Online Supplement for "Supervised Clustered Interpretability: Explainable Subgroup Discovery via Cluster Separation and Partial Dependence Disparity"

Matt Baucum & Meysam Rabiee

Appendix A: Theoretical analysis of the SD index

A.1. Non-negativity and finiteness

We now show that, for any finite sample, $0 \leq SD(X, \mathbf{c}) < \infty$, given the following assumption:

Assumption 1: *At least two clusters differ in their estimated PD functions, for at least one feature.*

Given this, it follows that for the SD index denominator, $0 < \mathrm{Disp}_X < \infty$, since it is an empirical variance over clusters' PD functions. Furthermore, the numerator terms Sep_X and $\mathrm{Sep}_{\hat{y}}$ are empirical variances, and therefore $0 \leq \mathrm{Sep}_X, \mathrm{Sep}_{\hat{y}} < \infty$. The ratio $\frac{(\mathrm{Sep}_X + \mathrm{Sep}_{\hat{y}})/2}{\mathrm{Disp}_X}$ is therefore nonnegative.

A.2. Monotonicity

With Disp_X held fixed, $SD(X, \mathbf{c})$ is strictly non-decreasing in both Sep_X and $\mathrm{Sep}_{\hat{y}}$; with the numerator fixed it is strictly non-increasing in Disp_X. Therefore, any clustering refinement that raises separation without raising disparity, or that reduces disparity without reducing separation, will increase the SD index.

A.3. Behavior under optimization

Because the SD index is optimized via metaheuristic algorithms, its improvement during optimization arises from the selection of candidate solutions with higher index values than the incumbent solution(s). We now analyze the conditions under which a candidate solution will be chosen over an incumbent solution.

Let $\mathrm{Sep'}_X$, $\mathrm{Sep'}_{\hat{y}}$, and $\mathrm{Disp'}_X$ represent candidate separation and disparity values to be compared with incumbent values Sep_X, $\mathrm{Sep}_{\hat{y}}$, and Disp_X. Moreover, let $\phi = \frac{\frac{1}{2}(\mathrm{Sep'}_X + \mathrm{Sep'}_{\hat{y}})}{\frac{1}{2}(\mathrm{Sep}_X + \mathrm{Sep}_{\hat{y}})}$ and $\psi = \frac{\mathrm{Disp}_X}{\mathrm{Disp'}_X}$ indicate the improvement factors in overall separation and disparity, respectively, such that $\phi > 1$ indicates improving (increasing) separation and $\psi > 1$ indicates improving (decreasing) disparity.

The improvement factor for the SD index during optimization, incorporating the user-set λ discount, is therefore $\frac{\frac{1}{2}(\mathrm{Sep'}_X + \mathrm{Sep'}_{\hat{y}})}{(\mathrm{Disp'}_X)^\lambda} \Big/ \frac{\frac{1}{2}(\mathrm{Sep}_X + \mathrm{Sep}_{\hat{y}})}{(\mathrm{Disp}_X)^\lambda} = \phi\psi^\lambda$. Thus, when $\phi\psi^\lambda > 1$, the candidate solution will be selected over the incumbent solution. As an example, consider a candidate solution that reduces separation by 40% ($\phi = 0.6$), and cuts disparity in half ($\psi = 2$). For $\lambda = 1$, $\phi\psi^\lambda = (0.6)(2)^1 = 1.2$, and this solution would be selected over the incumbent. If $\lambda = 0.5$, then $\phi\psi^\lambda = (0.6)(2)^{0.5} = 0.849$, and this solution would not be selected. Thus, lower λ values raise the threshold of necessary disparity improvements for solution selection.

