On the Normalization of Confusion Matrices: Methods and Geometric Interpretations

Johan Erbani
LIRIS, INSA Lyon,
CNRS (UMR 5205), France

Sonia Ben Mokhtar
LIRIS, INSA Lyon,
CNRS (UMR 5205), France

Pierre-Edouard Portier
Caisse d'Epargne Rhône Alpes,
Tour Incity, Lyon, France

Elöd Egyed-Zsigmond
LIRIS, INSA Lyon,
CNRS (UMR 5205), France

Diana Nurbakova
LIRIS, INSA Lyon,
CNRS (UMR 5205), France

Abstract

The confusion matrix is a standard tool for evaluating classifiers, providing a detailed view of model errors. In heterogeneous settings, its entries are influenced by two main factors: class similarity, reflecting how easily the model confuses certain classes, and distribution bias, stemming from imbalanced training or test distributions. Because confusion matrix values jointly reflect both factors, it is difficult to disentangle their individual effects. To address this issue, we introduce bi-normalization via Iterative Proportional Fitting, a generalization of row and column normalization. Unlike standard approaches, this method recovers the underlying structure of class similarity. By disentangling error sources, it enables a more precise diagnosis of model behavior and facilitates classifier improvement. We further establish connections between normalization, importance sampling, and class representations in the model's latent space, thus offering a clearer interpretation of normalization schemes. Our implementation is publicly available[1].

1 INTRODUCTION

The confusion matrix is a key tool for understanding, evaluating, and improving a model's behavior (Krstinić et al., 2020, 2024; Görtler et al., 2022; Erbani et al., 2024). It is a $C \times C$ square matrix that provides a detailed view of model errors, where C is the number of classes. Each entry (i, j) counts the number of instances with true label i that the model predicted as j, for all $i, j \in \{1, \ldots, C\}$. It is typically built from the test set (Sammut and Webb, 2011): starting from a zero matrix, the entry (i, j) is incremented by 1 whenever a sample with label i is predicted as j.

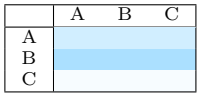
Two main factors shape confusion matrix values: (i) class similarity and (ii) distribution bias (Erbani et al., 2024), as illustrated in Figure 1. (i) Class similarity refers to confusion between similar classes—the more alike two classes i and j are, the more frequently they are misclassified as each other, increasing values in entries (i, j) and/or (j, i) (Subfigure 1c). While this source of error is intrinsic to the classes themselves, it also reflects inherent model properties: regardless of training, these similarities persist. (ii) Distribution bias refers to errors caused by class imbalance in the predictions or test set. Models trained on imbalanced data tend to overpredict majority classes and underpredict minority ones (Buda et al., 2018; Leevy et al., 2018), inflating the columns of majority classes (Subfigure 1b). Similarly, imbalance in the test set affects the rows: majority classes have higher counts, while minority classes have lower ones (Subfigure 1a). This source of confusion is accidental and varies with training.

Understanding the sources of classification errors enables targeted improvements. For example, errors from class similarity can be reduced by enhancing preprocessing to better distinguish similar classes or by augmenting the training set with more examples of those classes (Ghosh et al., 2024; Tian et al., 2024; Temraz and Keane, 2022; Aggarwal et al., 2021). Overprediction errors can be reduced by adjusting the

[1] https://github.com/JohanErbani/Bi-Normalization

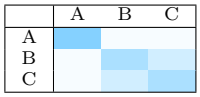
Figure 1: How class similarity and distribution bias affect confusion matrices. The first three subfigures isolate each factor, while the last combines them: (a) Test set imbalance, with B as the majority class and C as the minority; (b) Prediction imbalance, with C over-predicted and B under-predicted; (c) Class similarities; (d) Combined influence of all factors. Darker colors indicate higher values.

(a) Test set: Imbalance, Predictions: Balance, Class similarity: No

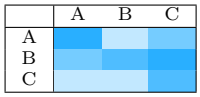


(b) Test set: Balance, Predictions: Imbalance, Class similarity: No

(c) Test set: Balance, Predictions: Balance, Class similarity: Yes



(d) Test set: Imbalance, Predictions: Imbalance, Class similarity: Yes



loss function, such as reweighting class contributions to reduce the influence of frequent classes (Fernando and Tsokos, 2021; Deepak and Ameer, 2023; Chamseddine et al., 2022; Wu et al., 2022). However, confusion matrix values reflect a mix of both class similarity and distribution bias (Subfigure 1d), making it difficult to disentangle their individual contributions.

In practice, machine learning workflows often apply row, column, or all normalization to confusion matrices:

$$\mathrm{row}(M)_{ij} = \frac{M_{ij}}{M_{i+}},\ \mathrm{col}(M)_{ij} = \frac{M_{ij}}{M_{+j}},\ \mathrm{all}(M)_{ij} = \frac{M_{ij}}{M_{++}},$$

where M is a confusion matrix, $M_{i+} = \sum_j M_{ij}$, $M_{+j} = \sum_i M_{ij}$, and $M_{++} = \sum_{i,j} M_{ij}$. Many papers present several normalized matrices, as each normalization highlights different aspects of the model behavior (Görtler et al., 2022). However, in imbalanced settings, these normalizations may fail to capture class similarities, as shown in our experiments.

Class similarities in the confusion matrix become visible when all rows and columns have the same total, which typically occurs when both the test and training sets are balanced. When both the test set and predictions are imbalanced, a double normalization of rows and columns at the same time is needed to reveal these similarities. This normalization can be achieved using the Iterative Proportional Fitting (IPF) procedure—also known as the Sinkhorn-Knopp algorithm, biproportional fitting, the RAS method, or simply matrix scaling (Idel, 2016). IPF is widely used in statistics, economics, and computer science (Idel, 2016). We refer to the normalization produced by IPF as *bi-normalization*, denoted by bi. This normalization yields a confusion matrix in which each row and each column sums to 1: $\mathrm{bi}(M)_{i+} = \mathrm{bi}(M)_{+j} = 1$.

Standard normalizations have a probabilistic interpretation: row, col, and all correspond to $\mathbb{P}(\hat{Y} = j \mid Y = i)$, $\mathbb{P}(Y = i \mid \hat{Y} = j)$, and $\mathbb{P}(Y = i, \hat{Y} = j)$, respectively (Görtler et al., 2022), where Y and $\hat{Y}$ denote the random variables of labels and predictions. We introduce two complementary perspectives that provide further insight into matrix normalization.

Importance sampling. Normalization can be viewed as applying an importance sampling strategy: it reweights each label–prediction pair to match a desired distribution—for instance, balancing labels via row normalization.

Model representation. Normalization is also related to how the model encodes class representations. We show empirically that the overlap of class clusters in the latent space corresponds—depending on how it is measured—to a particular form of normalized confusion matrix.

Standard and bi normalizations can thus be interpreted through these perspectives, providing additional probabilistic and geometric meaning.

Our main contributions are:

- We show that bi-normalization generalizes row and column normalizations while satisfying key expected properties.
- We establish connections between confusion matrix normalizations—row, col, all, and bi—and both importance sampling and class representations in the model's latent space, offering clearer interpretations of normalized confusion matrices.
- We provide empirical evidence that bi-normalization reveals class similarities more effectively than other normalization methods and supports our geometric interpretation.

2 RELATED WORK

The first part of this state-of-the-art reviews the application of IPF procedure to normalize contingency tables, while the second focuses on the impact of distribution bias on the confusion matrix.

2.1 Normalization Using Iterative Proportional Fitting

Deming and Stephan (1940) proposed the use of the Iterative Proportional Fitting (IPF) procedure to es-

timate cell probabilities in a contingency table under given marginal constraints.

Ireland and Kullback (1968) address the problem of estimating a new contingency table q from a given contingency table p, where each cell q_{ij} represents a probability, subject to known and fixed marginal probabilities $\sum_j q_{ij} = u_i$ and $\sum_i q_{ij} = v_j$. The authors introduce an algorithm minimizing the KL divergence from p to q. This procedure is equivalent to IPF (Idel, 2016).

In the fields of remote sensing and geographic modeling, disagreement between maps and reality is commonly displayed in a confusion matrix (Congalton, 2001; Hardin and Shumway, 1997). When multiple classification or modeling methods are used, the resulting confusion matrices are typically compared to assess significant differences (Hardin and Shumway, 1997). Confusion matrix normalization using IPF is a standard analytical technique (Congalton, 1991). In this way, differences in sample sizes used to generate the matrices are eliminated and, therefore, individual cell values within the matrix are directly comparable (Congalton, 1991).

For speaker verification system improvements, Nagineni and Hegde (2010) use the IPF procedure to normalize confusion matrices and select a cohort set based on similarity modeling for each client speaker.

