcal{Y}} \mathbb{Q}(h(X) = 1 \mid Y = i) \cdot \mathbb{Q}(Y = i) \quad \text{(law of total probability)}$$

Binary Adjusted Classify & Count

Preliminaries:

- let $\mathcal{Y} = \{1, 2\}$ (binary quantification)
- assume a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$

$$
\begin{aligned}
\Rightarrow \quad \mathbb{Q}(h(X) = 1) &= \sum_{i \in \mathcal{Y}} \mathbb{Q}(h(X) = 1 \mid Y = i) \cdot \mathbb{Q}(Y = i) \quad \text{(law of total probability)} \\
&= \underbrace{\mathrm{TPR}}_{\mathbb{Q}(h(X) = 1 \mid Y = 1)} \cdot \mathbb{Q}(Y = 1) + \underbrace{\mathrm{FPR}}_{\mathbb{Q}(h(X) = 1 \mid Y = 2)} \cdot (1 - \mathbb{Q}(Y = 1))
\end{aligned}
$$

Binary Adjusted Classify & Count

Preliminaries:

- let $\mathcal{Y} = \{1, 2\}$ (binary quantification)
- assume a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$

$$\Rightarrow \quad \mathbb{Q}(h(X) = 1) = \sum_{i \in \mathcal{Y}} \mathbb{Q}(h(X) = 1 \mid Y = i) \cdot \mathbb{Q}(Y = i) \quad \text{(law of total probability)}$$

$$= \underbrace{\mathrm{TPR}}_{\mathbb{Q}(h(X) = 1 \mid Y = 1)} \cdot \mathbb{Q}(Y = 1) + \underbrace{\mathrm{FPR}}_{\mathbb{Q}(h(X) = 1 \mid Y = 2)} \cdot (1 - \mathbb{Q}(Y = 1))$$

$$\Rightarrow \quad \mathbb{Q}(Y = 1) = \frac{\mathbb{Q}(h(X) = 1) - \mathrm{FPR}}{\mathrm{TPR} - \mathrm{FPR}}$$

where:

- $\mathbb{Q}(h(X) = 1)$ can be estimated by counting the predictions $h(\mathbf{x})\ \forall\, \mathbf{x} \in \mathrm{B}$
- TPR and FPR can be estimated with the training data D (due to PPS)

Binary Adjusted Classify & Count

Definition (binary ACC):

$$\hat{\mathbb{Q}}(Y = 1) \;=\; \frac{\hat{\mathbb{Q}}(h(X) = 1) - \mathrm{F\hat{P}R}}{\mathrm{T\hat{P}R} - \mathrm{F\hat{P}R}}$$

is Fisher-consistent,[4] where

- $\hat{\mathbb{Q}}(h(X) = 1) \;=\; \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x} \in \mathrm{B}} \mathbb{1}_{h(\mathbf{x})=1}$
- $\mathrm{T\hat{P}R} \;=\; \frac{1}{|\mathrm{D}_1|} \sum_{\mathbf{x} \in \mathrm{D}_1} \mathbb{1}_{h(\mathbf{x})=1}$
- $\mathrm{F\hat{P}R} \;=\; \frac{1}{|\mathrm{D}_2|} \sum_{\mathbf{x} \in \mathrm{D}_2} \mathbb{1}_{h(\mathbf{x})=1}$
- $\mathrm{D}_i \;=\; \{(\mathbf{x}, y) \in \mathrm{D} \,:\, y = i\} \;\forall\, i \in \mathcal{Y}$

[4] Tasche, “Fisher consistency for prior probability shift”, 2017.

Binary Adjusted Classify & Count

Definition (binary ACC):

$$\hat{\mathbb{Q}}(Y = 1) \; = \; \frac{\hat{\mathbb{Q}}(h(X) = 1) - \mathrm{F\hat{P}R}}{\mathrm{T\hat{P}R} - \mathrm{F\hat{P}R}}$$

is Fisher-consistent,[4] where

- $\hat{\mathbb{Q}}(h(X) = 1) \; = \; \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x} \in \mathrm{B}} \mathbb{1}_{h(\mathbf{x})=1}$
- $\mathrm{T\hat{P}R} \; = \; \frac{1}{|\mathrm{D}_1|} \sum_{\mathbf{x} \in \mathrm{D}_1} \mathbb{1}_{h(\mathbf{x})=1}$
- $\mathrm{F\hat{P}R} \; = \; \frac{1}{|\mathrm{D}_2|} \sum_{\mathbf{x} \in \mathrm{D}_2} \mathbb{1}_{h(\mathbf{x})=1}$
- $\mathrm{D}_i \; = \; \{(\mathbf{x}, y) \in \mathrm{D} \, : \, y = i\} \; \forall \, i \in \mathcal{Y}$

Definition (binary probabilistic ACC / PACC):

Replace each occurence of $\mathbb{1}_{h(\mathbf{x})=1}$ with the soft classification $s(\mathbf{x}) \in [0, 1]$

Problem: $\hat{\mathbb{Q}}(Y = 1)$ might be undefined or outside of $[0, 1]$

[4] Tasche, “Fisher consistency for prior probability shift”, 2017.

Counter example: Classify & Count

Assignment [2 min]:

What would happen if we simply returned $\hat{\mathbb{Q}}(h(X) = 1)$ as our estimate of $\hat{\mathbb{Q}}(Y = 1)$?

Counter example: Classify & Count

Assignment [2 min]:

What would happen if we simply returned $\hat{\mathbb{Q}}(h(X) = 1)$ as our estimate of $\hat{\mathbb{Q}}(Y = 1)$?

Answer: on the population level, we would obtain

$$h\Big(\mathbb{Q}(X)\Big) \;=\; \mathbb{Q}(h(X) = 1)$$

Counter example: Classify & Count

Assignment [2 min]:

What would happen if we simply returned $\hat{\mathbb{Q}}(h(X) = 1)$ as our estimate of $\hat{\mathbb{Q}}(Y = 1)$?

Answer: on the population level, we would obtain

$$\begin{aligned} h\big(\mathbb{Q}(X)\big) &= \mathbb{Q}(h(X) = 1) \\ &= \mathrm{TPR} \cdot \mathbb{Q}(Y = 1) + \mathrm{FPR} \cdot (1 - \mathbb{Q}(Y = 1)) \end{aligned}$$

Counter example: Classify & Count

Assignment [2 min]:

What would happen if we simply returned $\hat{\mathbb{Q}}(h(X) = 1)$ as our estimate of $\hat{\mathbb{Q}}(Y = 1)$?

Answer: on the population level, we would obtain

$$\begin{aligned} h\big(\mathbb{Q}(X)\big) \; &= \; \mathbb{Q}(h(X) = 1) \\ &= \; \mathrm{TPR} \cdot \mathbb{Q}(Y = 1) \, + \, \mathrm{FPR} \cdot (1 - \mathbb{Q}(Y = 1)) \\ &\neq \; \mathbb{Q}(Y = 1) \end{aligned}$$

if $\mathrm{TPR} \neq 1$ or if $\mathrm{FPR} \neq 0$.

Hence, CC is **not** Fisher-consistent under PPS.