A.4. Stability to data perturbations

Let datasets X and X' differ in exactly one row $\boldsymbol{x}_\ell$ for $\ell \in \{1, ..., N\}$, such that $\boldsymbol{x}_\ell$ is the original data point and $\boldsymbol{x}'_\ell$ is the perturbed data point. We assume $\boldsymbol{x}'_\ell$ and $\boldsymbol{x}_\ell$ both belong to cluster $C_k^{(\ell)}$. Let $F'()$ denote the trained ML model on the perturbed dataset X', $PD_j^{'(k)}$ denote its cluster-specific feature effects, and $\bar{PD'}_j$ denote its averaged (global) feature effects. We now bound the change between $SD(X, \mathbf{c})$ and $SD(X', \mathbf{c})$.

Assumption 2: Uniform stability of feature effects and model estimator. *We assume that the additive feature effects of $F()$ have uniform stability under single-row perturbation. Specifically, there is some β for which $|PD_j^{'(k)}(x_{ij}) - PD_j^{(k)}(x_{ij})| \leq \beta$ for all i and $|PD_j^{'(k)}(x'_{\ell j}) - PD_j^{(k)}(x'_{\ell j})| \leq \beta$. We also assume $|F'(\boldsymbol{x}_i) - F(\boldsymbol{x}_i)| \leq \eta \leq J\beta$ for all i and $|F'(\boldsymbol{x}_\ell) - F(\boldsymbol{x}_\ell)| \leq \eta \leq J\beta$ for all i.*

Assumption 3: Perturbation bounds.. *We assume that $|PD_j^{(k)}(x'_{\ell j}) - PD_j^{(k)}(x_{\ell j})| \leq \Delta$ for all features j and clusters k. That is, we assume that the difference between the (original) PD functions evaluated on $\boldsymbol{x}_{\ell j}$ and $\boldsymbol{x}'_{\ell j}$ differ at most by Δ. Similarly, we assume $|F(\boldsymbol{x}'_\ell) - F(\boldsymbol{x}_\ell)| \leq \gamma \leq J\Delta$.*

Lemma 1: *Let $\mathbf{z}$ and $\mathbf{z}'$ be two vectors for which all $|z_i - z'_i| \leq d$. Because $|z'_i - \mu_{\mathbf{z}'}| \leq |z_i - \mu_{\mathbf{z}}| + 2d$, then $\mathrm{Var}(\mathbf{z}') = \mathrm{Var}(\mathbf{z}) + 4d^2 + 4d\sum |z_i - \mu_{\mathbf{z}}|/N \leq \mathrm{Var}(\mathbf{z}) + 4d^2 + 4d\sigma_{\mathbf{z}}$, where $\sigma_{\mathbf{z}} = \sqrt{\mathrm{Var}(\mathbf{z})}$. Therefore, $|\mathrm{Var}(\mathbf{z}) - \mathrm{Var}(\mathbf{z}')| \leq 4d^2 + 4d\sigma_{\mathbf{z}}$.*

Proof: Using Assumption 3 and Lemma 1, each of the SD index terms can be found as follows:

