Divide and Imitate: Multi-Cluster Identification and Mitigation of Selection Bias

Supplementary Material

In addition to our paper, we provide (i) a technical appendix (Sec. A) containing further details of our method, Mimic, as well as proofs to claims made, (ii) dataset descriptions and explanations on the bias generation (Sec. B), (iii) additional experimental results (Sec. C), and (iv) our full Python+SciPy implementation together with all scripts, results and plots under:
dropbox.com/sh/ce6jl7aw6edio3e/AAA9EMTduTet2VJ1oPg8Kabwa?dl=0

A Technical Appendix

In this appendix, we provide proofs for the claims made in the paper, i.e., in Section 4.

From Univariate to Multivariate Gaussian. The Imitate algorithm provides us with estimates of univariate Gaussians (μ_i, σ_i^2) for each of the d independent components i, that is, with univariate densities

$$f_i(x_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x_i - \mu}{\sigma}\right)^2\right].$$

The joint probability function f for independent densities $f_1, \ldots, f_d$ is the product $f(x) = \prod_i f_i(x_i)$ for a data point $x \in \mathbb{R}^d$. This term can be transformed to

$$\begin{aligned}
f(x) &= \prod_{i=1}^{d} f_i(x_i) \\
&= \prod_{i=1}^{d} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right] \\
&= \frac{1}{(\prod_i \sigma_i) \cdot (\sqrt{2\pi})^d} \exp\left[\sum_i -\frac{1}{2}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right] \\
&= \frac{1}{\sqrt{(\prod_i \sigma_i^2) \cdot (2\pi)^d}} \exp\left[-\frac{1}{2}\sum_i \frac{(x_i - \mu_i)^2}{\sigma_i^2}\right] \\
&= \frac{1}{\sqrt{\det \mathit{\Sigma} \cdot (2\pi)^d}} \cdot \exp\left[-\frac{1}{2}(x - \mu)^{\mathrm{T}} \begin{bmatrix} \frac{1}{\sigma_1^2} & & \\ & \ddots & \\ & & \frac{1}{\sigma_d^2} \end{bmatrix} (x - \mu)\right] \\
&= \frac{1}{\sqrt{\det \mathit{\Sigma} \cdot (2\pi)^d}} \exp\left[-\frac{1}{2}(x - \mu)^{\mathrm{T}} \mathit{\Sigma}^{-1} (x - \mu)\right]
\end{aligned}$$

for $\mu = (\mu_1, \ldots, \mu_d)$ and $\Sigma \in \mathbb{R}^{d \times d}$ with diagonal $(\sigma_1^2, \ldots, \sigma_d^2)$ and 0 elsewhere. This is the density formula of a multivariate Gaussian parameterized by (μ, Σ).

ICA Back-Transformation. Assuming that a multivariate Gaussian (μ, Σ) has been found in the ICA-space obtained by transformation $x \mapsto Ix =: x'$ for a data point $x \in \mathbb{R}^d$ in the original space and the ICA matrix $I \in \mathbb{R}^{d \times d}$, we need to transform the Gaussian back to the original data space. We can now insert the transformation term into the definitions of mean and covariance matrix and exploit the linearity of the expectation:

$$\begin{aligned}
\mu &= \mathbb{E}[x'] = \mathbb{E}[Ix] = I\mathbb{E}[x] \\
\Leftrightarrow I^{-1}\mu &= \mathbb{E}[x] \\
\Sigma &= \mathbb{E}\left[(x' - \mathbb{E}[x'])^{\mathrm{T}}(x' - \mathbb{E}[x'])\right] \\
&= \mathbb{E}\left[(Ix - \mathbb{E}[Ix])^{\mathrm{T}}(Ix - \mathbb{E}[Ix])\right] \\
&= I\mathbb{E}\left[(x - \mathbb{E}[x])^{\mathrm{T}}(x - \mathbb{E}[x])\right] I^{\mathrm{T}} \\
\Leftrightarrow I^{-1}\Sigma(I^{\mathrm{T}})^{-1} &= \mathbb{E}\left[(x - \mathbb{E}[x])^{\mathrm{T}}(x - \mathbb{E}[x])\right]
\end{aligned}$$

which yields a Gaussian in the original space with parameters $(I^{-1}\mu, I^{-1}\Sigma(I^{\mathrm{T}})^{-1})$.