4. Multi-class quantifiers

Example: from binary to multi-class (P)ACC

Preliminaries:

- let $\mathcal{Y} = \{1, 2, \dots C\}$
- assume a classifier $h : \mathcal{X} \to \mathcal{Y}$

$$\Rightarrow \quad \mathbb{Q}(h(X) = i) \; = \sum_{j \in \mathcal{Y}} \mathbb{Q}(h(X) = i \mid Y = j) \cdot \mathbb{Q}(Y = j) \;\; \forall \; i \in \mathcal{Y}$$ (just like before)

Example: from binary to multi-class (P)ACC

Preliminaries:

- let $\mathcal{Y} = \{1, 2, \dots C\}$
- assume a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$

$$\Rightarrow \quad \mathbb{Q}(h(X) = i) = \sum_{j \in \mathcal{Y}} \mathbb{Q}(h(X) = i \mid Y = j) \cdot \mathbb{Q}(Y = j) \quad \forall \; i \in \mathcal{Y} \quad \text{(just like before)}$$

$$= \mathbf{M}_i^\top \mathbf{p}$$

where

$$\mathbf{M}_i = \begin{pmatrix} \mathbb{Q}(h(X) = i \mid Y = 1) \\ \mathbb{Q}(h(X) = i \mid Y = 2) \\ \vdots \\ \mathbb{Q}(h(X) = i \mid Y = C) \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} \mathbb{Q}(Y = 1) \\ \mathbb{Q}(Y = 2) \\ \vdots \\ \mathbb{Q}(Y = C) \end{pmatrix}$$

Example: from binary to multi-class (P)ACC

Preliminaries:

- let $\mathcal{Y} = \{1, 2, \dots C\}$
- assume a classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$

$$\Rightarrow \quad \mathbb{Q}(h(X) = i) \; = \sum_{j \in \mathcal{Y}} \mathbb{Q}(h(X) = i \mid Y = j) \cdot \mathbb{Q}(Y = j) \quad \forall \; i \in \mathcal{Y} \quad \text{(just like before)}$$

$$= \mathbf{M}_i^\top \mathbf{p}$$

where

$$\mathbf{M}_i = \begin{pmatrix} \mathbb{Q}(h(X) = i \mid Y = 1) \\ \mathbb{Q}(h(X) = i \mid Y = 2) \\ \vdots \\ \mathbb{Q}(h(X) = i \mid Y = C) \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} \mathbb{Q}(Y = 1) \\ \mathbb{Q}(Y = 2) \\ \vdots \\ \mathbb{Q}(Y = C) \end{pmatrix} \qquad \mathbf{M} = \begin{pmatrix} \mathbf{M}_1^\top \\ \mathbf{M}_2^\top \\ \vdots \\ \mathbf{M}_C^\top \end{pmatrix} \qquad \mathbf{q} = \begin{pmatrix} \mathbb{Q}(h(X) = 1) \\ \mathbb{Q}(h(X) = 2) \\ \vdots \\ \mathbb{Q}(h(X) = C) \end{pmatrix}^\top$$

such that $\mathbf{q} = \mathbf{M}\mathbf{p}$

Example: from binary to multi-class (P)ACC

Synopsis:

- we have just seen how multi-class ACC and PACC yield systems of equations
- we have also seen how binary ACC and PACC differ in their computation of $\mathrm{T\hat{P}R}, \mathrm{F\hat{P}R},$ and $\hat{\mathbb{Q}}(h(X) = 1)$

  $$\text{e.g.,} \qquad \hat{\mathbb{Q}}(h(X) = 1) = \begin{cases} \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x} \in \mathrm{B}} \mathbb{1}_{h(\mathbf{x})=1} & \text{(ACC)} \\ \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x} \in \mathrm{B}} s(\mathbf{x}) & \text{(PACC)} \end{cases}$$

  (they represent the data differently, either through $h(\mathbf{x})$ or $s(\mathbf{x})$)
- we have not yet discussed how $\mathbf{q} = \mathbf{M}\mathbf{p}$ can be solved

Example: from binary to multi-class (P)ACC

Synopsis:

- we have just seen how multi-class ACC and PACC yield systems of equations
- we have also seen how binary ACC and PACC differ in their computation of $\hat{\mathrm{TPR}}, \hat{\mathrm{FPR}}$, and $\hat{\mathbb{Q}}(h(X) = 1)$

$$\text{e.g.,} \qquad \hat{\mathbb{Q}}(h(X) = 1) \; = \begin{cases} \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x} \in \mathrm{B}} \mathbb{1}_{h(\mathbf{x})=1} & \text{(ACC)} \\ \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x} \in \mathrm{B}} s(\mathbf{x}) & \text{(PACC)} \end{cases}$$

  (they represent the data differently, either through $h(\mathbf{x})$ or $s(\mathbf{x})$)
- we have not yet discussed how $\mathbf{q} = \mathbf{M}\mathbf{p}$ can be solved

Next steps [30 min]:

- generalize these concepts towards arbitrary data representations [10 min]
- discuss ways of solving $\mathbf{q} = \mathbf{M}\mathbf{p}$
- define concrete representations (in addition to those of ACC and PACC)

General systems of linear equations

More generally: any data representation $\Phi : \mathcal{X} \rightarrow \mathcal{Z}$ yields another system of equations $\mathbf{q} = \mathbf{M}\mathbf{p}$ via

$$\mathbb{Q}(\Phi(X) = z) \; = \sum_{i \in \mathcal{Y}} \mathbb{Q}(\Phi(X) = z \mid Y = i) \cdot \mathbb{Q}(Y = i) \quad \forall \; z \in \mathcal{Z}$$

and any of these systems can suit the purpose of finding $\mathbf{p}$.

[5] Dussap, Blanchard, and Chérief-Abdellatif, "Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching", 2023.

General systems of linear equations

More generally: any data representation $\Phi : \mathcal{X} \to \mathcal{Z}$ yields another system of equations $\mathbf{q} = \mathbf{M}\mathbf{p}$ via

$$\mathbb{Q}(\Phi(X) = z) \;=\; \sum_{i \in \mathcal{Y}} \mathbb{Q}(\Phi(X) = z \mid Y = i) \cdot \mathbb{Q}(Y = i) \quad \forall\; z \in \mathcal{Z}$$

and any of these systems can suit the purpose of finding $\mathbf{p}$.

The solution of the system $\mathbf{q} = \mathbf{M}\mathbf{p}$

- is Fisher-consistent by construction
- is asymptotically consistent:[5]

$$\left\| \lambda(\mathrm{B}) - \mathbf{p}^* \right\|_2 \;\leq\; \underbrace{\frac{2k(2 + \sqrt{2 \log \frac{2C}{\delta}})}{\sqrt{\lambda_2}}}_{\text{representation } \Phi} \left(\underbrace{\frac{\| \frac{\mathbf{p}^*}{\mathbf{p}_{\mathrm{trn}}} \|_2}{\sqrt{|\mathrm{D}|}}}_{\text{shift \& volume}} + \underbrace{\frac{1}{\sqrt{|\mathrm{B}|}}}_{\text{volume}} \right) \quad \text{where} \begin{cases} k & \text{constant s.t. } \|\Phi(\mathbf{x})\|_2 \leq k \;\; \forall\, \mathbf{x} \in \mathcal{X} \\ \lambda_2 & \text{second-smallest eigenvalue of } \mathbf{G} \\ \delta & \text{desired probability} \end{cases}$$

[5] Dussap, Blanchard, and Chérief-Abdellatif, “Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching”, 2023.

Counter example: one-vs-rest quantification

Can we not just use binary (P)ACC with for each binary task in an OVR decomposition?

$$\hat{\mathbb{Q}}(Y = i) \;=\; \frac{\hat{\mathbb{Q}}(h(X) = 1) - \hat{\mathrm{FPR}}_i}{\hat{\mathrm{TPR}}_i - \hat{\mathrm{FPR}}_i} \quad \forall\, i \in \mathcal{Y}$$

[6] Gövert, "Fisher-Konsistenz für Quantification-Algorithmen", 2023, supervised by M. Bunse and S. Mücke.

[7] Donyavi, Serapião, and Batista, "MC-SQ: A Highly Accurate Ensemble for Multi-class Quantification", 2023.

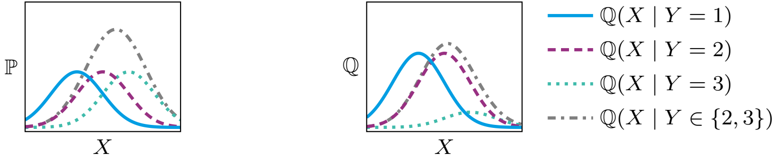
Counter example: one-vs-rest quantification

Can we not just use binary (P)ACC with for each binary task in an OVR decomposition?

$$\hat{\mathbb{Q}}(Y = i) \;=\; \frac{\hat{\mathbb{Q}}(h(X) = 1) - \mathrm{F\hat{P}R}_i}{\mathrm{T\hat{P}R}_i - \mathrm{F\hat{P}R}_i} \quad \forall\, i \in \mathcal{Y}$$

Assignment [2 min]: What is the problem in the following situation?

training: $\mathbb{P}(Y) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ testing: $\mathbb{Q}(Y) = (\frac{5}{11}, \frac{5}{11}, \frac{1}{11})$



[6] Gövert, “Fisher-Konsistenz für Quantification-Algorithmen”, 2023, supervised by M. Bunse and S. Mücke.
[7] Donyavi, Serapião, and Batista, “MC-SQ: A Highly Accurate Ensemble for Multi-class Quantification”, 2023.

