Unification of Algorithms for Quantification and Unfolding*

Mirko Bunse[0000−0002−5515−6278] (✉) and Katharina Morik[0000−0003−1153−5986]

Artificial Intelligence Unit, TU Dortmund University, 44227 Dortmund, Germany
{firstname.lastname}@cs.tu-dortmund.de

Abstract. Quantification is the supervised learning task of predicting the prevalence values of classes in a data sample. Physics literature knows the same task under a different name: unfolding. However, the literature on quantification and the literature on unfolding are largely disconnected from each other, likely due to an interdisciplinary gap. We bridge this gap by proposing a common framework that integrates algorithms from both fields in a unified form. Instantiations of our framework differ from each other in terms of the loss functions, the regularizers, and the feature transformations they employ.

Keywords: Quantification · Unfolding · Classification · Experimental physics · Machine learning.

1 Introduction

Many applications of supervised learning require a prediction of the *distribution* of the target quantity, as exhibited by some data sample. In these applications, predictions for individual data instances are only secondary; they are issued as a means from which the distribution can be reconstructed. Examples of such applications are text sentiment analyses [11], technical support log analyses [10], social sciences [15], the reconstruction of energy spectra in astroparticle physics [6], and several other areas.

Supervised learning for the prediction of target distributions is known as *quantification learning* [10,12]. Within experimental physics, however, the same problem is called *unfolding* [2,14,7] or *deconvolution* [6]. As of today, the literature from quantification research and the literature from unfolding research are largely disconnected from each other, despite their substantial similarities in terms of their problem statements and their solutions.

Contributions We propose a common framework for algorithms that stem from quantification literature and from unfolding literature. This framework reveals several similarities between existing methods from the two research fields. Moreover, it paves the way for strengthening interdisciplinary efforts on the subject. Our presentation completes a similar unification attempt by Firat [9] in terms

* This paper is a slightly modified resubmission of a recent publication by us [5]

of i) taking unfolding algorithms into consideration and ii) giving formal proofs about the correctness of our framework. Our reusable implementation of all methods is available online.[1]

Sec. 2 details unfolding algorithms within our unifying framework. In Sec. 3, we integrate algorithms from quantification literature. We summarize our findings in Tab. 1 before Sec. 4 concludes.

2 Unfolding

A frequent objective in experimental physics is to estimate the spectrum of a physical quantity that cannot be measured directly. In this case, the spectrum needs to be reconstructed from correlated quantities which are measured instead.

To this end, assume that we can measure the distribution $q(\vec{x}) = \mathbb{P}(X = \vec{x})$ of some quantity X within a sample. Moreover, let the measurement process be characterized through the conditional probabilities $M(\vec{x} \mid y_c) = \mathbb{P}(X = \vec{x} \mid Y_c = y)$ of measuring some $\vec{x} \in \mathcal{X}$ when the relevant quantity has the (possibly continuous) value $y \in \mathcal{Y}_c$. The objective of any unfolding algorithm is then to reconstruct the relevant distribution $p(y) = \mathbb{P}(Y_c = y)$ from the distributions q and M, according to the integral

$$q(\vec{x}) \;=\; \int_{\mathcal{Y}_c} M(\vec{x} \mid y) \cdot p(y) \; \mathrm{d}y. \tag{1}$$

The estimation of $p(y)$ from data is enabled through the discretization of Eq. 1. In case of a continuous target interval $\mathcal{Y}_c = [a, b)$, we first need to map each continuous label to a discrete class index $\mathcal{Y} = \{1, \ldots, C\}$. For instance, the estimation of an energy spectrum requires a binning of the interval $\mathcal{Y}_c$ into C bins [3,7]. We proceed similarly with the feature space $\mathcal{X} \subseteq \mathbb{R}^d$, in mapping it to a discrete feature representation $f(\vec{x}) \in \{1, \ldots, F\}$, which is still to be defined for each unfolding algorithm in particular.

The discretization of y and $\vec{x}$ gives rise to a straightforward representation of distributions in terms of histograms. Consider a data sample $D = \{(\vec{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq N\}$ in which the classes y_i are not observed. Estimating the quantities from Eq. 1 in terms of histograms

$$\vec{p} \;=\; \frac{1}{N} \sum_{i=1}^{N} \delta_{y_i}, \qquad \vec{q} \;=\; \frac{1}{N} \sum_{i=1}^{N} \delta_{f(\vec{x}_i)}, \qquad [\delta_j]_k \;=\; \begin{cases} 1 & \text{if } j = k, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

leads to the system of linear equations

$$\vec{q} \;=\; M \cdot \vec{p}, \tag{3}$$

where the transfer matrix $M \in \mathbb{R}^{C \times F}$ is estimated by counting and normalizing the co-occurrences of labels y and transformed features $f(\vec{x})$ in a training set. Advanced algorithms are required to estimate $\vec{p}$ because a direct solution $M^{-1}\vec{q}$ is not guaranteed to exist.