2.2 Causes of Errors

In the context of neural networks, Erbani et al. (2024) examine how imbalanced training and test sets influence confusion matrix entries. They introduce the test–training ranking to distinguish sources of error. When some entries deviate from this criterion, it suggests strong similarity between the corresponding classes. However, class similarities can still impact errors even when the ranking is preserved. Moreover, deviations from the ranking are ambiguous—they do not indicate which entry is too high or too low (Erbani et al., 2024).

2.3 Key Takeaways

Normalizing contingency tables—particularly confusion matrices—using IPF procedure is not a new idea; it is even considered a standard approach in some disciplines. However, to the best of our knowledge, no existing work in machine learning employs this normalization to separate the effects of distribution bias and class similarity within confusion matrix. This decomposition provides deeper insights into the sources of errors, helping to assess and improve the classifier.

To the best of our knowledge, no prior work has connected confusion matrix normalization to importance sampling or to the structure of class representations in a model's latent space. We introduce two perspectives—probabilistic and geometric—that offer deeper insight into classifier behavior.

3 BI-NORMALIZATION

In this section, we define the bi-normalization, outline the key properties it should satisfy, and show how the IPF procedure can compute it accordingly.

Theoretical analysis is simpler with positive matrices. Since a non-negative confusion matrix M and its strictly positive counterpart $M + \epsilon$ (for small ϵ) exhibit similar model behavior, we propose using $M + \epsilon$ instead. We now assume that M is a positive confusion matrix.

3.1 Empirical Definition & Desirable Properties

Normalization approximates the confusion matrix that would arise under specific experimental conditions. Row normalization corresponds to the confusion matrix obtained from a balanced test set, whereas column normalization corresponds to that obtained from a model producing balanced predictions.

Empirical definition. Bi-normalization captures class similarities by approximating the confusion matrix under balanced label distributions in both the training and test sets. We formalize this by requiring both row and column normalization: $\mathrm{bi}(M)_{i+} = \mathrm{bi}(M)_{+j} = 1$.

Standard normalization methods all, row, and col satisfy three properties—idempotence, class distribution invariance, and information preservation—described below. By extension, bi-normalization is expected to satisfy these properties as well.

Idempotence. Normalization maps the matrix to specific experimental conditions. Once these conditions are reached, further normalization does not modify the matrix. Accordingly, bi-normalization is expected to satisfy $\mathrm{bi} \circ \mathrm{bi}(M) = \mathrm{bi}(M)$, where $\circ$ denotes composition.

Class Distribution Invariance. If two confusion matrices differ only in the experimental conditions targeted by the normalization operator, then normalization yields similar matrices.

For instance, row normalization targets a balanced test set. Thus, regardless of the test set label distribution, a trained model yields similar row-normalized confusion matrices. Formally, let S and T be the confusion

matrices obtained from Φ on two large test sets with different label distributions (without class extinction). Then, the error patterns across rows remain similar, i.e., $S_{i,:} \propto T_{i,:}$ for all i, where $\propto$ denotes equality up to a positive scaling factor[2]. This implies that $\mathrm{row}(S) \approx \mathrm{row}(T)$, showing that the row operator is invariant to variations in the test set label distribution.

This invariance is formalized as invariance under left multiplication by a positive diagonal matrix: $\mathrm{row}(AM) = \mathrm{row}(M)$ for any diagonal matrix $A \in \mathbb{R}_{>0}^{C\times C}$. Similarly, column normalization is invariant under right multiplication by a positive diagonal matrix: $\mathrm{col}(MB) = \mathrm{col}(M)$ for any diagonal matrix $B \in \mathbb{R}_{>0}^{C\times C}$.

Bi-normalization should be invariant under both operations, as it combines row and column normalization, i.e., $\mathrm{bi}(AMB) = \mathrm{bi}(M)$.

Information Preservation. Normalization infers, from the original confusion matrix, the one obtained under other experimental conditions. It should therefore preserve as much of the original information as possible, altering it only when necessary.

This requirement can be formalized as an optimization problem that promotes similarity between the original and normalized matrices. We use the Kullback–Leibler (KL) divergence as a dissimilarity measure. Given row and/or column constraints, the normalized matrix is defined as the one closest to the original in terms of KL divergence.

For instance, row normalization satisfies

$$\mathrm{row}(M) \in \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}\,:\\ P_{i+} = 1,\ P_{+j} = \sum_i \frac{M_{ij}}{M_{i+}}\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|M),$$

where $M \in \mathbb{R}_{>0}^{C\times C}$ is the original confusion matrix, and $D_{\mathrm{KL}}(P\|M)$ denotes the KL divergence from P to M. Proofs and extensions to other normalization methods are provided in the Appendix.

Bi-normalization should minimize this optimization problem under the constraints that each row and each column sums to one.

3.2 Theoretical Definition, Properties & Estimation

This subsection provides the formal definition of the bi-normalization, its properties and how to estimate it (see Appendix for details).

[2]For vectors or matrices U and V of the same size, $U \propto V$ means $U \approx aV$ for some $a \in \mathbb{R}_{>0}$.

Definition 1. $\mathrm{bi}(M)$ *is the unique minimizer of the following constrained problem*[3]*:*

$$\mathrm{bi}(M) \in \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}\,:\\ P_{i+} = P_{+j} = 1\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|M)$$

We now establish key properties:

Proposition 1. $\mathrm{bi}(M)$ *satisfies idempotence, class distribution invariance, and information preservation as described in Subsection Empirical Definition & Desirable Properties.*

The last property is satisfied by definition, while the two others follow directly from it, as shown in the appendix.

Although $\mathrm{bi}(M)$ has no closed form, it can be estimated using the IPF algorithm (Algorithm in Appendix). In our case, it consists of normalizing the matrix alternately by rows and columns until convergence[4]. For example, if convergence is reached in two iterations, then $\mathrm{col}\circ\mathrm{row}\Big(\mathrm{col}\circ\mathrm{row}(M)\Big) \approx \mathrm{bi}(M)$.

The IPF procedure always converges when the input matrix M is positive (Idel, 2016). Let $M_{i+} = u_i$ and $M_{+j} = v_j$ for all i and j, and denote by Q the limit of the IPF procedure[5]. Then, Q solves the following optimization problem (Kurras, 2015; Idel, 2016):

$$Q \in \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}\,:\\ P_{i+} = u_i, P_{+j} = v_j\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|M)$$

This directly implies the following:

Proposition 2. $\mathrm{bi}(M)$ *can be approximated using the IPF procedure with row and column constraints set to* 1*. We have* $\mathrm{bi}(M) = Q \approx \hat{Q}$*, where* Q *is the theoretical IPF limit and* $\hat{Q}$ *is its empirical estimate.*

Since $\mathrm{bi}(M)$ is doubly stochastic[6] and M is positive, the IPF procedure converges linearly (Idel, 2016).

4 PROBABILISTIC & GEOMETRIC VIEWS OF NORMALIZATION

This section introduces two perspectives on normalization: (i) importance sampling and (ii) class repre-

[3]This optimization problem always has a solution, and it is unique, as shown in the appendix.

[4]Let $Q^{(t)}$ be the matrix produced by IPF at iteration t. Convergence is reached when $Q_{i+}^{(t)}$ and $Q_{+j}^{(t)}$ are sufficiently close to 1 under a fixed tolerance.

[5]Let $Q^{(t)}$ be the matrix produced by IPF at iteration t. Then $Q^{(t)} \to Q$ as $t \to \infty$.

[6]A doubly stochastic matrix is a square matrix with nonnegative entries such that each row and each column sums to 1.

sentations. We first show that normalization can be interpreted as importance sampling. We then propose to model class representations as volumes, and finally link the overlaps of these volumes to normalized confusion matrices.

4.1 Normalization as Importance Sampling

The normalization process can be viewed as an importance sampling strategy to match the desired distribution.

Let $(x_1, y_1), \ldots, (x_N, y_N)$ be the dataset used to construct the confusion matrix M, where $(x_k, y_k) \in \mathcal{X} \times [C]$, $\mathcal{X}$ is the input space, C is the number of classes, and $[C] = \{1, \ldots, C\}$ is the set of class indices. For each $k = 1, \ldots, N$, the model prediction $\hat{y}_k \in [C]$ is defined by

$$\hat{y}_k \in \operatorname{argmax}_{i=1}^{C} \Phi(x_k)_i,$$

where Φ denotes the model.

Weighted sum. By definition, the confusion matrix is $M = \sum_{k=1}^{N} E_{y_k \hat{y}_k}$, where E_{ij} denotes the $C \times C$ matrix with a 1 at entry (i, j) and 0 elsewhere. Each matrix $E_{y_k \hat{y}_k}$ is a confusion matrix constructed from the single pair $(y_k, \hat{y}_k)$, capturing its individual contribution to M.