Likelihood of Model Given Data. Mimic is searching for those candidate data points C that, when added to a preliminary cluster B_L, improve the likelihood of the model (i.e., the parameters θ describing the cluster) given the data (i.e., the already assigned data points together with the candidates; $B_L \cup C$). In other words, we aim to find $\arg\max_C \mathbb{P}[\theta \mid B_L \cup C]$. For the sake of brevity and readability, we denote $X_C := B_L \cup C$. In order to avoid underflow errors, we use logarithms. Exploiting Bayes' Theorem and since log-transformation preserves maxima, this term can expressed as:

$$\begin{aligned}
&\arg\max_C \mathbb{P}[\theta \mid X_C] \\
&\qquad = \arg\max_C \log\left(\mathbb{P}[\theta \mid X_C]\right) \\
&\qquad = \arg\max_C \log\left(\frac{\mathbb{P}[X_C \mid \theta] \cdot \mathbb{P}[\theta]}{\mathbb{P}[X_C]}\right) \\
&\qquad = \arg\max_C \log\left(\frac{\mathbb{P}[X_C \mid \theta]}{\mathbb{P}[X_C]}\right) \\
&\qquad = \arg\max_C \left(\log \mathbb{P}[X_C \mid \theta] - \log \mathbb{P}[X_C]\right)
\end{aligned}$$

since $\mathbb{P}[\theta]$ does not depend on C and can hence be omitted in $\arg\max_C$.

Using the Imitate output, we obtain histogram values h and fitted Gaussian densities g over a grid. With these, the first term, $\log \mathbb{P}[X_C \mid \theta]$, can be approximated via the grid representation as follows:

$$\begin{aligned}\log \mathbb{P}[X_C \mid \theta] &= \log \prod_{p \in X_C} \mathbb{P}[p \mid \theta] \\ &\approx \log \prod_{\text{grid cells } c} g(c)^{h(c)} \\ &= \sum_{\text{grid cells } c} h(c) \cdot \log g(c) \end{aligned} \tag{1}$$

which is a term that can be efficiently computed without the risk of underflow errors. The second term, $\log \mathbb{P}[X_C]$, can be transformed into $\log \int_{\theta_i} \mathbb{P}[X_C \mid \theta_i] \cdot \mathbb{P}[\theta_i] \,\mathrm{d}\theta_i$ using the law of total probability. We simplify the term by assuming that only one data generating model exists that we parameterize as follows: The mean μ_0 is the grid center, and the covariance matrix Cov_0 is set up as a diagonal matrix ensuring the grid borders are the minimal axis-aligned bounding box for a Gaussian around μ_0 truncated at the usual 3 standard deviations. We denote the parameters of the Gaussian (μ_0, Cov_0) as θ_0 and obtain the simplified term $\log \mathbb{P}[X_C \mid \theta_0]$. By evaluating the corresponding probability density f_0 for all grid cell centers, this term can be calculated similar to Eq. 1 as follows:

$$\begin{aligned}\log \mathbb{P}[X_C] &\approx \log \mathbb{P}[X_C \mid \theta_0] \\ &\approx \log \prod_{p \in X_C} f_0(\text{cell\_center}(p)) \cdot s \\ &= |X_C| \cdot \log s + \sum_{p \in X_C} \log f_0(\text{cell\_center}(p)) \end{aligned}$$

where s is the size of each grid cell. Although only an approximate calculation, this term is able to balance off Equation 1 and yields easily computable meaningful results.