[1] https://github.com/mirkobunse/QUnfold.jl

A Common Framework for Unfolding and Quantification Unfolding algorithms solve Eq. 3 for $\vec{p}$, a histogram estimate of the (continuous) distribution $p(y)$ from Eq. 1. However, M is not invertible in general. A general regularized solution for the unfolding / quantification problem, with a regularization strength $\tau \geq 0$, is

$$\vec{p}^* \;=\; \underset{\vec{p}\geq 0 \,\text{s.t.}\, \mathbb{1}^\top \vec{p}=1}{\arg\min} \quad \mathcal{L}(\vec{p};\, \vec{q}, M) + \tau \cdot r(\vec{p}), \tag{4}$$

where the loss function $\mathcal{L} : \mathbb{R}^C \to \mathbb{R}$ and the regularization function $r : \mathbb{R}^C \to \mathbb{R}$ are still to be defined for each particular unfolding / quantification method. The constraints in Eq. 4 ensure that $\vec{p}^*$ represents a valid probability density. Our framework extends the one by Firat [9] with regularization functions $r(\vec{p})$.

Adhering to this framework are the most important unfolding algorithms, namely

Regularized Unfolding (RUN) [3,2] RUN models the likelihood of solutions in terms of Poisson-distributed counts. Namely, we observe a histogram of counts $\bar{q} = N \cdot \vec{q} \in \mathbb{N}^F$, each element of which is modelled as being Poisson-distributed with the rate $\lambda_i = [M\bar{p}]_i$. This modelling gives rise to the negative log-likelihood function

$$\mathcal{L}^{\text{RUN}}(\vec{p};\, \vec{q}, M) \;=\; \sum_{i=1}^{F} [M\bar{p}]_i - \bar{q}_i \ln [M\bar{p}]_i, \tag{5}$$

which RUN minimizes.
To ensure smooth solutions, RUN employs Tikhonov regularization. The Tikhonov matrix $T \in \mathbb{R}^{C\times C}$ is defined such that

$$r^{\text{RUN}}(\vec{p}) \;=\; \frac{1}{2}\left(T\vec{p}\right)^2 \;=\; \frac{1}{2}\sum_{i=2}^{C-1}\left([\vec{p}]_{i-1} - 2[\vec{p}]_i + [\vec{p}]_{i+1}\right)^2. \tag{6}$$

Unfolding via Singular Value Decomposition (SVD) [14] This method employs the regularizer from Eq. 6 with a least squares loss

$$\mathcal{L}^{\text{SVD}}(\vec{p};\, \vec{q}, M) \;=\; \left\| \frac{\vec{q} - M\vec{p}}{\vec{w}} \right\|_2^2, \tag{7}$$

which is weighted by a vector $\vec{w} \in \mathbb{R}^F$. For instance, a Poisson model can be realized through Poisson variances $\vec{w} = \sqrt{\bar{q}}$.

Iterative Bayesian Unfolding (IBU) [8,7] IBU revolves around an expectation maximization approach. Starting from a prior $\vec{p}^{(0)}$, it repeatedly updates the estimate $\vec{p}^{(k)}$ according to Bayes' theorem

$$[\vec{p}^{(k)}]_i \;=\; \sum_{j=1}^{F} \frac{[M]_{ij}[\vec{p}^{(k-1)}]_i}{\sum_{i'=1}^{C}[M]_{i'j}[\vec{p}^{(k-1)}]_{i'}}\,[\vec{q}]_j. \tag{8}$$

IBU implements regularization in two ways. First, through early stopping in combination with a smooth prior. For instance, starting from $\vec{p}^{(0)} = \frac{1}{C}$ and stopping before Eq. 8 converges will maintain the smoothness of $\vec{p}^{(0)}$ to some degree. Second, the intermediate estimates $\vec{p}^{(k)}$ are smoothed with a low-order polynomial.