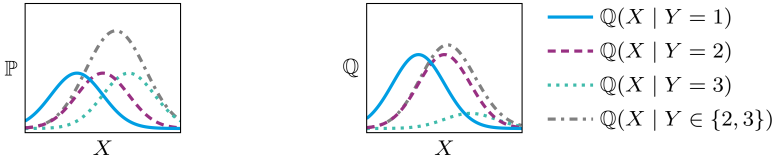
Counter example: one-vs-rest quantification

Can we not just use binary (P)ACC with for each binary task in an OVR decomposition?

$$\hat{\mathbb{Q}}(Y = i) \;=\; \frac{\hat{\mathbb{Q}}(h(X) = 1) - \hat{\mathrm{FPR}}_i}{\hat{\mathrm{TPR}}_i - \hat{\mathrm{FPR}}_i} \quad \forall\, i \in \mathcal{Y}$$

Assignment [2 min]: What is the problem in the following situation?

training: $\mathbb{P}(Y) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ testing: $\mathbb{Q}(Y) = (\frac{5}{11}, \frac{5}{11}, \frac{1}{11})$



- PPS among $C > 2$ classes leads to concept shift in OVR decompositions[6,7]
- hence, OVR quantification with Fisher-consistent binary quantifiers is not Fisher-consistent

[6] Gövert, “Fisher-Konsistenz für Quantification-Algorithmen”, 2023, supervised by M. Bunse and S. Mücke.
[7] Donyavi, Serapião, and Batista, “MC-SQ: A Highly Accurate Ensemble for Multi-class Quantification”, 2023.

Synopsis

We learned how to achieve Fisher consistency:

- define some data representation $\Phi : \mathcal{X} \rightarrow \mathcal{Z}$
- solve $\mathbf{q} = \mathbf{M}\mathbf{p}$

Next steps:

- discuss ways of solving $\mathbf{q} = \mathbf{M}\mathbf{p}$
- define concrete representations Φ (in addition to those of ACC and PACC)

1. Problem statement
2. Desirable properties of quantifiers
3. Binary quantifiers
4. Multi-class quantifiers

5. Numerical optimization

6. Loss functions & data representations
7. Beyond linear systems of equations

Matrix inversion

Goal: solve $\mathbf{q} = \mathbf{M}\mathbf{p}$

- in other words, find $\mathbf{p}$, given $\mathbf{q}$ and $\mathbf{M}$
- $\mathbf{q}$ and $\mathbf{M}$ are fully defined through Φ, B, and D

[8] Mueller and Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*, 2012.

Matrix inversion

Goal: solve $\mathbf{q} = \mathbf{M}\mathbf{p}$

- in other words, find $\mathbf{p}$, given $\mathbf{q}$ and $\mathbf{M}$
- $\mathbf{q}$ and $\mathbf{M}$ are fully defined through Φ, B, and D

Naive solution: choose $\hat{\mathbf{p}} = \mathbf{M}^{-1}\mathbf{q}$

- the inverse $\mathbf{M}^{-1}$ is not guaranteed to exist ($\mathbf{M}$ might not even be square)
- if $\mathbf{M}^{-1}$ exists, $\hat{\mathbf{p}}$ is not guaranteed to be in Δ^{C-1} (an ad-hoc projection is necessary)

[8] Mueller and Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*, 2012.

Matrix inversion

Goal: solve $\mathbf{q} = \mathbf{M}\mathbf{p}$

- in other words, find $\mathbf{p}$, given $\mathbf{q}$ and $\mathbf{M}$
- $\mathbf{q}$ and $\mathbf{M}$ are fully defined through Φ, B, and D

Naive solution: choose $\hat{\mathbf{p}} = \mathbf{M}^{-1}\mathbf{q}$

- the inverse $\mathbf{M}^{-1}$ is not guaranteed to exist ($\mathbf{M}$ might not even be square)
- if $\mathbf{M}^{-1}$ exists, $\hat{\mathbf{p}}$ is not guaranteed to be in Δ^{C-1} (an ad-hoc projection is necessary)

Naive improvement: choose $\hat{\mathbf{p}} = \mathbf{M}^{\dagger}\mathbf{q}$ with the Moore-Penrose pseudo-inverse $\mathbf{M}^{\dagger}$

- $\mathbf{M}^{\dagger}$ always exists and $\hat{\mathbf{p}}$ is a unique solution
- however, $\hat{\mathbf{p}}$ is still not guaranteed to be in Δ^{C-1}
- $\hat{\mathbf{p}}$ is a minimum-norm least-squares solution[8] (while a minimum norm does not relate to quantification)

[8] Mueller and Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*, 2012.

Constrained optimization

Proper solution: constrain $\hat{\mathbf{p}}$ to always be in Δ^{C-1}, i.e.,

$$\hat{\mathbf{p}} = \operatorname*{arg\,min}_{\mathbf{p} \in \Delta^{C-1}} \ell(\mathbf{q}, \mathbf{M}\mathbf{p})$$

where $\ell : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$ is a loss function, e.g., choose

$$\ell(\mathbf{q}, \mathbf{M}\mathbf{p}) = \left|\left|\mathbf{q} - \mathbf{M}\mathbf{p}\right|\right|_2^2 \qquad \text{(least squares)}$$

Remarks:

- we will soon learn about other loss functions
- a straightforward implementation requires **constrained optimization algorithms**

Implicit constraints

Can we use **unconstrained** optimization algorithms?

Yes:[9] use the soft-max operator $\sigma : \mathbb{R}^{C-1} \to \Delta^{C-1}$ and optimize over log-odds $\mathbf{l} \in \mathbb{R}^{C-1}$, i.e.,

$$\hat{\mathbf{p}} = \sigma(\mathbf{l}^*)$$

$$\mathbf{l}^* = \operatorname*{arg\,min}_{\mathbf{l} \in \mathbb{R}^{C-1}} \ell\Big(\mathbf{q},\, \mathbf{M}\sigma(\mathbf{l})\Big)$$

[9] Bunse, "On Multi-Class Extensions of Adjusted Classify and Count", 2022.

Implicit constraints

Can we use **unconstrained** optimization algorithms?

Yes:[9] use the soft-max operator $\sigma : \mathbb{R}^{C-1} \rightarrow \Delta^{C-1}$ and optimize over log-odds $\mathbf{l} \in \mathbb{R}^{C-1}$, i.e.,

$$\hat{\mathbf{p}} = \sigma(\mathbf{l}^*)$$

$$\mathbf{l}^* = \operatorname*{arg\,min}_{\mathbf{l} \in \mathbb{R}^{C-1}} \ell\Big(\mathbf{q},\, \mathbf{M}\sigma(\mathbf{l})\Big)$$

$$\Big[\sigma(\mathbf{l})\Big]_i = \begin{cases} \frac{1}{1+\sum_{j=1}^{C-1} \exp(\mathbf{l}_j)} & \text{if } i = 1 \\ \frac{\exp(\mathbf{l}_{i-1})}{1+\sum_{j=1}^{C-1} \exp(\mathbf{l}_j)} & \forall\, i \in \{2, 3, \dots C\} \end{cases}$$

[9] Bunse, “On Multi-Class Extensions of Adjusted Classify and Count”, 2022.

Synopsis

Components of a quantification algorithm:

- a data representation $\Phi : \mathcal{X} \rightarrow \mathcal{Z}$
- a loss function $\ell : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$
- an optimization algorithm (matrix inversion does not suffice!)

Next steps:

- define concrete choices of Φ and ℓ (in addition to those of ACC and PACC)
- discuss algorithms beyond solutions of $\mathbf{q} = \mathbf{M}\mathbf{p}$

6. Loss functions & data representations

(Probabilistic) Adjusted Classify & Count

Loss function:

$$\ell(\mathbf{q}, \mathbf{M}\mathbf{p}) = \left\| \mathbf{q} - \mathbf{M}\mathbf{p} \right\|_2^2 \qquad \text{(least squares)}$$

Representation:[10,11]

$$\Phi(\mathbf{x}) = \begin{cases} \mathbb{1}_{h(\mathbf{x})} \in \{0,1\}^C & \text{ACC} \quad \text{(a one-hot encoding of } h(\mathbf{x})\text{)} \\ s(\mathbf{x}) \in \Delta^{C-1} & \text{PACC} \end{cases}$$

where

- $h : \mathcal{X} \rightarrow \mathcal{Y}$ a "hard" classifier such that $h(\mathbf{x}) = \hat{y}$
- $s : \mathcal{X} \rightarrow \Delta^{C-1}$ a "soft" classifier such that $s(\mathbf{x}) \approx \mathbb{P}(Y \mid \mathbf{x})$

[10] Firat, "Unified Framework for Quantification", 2016.
[11] Bunse, "Unification of Algorithms for Quantification and Unfolding", 2022.