Merging. For every pair of clusters i and j, the overlap $o(i, j)$ can be quantified by counting the points in the dataset for which the cluster membership is not entirely clear and weighting them using their probabilities:

$$o(i, j) = \frac{\sum_p \mathbb{P}[p \mid \theta_i] \mathbb{1}_{\mathbb{P}[p \mid \theta_i] < \alpha \mathbb{P}[p \mid \theta_j]}}{\sum_p \mathbb{P}[p \mid \theta_i]}$$

for a factor $\alpha > 1$ ($\alpha = 10$ in our implementation) where $\mathbb{1}$ denotes the indicator function. Note that o is not symmetric in its arguments which follows the intuition of overlap. Imagine two clusters in 2D arranged like a fried egg: while the yolk fully overlaps with the eggwhite, the reverse direction would not hold true.

Based on the parameters for all clusters, Mimic calculates the overlap between each (ordered) pair of clusters. The ones with high symmetric overlap (that is, $o(i, j) > \beta$ and $o(j, i) > \beta$ for $\beta \in [0, 1]$; $\beta = 0.8$ in our implementation) are merged right away into a cluster with parameters $((\mu_i + \mu_j)/2, (\Sigma_i + \Sigma_j)/2)$ since they can be expected to be duplicates. All other clusters with a small, possibly one-sided overlap (that is, $o(i, j) > \gamma$ and $o(j, i) > \gamma$ for a small $\gamma \in [0, \beta]$)

Dataset	**Predicted Attr.**	**Biased Set B**	**Omitted Features**
Wholesale Customers	Region	Frozen > 409.5	Channel
Vertebral Column	Normal/Abnormal	Spondylolisthesis Grade ≤ 14.855	-
Banknote	Class	Variance > 0.32	-
Diabetes (130 US hospitals)	Diabetes Med	#Medications > 9.5	All but Age*, #LabProcedures, #Procedures, #Medications, #Outpatient, #Emergency, #Inpatient, #Diagnoses
Skin Segmentation	Class	R ≤ 170.5	R**

Table 1. Description of real-world datasets used in the experiments. *The categorical values were transformed into numerical variables. **This attribute was omitted to achieve consistency with the Imitate paper.

are merged only if, after a probabilistic cluster assignment, the Imitate fitting error e of the merged cluster is lower than the weighted sum of the individual errors, i.e., if $e_{i \cup j} < (\#i \cdot e_i + \#j \cdot e_j)/(\#i + \#j)$ where $\#i$ counts the points with label i. In order to address the randomness of the involved ICA, we repeat this test 10 times and use a majority vote for the final decision. The merging procedure is repeated until no further clusters are merged.

In our implementation, we use hardcoded $\alpha = 10$, $\beta = 0.8$, and $\gamma = 0.2$. Note that these values reduce the number of merge tests that need to be carried out and hence reduce the computational burden while sacrificing little to no quality. Parameter tuning is not necessary as the merge tests determine the results, not the parameters.

B Datasets

For our experiments, we used synthetic as well as real-world datasets. Synthetic datasets are generated using a specified number of clusters per class and dimensions. Each cluster is generated as a multivariate Gaussian with random covariance matrix and mean. All means are generated within the unit cube and pushed away from the center using a parameter that controls the *spread* of the clusters. If not explicitely mentioned, we used a medium spread of 100. Biases are created as described in [5] for two randomly selected dimensions per cluster: A hyperplane is rotated through the cluster center by a random angle. Data points above that plane associated with the cluster are omitted in B. Note that this is a hard bias which we decided to use as it challenges our method further (see the Imitate paper for a study on the impact). In order to ensure that the bias has an impact on the classification accuracy, we select only those randomly generated datasets and biases that inflict at least a 10% accuracy drop with the SVM classifier. We generated datasets of size 5000 and provide all methods, parameters, and seeds necessary for the generation in the provided code. The real-world datasets are all taken from the UCI Machine Learning Repository. We specify the predicted attribute as well the created bias in Table 1.