Standard and bi normalizations can be obtained by multiplying M on the left and right by diagonal matrices. Specifically, let D^l and D^r denote diagonal matrices (with l for left and r for right), and let M be the original confusion matrix. Then, $D^l M D^r$ yields the normalized version of M.

For instance, setting

$$D^l = \operatorname{diag}\left(\frac{1}{M_{1+}}, \ldots, \frac{1}{M_{C+}}\right) \text{ and } D^r = I$$

yields the row-normalized matrix, where $\operatorname{diag}(d_1, \ldots, d_C)$ denotes a diagonal matrix with entries $d_1, \ldots, d_C$ on its diagonal, and I the identity. Other normalizations are presented in the Appendix.

As a result, normalization can be expressed as a weighted sum of the individual contributions $E_{y_k \hat{y}_k}$:

$$D^l M D^r = \sum_{k=1}^{N} D^l_{y_k} D^r_{\hat{y}_k} E_{y_k \hat{y}_k},$$

where each contribution $E_{y_k \hat{y}_k}$ is weighted by the product $D^l_{y_k} D^r_{\hat{y}_k}$.

For clarity, the weight $D^l_{y_k} D^r_{\hat{y}_k}$ can be expressed through a weight function ω. For instance, the weight function ω_r (with r for row) for row-normalization is

$$\omega_r : (y, \hat{y}) \mapsto \frac{1}{M_{y+}}, \ \operatorname{row}(M) = \sum_{k=1}^{N} \omega_r(y_k, \hat{y}_k) E_{y_k \hat{y}_k}.$$

In the same way, we define ω_a (a for all), ω_c (c for column) and ω_b (b for bi), yielding all, column, and bi normalizations, respectively (see Appendix for explicit definitions).

When the dataset is viewed as a realization of random variables, this process corresponds to importance sampling.

Importance Sampling. Let $(Y_1, \hat{Y}_1), \ldots, (Y_N, \hat{Y}_N)$ be independent and identically distributed (i.i.d.) random pairs, representing label–prediction pairs, and f be their joint distribution $f(i, j) = \mathbb{P}(Y = i, \hat{Y} = j)$. The random counterpart of all(M) converges almost surely (a.s.) to

$$\frac{1}{N} \sum_{k=1}^{N} E_{Y_k \hat{Y}_k} \xrightarrow[N \to \infty]{\text{a.s.}} \mathbb{E}[E_{Y \hat{Y}}],$$

where

$$\mathbb{E}[E_{Y \hat{Y}}]_{ij} = \mathbb{E}[\mathbb{1}_{Y=i, \hat{Y}=j}] = \mathbb{P}(Y = i, \hat{Y} = j) = f(i, j),$$

with $\mathbb{1}$ the indicator function.

Importance sampling refers to Monte Carlo methods that approximate expectations under a target distribution g by reweighting samples drawn from a proposal distribution f (Tokdar and Kass, 2010). This approach is, for instance, useful when direct samples from g are unavailable. To approximate the expectation of $E_{Y,\hat{Y}}$ when $(Y, \hat{Y})$ is distributed according to g rather than f, we use the estimator $\hat{\mu}_g$, defined as

$$\begin{aligned} \hat{\mu}_g = & \frac{1}{N} \sum_{k=1}^{N} \frac{g(Y_k, \hat{Y}_k)}{f(Y_k, \hat{Y}_k)} E_{Y_k \hat{Y}_k} \\ & \xrightarrow[N \to \infty]{\text{a.s.}} \mathbb{E}\left[\frac{g(Y, \hat{Y})}{f(Y, \hat{Y})} E_{Y \hat{Y}}\right] = \mathbb{E}_g[E_{Y \hat{Y}}] \end{aligned}$$

with

$$\mathbb{E}_g[E_{Y \hat{Y}}]_{ij} = \mathbb{E}_g[\mathbb{1}_{Y=i, \hat{Y}=j}] = \mathbb{P}_g(Y = i, \hat{Y} = j) = g(i, j),$$

where $\mathbb{E}_g$ and $\mathbb{P}_g$ denote expectation and probability assuming the variables follow the law g.

For importance sampling to remain accurate, the variance of each entry of the estimator $\hat{\mu}_g$ must remain small, which occurs when f is approximately proportional to g (Tokdar and Kass, 2010).

By selecting different choices of g, standard normalizations arise naturally. As an example, consider row normalization (see Appendix for other normalizations). Setting $g(i, j) = \mathbb{P}(\hat{Y} = j \mid Y = i)$[7] recovers the row-normalized confusion matrix.

[7] Note that g is not a normalized distribution, since $\sum_{i,j} g(i, j) = C$ rather than 1.

In this case, the importance sampling ratio becomes

$$\frac{g(i,j)}{f(i,j)} = \frac{\mathbb{P}(\hat{Y}=j \mid Y=i)}{\mathbb{P}(Y=i,\hat{Y}=j)} = \frac{1}{\mathbb{P}(Y=i)} \approx \frac{N}{M_{i+}}.$$

Accordingly, the empirical estimator is

$$\frac{1}{N}\sum_{k=1}^{N} \frac{N}{M_{y_k+}} E_{y_k\hat{y}_k} = \sum_{k=1}^{N} \omega_r(y_k,\hat{y}_k) E_{y_k\hat{y}_k} = \mathrm{row}(M).$$

For row normalization to be reliable, the importance sampling ratio should remain controlled. In particular, this requires that each class i is sufficiently represented in the test set; otherwise, the estimator exhibits high variance and becomes unstable.

Key Takeaways. Normalization is an importance sampling procedure: each pair is reweighted to shift the empirical distribution toward a target one. Reliability requires that the two distributions are not too dissimilar.

4.2 Class Representations as Volumes

This subsection introduces a geometric perspective on class representations in the model's latent space. The key idea is to characterize class clusters as multidimensional histograms.

Representation Space & Dimensionality Reduction. We use the output of the final pre-logit layer as the class representation space, a procedure already employed in previous work (Beery et al., 2020; Krizhevsky et al., 2012; Zhu et al., 2022). Let ε be the embedding function, so that $\Phi(x) = \mathrm{Softmax} \circ \mathrm{Logit} \circ \varepsilon(x)$, where $\varepsilon(x) \in \mathbb{R}^n$ for each (x, y) in the training set $\mathcal{D}$.

To facilitate histogram construction, we project the embeddings into a lower-dimensional space using Principal Component Analysis (PCA). Let P be the projection matrix with $m < n$, so that $P\varepsilon(x) \in \mathbb{R}^m$.

Class Clusters & Histograms. For each label i, we define the cluster of projected embedded points as $\mathcal{C}_i = \{P\varepsilon(x) : (x,y) \in \mathcal{D},\ y = i\}$. Similarly, the cluster of points predicted as class j is $\hat{\mathcal{C}}_j = \{P\varepsilon(x) : (x,y) \in \mathcal{D},\ \hat{y} = j\}$.

For each label and prediction clusters, we construct a multidimensional, non-overlapping histogram (Thaper et al., 2002) (see Appendix for details). Each histogram is interpreted as a volume in $\mathbb{R}^{m+1}$, providing a geometric view of the clusters. Figure 2 shows a toy example illustrating histograms of class clusters.

Weighted Histograms. Following the importance sampling view (Subsection 4.1), we weight each point to construct weighted histograms, using the schemes

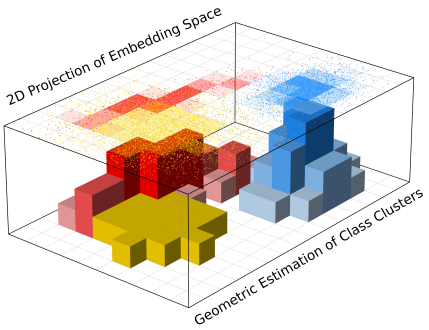


Figure 2: Toy example showing a 2D projection of embedded observations forming class clusters (top map). Each cluster is associated with a histogram indicating the spatial distribution of points (bottom map).

ω_a, ω_r, ω_c, and ω_b. For a given bin, its height is defined not by the number of points it contains, but by the sum of their weights.

For instance, consider cluster $\mathcal{C}_i$ and the weighting ω_c. The height of a given bin b is $\sum_{(x,y)\in\mathcal{D}} \omega_c(y,\hat{y}) \mathbb{1}_{y=i} \mathbb{1}_{P\varepsilon(x)\in b}$, whereas in the unweighted histogram it is $\sum_{(x,y)\in\mathcal{D}} \mathbb{1}_{y=i} \mathbb{1}_{P\varepsilon(x)\in b}$.

We denote by $V(\mathcal{C},\omega)$ the hypervolume in $\mathbb{R}^{m+1}$ of the weighted histogram built from cluster $\mathcal{C}$ using the weighting ω.