The above algorithms do not specify the feature transformation $f(\vec{x}) \in \{1, \ldots, F\}$ through which $\vec{q}$ and M are defined; they solely focus on the estimation of $\vec{p}$ from any given $\vec{q}$ and M. In this sense, these algorithms are open to any feature transformation. Physicists have proposed

- to bin a single feature that is well correlated with the target quantity [2],
- to cluster the features in order to map instances to cluster indices [6],
- or to optimally partition the feature space by means of decision trees [4]

in order to obtain histograms $\vec{q}$ which represent the data sample.

3 Quantification

In the following, we show that several algorithms from quantification literature are indeed instances of the unified framework we have presented above. A summary of these findings is displayed in Tab. 1. We prove the correctness of our unifying notation in the Appendix.

Table 1. Algorithms for unfolding and quantification within the framework of Eq. 4.

	loss function $\mathcal{L}$	regularizer r	feature transformation f
RUN [3,2]	$\sum_{i=1}^{d} [M\bar{p}]_i - \bar{q}_i \ln[M\bar{p}]_i$	$\frac{1}{2}\,(T\vec{p})^2$	not specified / any
SVD [14]	$\left\lVert \frac{\vec{q}-M\vec{p}}{\vec{w}} \right\rVert_2^2$	$\frac{1}{2}\,(T\vec{p})^2$	not specified / any
IBU [8,7]	expectation maximization	smoothing	not specified / any
ACC [10,15]	$\lVert\vec{q} - M\vec{p}\rVert_2^2$	none	$\delta_{\arg\max_i [h(\vec{x})]_i}$
PACC [1]	$\lVert\vec{q} - M\vec{p}\rVert_2^2$	none	$h(\vec{x})$
ReadMe [15]	$\lVert\vec{q} - M\vec{p}\rVert_2^2$	none	$\delta_{\vec{x}=(X_1,\ldots,X_{2^d})}$
HDx [13]	$\frac{1}{d}\sum_{i=1}^{d} \mathrm{HD}_i(\vec{q},\, M\vec{p})$	none	$\left(\delta_{b(\vec{x};1)}, \ldots, \delta_{b(\vec{x};d)}\right)$
HDy [13]	$\frac{1}{d}\sum_{i=1}^{d} \mathrm{HD}_i(\vec{q},\, M\vec{p})$	none	$\left(\delta_{b(h(\vec{x});1)}, \ldots, \delta_{b(h(\vec{x});C)}\right)$
CC [10]	none (assume $M = \mathbb{I}$)	none	$\delta_{\arg\max_i [h(\vec{x})]_i}$
PCC [1]	none (assume $M = \mathbb{I}$)	none	$h(\vec{x})$

Namely, our framework from Eq. 4 accommodates the following algorithms:

Adjusted Classify and Count (ACC) [10,15] Hopkins and King [15] present a method that extends the binary adjustment by Forman [10] to multi-class settings. Their extension represents a data sample as the counts of classification outcomes $\arg\max_i [h(\vec{x})]_i$, as returned by a multi-class classifier $h : \mathcal{X} \rightarrow \mathbb{R}^C$. In this case, M is simply the normalized confusion matrix of h, as estimated on held-out training data. Hopkins and King [15] propose to solve Eq. 3 via constrained least squares regression, hence

$$\mathcal{L}^{\mathrm{ACC}}(\vec{p};\, \vec{q}, M) \;=\; \|\vec{q} - M\vec{p}\,\|_2^2 \tag{9}$$

and no regularization is employed.

Others [17,16] have proposed to solve Eq. (3) through matrix inversion,

$$\vec{p}^{\,\mathrm{inv}} = M^{-1}\vec{q}.$$

However, there is no guarantee that M is indeed invertible. Therefore, $\vec{p}^{\,\mathrm{inv}}$ might be undefined and the method by Hopkins and King [15] should be the prefered multi-class version of ACC.

Probabilistic ACC (PACC) [1] This method employs the same adjustment as ACC, hence the same loss. However, PACC averages soft classifications $h(\vec{x}) \in \mathbb{R}^C$ instead of counting the crisp outcomes $\arg\max_i [h(\vec{x})]_i$.