Distribution matching: HDx and HDy

Loss function:

$$\ell(\mathbf{q}, \mathbf{M}\mathbf{p}) = \frac{1}{d} \sum_{i=1}^{d} \mathrm{HD}(\mathbf{q}_{i\bullet}, \mathbf{M}_{i\bullet\bullet}\mathbf{p}) \qquad \text{where} \qquad \mathrm{HD}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i}^{b} \left(\sqrt{\mathbf{a}_i} - \sqrt{\mathbf{b}_i} \right)^2}$$

Representation:

$$\Phi(\mathbf{x}) = \begin{cases} \left(\mathbb{1}_{b_1(\mathbf{x}_1)}, \mathbb{1}_{b_2(\mathbf{x}_2)}, \dots \mathbb{1}_{b_d(\mathbf{x}_d)} \right) \in \{0,1\}^{Bd} & \text{HDx} \\ \left(\mathbb{1}_{b_1([s(\mathbf{x})]_1)}, \mathbb{1}_{b_2([s(\mathbf{x})]_2)}, \dots \mathbb{1}_{b_d([s(\mathbf{x})]_d)} \right) \in \{0,1\}^{BC} & \text{HDy} \end{cases}$$

where $b_i : \mathbb{R} \to \{1, 2, \dots B\}$ is a binning of the i-th feature (or class probability)

Problem: HD is not twice differentiable $\Rightarrow$ prefer HD^2 instead.[12]

[12] Bunse, "qunfold: Composable Quantification and Unfolding Methods in Python", 2023.

Kernel mean matching: EDx, EDy, and others

Representation:

$$[\Phi(\mathbf{x})]_i = \frac{1}{|\mathrm{D}_i|} \sum_{\mathbf{x}' \in \mathrm{D}_i} K(\mathbf{x}, \mathbf{x}')$$

where $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is some kernel between data items, e.g.,

- $\|\mathbf{x} - \mathbf{x}'\|_2$ (Euclidean distance; EDx[13])
- $\sum_{i=1}^{C-1} \left| \sum_{j=1}^{i} [s(\mathbf{x})]_j - [s(\mathbf{x}')]_j \right|$ (Earth Mover's Distance; EDy[14])
- $\exp\left(\frac{-\|\mathbf{x}-\mathbf{x}'\|_2}{(2\sigma)^2}\right)$ (Gaussian kernel)

[13] Kawakubo, Plessis, and Sugiyama, "Computationally Efficient Class-Prior Estimation under Class Balance Change Using Energy Distance", 2016.

[14] Castaño et al., "Matching Distributions Algorithms Based on the Earth Mover's Distance for Ordinal Quantification", 2022.

[15] Dussap, Blanchard, and Chérief-Abdellatif, "Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching", 2023.

Kernel mean matching: EDx, EDy, and others

Representation:

$$[\Phi(\mathbf{x})]_i \;=\; \frac{1}{|\mathrm{D}_i|} \sum_{\mathbf{x}' \in \mathrm{D}_i} K(\mathbf{x}, \mathbf{x}')$$

where $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is some kernel between data items, e.g.,

- $\|\mathbf{x} - \mathbf{x}'\|_2$ (Euclidean distance; EDx[13])
- $\sum_{i=1}^{C-1} \left| \sum_{j=1}^{i} [s(\mathbf{x})]_j - [s(\mathbf{x}')]_j \right|$ (Earth Mover's Distance; EDy[14])
- $\exp\left(\frac{-\|\mathbf{x}-\mathbf{x}'\|_2}{(2\sigma)^2}\right)$ (Gaussian kernel)

Problem: a high computational cost $\Rightarrow$ use a random Fourier approximation[15]

13 Kawakubo, Plessis, and Sugiyama, "Computationally Efficient Class-Prior Estimation under Class Balance Change Using Energy Distance", 2016.

14 Castaño et al., "Matching Distributions Algorithms Based on the Earth Mover's Distance for Ordinal Quantification", 2022.

15 Dussap, Blanchard, and Chérief-Abdellatif, "Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching", 2023.

Kernel mean matching: EDx, EDy, and others

Representation:

$$[\Phi(\mathbf{x})]_i = \frac{1}{|\mathrm{D}_i|} \sum_{\mathbf{x}' \in \mathrm{D}_i} K(\mathbf{x}, \mathbf{x}')$$

where $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is some kernel between data items, e.g.,

- $\|\mathbf{x} - \mathbf{x}'\|_2$ (Euclidean distance; EDx[13])
- $\sum_{i=1}^{C-1} \left| \sum_{j=1}^{i} [s(\mathbf{x})]_j - [s(\mathbf{x}')]_j \right|$ (Earth Mover's Distance; EDy[14])
- $\exp\left(\frac{-\|\mathbf{x}-\mathbf{x}'\|_2}{(2\sigma)^2} \right)$ (Gaussian kernel)

Problem: a high computational cost $\Rightarrow$ use a random Fourier approximation[15]

Loss function (EDx and EDy): $\ell(\mathbf{q}, \mathbf{M}\mathbf{p}) = 2\mathbf{p}^T\mathbf{q} - \mathbf{p}^\top \mathbf{M}\mathbf{p}$ (any other loss is possible)

[13] Kawakubo, Plessis, and Sugiyama, "Computationally Efficient Class-Prior Estimation under Class Balance Change Using Energy Distance", 2016.

[14] Castaño et al., "Matching Distributions Algorithms Based on the Earth Mover's Distance for Ordinal Quantification", 2022.

[15] Dussap, Blanchard, and Chérief-Abdellatif, "Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching", 2023.

Regularization

So far, we've been very concerned about consistency. But what if

- the data volume is small?
- we've strong assumptions about how the predictions should look like?

[16] Bunse et al., "Regularization-based Methods for Ordinal Quantification", 2024.

Regularization

So far, we've been very concerned about consistency. But what if

- the data volume is small?
- we've strong assumptions about how the predictions should look like?

Regularization:

$$\ell'(\mathbf{q}, \mathbf{M}\mathbf{p}) \;=\; \ell(\mathbf{q}, \mathbf{M}\mathbf{p}) \;+\; \tau \cdot r(\mathbf{p})$$

- $\tau \geq 0$ is the regularization impact (i.e., a hyper-parameter that needs to be optimized)
- $r : \Delta^{C-1} \to \mathbb{R}$ is a regularization term that penalizes any deviation from our assumptions

[16] Bunse et al., "Regularization-based Methods for Ordinal Quantification", 2024.

Regularization

So far, we've been very concerned about consistency. But what if

- the data volume is small?
- we've strong assumptions about how the predictions should look like?

Regularization:

$$\ell'(\mathbf{q}, \mathbf{M}\mathbf{p}) \;=\; \ell(\mathbf{q}, \mathbf{M}\mathbf{p}) \;+\; \tau \cdot r(\mathbf{p})$$

- $\tau \geq 0$ is the regularization impact (i.e., a hyper-parameter that needs to be optimized)
- $r : \Delta^{C-1} \to \mathbb{R}$ is a regularization term that penalizes any deviation from our assumptions

Tikhonov regularization:[16]

$$r(\mathbf{p}) \;=\; \frac{1}{2}(\mathbf{C}\mathbf{p})^2 \;=\; \begin{cases} \frac{1}{2}\sum_{i=2}^{C-1}(-\mathbf{p}_{i-1} + 2\mathbf{p}_i - \mathbf{p}_{i+1})^2 & \text{ordinal quantification} \\ \frac{1}{2}\sum_{i=1}^{C-1}(\mathbf{p}_i - \mathbf{p}_{i+1})^2 & \text{deviation from } \mathbf{p}_i = \frac{1}{C} \end{cases}$$

[16] Bunse et al., "Regularization-based Methods for Ordinal Quantification", 2024.