4.3 Geometric View of Normalized Confusion Matrices

This subsection introduces a new matrix, referred to as Geometric Confusion Matrix (GCM), which quantifies the overlap between the weighted histogram of label and prediction clusters:

$$\Big(\mathrm{GCM}_\omega\Big)_{ij} = \lambda\Big(V(\mathcal{C}_i,\omega) \cap V(\hat{\mathcal{C}}_j,\omega)\Big),$$

for $i, j = 1, \ldots, C$, where λ denotes the Lebesgue measure. The Lebesgue measure of $V(\mathcal{C}_i,\omega) \cap V(\hat{\mathcal{C}}_j,\omega)$ corresponds to the volume of the intersection between the ω-weighted histograms of $\mathcal{C}_i$ and the one of $\hat{\mathcal{C}}_j$. Unlike the standard confusion matrix, the GCM takes into account the spatial organization of embedded points (see Appendix for details).

We now provide geometric views of confusion matrix normalizations, all claims are supported by experimental results (see Section 6).

All-normalization. All-normalized matrix approximates the overlap between the unweighted histograms of label and prediction clusters: $\mathrm{all}(M) \propto \mathrm{GCM}_{\omega_a}$, where $\propto$ means approximately equal up to a scaling

factor[8].

Row-normalization. Row-normalized matrix approximates the overlap between normalized label histograms and the resulting weighted prediction histograms: $\mathrm{row}(M) \propto\!\!\!\sim \mathrm{GCM}_{\omega_r}$.

More precisely, up to a known scaling factor, the label histograms are normalized in the sense that $\lambda\big(V(\mathcal{C}_i, \omega_r)\big) = 1$, whereas the volume of the prediction histograms, $\lambda\big(V(\hat{\mathcal{C}}_j, \omega_r)\big)$ are not fixed a priori (see Appendix for details). Accordingly, label histograms are normalized, while prediction histograms are scaled according to ω_r.

Col-normalization. Similarly, column-normalized matrix approximates the overlap between normalized prediction histograms and resulting weighted label histograms: $\mathrm{col}(M) \propto\!\!\!\sim \mathrm{GCM}_{\omega_c}$.

Bi-normalization. Finally, bi-normalized matrix approximates the overlap between normalized histograms of both label and prediction clusters: $\mathrm{bi}(M) \propto\!\!\!\sim \mathrm{GCM}_{\omega_b}$.

These correspondences provide a geometric interpretation of normalized confusion matrices and illustrate how confusion matrices capture the organization of classes in the model's latent space.

5 EXPERIMENTAL SETUP

This section describes our experimental setup.

5.1 Datasets & Heterogeneity

We conduct experiments on four datasets: MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky, 2009), and STL-10 (Coates et al., 2011).

Normalizations methods differ mainly on non-diagonal matrices. For example, row, column, and bi normalizations yield the same result on diagonal matrices. To encourage non-diagonal confusion matrices, we make MNIST and Fashion-MNIST harder by applying random rotations of 0°, 90°, 180°, or 270° during preprocessing, applied to both the training and test images.

Bi-normalization is particularly useful under heterogeneous settings. To simulate data heterogeneity, we sample from the original datasets with a Dirichlet distribution of concentration α, following (Allouah et al., 2023; Hsu et al., 2019; Erbani et al., 2025). We consider five levels—very low, low, medium, high, and extreme—corresponding to $\alpha \in \{10, 3, 1, 0.3, 0.1\}$.

[8]Let M and N be matrices of the same size. We write $M \propto\!\!\!\sim N$ to indicate that $M \approx \alpha N$ for some scalar $\alpha \in \mathbb{R}_{>0}$.

To avoid class extinction, we enforce a minimum representation of 20% of the original class size. For example, in a dataset with 1000 samples per class, at least 200 samples are retained for each class, while the remaining samples are allocated according to a Dirichlet distribution.

5.2 Models & Training

CNN models (see Appendix for details) are trained using stochastic gradient descent with cross-entropy loss, a batch size of 32, a learning rate of 10^{-3}, a momentum of 0.9, and a weight decay of 10^{-4}.

As mentioned above, the main differences between normalizations appear in the off-diagonal confusion matrices. Therefore, training is limited to a maximum of 10 epochs, and stops earlier if the classifier reaches 60% balanced test accuracy.

5.3 Metric

Let S and T be two confusion matrices in $\mathbb{R}_{\geq 0}^{C \times C}$, possibly already normalized.

To measure the similarity between confusion matrices, we use their overlap:

$$\mathrm{Overlap}(S, T) = \sum_{ij} \min\left(\mathrm{all}(S)_{ij}, \mathrm{all}(T)_{ij}\right),$$

where min denotes the element-wise minimum (see Appendix for the rationale behind reusing all normalization). This score ranges from 0 to 1, with higher values indicating greater similarity. It reaches 1 if and only if $T = S$.

This metric is strictly equivalent to the L^1 distance, since $\|\,\mathrm{all}(S) - \mathrm{all}(T)\|_1 = 2 - 2\,\mathrm{Overlap}(S, T)$ (see Appendix for proof).

5.4 Baselines & Experiments

We compare bi-normalization with standard normalizations: row, col, and all. Each experiment is repeated over MNIST, Fashion-MNIST, CIFAR-10, and STL-10, with five heterogeneity levels and 30 random seeds.

Experiment 1. This experiment compares the confusion matrix obtained from balanced datasets (denoted as M_1) with normalized versions of the confusion matrix obtained from imbalanced datasets (denoted as M_2).

1. Initialize a model with a fixed random seed, then create two deep copies: model 1 and model 2.
2. Train model 1 on a balanced training set and compute its confusion matrix M_1 using a balanced test

set.
3. Sample an imbalanced training set and test set, train model 2 on the imbalanced training set, and compute its confusion matrix M_2 on the imbalanced test set.
4. Apply normalization techniques to M_2 and evaluate their similarity to M_1.

This experiment tests whether bi-normalization reveals class similarities under imbalanced data more effectively than other approaches.

Experiment 2. This experiment, conducted in an imbalanced setting, compares different versions of the GCM—GCMω_a, GCMω_r, GCMω_c, GCMω_b—with normalized confusion matrices derived from M—row(M), col(M), all(M), and bi(M).

1. Initialize the model with a fixed seed.
2. Sample imbalanced training and test sets. Train the model. Compute both the GCM variants and the normalized confusion matrices using the test set.
3. Measure pairwise similarities.

This experiment evaluates how each normalization reveals the structure of class representations in the model's latent space under a specific weighted histogram.

Details on the construction of multivariate histograms required for GCM are provided in the Appendix.

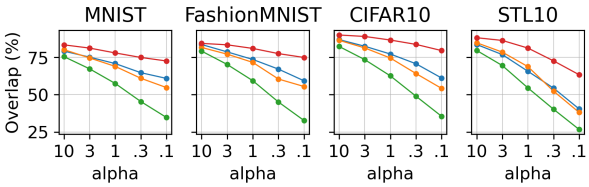
6 EMPIRICAL RESULTS

Figure 3 shows the results of Experiment 1 (Subfigure 3a) and Experiment 2 (Subfigures 3b and 3c), described in Section 5.4.

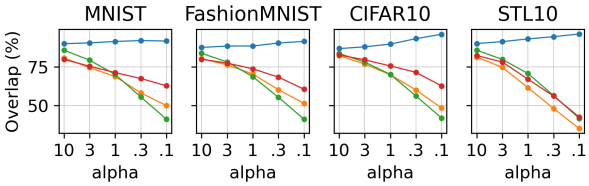
Experiment 1. This experiment evaluates how well different normalizations recover class similarities. Starting from a confusion matrix obtained under an imbalanced setting, normalization aims to approximate the one obtained under a balanced setting.

Across datasets and heterogeneity levels, Subfigure 3a shows that bi-normalization consistently achieves the highest overlap, outperforming other methods. As heterogeneity increases, (i) the gap between bi-normalization and the other methods tends to widen, and (ii) the ability to recover class similarity diminishes regardless of the normalization method.

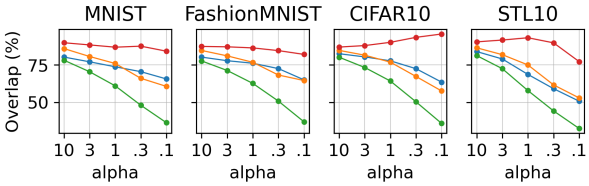
This behavior is expected: the greater the imbalance between the training and test sets, the more distribution bias is introduced into the matrix. This bias tends to obscure class similarities, making row, col, and all normalizations unsuitable. Moreover, higher heterogeneity leads to prediction and test class distributions



(a) Target: Matrix from balanced setting



(b) Target: GCMω_r



(c) Target: GCMω_b

Figure 3: Overlap (%) between target matrices and normalized matrices. Legend: ● bi ● row ● col ● all

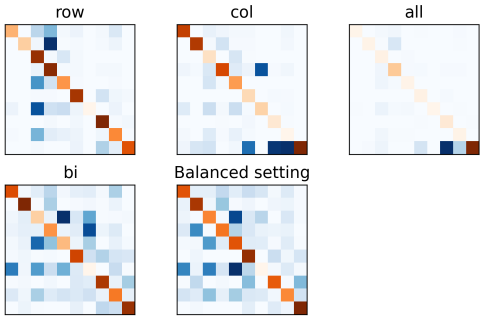


Figure 4: One instance (seed 0) on Fashion-MNIST with $\alpha = 0.3$. Bi-normalization best recovers class relationships under distribution shifts. Diagonal variations are shown in orange; off-diagonal in blue, with darker shades indicating higher values.

that deviate further from a balanced setting, making normalization procedures less reliable, as discussed in Section 4.1.