1. **Change in $\mathbf{Sep}_X$**: The original and perturbed cluster- and feature-specific average PD value, $\frac{1}{|C_k|}\sum_{i \in C_k} \bar{PD}_j(x_{ij})$ and $\frac{1}{|C_k|}\sum_{i \in C_k} \bar{PD'}_j(x_{ij})$, will differ by no more than $\beta + \frac{\Delta}{|C_k^{(\ell)}|}$. This is because $|\bar{PD}_j(x_{ij}) - \bar{PD'}_j(x_{ij})| \leq \beta$ for $i \neq \ell$ (i.e., model change only), and $|\bar{PD}_j(x_{\ell j}) - \bar{PD'}_j(x_{\ell j})| \leq \beta + \Delta$ (i.e., model change and value perturbation), with the latter change being averaged over its cluster $C_k^{(\ell)}$. If we let $\sigma_{\max}^{(X)} = \max_j \sqrt{\mathrm{Var}_k(\frac{1}{|C_k|}\sum_{i \in C_k} \bar{PD}_j(x_{ij}))}$, then by Lemma 1:
$$|\mathrm{Sep}_{X'} - \mathrm{Sep}_X| \leq 4(\beta + \frac{\Delta}{|C_k^{(\ell)}|})^2 + 4\sigma_{\max}^{(X)}(\beta + \frac{\Delta}{|C_k^{(\ell)}|}) \tag{4}$$
2. **Change in $\mathbf{Sep}_{\hat{y}}$**: The original and perturbed cluster-specific average predictions, $\frac{1}{|C_k|}\sum_{i \in C_k} F(\boldsymbol{x}_i)$ and $\frac{1}{|C_k|}\sum_{i \in C_k} F'(\boldsymbol{x}_i)$, will differ by no more than $\eta + \frac{\gamma}{|C_k^{(\ell)}|}$, by the same logic as for Sep_X. If we let $\sigma^{(\hat{y})} = \sqrt{\mathrm{Var}_k \frac{1}{|C_k|}\sum_{i \in C_k} F(\boldsymbol{x}_i)}$, then by Lemma 1:
$$\left|\mathrm{Sep}_{\hat{y}'} - \mathrm{Sep}_{\hat{y}}\right| \leq \frac{4}{J}(\eta + \frac{\gamma}{|C_k^{(\ell)}|})^2 + \frac{4}{J}\sigma^{(\hat{y})}(\eta + \frac{\gamma}{|C_k^{(\ell)}|}) \tag{5}$$
3. **Change in $\mathbf{Disp}_X$**: The original and perturbed cluster-specific, feature-specific PD values $PD_j^{'(k)}(x_{ij})$ and $PD_j^{(k)}(x_{ij})$ will differ by no more than β for rows $i \neq \ell$ and by up to $\beta + \Delta$ for row ℓ. If we let $\sigma_{\max(PD)} = \max_{i,j} \sqrt{\mathrm{Var}_k(PD_j^{(k)}(x_{ij}))}$, then
$$|\mathrm{Var}_k\left(PD_j^{'(k)}(x_{ij})\right) - \mathrm{Var}_k\left(PD_j^{(k)}(x_{ij})\right)| \leq 4\beta^2 + 4\sigma_{\max}^{(PD)}\beta \quad \forall i \neq \ell \tag{6}$$
$$|\mathrm{Var}_k\left(PD_j^{'(k)}(x_{\ell j})\right) - \mathrm{Var}_k\left(PD_j^{(k)}(x_{\ell j})\right)| \leq 4(\beta + \Delta)^2 + 4\sigma_{\max}^{(PD)}(\beta + \Delta) \tag{7}$$

Multiplying out the $(\beta + \Delta)^2$ and aggregating over J and N leads to

$$|\text{Disp}_{X'} - \text{Disp}_X| \leq 4\beta^2 + 4\sigma_{\max}^{(PD)}\beta + \frac{4}{N}[2\beta\Delta + \Delta^2 + \sigma_{\max}^{(PD)}\Delta] \tag{8}$$

4. **Relative difference between $SD(X', \mathbf{c})$ and $SD(X, \mathbf{c})$**: The relative difference between $SD(X', \mathbf{c})$ and $SD(X, \mathbf{c})$ can be bounded by the sum of the absolute relative differences in its numerator and denominator, times the ratio $\frac{\text{Disp}_X}{\text{Disp}_{X'}}$:

$$\left|\frac{SD(X', \mathbf{c}) - SD(X, \mathbf{c})}{SD(X, \mathbf{c})}\right| \leq \frac{\text{Disp}_X}{\text{Disp}_{X'}} \times \left(\frac{4\beta^2 + 4\sigma_{\max}^{(PD)}\beta + \frac{4}{N}[2\beta\Delta + \Delta^2 + \sigma_{\max}^{(PD)}\Delta]}{\text{Disp}_X} + \right. \tag{9}$$

$$\left. \frac{\frac{1}{2}\left([4(\beta + \frac{\Delta}{|C_k^{(\ell)}|})^2 + 4\sigma_{\max}^{(X)}(\beta + \frac{\Delta}{|C_k^{(\ell)}|})] + [\frac{4}{J}(\eta + \frac{\gamma}{|C_k^{(\ell)}|})^2 + \frac{4}{J}\sigma^{(\hat{y})}(\eta + \frac{\gamma}{|C_k^{(\ell)}|})] \right)}{\frac{1}{2}(\text{Sep}_X + \text{Sep}_{\hat{y}})} \right) \tag{10}$$