Synopsis

Many quantification algorithms are combinations of:

- a data representation $\Phi : \mathcal{X} \to \mathcal{Z}$
- a loss function $\ell : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$
- an optimization algorithm

We have still omitted many methods from this family (ReadMe, PDF, unfolding methods, ...)

Synopsis

Many quantification algorithms are combinations of:

- a data representation $\Phi : \mathcal{X} \to \mathcal{Z}$
- a loss function $\ell : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$
- an optimization algorithm

We have still omitted many methods from this family (ReadMe, PDF, unfolding methods, ...)

We can alter these algorithms by:

- re-combining their Φ and ℓ
- approximating their representations
- adding regularization

Next step: discuss algorithms beyond solutions of $\mathbf{q} = \mathbf{M}\mathbf{p}$

7. Beyond linear systems of equations

Expectation maximization

Preliminaries:

$$\mathbb{Q}(\mathbf{x} \mid y) \overset{\text{Bayes}}{=} \frac{\mathbb{Q}(y \mid \mathbf{x}) \cdot \mathbb{Q}(\mathbf{x})}{\mathbb{Q}(y)} \overset{\text{PPS}}{=} \mathbb{P}(\mathbf{x} \mid y) \overset{\text{Bayes}}{=} \frac{\mathbb{P}(y \mid \mathbf{x}) \cdot \mathbb{P}(\mathbf{x})}{\mathbb{P}(y)} \qquad \forall\, (\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$$

[17] Saerens, Latinne, and Decaestecker, "Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure", 2002.

Expectation maximization

Preliminaries:

$$\mathbb{Q}(\mathbf{x} \mid y) \overset{\text{Bayes}}{=} \frac{\mathbb{Q}(y \mid \mathbf{x}) \cdot \mathbb{Q}(\mathbf{x})}{\mathbb{Q}(y)} \overset{\text{PPS}}{=} \mathbb{P}(\mathbf{x} \mid y) \overset{\text{Bayes}}{=} \frac{\mathbb{P}(y \mid \mathbf{x}) \cdot \mathbb{P}(\mathbf{x})}{\mathbb{P}(y)} \qquad \forall\ (\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$$

$$\Rightarrow \quad \mathbb{Q}(y \mid \mathbf{x}) = \frac{\frac{\mathbb{Q}(y)}{\mathbb{P}(y)} \cdot \mathbb{P}(y \mid \mathbf{x})}{\sum_{y' \in \mathcal{Y}} \frac{\mathbb{Q}(y')}{\mathbb{P}(y')} \cdot \mathbb{P}(y' \mid \mathbf{x})}$$

(the original purpose of this method)

[17] Saerens, Latinne, and Decaestecker, “Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure”, 2002.

Expectation maximization

Preliminaries:

$$\mathbb{Q}(\mathbf{x} \mid y) \overset{\text{Bayes}}{=} \frac{\mathbb{Q}(y \mid \mathbf{x}) \cdot \mathbb{Q}(\mathbf{x})}{\mathbb{Q}(y)} \overset{\text{PPS}}{=} \mathbb{P}(\mathbf{x} \mid y) \overset{\text{Bayes}}{=} \frac{\mathbb{P}(y \mid \mathbf{x}) \cdot \mathbb{P}(\mathbf{x})}{\mathbb{P}(y)} \qquad \forall\, (\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$$

$$\Rightarrow \quad \mathbb{Q}(y \mid \mathbf{x}) = \frac{\frac{\mathbb{Q}(y)}{\mathbb{P}(y)} \cdot \mathbb{P}(y \mid \mathbf{x})}{\sum_{y' \in \mathcal{Y}} \frac{\mathbb{Q}(y')}{\mathbb{P}(y')} \cdot \mathbb{P}(y' \mid \mathbf{x})} \qquad \text{(the original purpose of this method)}$$

SLD / EMQ:[17] repeat the E-step and the M-step until convergence.

Initialize: $\mathbf{p}^{(0)} \leftarrow \hat{\mathbb{P}}(Y)$

$\hat{\mathbb{Q}}^{(0)}(y \mid \mathbf{x}) \leftarrow \hat{\mathbb{P}}(y \mid \mathbf{x}) \quad \forall\, y \in \mathcal{Y},\, \mathbf{x} \in \mathrm{B}$

E-step: $[\mathbf{p}^{(k)}]_i \leftarrow \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x} \in \mathrm{B}} \hat{\mathbb{Q}}^{(k-1)}(Y = i \mid \mathbf{x})$

M-step: $\hat{\mathbb{Q}}^{(k)}(y \mid \mathbf{x}) \leftarrow \dfrac{\frac{[\mathbf{p}^{(k)}]_y}{\hat{\mathbb{P}}(y)} \cdot \hat{\mathbb{P}}(y|\mathbf{x})}{\sum_{y' \in \mathcal{Y}} \frac{[\mathbf{p}^{(k)}]_{y'}}{\hat{\mathbb{P}}(y')} \cdot \hat{\mathbb{P}}(y'|\mathbf{x})} \quad \forall\, y \in \mathcal{Y},\, \mathbf{x} \in \mathrm{B}$

[17] Saerens, Latinne, and Decaestecker, “Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure”, 2002.

Expectation maximization

Properties: SLD / EMQ is...

- Fisher consistent[18]
- equivalent to the maximum likelihood quantifier[19] (which is to be presented next)
- and it maintains per-example contributions $\hat{\mathbb{Q}}(Y = i \mid \mathbf{x})$

[18] Tasche, "Fisher consistency for prior probability shift", 2017.

[19] Alexandari, Kundaje, and Shrikumar, "Maximum Likelihood with Bias-Corrected Calibration is Hard-To-Beat at Label Shift Adaptation", 2020.

Maximum likelihood

Likelihood principle:

$$
\begin{aligned}
\mathcal{L}(\mathbf{p} \mid \mathrm{B}) &= \mathbb{Q}(\mathrm{B} \mid \mathbf{p}) \\
&= \prod_{\mathbf{x} \in \mathrm{B}} \mathbb{Q}(\mathbf{x} \mid \mathbf{p}) \\
&\overset{\text{PPS}}{=} \prod_{\mathbf{x} \in \mathrm{B}} \sum_{y \in \mathcal{Y}} \mathbb{P}(\mathbf{x} \mid y) \cdot \mathbf{p}_y
\end{aligned}
$$

Maximum likelihood

Likelihood principle:

$$\begin{aligned}
\mathcal{L}(\mathbf{p} \mid \mathrm{B}) &= \mathbb{Q}(\mathrm{B} \mid \mathbf{p}) \\
&= \prod_{\mathbf{x} \in \mathrm{B}} \mathbb{Q}(\mathbf{x} \mid \mathbf{p}) \\
&\overset{\text{PPS}}{=} \prod_{\mathbf{x} \in \mathrm{B}} \sum_{y \in \mathcal{Y}} \mathbb{P}(\mathbf{x} \mid y) \cdot \mathbf{p}_y
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad -\log \mathcal{L}(\mathbf{p} \mid \mathrm{B}) &= -\sum_{\mathbf{x} \in \mathrm{B}} \log \sum_{y \in \mathcal{Y}} \mathbb{P}(\mathbf{x} \mid y) \cdot \mathbf{p}_y \\
&\propto -\sum_{\mathbf{x} \in \mathrm{B}} \log \sum_{y \in \mathcal{Y}} \frac{\mathbb{P}(y|\mathbf{x})}{\mathbb{P}(y)} \cdot \mathbf{p}_y
\end{aligned}$$

Maximum likelihood

Likelihood principle:

$$\begin{aligned}
\mathcal{L}(\mathbf{p} \mid \mathrm{B}) &= \mathbb{Q}(\mathrm{B} \mid \mathbf{p}) \\
&= \prod_{\mathbf{x} \in \mathrm{B}} \mathbb{Q}(\mathbf{x} \mid \mathbf{p}) \\
&\overset{\text{PPS}}{=} \prod_{\mathbf{x} \in \mathrm{B}} \sum_{y \in \mathcal{Y}} \mathbb{P}(\mathbf{x} \mid y) \cdot \mathbf{p}_y
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad -\log \mathcal{L}(\mathbf{p} \mid \mathrm{B}) &= -\sum_{\mathbf{x} \in \mathrm{B}} \log \sum_{y \in \mathcal{Y}} \mathbb{P}(\mathbf{x} \mid y) \cdot \mathbf{p}_y \\
&\propto -\sum_{\mathbf{x} \in \mathrm{B}} \log \sum_{y \in \mathcal{Y}} \frac{\mathbb{P}(y|\mathbf{x})}{\mathbb{P}(y)} \cdot \mathbf{p}_y
\end{aligned}$$