Figure 4 further illustrates that bi-normalization captures the class similarity patterns present in the balanced confusion matrix, whereas other methods often produce vertical or horizontal stripes.

By definition, bi-normalization is expected to achieve the lowest KL divergence. Figure 6 in the Appendix

confirms this, providing further evidence of its ability to recover class similarities.

Experiment 2. This experiment evaluates the correspondence between the organization of class representations in the model's latent space and normalized confusion matrices.

Subfigure 3b demonstrates a clear correspondence between GCMω_r and row normalization. Overlap increases as heterogeneity grows. We observe similar results for GCMω_a with all normalizations, and for GCMω_c with column normalization (see Appendix).

Subfigure 3c shows that the correspondence between GCMω_b and bi-normalization is weaker. This makes sense, as the granularity differs from the other approaches. For instance, in GCMω_r, all embedded points of the same label receive the same weight, while in GCMω_b, embedded points are weighted by both label and prediction. This indicates that spatial separation is finer under GCMω_b weighting than with other weightings. Nonetheless, bi-normalization still exhibits higher correspondence than other methods.

This experiment confirms the validity of the geometric interpretation of normalizations described in Section 4.3.

7 CONCLUSION

We revisited confusion matrix normalization and showed that bi-normalization provides a principled way to isolate class similarity from distribution bias. While the use of IPF for contingency tables is well established in other fields, its value for interpreting classifier behavior in machine learning has been largely overlooked. We demonstrated that bi-normalization reveals class similarities more clearly than standard methods, especially in heterogeneous settings.

We also showed that normalization corresponds to an importance sampling strategy and introduced a geometric interpretation of class representations in latent space, offering deeper insight into what normalization reveals about model behavior.

Acknowledgements

We would like to thank Olivier Mbarek and Eric Lombardi, who are responsible for the PAGODA platform at LIRIS, for their valuable support and assistance. This work was supported by the French government managed by the Agence Nationale de la Recherche (ANR) through France 2030 program with the reference ANR-23-PEIA-005 (REDEEM project).

References

Umang Aggarwal, Adrian Popescu, and Céline Hudelot. Minority class oriented active learning for imbalanced datasets. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9920–9927. IEEE, 2021.

Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and John Stephan. Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, pages 1232–1300. PMLR, 2023.

Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Neel Joshi, Markus Meister, and Pietro Perona. Synthetic examples improve generalization for rare classes. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pages 863–873, 2020.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

Ekram Chamseddine, Nesrine Mansouri, Makram Soui, and Mourad Abed. Handling class imbalance in covid-19 chest x-ray images classification: Using smote and weighted loss. *Applied Soft Computing*, 129:109588, 2022.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

Russell G Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote sensing of environment*, 37(1):35–46, 1991.

Russell G Congalton. Accuracy assessment and validation of remotely sensed and other spatial information. *International journal of wildland fire*, 10 (4):321–328, 2001.

S Deepak and PM Ameer. Brain tumor categorization from imbalanced mri dataset using weighted loss and deep feature fusion. *Neurocomputing*, 520:94–102, 2023.

W Edwards Deming and Frederick F Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, 11(4):427–444, 1940.

Johan Erbani, Pierre-Édouard Portier, Elöd Egyed-Zsigmond, and Diana Nurbakova. Confusion matrices: A unified theory. *IEEE Access*, 2024.

Johan Erbani, Sonia Ben Mokhtar, Pierre-Edouard Portier, Elod Egyed-Zsigmond, and Diana Nurbakova. Weighted loss methods for robust federated learning under data heterogeneity. *arXiv preprint arXiv:2506.09824*, 2025.

K Ruwani M Fernando and Chris P Tsokos. Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2940–2951, 2021.

Kushankur Ghosh, Colin Bellinger, Roberto Corizzo, Paula Branco, Bartosz Krawczyk, and Nathalie Japkowicz. The class imbalance problem in deep learning. *Machine Learning*, 113(7):4845–4901, 2024.

Jochen Görtler, Fred Hohman, Dominik Moritz, Kanit Wongsuphasawat, Donghao Ren, Rahul Nair, Marc Kirchner, and Kayur Patel. Neo: Generalizing confusion matrix visualization to hierarchical and multi-output labels. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2022.

Perry J Hardin and J Matthew Shumway. Statistical significance and normalized confusion matrices. *Photogrammetric engineering and remote sensing*, 63(6):735–739, 1997.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Martin Idel. A review of matrix scaling and sinkhorn's normal form for matrices and positive maps. *arXiv preprint arXiv:1609.06349*, 2016.

C Terrance Ireland and Solomon Kullback. Contingency tables with given marginals. *Biometrika*, 55 (1):179–188, 1968.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Damir Krstinić, Maja Braović, Ljiljana Šerić, and Dunja Božić-Štulić. Multi-label classifier performance evaluation with confusion matrix. *Computer Science & Information Technology*, 1, 2020.

Damir Krstinić, Ana Kuzmanić Skelin, Ivan Slapničar, and Maja Braović. Multi-label confusion tensor. *IEEE access*, 2024.

Sven Kurras. Symmetric iterative proportional fitting. In *Artificial Intelligence and Statistics*, pages 526–534. PMLR, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

Joffrey L Leevy, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):1–30, 2018.

Srikanth Nagineni and Rajesh M Hegde. On line client-wise cohort set selection for speaker verification using iterative normalization of confusion matrices. In *2010 18th European Signal Processing Conference*, pages 576–580. IEEE, 2010.

Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

Robert I Saye. High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. *SIAM Journal on Scientific Computing*, 37 (2):A993–A1019, 2015.

David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

Mohammed Temraz and Mark T Keane. Solving the class imbalance problem using a counterfactual method for data augmentation. *Machine Learning with Applications*, 9:100375, 2022.

Nitin Thaper, Sudipto Guha, Piotr Indyk, and Nick Koudas. Dynamic multidimensional histograms. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 428–439, 2002.

Jilun Tian, Yuchen Jiang, Jiusi Zhang, Hao Luo, and Shen Yin. A novel data augmentation approach to fault diagnosis with class-imbalance problem. *Reliability Engineering & System Safety*, 243:109832, 2024.

Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

Na Wu, Shizhuang Weng, Jinxin Chen, Qinlin Xiao, Chu Zhang, and Yong He. Deep convolution neural network with weighted loss to detect rice seeds vigor based on hyperspectral imaging under the sample-imbalanced condition. *Computers and Electronics in Agriculture*, 196:106850, 2022.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmark-

ing machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Enwei Zhu, Yiyang Liu, and Jinpeng Li. Deep span representations for named entity recognition. *arXiv preprint arXiv:2210.04182*, 2022.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable] The only algorithm used is IPF, which, as established in the literature, converges linearly in our setting.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable] Publicly available.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable] All proofs are provided in Appendix.
   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/**Not Applicable**]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable]
   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not Applicable**]
   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

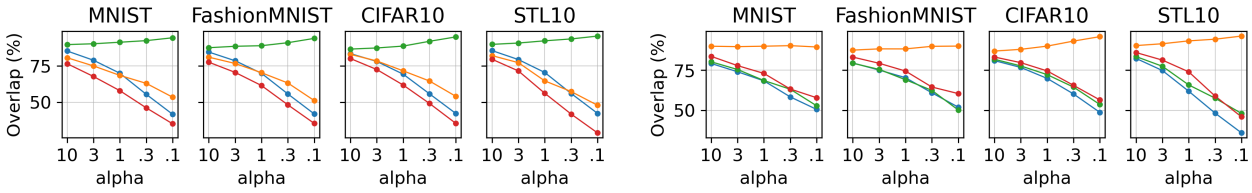
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

The supplementary materials are organized as follows: **Section 1** provides additional results that further support our propositions. **Section 2** describes the Iterative Proportional Fitting (IPF) algorithm and an equivalent formulation known as the RAS algorithm, whose variations with respect to IPF are later exploited. **Section 3** details the overlap metric used in our experiments. **Section 4** presents experimental details, including the multidimensional histogram setup and the model architectures. **Section 5** provides proofs of key properties such as information preservation, idempotence, and class distribution invariance. **Section 6** describes additional weighting schemes mentioned in Subsection Normalization as Importance Sampling. **Section 7** derives and proves the explicit formula for the Geometric Confusion Matrix (GCM). **Section 8** demonstrates an auxiliary lemma used in the preceding sections.