C Additional Experimental Results

Due to space limitations, we show compressed versions of the results in our paper. That means that variances are often omitted in order to combine several plots into one. We provide the extended results (including variances) here.

Unbiased Datasets. We counted the number of points being generated for unbiased and corresponding biased 2D datasets with random spreads between 100 and 200, and we normalized the counts with the dataset size for comparability. Figure 1 shows that substantially less data points are generated for the unbiased datasets. We suspected that these data points result from histogram inaccuracies and confirmed that suspicion by applying Imitate's purging strategy (that is, it removes the generated data points that are not distributed densely enough): the generated points for the unbiased datasets do not focus on certain areas and are hence removed as noise. Almost no points remain on the unbiased dataset while there are about 20% of generated points left on the biased datasets.

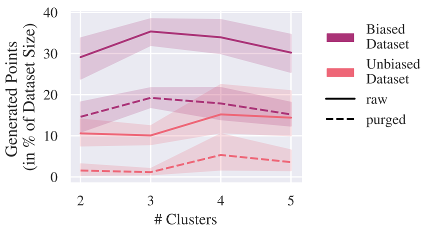

Fig. 1. Comparison of Mimic's behavior with present and absent bias.

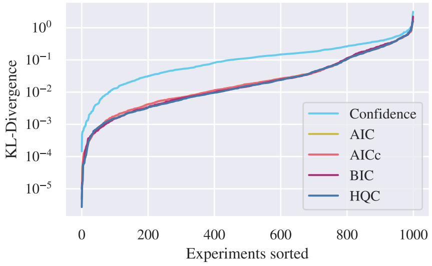

Fig. 2. Test of different information criteria to determine the optimal number of histogram bins for Imitate

Dimensionality. Figure 3 shows the variance for each line contained in Figure 3 of the main paper.

Cluster Overlap. The center-to-cluster distances of the clusters directly affect the difficulty of the clustering task as they control the overlap. In order to investigate the influence, we adjust the spread parameter in the dataset generation. Figure 4 shows the compressed version of our experimental results. The variance can be seen in Figure 5.

Information Criteria for Histogram Bins. As mentioned in Section 4 – "Adapting Imitate to Our Needs", choosing an appropriate number of histogram bins for the Imitate algorithm is crucial to balance off accuracy versus the danger to overfit. The original paper uses a confidence measure that evaluates the final output to determine a suitable number of bins. Given the large amount of Imitate calls used when executing Mimic, this procedure is not feasible and

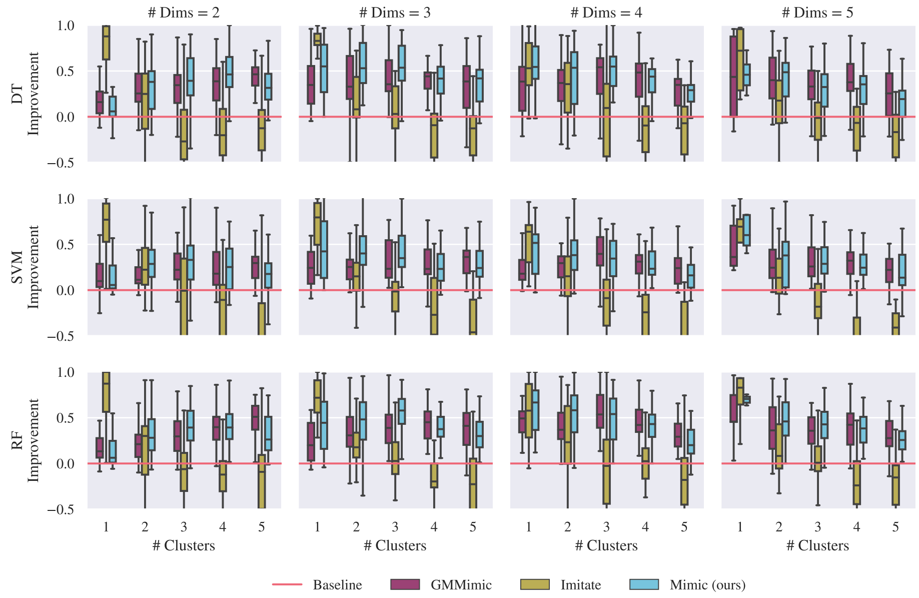

Fig. 3. Extended dimensionality experiment (Main paper Fig. 3)