Therefore, choose $\quad \hat{\mathbf{p}} = \arg\min_{\mathbf{p} \in \Delta^{C-1}} -\sum_{\mathbf{x} \in \mathrm{B}} \log \sum_{y \in \mathcal{Y}} \frac{\hat{\mathbb{P}}(y|\mathbf{x})}{\hat{\mathbb{P}}(y)} \cdot \mathbf{p}_y$

Continuous representations

KDEy:[20] represent all probabilities through kernel density estimates (KDEs), i.e.,

$$\hat{\mathbb{Q}}(\mathbf{x}) = \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x}' \in \mathrm{B}} K(s(\mathbf{x}), s(\mathbf{x}')) \quad \text{and} \quad \hat{\mathbb{Q}}(\mathbf{x} \mid y) = \frac{1}{|\mathrm{D}_y|} \sum_{\mathbf{x}' \in \mathrm{D}_y} K(s(\mathbf{x}), s(\mathbf{x}')) \qquad \forall\, \mathbf{x} \in \mathcal{X}$$

where $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel function (e.g., a Gaussian kernel with some bandwidth)

[20] Moreo, González, and Coz, “Kernel Density Estimation for Multiclass Quantification”, 2024.

Continuous representations

KDEy:[20] represent all probabilities through kernel density estimates (KDEs), i.e.,

$$\hat{\mathbb{Q}}(\mathbf{x}) = \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x}' \in \mathrm{B}} K(s(\mathbf{x}),\, s(\mathbf{x}')) \quad \text{and} \quad \hat{\mathbb{Q}}(\mathbf{x} \mid y) = \frac{1}{|\mathrm{D}_y|} \sum_{\mathbf{x}' \in \mathrm{D}_y} K(s(\mathbf{x}),\, s(\mathbf{x}')) \qquad \forall\, \mathbf{x} \in \mathcal{X}$$

where $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a kernel function (e.g., a Gaussian kernel with some bandwidth)

Remark: this is different from KMM, where $[\Phi(\mathbf{x})]_i = \frac{1}{|\mathrm{D}_i|} \sum_{\mathbf{x}' \in \mathrm{D}_i} K(s(\mathbf{x}), s(\mathbf{x}')) \quad \forall\, \mathbf{x} \in \mathrm{D} \cup \mathrm{B}$

[20] Moreo, González, and Coz, “Kernel Density Estimation for Multiclass Quantification”, 2024.

Continuous representations

KDEy:[20] represent all probabilities through kernel density estimates (KDEs), i.e.,

$$\hat{\mathbb{Q}}(\mathbf{x}) = \frac{1}{|\mathrm{B}|} \sum_{\mathbf{x}' \in \mathrm{B}} K(s(\mathbf{x}), s(\mathbf{x}')) \quad \text{and} \quad \hat{\mathbb{Q}}(\mathbf{x} \mid y) = \frac{1}{|\mathrm{D}_y|} \sum_{\mathbf{x}' \in \mathrm{D}_y} K(s(\mathbf{x}), s(\mathbf{x}')) \qquad \forall\, \mathbf{x} \in \mathcal{X}$$

where $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel function (e.g., a Gaussian kernel with some bandwidth)

Remark: this is different from KMM, where $[\Phi(\mathbf{x})]_i = \frac{1}{|\mathrm{D}_i|} \sum_{\mathbf{x}' \in \mathrm{D}_i} K(s(\mathbf{x}), s(\mathbf{x}')) \quad \forall\, \mathbf{x} \in \mathrm{D} \cup \mathrm{B}$

Optimization task: is determined by the choice of loss function.

- **Losses with closed-form solutions** lead to specific tasks (e.g., Cauchy-Schwarz)
- **Negative log-likelihood** leads to the maximum likelihood estimator (with a KDE representation)
- **MC-sampled losses** lead to $\mathbf{q} = \mathbf{M}\mathbf{p}$ tasks

[20] Moreo, González, and Coz, "Kernel Density Estimation for Multiclass Quantification", 2024.

Symmetric learning

So far, we have assumed a training set $\mathrm{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n}$, but what if we have

$$\mathrm{D}' = \left\{(\mathrm{B}_i, \mathbf{p}_i) \in \cup_{m=1}^{\infty} \mathcal{X}^m \times \Delta^{C-1}\right\}_{i=1}^{n}$$

[21] Pérez-Mon et al., “Quantification using Permutation-Invariant Networks based on Histograms”, 2024.

Symmetric learning

So far, we have assumed a training set $\mathrm{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n}$, but what if we have

$$\mathrm{D}' = \left\{(\mathrm{B}_i,\, \mathbf{p}_i) \in \cup_{m=1}^{\infty} \mathcal{X}^m \times \Delta^{C-1}\right\}_{i=1}^{n}$$

Requirements: the representations of the B_i need to be...

[21] Pérez-Mon et al., "Quantification using Permutation-Invariant Networks based on Histograms", 2024.

Symmetric learning

So far, we have assumed a training set $\mathrm{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{n}$, but what if we have

$$\mathrm{D}' = \left\{(\mathrm{B}_i,\, \mathbf{p}_i) \in \cup_{m=1}^{\infty} \mathcal{X}^m \times \Delta^{C-1}\right\}_{i=1}^{n}$$

Requirements: the representations of the B_i need to be...

- variable-size
- permutation-invariant

[21] Pérez-Mon et al., "Quantification using Permutation-Invariant Networks based on Histograms", 2024.

Symmetric learning

So far, we have assumed a training set $\mathrm{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$, but what if we have

$$\mathrm{D}' = \left\{(\mathrm{B}_i,\, \mathbf{p}_i) \in \cup_{m=1}^{\infty} \mathcal{X}^m \times \Delta^{C-1}\right\}_{i=1}^n$$

Requirements: the representations of the B_i need to be...

- variable-size
- permutation-invariant

HistNetQ:[21]

$$\ell(\theta) = \frac{1}{|\mathrm{D}'|} \sum_{(\mathrm{B},\, \mathbf{p}) \in \mathrm{D}'} \mathrm{RAE}(\lambda_\theta(\mathrm{B}),\, \mathbf{p})$$

where λ_θ a neural network with a differentiable histogram layer

[21] Pérez-Mon et al., “Quantification using Permutation-Invariant Networks based on Histograms”, 2024.

Ensembling

Idea: compute a central tendency (mean, median) of multiple predictions.

- multiple classifiers within different quantifiers (MC-MQ) or within duplicates of the same (MC-SQ)[22]

[22] Donyavi, Serapião, and Batista, "MC-SQ and MC-MQ: Ensembles for Multi-class Quantification", 2024.
[23] Pérez-Gállego, Quevedo, and Coz, "Using ensembles for problems with characterizable changes in data distribution: A case study on quantification", 2017.
[24] Pérez-Gállego et al., "Dynamic ensemble selection for quantification tasks", 2019.
[25] Janssen, "Ensembles für Quantification durch Konkatenieren von Quantifier-Modellen", 2024, supervised by M. Bunse and S. Buschjäger.

Ensembling

Idea: compute a central tendency (mean, median) of multiple predictions.

- multiple classifiers within different quantifiers (MC-MQ) or within duplicates of the same (MC-SQ)[22]
- for each member, use all data or use different subsamples[23]

[22] Donyavi, Serapião, and Batista, “MC-SQ and MC-MQ: Ensembles for Multi-class Quantification”, 2024.
[23] Pérez-Gállego, Quevedo, and Coz, “Using ensembles for problems with characterizable changes in data distribution: A case study on quantification”, 2017.
[24] Pérez-Gállego et al., “Dynamic ensemble selection for quantification tasks”, 2019.
[25] Janssen, “Ensembles für Quantification durch Konkatenieren von Quantifier-Modellen”, 2024, supervised by M. Bunse and S. Buschjäger.

Ensembling

Idea: compute a central tendency (mean, median) of multiple predictions.