A ADDITIONAL RESULTS

Figure 5 presents the remaining results of Experiment 2 (see Subsection Baselines & Experiments) with target matrices GCM_a and GCM_c. We observe trends consistent with those obtained for GCM_r: GCM_a aligns closely with the all normalization (Subfigure 5a), while GCM_c aligns closely with the col normalization (Subfigure 5b). These results further support the geometric interpretation of normalizations.

Figure 6 reports the results for Experiment 1 (see Subsection Baselines & Experiments) using the KL divergence instead of the overlap metric. It shows that bi-normalization still achieves the highest similarity with the confusion matrix under a balanced setting, as expected.



(a) Target: GCM_{ω_a} (b) Target: GCM_{ω_c}

Figure 5: Overlap (%) between target matrices and normalized matrices. Legend: ● bi ● row ● col ● all

B IPF & RAS ALGORITHMS

This section presents the IPF (Algorithm 1) and RAS (Algorithm 2) procedures.

According to Lemma 2 (Section H), there exist positive diagonal matrices D^l and D^r such that $\mathrm{bi}(M) = D^l M D^r$. The RAS method estimates D^l and D^r. This algorithm will be used in the following section and is equivalent to Algorithm 1; see (Idel, 2016) for further details.

C OVERLAP METRIC

This section explains the rationale for normalizing confusion matrices when computing the Overlap metric, and shows its equivalence with the L_1 distance.

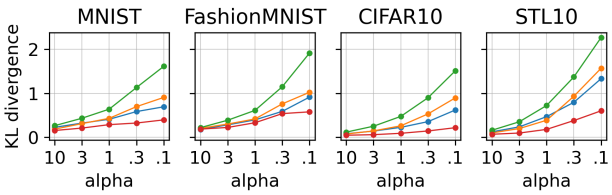


Figure 6: KL divergence (%) between matrix from balanced setting and normalized matrices. Legend: ● bi ● row ● col ● all

Algorithm 1 Iterative Proportional Fitting (IPF)

Require: Initial matrix $M \in \mathbb{R}_{>0}^{C\times C}$, target row sums $u \in \mathbb{R}_{>0}^{C}$, target column sums $v \in \mathbb{R}_{>0}^{C}$ such that $u_+ = v_+$, tolerance $\epsilon > 0$, maximum number of steps T

Initialize $Q^{(0)} \leftarrow M$, $t \leftarrow 0$
repeat
  for each row $i = 1$ to C **do**
    $Q_{i,:}^{(t+1)} \leftarrow Q_{i,:}^{(t)} \cdot \frac{u_i}{Q_{i+}^{(t)}}$
  end for
  for each column $j = 1$ to C **do**
    $Q_{:,j}^{(t+2)} \leftarrow Q_{:,j}^{(t+1)} \cdot \frac{v_j}{Q_{+j}^{(t+1)}}$
  end for
  $t \leftarrow t + 2$
until $\|Q_{:+}^{(t)} - u\|_1 + \|Q_{+:}^{(t)} - v\|_1 \leq \epsilon$ **or** $t \geq T$
return $\widehat{Q} \leftarrow Q^{(t)}$

Algorithm 2 RAS method

Require: Positive matrix $M \in \mathbb{R}_{>0}^{C\times C}$, target row sums $r \in \mathbb{R}_{>0}^{C}$, target column sums $c \in \mathbb{R}_{>0}^{C}$ such that $r_+ = c_+$, tolerance $\epsilon > 0$, maximum number of steps T

Initialize $D^{r\,(0)} \leftarrow I$, $t \leftarrow 0$
repeat
  for each $i = 1$ to C **do**
    $D_{ii}^{l\,(t+1)} \leftarrow \dfrac{r_i}{\sum_j M_{ij} D_{jj}^{r\,(t)}}$
  end for
  for each $j = 1$ to C **do**
    $D_{jj}^{r\,(t+2)} \leftarrow \dfrac{c_j}{\sum_i M_{ij} D_{ii}^{l\,(t+1)}}$
  end for
  $t \leftarrow t + 2$
until $\|(D^{l\,(t)} M D^{r\,(t)})_{:+} - r\|_1 + \|(D^{l\,(t)} M D^{r\,(t)})_{+:} - c\|_1 \leq \epsilon$ **or** $t \geq T$
return $\widehat{D}^l \leftarrow D^{l\,(t)}$, $\widehat{D}^r \leftarrow D^{r\,(t)}$

C.1 Why Normalize?

The entries of a confusion matrix are only meaningful relative to each other; absolute values lack interpretability. For example, $P_{ii} = 3$ indicates perfect classification if $P_{i+} = 3$, but poor performance if $P_{i+} = 30$. More generally, for any $\lambda > 0$, P and λP represent the same model behavior.

Comparisons between two matrices are meaningful when they share the same total sum, i.e., $P_{++} = Q_{++}$. For instance, $P_{ii} = 1$ may seem worse than $Q_{ii} = 10$, but if $P_{++} = 3$ and $Q_{++} = 30$, the relative accuracies are identical: $P_{ii}/P_{++} = Q_{ii}/Q_{++}$.

For consistency, we compare normalized confusion matrices: $\mathrm{all}(P) = P/P_{++}$ and $\mathrm{all}(Q) = Q/Q_{++}$. This ensures that behaviors are preserved and enables consistent comparisons.

C.2 Equivalence with L_1 Distance

Let P and Q be two matrices such that $P_{++} = Q_{++} = 1$. Then,

$$\begin{aligned}\|P - Q\|_1 = \sum_{ij} |P_{ij} - Q_{ij}| &= \sum_{ij} \max(P_{ij}, Q_{ij}) - \min(P_{ij}, Q_{ij}) \\ &= \sum_{ij:P_{ij} \geq Q_{ij}} P_{ij} - \min(P_{ij}, Q_{ij}) \qquad + \sum_{ij:Q_{ij} > P_{ij}} Q_{ij} - \min(P_{ij}, Q_{ij}) \\ &= \sum_{ij} P_{ij} - \min(P_{ij}, Q_{ij}) + \sum_{ij} Q_{ij} - \min(P_{ij}, Q_{ij}) \\ &= 2 - 2\,\mathrm{Overlap}(P, Q)\end{aligned}$$

Thus, the Overlap metric is directly related to the L_1 distance.

D EXPERIMENTS DETAILS

This section presents details on the multidimensional histogram setup and the model architectures.

D.1 HISTOGRAM IN EXPERIMENTS

First, the projection space is divided into a regular grid of m-dimensional hyperrectangles,[9] which define the histogram bins. We then construct non-overlapping histograms (Thaper et al., 2002) associated with the clusters $\mathcal{C}_k$ and $\widehat{\mathcal{C}}_k$ for $k = 1, \ldots, C$.

Considering a cluster $\mathcal{C}$ and its unweighted histogram, each bin height represents the number of points from $\mathcal{C}$ that fall within the bin (i.e., hyperrectangle) under consideration. In the weighted case, the bin height instead corresponds to the sum of the point weights, using one of these weighting schemes ω_a, ω_r, ω_c, or ω_b.

In the experiments, we fix the projected space dimensionality to $m = 10$ and specify the bin width accordingly. Following the recommendation of (Scott, 2015) (see Subsection 3.4, Eq. (3.66)), we adopt the optimal bin width $3.5\ \sigma_k\ n^{-1/(2+m)}$, where σ_k denotes the empirical standard deviation along the k-th dimension, n is the sample size, and m is the data dimensionality.

D.2 MODEL ARCHITECTURES

We use the following abbreviations to describe our architectures: $L(n)$ denotes a fully connected linear layer with n outputs; R is a ReLU activation; $C(c)$ represents a 2D convolutional layer with c output channels, kernel size 5 for CIFAR datasets and 3 otherwise, padding 0 for CIFAR and 1 otherwise, and stride 1; M denotes 2D max pooling with kernel size 2; B stands for batch normalization; and D represents dropout with a fixed probability of 0.25.

In line with Allouah et al. (2023), the model architectures are defined as follows. For MNIST and Fashion-MNIST, the architecture is $(1, 28 \times 28) - C(8) - R - M - C(16) - R - M - L(64) - R - L(10)$. For CIFAR-10 and STL-10, the architecture is $(3, 32 \times 32) - C(64) - R - B - C(64) - R - B - M - D - C(128) - R - B - C(128) - R - B - M - D - L(128) - R - D - L(10)$.

E INFORMATION PRESERVATION, IDEMPOTENCE & CLASS DISTRIBUTION INVARIANCE

This section demonstrates that normalization methods satisfy the properties of information preservation, idempotence, and class distribution invariance.