Finally, assuming that $|\text{Disp}'_X - \text{Disp}_X| \ll \text{Disp}_X$, the ratio $\frac{\text{Disp}_X}{\text{Disp}_{X'}}$ can be approximated as one.

Given this, the SD index change under perturbation is $O(\beta^2) + O(\beta/N) + O(1/N) + O(\beta/|C_k^{(\ell)}|) + O(1/|C_k^{(\ell)}|^2) + O(\eta^2/J) + O(1/(J|C_k^{(\ell)}|^2)) + O(\eta/(J|C_k^{(\ell)}|))$, treating the single-point perturbation bounds Δ and γ as fixed. Thus, the change in SD index is decreasing with the square of the perturbed row's cluster size $|C_k^{(\ell)}|$, the feature count J, and sample size N, and increasing quadratically with the model stability bounds β and η. The former is to be expected since the influence of any point will diminish as the sample and cluster sizes increase, and the latter is to be expected given the variance computation over model-derived predictions.

A.5. Asymptotic consistency (under stable cluster labels)

We show that the SD index converges in probability to its population value with N. We assume that rows of X are independent and identically distributed (iid), and are assigned to the same clusters they would be under a population model. We rely on the Continuous Mapping Theorem (CMT), which states that if a sequence of random variables converges in probability, any continuous function of that sequence also converges in probability to the same function of the limit.

Assumption 4: Consistency of additive PD functions $PD_j^{(k)}$: *We assume that the PD functions of our additive ML model are consistent, such that for any data value x_{ij},*

$$\lim_{n\to\infty} P(|PD_j^{(k)}(x_{ij}) - \tilde{PD}_j^{(k)}(x_{ij})| > \epsilon) = 0 \quad \forall \epsilon > 0 \tag{11}$$

where $\tilde{PD}$ denotes the feature effect under the ground-truth data generating function. For tree-based additive models, which we use for our analyses, the PD functions are the output of single-feature (or bivariate

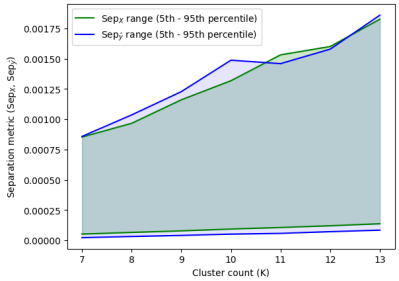

Figure 14 Range of simulated values for Sep_X and $\text{Sep}_{\hat{y}}$.

interaction) tree estimators. Note that past work has demonstrated consistency for random forests (Scornet et al. 2015) and decision trees approximating additive data-generating functions (Klusowski 2021).

Proof: Given Assumption 4, the following components of the SD index are consistent by the CMT:

1. **Consistency of global PD functions and model predictions:** The average cross-cluster feature effects $\bar{PD}_j()$ are consistent, given that they are an empirical average of the $PD_J^{(k)}$. Similarly, model predictions $F()$, which are the sum of feature effects for each row, are also consistent.
2. **Consistency of Sep_X and $\text{Sep}_{\hat{y}}$:** Sep_X and $\text{Sep}_{\hat{y}}$ are consistent, given that they are continuous functions (combination of averages and variances) of $\bar{PD}_j()$ and $F()$, respectively.
3. **Consistency of Disp_X:** Disp_X is consistent, given it is a continuous function of $PD_j^{(k)}()$.
4. **Consistency of the SD index:** The SD index ratio is a well-defined and continuous function of the separation and disparity components as long as Disp_X does not converge to zero. If Assumption 1 (at least two clusters differ in their PD functions) holds for the population, then Disp_X will converge to a positive values, in which case the SD index is consistent by the CMT.