- multiple classifiers within different quantifiers (MC-MQ) or within duplicates of the same (MC-SQ)[22]
- for each member, use all data or use different subsamples[23]
- maintain all members, select a subset at training time, or select a subset at prediction time[24]

[22] Donyavi, Serapião, and Batista, “MC-SQ and MC-MQ: Ensembles for Multi-class Quantification”, 2024.
[23] Pérez-Gállego, Quevedo, and Coz, “Using ensembles for problems with characterizable changes in data distribution: A case study on quantification”, 2017.
[24] Pérez-Gállego et al., “Dynamic ensemble selection for quantification tasks”, 2019.
[25] Janssen, “Ensembles für Quantification durch Konkatenieren von Quantifier-Modellen”, 2024, supervised by M. Bunse and S. Buschjäger.

Ensembling

Idea: compute a central tendency (mean, median) of multiple predictions.

- multiple classifiers within different quantifiers (MC-MQ) or within duplicates of the same (MC-SQ)[22]
- for each member, use all data or use different subsamples[23]
- maintain all members, select a subset at training time, or select a subset at prediction time[24]
- concatenate $\Phi(\mathbf{x}) = \big(\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}), \dots \Phi_E(\mathbf{x})\big)$ and minimize the loss once[25]

Open issue: under which circumstances are ensembles *provably* better than single models?

[22] Donyavi, Serapião, and Batista, "MC-SQ and MC-MQ: Ensembles for Multi-class Quantification", 2024.
[23] Pérez-Gállego, Quevedo, and Coz, "Using ensembles for problems with characterizable changes in data distribution: A case study on quantification", 2017.
[24] Pérez-Gállego et al., "Dynamic ensemble selection for quantification tasks", 2019.
[25] Janssen, "Ensembles für Quantification durch Konkatenieren von Quantifier-Modellen", 2024, supervised by M. Bunse and S. Buschjäger.

Conclusion:
supervised methods for quantification

Conclusion: supervised methods for quantification

Goal: under PPS, find a quantifier $\lambda : \cup_{m=1}^{\infty} \mathcal{X}^m \rightarrow \Delta^{C-1}$ that is

- Fisher-consistent
- has low estimation error // what about other settings than PPS?

Conclusion: supervised methods for quantification

Goal: under PPS, find a quantifier $\lambda : \cup_{m=1}^{\infty} \mathcal{X}^m \rightarrow \Delta^{C-1}$ that is

- Fisher-consistent
- has low estimation error // what about other settings than PPS?

Linear systems of equations: most algorithms can be expressed as solutions of some $\mathbf{q} = \mathbf{M}\mathbf{p}$

- choose a data representation $\Phi : \mathcal{X} \rightarrow \mathcal{Z}$ // what makes a good Φ?
- choose a loss function $\ell : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$ // what makes a good ℓ?
- choose an optimization algorithm (constrained or soft-max)

Conclusion: supervised methods for quantification

Goal: under PPS, find a quantifier $\lambda : \cup_{m=1}^{\infty} \mathcal{X}^m \rightarrow \Delta^{C-1}$ that is

- Fisher-consistent
- has low estimation error // what about other settings than PPS?

Linear systems of equations: most algorithms can be expressed as solutions of some $\mathbf{q} = \mathbf{M}\mathbf{p}$

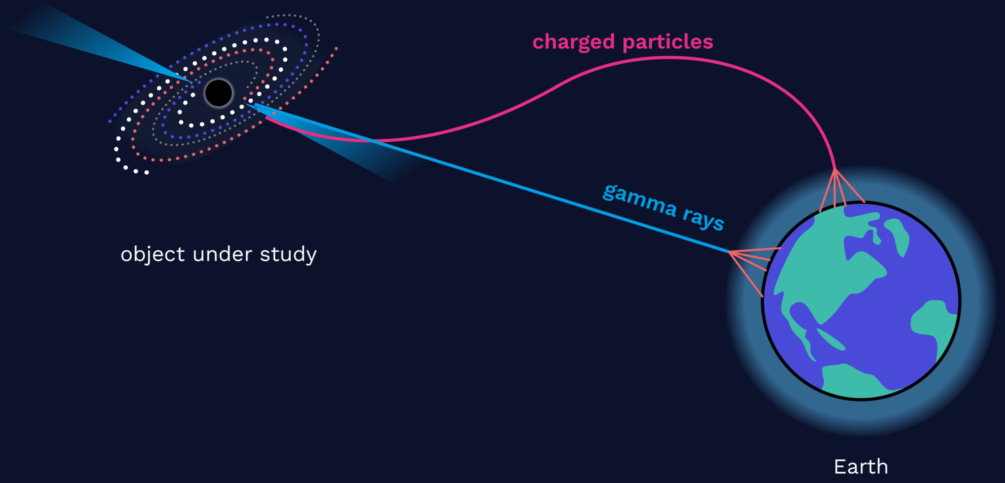
- choose a data representation $\Phi : \mathcal{X} \rightarrow \mathcal{Z}$ // what makes a good Φ?
- choose a loss function $\ell : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$ // what makes a good ℓ?
- choose an optimization algorithm (constrained or soft-max)

Notable other methods:

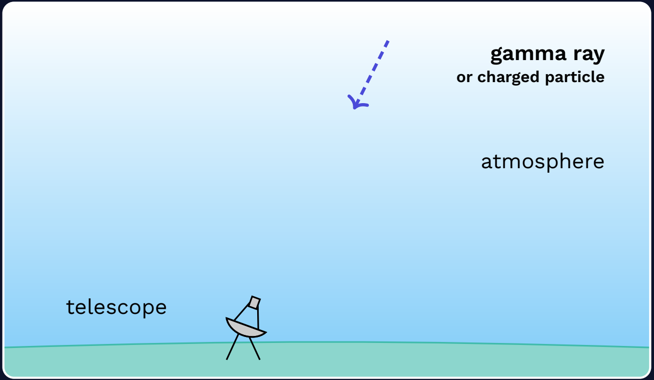
- SLD / EMQ
- maximum likelihood // can we bound its error?
- continuous representations
- symmetric learning
- ensembles // are they provably better?

Advanced topics for experimental physics

Example: astro-particle physics



Example: astro-particle physics



[26] **Fig:** Bockermann et al., "Online Analysis of High-Volume Data Streams in Astroparticle Physics", 2015

Example: astro-particle physics

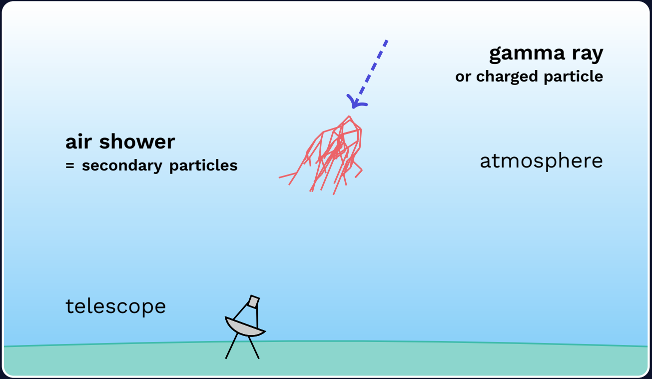


Fig: Bockermann et al., "Online Analysis of High-Volume Data Streams in Astroparticle Physics", 2015

Example: astro-particle physics

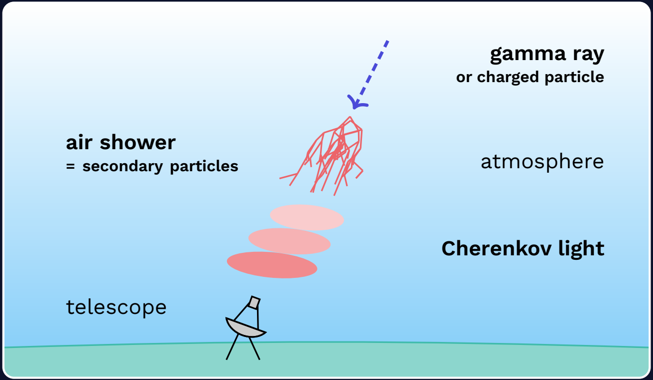
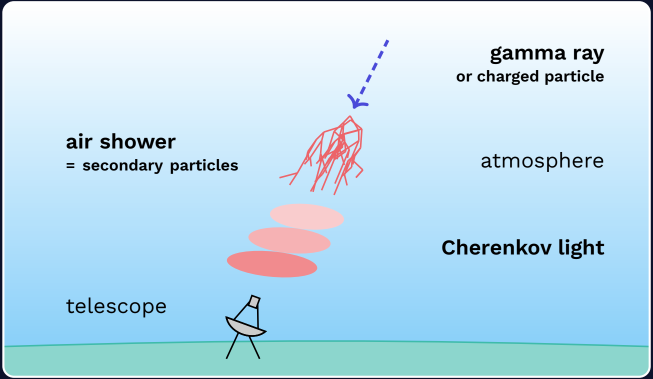