[9] In geometry, a hyperrectangle generalizes a rectangle (in 2D) and a rectangular cuboid (in 3D) to higher dimensions (Saye, 2015).

E.1 Information Preservation for Standards Normalizations

We state that, for a positive confusion matrix M, the normalizations minimize the KL divergence under specific constraints:

$$\mathrm{all}(M) \in \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}: \\ P_{i+} = \frac{M_{i+}}{M_{++}}, P_{+j} = \frac{M_{+j}}{M_{++}}\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|M), \quad \mathrm{row}(M) \in \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}: \\ P_{i+} = 1, P_{+j} = \sum_i \frac{M_{ij}}{M_{i+}}\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|M), \quad \mathrm{col}(M) \in \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}: \\ P_{i+} = \sum_j \frac{M_{ij}}{M_{+j}}, P_{+j} = 1\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|M).$$

We observe that these normalizations can be expressed as the product of diagonal matrices and the original confusion matrix:

$$\mathrm{row}(M) = \mathrm{diag}(\tfrac{1}{M_{1+}}, \ldots, \tfrac{1}{M_{C+}})MI, \quad \mathrm{col}(M) = IM\,\mathrm{diag}(\tfrac{1}{M_{+1}}, \ldots, \tfrac{1}{M_{+C}}), \text{ and } \mathrm{all}(M) = IMI/M_{++}. \tag{1}$$

According to Lemma 2 (Section H), these normalization procedures indeed minimize the KL divergence, which concludes the proof.

E.2 Idempotence & Class Distribution Invariance of Bi-Normalization

In Proposition 1, we state that the bi-normalization satisfies idempotence,

$$\mathrm{bi} \circ \mathrm{bi}(M) = \mathrm{bi}(M),$$

and class distribution invariance, that is, for any diagonal matrices $A, B \in \mathbb{R}_{>0}^{C\times C}$,

$$\mathrm{bi}(AMB) = \mathrm{bi}(M).$$

We prove these properties in what follows.

Idempotence. By definition, $\mathrm{bi} \circ \mathrm{bi}(M)$ is the solution to the problem

$$\underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}: \\ P_{i+} = P_{+j} = 1\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|\,\mathrm{bi}(M))$$

Moreover, we observe that $I\,\mathrm{bi}(M)I$ satisfies the marginal constraints. By Lemma 2, $I\,\mathrm{bi}(M)I$ solves the above problem, and since the solution is unique, we have

$$\mathrm{bi} \circ \mathrm{bi}(M) = I\ \mathrm{bi}(M)\, I = \mathrm{bi}(M),$$

which establishes the idempotence property.

Class Distribution Invariance. By Lemma 2, there exist diagonal matrices D^l and D^r such that

$$\mathrm{bi}(AMB) = D^l AMBD^r.$$

By definition of the bi-normalization, the row and column sums satisfy

$$(D^l AMBD^r)_{i+} = (D^l AMBD^r)_{+j} = 1 \quad \forall i,j.$$

Additionally, $D^l A$ and BD^r are diagonal. Therefore, by Lemma 2, $\mathrm{bi}(AMB)$ is the unique solution to

$$\underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}: \\ P_{i+} = 1, P_{+j} = 1\ \forall i,j}}{\operatorname{argmin}} D_{\mathrm{KL}}(P\|M)$$

This implies $\mathrm{bi}(AMB) = \mathrm{bi}(M)$, completing the proof.

F IMPORTANCE SAMPLING

This section details the weighting schemes mentioned in Subsection Normalization as Importance Sampling.

From the equalities presented in Equation (1) (Subsection E.1) and by applying the same procedure described in the main part of the paper, it directly follows that

$$\omega_a : (y, \hat{y}) \mapsto \frac{1}{M_{++}} \quad \text{and} \quad \text{all}(M) = \sum_{k=1}^{N} \omega_a(y_k, \hat{y}_k) E_{y_k \hat{y}_k},$$

and

$$\omega_c : (y, \hat{y}) \mapsto \frac{1}{M_{+\hat{y}}} \quad \text{and} \quad \text{col}(M) = \sum_{k=1}^{N} \omega_c(y_k, \hat{y}_k) E_{y_k \hat{y}_k}.$$

According to Lemma 2 (Section H), there exist positive diagonal matrices D^l and D^r such that $\text{bi}(M) = D^l M D^r$, using Algorithm 2 (Section B), we estimate these matrices by matrices $\widehat{D}^l$ and $\widehat{D}^r$. It follows

$$\omega_b : (y, \hat{y}) \mapsto \widehat{D}^l_y \widehat{D}^r_{\hat{y}} \quad \text{and} \quad \text{bi}(M) = \sum_{k=1}^{N} \omega_b(y_k, \hat{y}_k) E_{y_k \hat{y}_k}.$$

G GEOMETRIC CONFUSION MATRIX

This section establishes the volumes of the various scaled histograms introduced previously, and compares the analytical form of the classical confusion matrix with that of the geometric confusion matrix.

G.1 Normalized Histograms Under Weighting Schemes

In Section Geometric View of Normalized Confusion Matrices, we state that, under specific weighting schemes, some histograms are normalized up to a scaling factor. We prove this statement below.

We first introduce the following lemma:

Lemma 1. *Let $\mathcal{C}$ be a cluster of points in the model's latent space, based on dataset $\mathcal{D}$. Let P be a projection matrix, and let ω be the weighting function. Then,*

$$\lambda(V(\mathcal{C}, \omega)) = r \sum_{(x,y)\in\mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P\varepsilon(x)\in\mathcal{C}},$$

where λ denotes the Lebesgue measure, $\mathbb{1}$ is the indicator function, and r the volume of hyperrectangles which split the projection space.

Proof. We begin by decomposing the histogram into a union of hyperrectangles which grid the projection space, denoted by $\mathcal{V}$. Since $\mathcal{D}$ contains a finite number of points, the covering set $\mathcal{V}$ is also finite. By construction of the weighted histogram, we have

$$\begin{aligned} V(\mathcal{C}, \omega) &= \bigcup_{v\in\mathcal{V}} (V(\mathcal{C}, \omega) \cap v) \\ &= \bigcup_{v\in\mathcal{V}} \underbrace{r_1^v \times \cdots \times r_m^v}_{\text{bin/hyperrectangle splitting the projection space}} \times \underbrace{\left[0, \sum_{(x,y)\in\mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P\varepsilon(x)\in\mathcal{C}\cap v}\right]}_{\text{bin height}} \end{aligned}$$

where $r_1^v, \ldots, r_m^v$ denote the intervals describing the sides of each hyperrectangle.

Its Lebesgue measure equals the sum of the measures of the individual hyperrectangles:

$$\lambda(V(\mathcal{C}, \omega)) = \sum_{v\in\mathcal{V}} \lambda(r_1^v \times \cdots \times r_m^v \times \left[0, \sum_{(x,y)\in\mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P\varepsilon(x)\in\mathcal{C}\cap v}\right])$$

The Lebesgue measure of a hyperrectangle is the product of its side lengths. Only the heights vary, while the other side lengths are bin-independent. Let $r = \prod_{i=1}^{m} \lambda(r_i^v)$. Then,

$$\lambda(V(\mathcal{C}, \omega)) = \sum_{v \in \mathcal{V}} r \sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C} \cap v} = r \sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}}$$

which concludes the proof. □

According to Lemma 1, we have

$$\begin{aligned}
\lambda(V(\mathcal{C}_i, \omega_r)) &= r \sum_{(x,y) \in \mathcal{D}} \omega_r(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i} = r \sum_{(x,y) \in \mathcal{D}} \frac{1}{M_{i+}} \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i} = r, \\
\lambda(V(\widehat{\mathcal{C}}_j, \omega_c)) &= r \sum_{(x,y) \in \mathcal{D}} \omega_c(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \widehat{\mathcal{C}}_j} = r \sum_{(x,y) \in \mathcal{D}} \frac{1}{M_{+j}} \mathbb{1}_{P_\varepsilon(x) \in \widehat{\mathcal{C}}_j} = r, \\
\lambda(V(\mathcal{C}_i, \omega_b)) &= r \sum_{(x,y) \in \mathcal{D}} \omega_b(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i} = r \sum_{(x,y) \in \mathcal{D}} \widehat{D}_i^l \widehat{D}_{\hat{y}}^r \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i} = r \,\mathrm{bi}(M)_{i+} = r, \text{ and} \\
\lambda(V(\widehat{\mathcal{C}}_j, \omega_b)) &= r \sum_{(x,y) \in \mathcal{D}} \omega_b(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \widehat{\mathcal{C}}_j} = r \sum_{(x,y) \in \mathcal{D}} \widehat{D}_y^l \widehat{D}_j^r \mathbb{1}_{P_\varepsilon(x) \in \widehat{\mathcal{C}}_j} = r \,\mathrm{bi}(M)_{+j} = r.
\end{aligned}$$

In that sense, these histograms are normalized using weighting schemes up to a scaling factor r, corresponding to the fixed volume of the hyperrectangles partitioning the projection space.