Appendix B: Comparison of Sep_X and $\text{Sep}_{\hat{y}}$ ranges

To justify the SD index's averaging of Sep_X and $\text{Sep}_{\hat{y}}$, we compute Sep_X and $\text{Sep}_{\hat{y}}$ over 500 randomly-initialized cluster centroids for cluster counts $K \in \{7, ..., 13\}$ on our ten-cluster datasets. We then plot the 5th and 9th percentiles of Sep_X and $\text{Sep}_{\hat{y}}$ as a function of K across these random simulations. Figure 14 plots these results, and suggests that Sep_X and $\text{Sep}_{\hat{y}}$ take on almost identical ranges of possible values.

Appendix C: Comparison of main results across disparity discount values λ

We provide an in-depth comparison of our main results trained with disparity discount values of $\lambda \in \{0.25, 0.5, 1\}$, followed by an overall evaluation of our method's behavior across a wide range of λ values.

Table 9 presents our main results trained with disparity discount values of $\lambda \in \{0.25, 0.5, 1\}$, along with a K-means benchmark for reference. As expected, solutions' final disparity values increase as λ decreases (because lower λ values down-weight the importance of reducing disparity during training).

SD index values are very similar between the $\lambda = 0.5$ and $\lambda = 1$ solutions. This is likely because the final SD index values are evaluated on a fully-trained ML model (rather than regression estimates of PD functions); thus, solutions that discount disparity during training can still achieve desirable disparity scores on out-of-sample evaluation. Notably, $\lambda = 1$ achieves the result via lower disparity and slightly lower separation, whereas $\lambda = 0.5$ preserves separation while accepting modestly higher disparity.

Finally, these results suggest that λ can be used to control the PD function heterogeneity of the final cluster solution. Figure 15 visualizes the PD functions for the $\lambda = 0.25$ solution, which can be compared with the PD functions from Figure 5. The $\lambda = 0.25$ solution introduces more heterogeneity between clusters' PD functions, especially for features 1 and 2, and the lower quantiles of feature 4, while still offering lower disparity than the K-means benchmark, and better separation than the interpretable tree-based benchmark.

Table 9 Synthetic data results for $\lambda \in \{0.25, 0.5, 1\}$, with a K-means benchmark for reference.

	SD index	**Separation**			**Disparity**
		Centroid Sep (X)	PD Centroid Sep (PD-X)	Outcome Sep (y)	PD Disparity
PSO, $\lambda = 1$	1.225 (0.917, 1.635)	0.49 (0.027)	0.088 (0.006)	0.495 (0.046)	0.106 (0.011)
PSO, $\lambda = 0.5$	1.265 (0.786, 2.036)	0.532 (0.027)	0.097 (0.008)	0.589 (0.047)	0.123 (0.016)
PSO, $\lambda = 0.25$	0.614 (0.416, 0.906)	0.542 (0.017)	0.11 (0.005)	0.598 (0.1)	0.167 (0.016)
K-means	0.238 (0.151, 0.376)	0.548 (0.008)	0.1 (0.005)	0.482 (0.069)	0.218 (0.01)

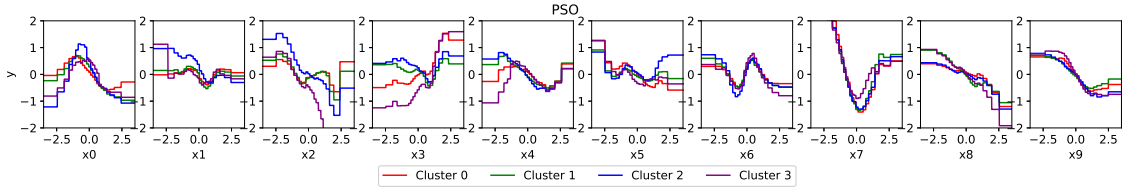

Figure 15 Synthetic data PD functions for SD-PSO under $\lambda = 0.25$.