ReadMe [15] Building on the multi-class version of ACC, ReadMe employs the loss function from Eq. 9. However, ReadMe transforms the features in a unique way that is motivated in text mining. In this application area, instances $\vec{x}$ are often represented as bags of words, i.e. by sparse indicator vectors $\{0,1\}^d$ for a vocabulary of size d. In ReadMe, $\vec{q}$ is a histogram over all 2^d possible incarnations X_i of these indicator vectors, i.e.

$$\begin{aligned} f^{\mathrm{ReadMe}}(\vec{x}) \;&=\; \delta_{\vec{x}=(X_1,\ldots,X_{2^d})}, \\ \text{where} \quad [\delta_{a=(a_1,\ldots,a_n)}]_i \;&=\; \begin{cases} 1 & \text{if } a = a_i, \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{10}$$

Since such a representation is only feasible with small d, ReadMe produces multiple estimates, each of which employs a different and small random selection of words. Finally, all of these estimates are averaged.

HDx [13] In this method, each feature is separately binned and a data sample is represented as a concatenation of all feature-wise histograms

$$f(\vec{x}) \;=\; (\delta_{b(\vec{x};1)}, \ldots, \delta_{b(\vec{x};d)}), \tag{11}$$

where $b(\vec{x}; i)$ is a binning function which maps the feature value $[\vec{x}]_i$ to the corresponding bin index $\{1, \ldots, B_i\}$.

The loss is measured as the average of feature-wise Hellinger distances,

$$\mathcal{L}(\vec{p};\, M, \vec{q}) \;=\; \frac{1}{d} \sum_{i=1}^{d} \mathrm{HD}_i(\vec{q},\, M\vec{p}), \tag{12}$$

$$\text{where} \quad \mathrm{HD}_i(\vec{q},\, M\vec{p}) \;=\; \sqrt{\sum_{j=1+\sum_{k=1}^{i-1} B_k}^{\sum_{k=1}^{i} B_k} \left(\sqrt{[\vec{q}]_j} - \sqrt{[M\vec{p}]_j} \right)^2} \,. \tag{13}$$

HDy [13] Originally, HDy has been proposed for binary quantification only. However, we can easily extend the method to the multi-class setting. In this setting, HDy replaces the separated binning of features $b(\vec{x}, i)$ in HDx with a separated binning of class-wise classifier outputs $b(h(\vec{x}), i)$. All other aspects of HDx are maintained.

(Probabilistic) Classify and Count (PCC/CC) [10,1] We also conceive these non-adjusted methods, which simply return $\vec{q}$ as their estimates for $\vec{p}$, as instances of our framework. Strictly speaking, CC and PCC do not require the minimization of a loss function. More loosely speaking, however, their disregard of M can be understood as the assumption of a perfect classifier, so that $M = \mathbb{I}$ is the identity matrix. Under this assumption, the least squares loss from Eq. 9 leads to the estimate $\vec{p}^{\,\mathrm{CC}} = \vec{q}$ and we can understand this estimate as an instance of Eq. 4.

Regarding $f(\vec{x})$, CC employs the feature transformation of ACC and PCC employs the feature transformation of PACC.

4 Conclusion and Outlook

We have presented the unfolding algorithms RUN, SVD, and IBU and the quantification algorithms ACC, PACC, ReadMe, HDx, HDy, CC, and PCC within a common framework. These algorithms differ in terms of the loss functions, the regularizers, and the feature transformations they employ.

Our unification demonstrates the similarity between the problems that are approached in unfolding and in quantification literature. Due to this similarity, we conceive adaptations of quantification algorithms to physics problems as a valuable endeavor for future work. Likewise, we suggest to adapt unfolding algorithms to problems outside of physics.

References

1. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Quantification via probability estimators. In: Int. Conf. on Data Mining. pp. 737–742. IEEE (2010). https://doi.org/10.1109/ICDM.2010.75
2. Blobel, V.: Unfolding methods in high-energy physics experiments. Tech. rep., CERN (1985). https://doi.org/10.5170/CERN-1985-009.88
3. Blobel, V.: An unfolding method for high energy physics experiments. In: Adv. Stat. Tech. in Part. Phys. pp. 258–267 (2002)