 Fig: Bockermann et al., “Online Analysis of High-Volume Data Streams in Astroparticle Physics”, 2015

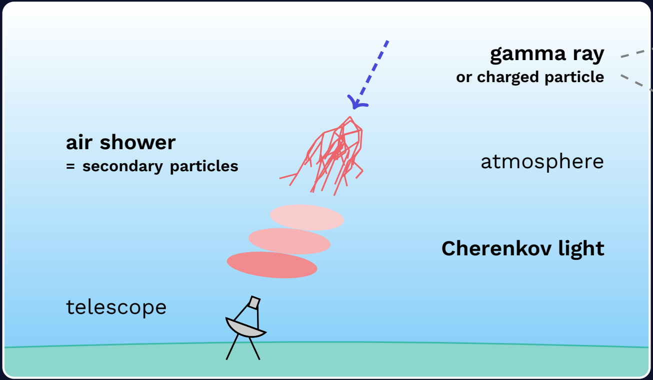
Example: astro-particle physics

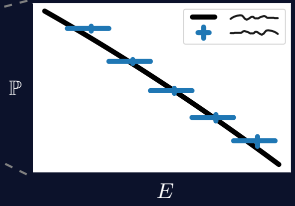


Preprocessing → Classification

Fig: Bockermann et al., “Online Analysis of High-Volume Data Streams in Astroparticle Physics”, 2015

Example: astro-particle physics





Preprocessing → Classification

26 **Fig:** Bockermann et al., “Online Analysis of High-Volume Data Streams in Astroparticle Physics”, 2015

Example: astro-particle physics

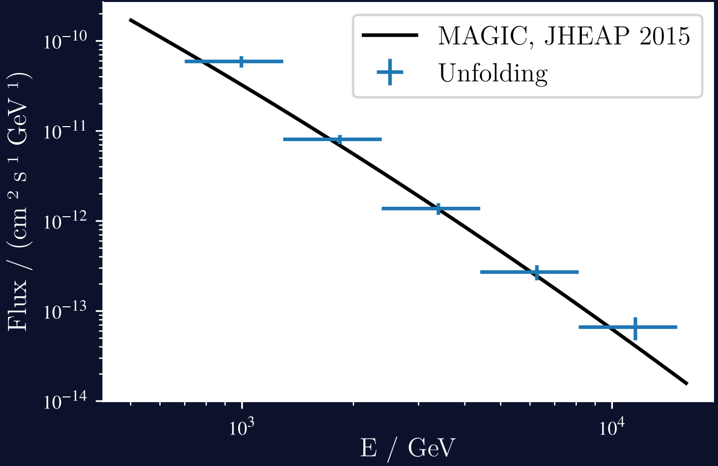


 Fig: Morik and Rhode, *Machine Learning under Resource Constraints – Discovery in Physics*, 2023

Advanced topics for experimental physics

Issues to be resolved:

- ordinality: $y_i \prec y_{i+1} \; \forall \, i \in \mathcal{Y}$ (to be covered through regularization for ordinal plausibility[28])

[28] Bunse et al., "Regularization-based Methods for Ordinal Quantification", 2024.
[29] Dussap, Blanchard, and Chérief-Abdellatif, "Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching", 2023.

Advanced topics for experimental physics

Issues to be resolved:

- ordinality: $y_i \prec y_{i+1} \;\forall\, i \in \mathcal{Y}$ (to be covered through regularization for ordinal plausibility[28])
- background: $\mathbb{Q}(\mathbf{x}) = \mathbb{Q}(\mathbf{x}, \varnothing) + \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (PPS with a noise class[29])

[28] Bunse et al., “Regularization-based Methods for Ordinal Quantification”, 2024.
[29] Dussap, Blanchard, and Chérief-Abdellatif, “Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching”, 2023.

Advanced topics for experimental physics

Issues to be resolved:

- ordinality: $y_i \prec y_{i+1} \; \forall \, i \in \mathcal{Y}$ (to be covered through regularization for ordinal plausibility[28])
- background: $\mathbb{Q}(\mathbf{x}) = \mathbb{Q}(\mathbf{x}, \varnothing) + \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (PPS with a noise class[29])
- class-conditional selection bias: $\mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_i) \neq \mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_j) \; \exists \, i \neq j$

[28] Bunse et al., “Regularization-based Methods for Ordinal Quantification”, 2024.
[29] Dussap, Blanchard, and Chérief-Abdellatif, “Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching”, 2023.

Advanced topics for experimental physics

Issues to be resolved:

- ordinality: $y_i \prec y_{i+1} \;\forall\, i \in \mathcal{Y}$ (to be covered through regularization for ordinal plausibility[28])
- background: $\mathbb{Q}(\mathbf{x}) = \mathbb{Q}(\mathbf{x}, \varnothing) + \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (PPS with a noise class[29])
- class-conditional selection bias: $\mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_i) \neq \mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_j) \;\exists\, i \neq j$
- changing environment: $\mathbb{Q}(\mathbf{x}, y) = \sum_{e \in \mathcal{E}} \mathbb{Q}(\mathbf{x}, y, e)$

[28] Bunse et al., “Regularization-based Methods for Ordinal Quantification”, 2024.
[29] Dussap, Blanchard, and Chérief-Abdellatif, “Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching”, 2023.

Advanced topics for experimental physics

Issues to be resolved:

- ordinality: $y_i \prec y_{i+1} \ \forall\, i \in \mathcal{Y}$ (to be covered through regularization for ordinal plausibility[28])
- background: $\mathbb{Q}(\mathbf{x}) = \mathbb{Q}(\mathbf{x}, \varnothing) + \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (PPS with a noise class[29])
- class-conditional selection bias: $\mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_i) \neq \mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_j) \ \exists\, i \neq j$
- changing environment: $\mathbb{Q}(\mathbf{x}, y) = \sum_{e \in \mathcal{E}} \mathbb{Q}(\mathbf{x}, y, e)$
- concept shift: $\mathbb{Q}(\mathbf{x} \mid y) \neq \mathbb{P}(\mathbf{x} \mid y)$ (in addition to PPS)

[28] Bunse et al., "Regularization-based Methods for Ordinal Quantification", 2024.
[29] Dussap, Blanchard, and Chérief-Abdellatif, "Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching", 2023.

Advanced topics for experimental physics

Issues to be resolved:

- ordinality: $y_i \prec y_{i+1} \; \forall \, i \in \mathcal{Y}$ (to be covered through regularization for ordinal plausibility[28])
- background: $\mathbb{Q}(\mathbf{x}) = \mathbb{Q}(\mathbf{x}, \varnothing) + \sum_{y=1}^{C} \mathbb{Q}(\mathbf{x}, y)$ (PPS with a noise class[29])
- class-conditional selection bias: $\mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_i) \neq \mathbb{Q}(\mathbf{x} \in \mathrm{B} \mid y_j) \; \exists \, i \neq j$
- changing environment: $\mathbb{Q}(\mathbf{x}, y) = \sum_{e \in \mathcal{E}} \mathbb{Q}(\mathbf{x}, y, e)$
- concept shift: $\mathbb{Q}(\mathbf{x} \mid y) \neq \mathbb{P}(\mathbf{x} \mid y)$ (in addition to PPS)
- inspect contributions of individual data items $\mathbf{x} \in \mathrm{B}$ to $\lambda(\mathrm{B})$ (data selection, human in the loop)

[28] Bunse et al., "Regularization-based Methods for Ordinal Quantification", 2024.
[29] Dussap, Blanchard, and Chérief-Abdellatif, "Label Shift Quantification with Robustness Guarantees via Distribution Feature Matching", 2023.