we decided to use an information criterion that selects an appropriate number of bins.

In preliminary experiments, we investigate the quality of the selected number of bins for several information criteria: the Akaike information criterion (AIC) and its corrected version AICc, the Bayesian information criterion (BIC), and the Hannan-Quinn criterion (HQC). See [8] for the definitions. We generated 1000 standard Gaussian datasets of random size $n \sim \mathcal{U}(5000, 50000)$ with artificial biases removing $h \sim \mathcal{U}(0, 100)\%$ of the data above a threshold $t \sim \mathcal{U}(0, 1)$. For all tested numbers of bins in $\{5, \ldots, 100\}$ we evaluate the KL-divergence between the ground-truth $\mathcal{N}(0, 1)$ and the Gaussian Imitate fits to the histogram representation of the biased data. The results are shown in Fig. 2 and indicate that all information criteria perform better than the confidence strategy described in the original paper.

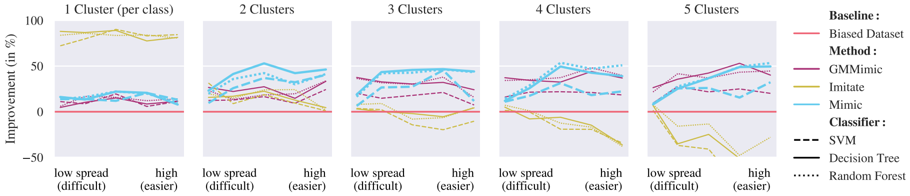

Fig. 4. Datasets in 2D with two classes have been generated with different numbers of clusters per class. The spread (on the x-axes) indicates how much the clusters are being pushed away from the center, and a low spread corresponds with a high overlap. Even with a large number of clusters, Mimic performs consistently well. However, high overlaps seem to be adressed better by GMMimic.

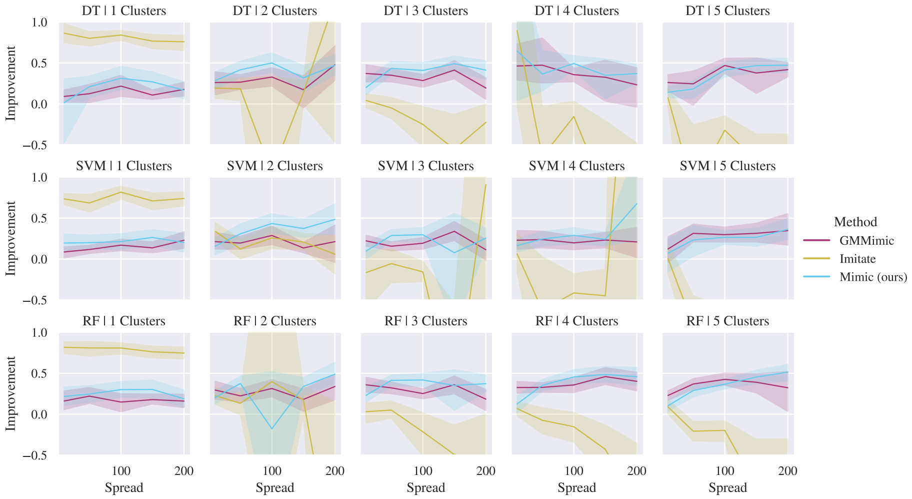

Fig. 5. Extended cluster spread experiment (Fig. 4)