4. Börner, M., Hoinka, T., Meier, M., Menne, T., Rhode, W., Morik, K.: Measurement/simulation mismatches and multivariate data discretization in the machine learning era. In: Astron. Data Anal. Softw. and Syst. pp. 431–434. ASP Conference Series, Astronomical Society of the Pacific (2017)
5. Bunse, M.: Unification of algorithms for quantification and unfolding. In: Workshop on Mach. Learn. for Astropart. Phys. and Astron. Gesellschaft für Informatik e.V. (2022), to appear
6. Bunse, M., Piatkowski, N., Morik, K., Ruhe, T., Rhode, W.: Unification of deconvolution algorithms for Cherenkov astronomy. In: Int. Conf. on Data Sci. and Adv. Anal. pp. 21–30. IEEE (2018). https://doi.org/10.1109/DSAA.2018.00012
7. D'Agostini, G.: A multidimensional unfolding method based on Bayes' theorem. Nucl. Instr. and Meth. in Phys. Res. Sect. A **362**(2-3), 487–498 (1995)
8. D'Agostini, G.: Improved iterative Bayesian unfolding. arXiv:abs/1010.0632 (2010)
9. Firat, A.: Unified framework for quantification. arXiv:abs/1606.00868 (2016)
10. Forman, G.: Quantifying counts and costs via classification. Data Mining and Knowl. Discov. **17**(2), 164–206 (2008). https://doi.org/10.1007/s10618-008-0097-y
11. Gao, W., Sebastiani, F.: From classification to quantification in tweet sentiment analysis. Soc. Netw. Anal. and Mining **6**(19), 1–22 (2016)
12. González, P., Castaño, A., Chawla, N.V., del Coz, J.J.: A review on quantification learning. ACM Comput. Surv. **50**(5), 74:1–74:40 (2017). https://doi.org/10.1145/3117807
13. González-Castro, V., Alaíz-Rodríguez, R., Alegre, E.: Class distribution estimation based on the Hellinger distance. Inf. Sci. **218**, 146–164 (2013). https://doi.org/10.1016/j.ins.2012.05.028
14. Hoecker, A., Kartvelishvili, V.: SVD approach to data unfolding. Nucl. Instr. and Meth. in Phys. Res. Sect. A **372**(3), 469–481 (1996)
15. Hopkins, D.J., King, G.: A method of automated nonparametric content analysis for social science. Amer. J. of Polit. Sci. **54**(1), 229–247 (2010). https://doi.org/10.1111/j.1540-5907.2009.00428.x
16. McLachlan, G.J.: Discriminant analysis and statistical pattern recognition. Wiley (1992)
17. Vucetic, S., Obradovic, Z.: Classification on data with biased class distribution. In: Eur. Conf. on Mach. Learn. pp. 527–538. Springer (2001). https://doi.org/10.1007/3-540-44795-4_45

A Proofs

We now detail the mapping from the original algorithms to our unified notation, to formally prove that our framework is consistent with the original proposals.

Regularized Unfolding (RUN) The loss function we present in Eq. 5 is a verbatim statement by Blobel [2, Eqs. (2.29), and (2.26)]. The original algorithm treats the elements of $\vec{p}$ as B-spline coefficients; however, a more recent version by the same author [3] employs histograms, which are consistent with our Eq. 2. Due to this change "the second derivative in bin j is proportional to $x_{j-1} - 2x_j + x_{j+1}$" [3], where $x_i = [\vec{p}]_i$. This derivative defines the regularization term from Eq. 6. □

Unfolding via Singular Value Decomposition (SVD) The loss function we present in Eq. 7 and the regularization term from Eq. 6 are verbatim statements by Hoecker and Kartvelishvili [14, Eqs. (29), (37), and (38)]. □

Iterative Bayesian Unfolding (IBU) D'Agostini [7, Eqs. (3), and (4)] estimates $[\vec{p}^{(k)}]_i$ as

$$\frac{1}{\epsilon_i} \sum_{j=1}^{n_E} n(E_j) \cdot \frac{P(E_j \mid C_i) \cdot P_0(C_i)}{\sum_{l=1}^{n_C} P(E_j \mid C_l) \cdot P_0(C_l)},$$

where we identify our notation as $F = n_E$, $C = n_C$, $M_{ij} = P(E_j \mid C_i)$, and $[\vec{p}^{(k-1)}]_i = P_0(C_i)$. In the original algorithm, $n(E_j) \in \mathbb{N}$ is the count observed in the j-th bin, i.e. $n(E_j) = N \cdot [\vec{q}]_j$. Moreover, $\epsilon_i > 0$ is an acceptance factor, which models the probability that an existing instance of class i is indeed part of the sample—and not hidden due to measurement complications. Setting $\epsilon_i = N$, we obtain $[\vec{q}]_j = \frac{n(E_j)}{\epsilon_i}$, which is consistent with our Eq. 8.