To demonstrate the behavior of separation and disparity values in response to λ, we re-run our main results for 20 values of $\lambda \in \{0.05, 0.1, ..., 0.95, 1\}$ across our ten synthetic datasets. For each run, we calculate the final solution's separation and disparity scores on our out-of-sample test set. Figure 16 demonstrates that square-root transformed separation (left) and disparity (right) scores decrease linearly with λ. Regressing each on λ yields significant negative linear terms ($p < 0.001$), with quadratic terms non-significant.

Importantly, we find that disparity's dependence on λ can be predicted using only the most extreme λ values. That is, if we define b_{Disp} as the slope predicting out-of-sample $\sqrt{Disp_X}$ from λ, then it can be

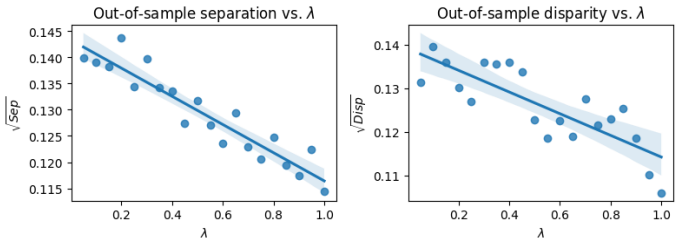

Figure 16 Out-of-sample separation and disparity by λ (averaged across ten synthetic datasets)

estimated by $\hat{b}_{\text{Disp}} = (\sqrt{\text{Disp}_{\lambda^{(max)}}} - \sqrt{\text{Disp}_{\lambda^{(min)}}})/(\lambda^{(max)} - \lambda^{(min)})$, where $\lambda^{(max)}$ and $\lambda^{(min)}$ are the highest and lowest tested λ values (in our case, 1 and 0.05). For our main results, b_{Disp} and $\hat{b}_{\text{Disp}}$ correlate at 0.84. Given this, analysts can use the following procedure to calibrate λ to their desired level of disparity:

1. Run SD-PSO with $\lambda^{(max)} = 1$ and a small λ (e.g., $\lambda^{(min)} = 0.05$); ensure $\text{Disp}_{\lambda^{(max)}} < \text{Disp}_{\lambda^{(min)}}$.
2. Compared with the $\lambda = 1$ solution, decide on the percentage Γ by which disparity should increase (e.g., $\Gamma = 20\%$ implies a desired disparity increase of 20%).
3. Calculate the estimated slope $\hat{b}_{\text{Disp}} = (\sqrt{\text{Disp}_{\lambda^{(max)}}} - \sqrt{\text{Disp}_{\lambda^{(min)}}})/(\lambda^{(max)} - \lambda^{(min)})$.
4. Calculate the desired increase in square-root disparity as $\delta = \sqrt{(1+\Gamma)\text{Disp}_{\lambda^{(max)}}} - \sqrt{\text{Disp}_{\lambda^{(max)}}}$.
5. Finally, calculate the new disparity discount $\lambda^* = \max\{\lambda^{(min)}, 1 + \frac{\delta}{\hat{b}_{\text{Disp}}}\}$. Because b_{Disp} will be negative given step 1, this formula provides the degree to which λ should decrease from its maximum value of 1 to achieve the desired increase in disparity, as long as it is within the tested range $(\lambda_{\min}, 1)$.

Appendix D: Detailed description of PSO modifications.

D.1. Residual-based PD estimation.

The novelty of our PSO implementation lies in its calculation of the SD index for each particle, without training cluster-specific ML models. We first train a state-of-the-art additive model (the Explainable Boosting Machine, Nori et al. 2019) on the dataset, and compute PD values $PD_j(x_{ij})$ for each point. We then use polynomial ridge regression to fit the residuals to X separately for each cluster. These cluster-specific regression models are then used to estimate cluster-specific PD functions.