G.2 Geometric Confusion Matrix

The Geometric Confusion Matrix is defined as

$$\Big(\mathrm{GCM}_\omega \Big)_{ij} = \lambda \Big(V(\mathcal{C}_i, \omega) \cap V(\widehat{\mathcal{C}}_j, \omega) \Big)$$

We begin by deriving an explicit formula for GCM_ω. Following the same approach as in proof of Lemma 1, and noting that the grid of hyperrectangles partitioning the projection space is shared across all histograms, the intersection of two histograms is given by:

$$\begin{aligned}
V(\mathcal{C}_i, \omega) \cap V(\widehat{\mathcal{C}}_j, \omega) &= \bigcup_{v \in \mathcal{V}} r_1^v \times \cdots \times r_m^v \times \left[0, \sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i \cap v} \right] \quad \bigcap \\
&\qquad \bigcup_{v \in \mathcal{V}} r_1^v \times \cdots \times r_m^v \times \left[0, \sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \widehat{\mathcal{C}}_j \cap v} \right] \\
&= \bigcup_{v \in \mathcal{V}} r_1^v \times \cdots \times r_m^v \times \left[0, \min \left(\sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i \cap v}, \sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \widehat{\mathcal{C}}_j \cap v} \right) \right]
\end{aligned}$$

As a result, the (i, j) entry of the geometric confusion matrix becomes:

$$\Big(\mathrm{GCM}_{l,p} \Big)_{ij} = r \sum_{v \in \mathcal{V}} \min \left(\sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i \cap v}, \sum_{(x,y) \in \mathcal{D}} \omega(y, \hat{y}) \mathbb{1}_{P_\varepsilon(x) \in \widehat{\mathcal{C}}_j \cap v} \right)$$

The normalized confusion matrix can also be expressed as a sum over grid cells, analogously to GCM_ω. Let norm denote a normalization operator among all, row, col, and bi, and let ω be its associated weighting scheme. It is straightforward to obtain:

$$\mathrm{norm}(M)_{ij} = \sum_{(x,y) \in \mathcal{D}} \omega(i, j) \mathbb{1}_{y=i} \mathbb{1}_{\hat{y}=j} = \sum_{(x,y) \in \mathcal{D}} \omega(i, j) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i \cap \widehat{\mathcal{C}}_j} = \sum_{v \in \mathcal{V}} \sum_{(x,y) \in \mathcal{D}} \omega(i, j) \mathbb{1}_{P_\varepsilon(x) \in \mathcal{C}_i \cap \widehat{\mathcal{C}}_j \cap v}$$

This explicit formula for GCM_ω highlights how the geometric confusion matrix incorporates spatial information. The geometric confusion matrix compares in each grid cell the number of points labeled i (regardless of prediction) with the number of points predicted as j (regardless of true label). As a result, in a grid cell containing only points with label i and prediction j, the contribution to GCM_ω matches that of the standard confusion matrix. In contrast, in a cell containing a mix of labels and predictions, the minimum operation produces a contribution that differs from the simple count of points labeled i and predicted as j—that is, from the contribution in the standard confusion matrix.

H EQUIVALENCE SCALING LEMMA

This section demonstrates the following lemma:

Lemma 2. *Let M be a positive matrix in $\mathbb{R}_{>0}^{C\times C}$, and r and c, be two positive vectors in $\mathbb{R}_{>0}^{C}$ such that $r_+ = c_+$. Then, any matrix M^* of the form $M^* = D^l M D^r$, where D^l and D^r are positive diagonal matrices such that*

$$(D^l M D^r)_{i+} = r_i \quad \textit{and} \quad (D^l M D^r)_{+j} = c_j \quad \textit{for all } i, j,$$

is the unique solution to the following problem:

$$\begin{array}{rc} M^* \in & \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}\,:\\ P_{i+} = r_i,\, P_{+j} = c_j\ \forall i,j}}{\operatorname{argmin}} \quad D_{\mathrm{KL}}(P\|M) \end{array}$$

Note that this result is already known in a different form, referred to as equivalence scaling (see Section 3. Different approaches to equivalence scaling in (Idel, 2016)). We do not claim to introduce a new result with this lemma; rather, we were unable to find this particular formulation in the literature.

H.1 Problem Reformulation

We use the method of Lagrange multipliers. We define the row constraints for $i = 1, \ldots, C$ as:

$$\begin{array}{rccc} h_i: & \mathbb{R}_{>0}^{C\times C} & \to & \mathbb{R} \\ & P & \mapsto & P_{i+} - r_i \end{array}$$

and the column constraints for $j = 1, \ldots, C$ as:

$$\begin{array}{rccc} g_j: & \mathbb{R}_{>0}^{C\times C} & \to & \mathbb{R} \\ & P & \mapsto & P_{+j} - c_j \end{array}$$

Since adding a constant to the objective function does not affect the arguments of the minima, we define $f(P) := D_{\mathrm{KL}}(P\|M) - 1$ to simplify the derivative calculations.

The optimization problem becomes:

$$\begin{array}{rc} M^* \in & \underset{\substack{P \in \mathbb{R}_{>0}^{C\times C}\,:\\ h_i(P) = g_j(P) = 0\ \forall i,j}}{\operatorname{argmin}} \quad f(P) \end{array}$$

The functions f, h_i, and g_j are continuously differentiable.

H.2 Karush-Kuhn-Tucker Conditions

To find a solution, we apply the Karush-Kuhn-Tucker (KKT) conditions (see *Section 5.5.3: KKT Optimality Conditions* in (Boyd and Vandenberghe, 2004)). These conditions are necessary for a point M^* to be a local minimum when a constraint qualification is satisfied. In our case, the Linearity Constraint Qualification (LCQ) holds, since the constraints h_i and g_j are affine.

Moreover, since f is convex (as shown in a subsection below), and the constraint set is convex, the KKT conditions are also sufficient for optimality (Boyd and Vandenberghe, 2004). Thus, any point satisfying the KKT conditions is a global minimum.

In other words, any feasible matrix P^* and multipliers μ^* and ν^* satisfying:

$$\nabla_P L(P^*, \mu^*, \nu^*) = 0$$

is a global minimum, where the Lagrangian is given by:

$$L(P, \mu, \nu) = f(P) + \sum_{i=1}^{C} \mu_i h_i(P) + \sum_{j=1}^{C} \nu_j g_j(P)$$

Since f is strictly convex (Subsection *Convexity of f*), this minimum is also unique.

H.3 Solution

Let P be an admissible point. We compute the gradient of the Lagrangian[10]:

$$\nabla_P L(P, \mu, \nu)_{ij} = \left[\ln \left(\frac{P_{ij}}{M_{ij}} \right) + \mu_i + \nu_j \right]_{ij}$$

Setting the gradient to zero gives:

$$\ln \left(\frac{P_{ij}}{M_{ij}} \right) + \mu_i + \nu_j = 0 \quad \Leftrightarrow \quad P_{ij} = \exp(-\mu_i) M_{ij} \exp(-\nu_j)$$

We define diagonal matrices D^l and D^r with entries:

$$(D^l)_{ii} = \exp(-\mu_i), \qquad (D^r)_{jj} = \exp(-\nu_j)$$

Then the solution takes the form:

$$P = D^l M D^r$$

In conclusion, any matrix of the form $D^l M D^r$ that satisfies the given row and column constraints is the unique solution to the minimization problem.

H.4 Convexity of f

Let $P, P' \in \mathbb{R}_{>0}^{C \times C}$ be two distinct matrices, and let $\gamma \in (0, 1)$. The function f is strictly convex if and only if

$$f(\gamma P + (1-\gamma)P') < \gamma f(P) + (1-\gamma) f(P').$$

We now prove this inequality. We first show that the KL divergence is strictly convex in P when M is fixed.

$$\begin{aligned} D_{\mathrm{KL}}(\gamma P + (1-\gamma)P' \| M) &= \sum_{ij} \left(\gamma P + (1-\gamma)P' \right) \ln \left(\frac{\gamma P + (1-\gamma)P'}{\gamma M + (1-\gamma)M} \right) \\ &< \sum_{ij} \gamma P \ln \left(\frac{P}{M} \right) + (1-\gamma) P' \ln \left(\frac{P'}{M} \right) \\ &= \gamma D_{\mathrm{KL}}(P \| M) + (1-\gamma) D_{\mathrm{KL}}(P' \| M) \end{aligned}$$

where the strict inequality follows from the log-sum inequality, which becomes strict when $P \neq P'$. This implies that f is strictly convex, which completes the proof.

[10] Though the gradient is a vector, we index it here by (i, j) for clarity.