For regularization, D'Agostini [7] proposes to "smooth the results of the unfolding before feeding them in the next step", for instance "by a polynomial fit of 3^{rd} degree" or by another low-order polynomial. □

Adjusted Classify and Count (ACC) Hopkins and King [15, Eq. (4)] marginalize over the true labels $D \in \{1, \ldots, J\}$ to yield the distribution of class predictions $\widehat{D}$,

$$P(\widehat{D} = j) \ = \ \sum_{j'=1}^{J} P(\widehat{D} = j \mid D = j')P(D = j).$$

The authors note that "this expression represents a set of J equations [...] that can be solved for the J elements in $P(D)$". Accordingly, we identify our notation as $\vec{p} = P(D)$, $\vec{q} = P(\widehat{D})$, and $M = P(\widehat{D} \mid D)$ in their presentation. To solve this set of equations, the authors propose a "standard constrained least squares to ensure that elements of $P(D)$ are each in [0,1] and collectively sum up to 1". This proposal defines the least squares loss from Eq. 9 and matches our constraints in Eq. 4. □

Note that Hopkins and King have developed their method independently of Forman's binary ACC. However, the basis of their work is precisely the adjustment by Forman [10, Eq. (1)], as can be seen in Hopkins and King [15, Eq. (3)]. Therefore, we call their method "multi-class ACC".

The other multi-class extension of ACC, $\vec{p}^{\,\mathrm{inv}}$, is presented in McLachlan [16, Eq. (2.3.4)] and in Vucetic and Obradovic [17, Eq. (3)].

Probabilistic ACC (PACC) The essential proposal by Bella et al. [1] is to replace hard classifications $\arg\max_i [h(\vec{x})]_i$ with probabilistic ones $h(\vec{x}) \in \mathbb{R}^C$; their adjustment is the same as in binary ACC. By applying this proposal to multi-class ACC [15], we obtain a multi-class PACC which employs the loss from Eq. 9. □

ReadMe Building on their multi-class design of ACC, Hopkins and King [15, Eq. (6)] set up a matrix equation $P(\mathbf{S}) = P(\mathbf{S} \mid D)P(D)$, which maps to our notation as $\vec{q} = P(\mathbf{S}) \in \mathbb{R}^{2^d}$, $M = P(\mathbf{S} \mid D) \in \mathbb{R}^{2^d \times C}$, and $\vec{p} = P(D) \in \mathbb{R}^C$. The authors note that "$P(\mathbf{S})$ is the probability of each of the 2^K possible word stem profiles" with $K = d$ being the number of word stems. To estimate this probability, "we merely compute the proportion of documents observed with each pattern of word profiles". This computation leads to a histogram

$$\vec{q} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\vec{x}_i=(X_1,\ldots,X_{2^d})},$$

which is consistent with our Eqs. 2 and 10. □

HDx González-Castro et al. [13, Eq. (9)] minimize the average of feature-wise Hellinger distances, as we have stated in Eq. 12. They present the distance with respect to a single feature j, binned into b bins, as

$$\sqrt{\sum_{i=1}^{b}\left(\sqrt{\frac{|V_{j,i}|}{|V|}}-\sqrt{\frac{|U_{j,i}|}{|U|}}\right)^2},$$

where $|U|$ is the total number of instances and $|U_{j,i}|$ is the number of instances whose feature j is mapped to the i-th bin [13, Eq. (10)]. $|V|$ and $|V_{j,i}|$ are the numbers of instances that are to be expected under class prevalence values $\vec{p}$, hence

$$\frac{|V_{j,i}|}{|V|} = [M\vec{p}]_{i+\sum_{k=1}^{j-1}B_k},$$

where $\sum_{k=1}^{j-1} B_k$ is the offset of the histogram of feature j within our concatenation of feature-wise histograms. Using the product $M\vec{p}$ at this point is consistent with the binary conception that is proposed by González-Castro et al. [13, Eq. (12)]. □

HDy The original HDy [13, Eqs. (13) and (14)] only addresses binary quantification. For this case, however, the only change with respect to HDx is that HDy employs soft classifier outputs $h(\vec{x})$ instead of features $\vec{x}$. A straightforward extension to the multi-class setting is therefore to bin the class-wise outputs $[h(\vec{x})]_i$ separately, as HDx does in case of features and as we propose in our presentation of HDy. □

(Probabilistic) Classify and Count (PCC/CC) Let $M = \mathbb{I}$. Recognize that the global minimum of the least squares loss,

$$\min_{\vec{p}} \|\vec{q} - M\vec{p}\|_2^2 \; = \; 0,$$

is now attained if and only if $\vec{p} = \vec{q}$. Therefore, under the assumption $M = \mathbb{I}$, the unique minimizer of the least squares loss is $\vec{q}$. In this sense, PCC and CC are proper instances of our framework. □