D.2. Discounted disparity term during training

We introduce an optimal hyperparameter $\lambda \in (0, 1]$ which discounts the disparity term during training, i.e., $\frac{(\text{Sep}_X + \text{Sep}_{\hat{y}})/2}{(\text{Disp}_X)^{\lambda}}$. This simple modification raises the threshold for disparity improvements that cause a candidate solution to be accepted/preferred. In Online Supplement C, we show that lower λ values can be used to preserve a greater degree of PD function heterogeneity, if desired by the user. We find that as λ decreases, both separation and disparity of the final solutions tend to increase.

D.3. Asymmetric particle updates.

We aim to prevent premature PSO convergence before low-disparity alternatives have been sufficiently explored, which we achieve through *asymmetric particle updates*. In standard PSO, particles update toward their own best solution (PBEST) and the overall best solution (GBEST) with respective velocities governed by θ_1 ('cognition parameter') and θ_2 ('social parameter'). We reduce θ_2 for solutions that achieve lower disparity scores than the GBEST particle, allowing them to 'continue exploring' low-disparity solutions. If Θ_2 is the predefined social parameter value, and $SD^{(min)}$ is the lowest SD index across particles, then the social parameter for a particle p during an epoch is given by

$$\theta_2{}^{(p)} = \begin{cases} \Theta_2 \cdot \left(1 - \frac{SD^{(p)} - SD^{(min)}}{SD^{(GBEST)} - SD^{(min)}}\right), & \text{if } \text{Disp}_X^{(p)} < \text{Disp}_X^{(GBEST)} \\ \Theta_2, & \text{otherwise} \end{cases} \tag{12}$$

D.4. Particle convergence

At any given PSO iteration, each particle will converge to a linear combination of the PBEST and $GBEST$ particles, provided the eigenvalues of the recurrence matrix have magnitude less than one (Van den Bergh and Engelbrecht 2006). This depends on the inertia parameter ω, social parameter θ_1, and cognition parameter θ_2, for which we use $\omega = 0.729$ and $\theta_1, \theta_2 = 1.49$, values which are known to meet this convergence condition (Van den Bergh and Engelbrecht 2006). This applies even if θ_1, θ_2 are merely upper bounds for the social and cognition parameters (Van den Bergh and Engelbrecht 2006); if θ_1, θ_2 are chosen to ensure particle convergence, decreasing either parameter will also lead to particle convergence. Under our asymmetric update procedure, $\theta_2^{(p)} \leq \Theta_2$ (where Θ_2 is a chosen parameter value), ensuring convergent update behavior.

Appendix E: Comparison of PSO and genetic algorithms

We now compare results from our main four-cluster synthetic dataset between PSO and genetic algorithms. The genetic algorithm is run according to our multi-seed procedure (Section 4.1.2) and uses $\lambda = 0.5$ during training (see Online Supplement C). It uses $M = 3$ seeds, crossover rate of 0.7, mutation rate of 0.02, tournament size of 5, and population size of 25 (Maulik and Bandyopadhyay 2000), trained for 25 generations.

Because solutions do not 'move' toward each other in a genetic algorithm in the same way as PSO, we adapt the asymmetric particle update step described in Section D.3 to the genetic algorithm's tournament selection step, by increasing low-disparity solutions' probability of winning tournament selections:

1. In each tournament, we identify *two* solutions: the solution with the highest SD index (the tournament winner in a standard genetic algorithm), and the solution with the lowest disparity value.
2. If these two solutions are not the same, the winner is selected probabilistically, with the low-disparity solution's probability of winning computed as $\frac{SD^{(p)} - SD^{(min)}}{SD^{(TBEST)} - SD^{(min)}}$, where $SD^{(p)}$ is the SD index of

the lowest-disparity solution, $SD^{(TBEST)}$ is the highest SD index in the tournament, and $SD^{(min)}$ is the lowest SD index across all solutions. This is adapted from the PSO update in Equation 12.

PSO and genetic algorithm results for our four-cluster synthetic datasets appear in Table 10, and suggest that PSO yields a higher SD index. Further investigation suggests that this is due to the genetic algorithm's lower stability across seeds, given that we omit non-reproducible solutions from consideration even if they perform well. For the PSO stability analysis in Online Supplement G, running the genetic algorithm under the same parameter set yields an Adjusted Rand Index (ARI) of 0.534, compared with ARI values of more than 0.8 for PSO. This supports our use of PSO for our main results, though it is of course possible that fine-tuning of other metaheuristic algorithms could yield competitive solutions as well.

Table 10 Comparison between PSO and genetic algorithm results for main four-cluster results.

	SD index		**Separation**		**Disparity**
		Centroid Sep (X)	PD Centroid Sep (PD-X)	Outcome Sep (y)	PD Disparity
PSO	1.265 (0.786, 2.036)	0.532 (0.027)	0.097 (0.008)	0.589 (0.047)	0.123 (0.016)
Genetic	0.887 (0.629, 1.25)	0.459 (0.013)	0.094 (0.006)	0.582 (0.066)	0.136 (0.015)
K-means	0.238 (0.151, 0.376)	0.548 (0.008)	0.1 (0.005)	0.482 (0.069)	0.218 (0.01)

Appendix F: Synthetic data generation procedure

We intentionally generate datasets with overlapping clusters and noisy feature-outcome relationships to mimic real-world data generating processes in which data are highly heterogeneous and for which ground-truth clusters cannot be known. For our main four-cluster synthetic dataset, we resample the original dataset (Handl and Knowles 2005) to produce training, validation, and test sets, each with $10,000$ rows. Then, to enforce greater cluster overlap, we add standard Gaussian noise to each feature. We generate a continuous outcome $\mathbf{y}$ using the synthetic function generator from Friedman (2001), in which the outcome variable is given by $\mathbf{y} = \sum_{t=1}^{T} a_t \cdot g_t(\mathbf{z}_t) + \epsilon$. The coefficients $a_t \sim U[-1, 1]$ are uniformly distributed, the functions g_t are multivariate normal with random means and covariance matrices (which allows for feature interactions), and the $\mathbf{z}_t$ are randomly-selected subsets of variables. We use $T = 80$ function terms, sample the $\mathbf{z}_t$ such that terms contain an average of 4.465 features, and scale the error to produce a signal-to-noise ratio of one. The output generating function $\mathbf{y} = \sum_{t=1}^{T} a_t \cdot g_t(\mathbf{z}_t) + \epsilon$ is defined separately for each cluster, producing K outcome vectors $\mathbf{y}^1, ..., \mathbf{y}^K$. Each row's final outcome is a linear combination of its entries in $\mathbf{y}^1, ..., \mathbf{y}^K$, weighted by that point's proximity to each cluster centroid using a Gaussian kernel. We repeat the data generating process above 10 times using a different $K = 4$ dataset from Handl and Knowles 2005, then repeat the entire procedure for ten 10-cluster datasets.

Appendix G: Multi-seed optimization procedure and solution stability analysis

All SD-PSO runs use the following multi-seed optimization procedure to ensure cluster solution stability:

1. From a given initialization, we re-run PSO across M random seeds. We then pool the P particle solutions across all seed runs, for a total of $M \times P$ candidate solutions.
2. For each solution, we compute the adjusted Rand index (ARI) between it and all other solutions from the other seeds, to quantify the degree to which each solution is similar to those found from other PSO runs. The ARI ranges from -1 to 1, with 1 indicating perfect agreement, -1 indicating perfect disagreement, and 0 indicating chance agreement.
3. We rank all $M \times P$ solutions according to their training reward (descending), and select a cutoff quantile $q \in [0, 1]$ for the ARI scores. We then select the top T highest-reward solutions with ARI scores above the q quantile for further validation. These T candidate solutions are then evaluated on a validation set, as described in Online Supplement H.

We now demonstrate that this multi-seed optimization procedure improves cluster solution stability. We use PSO to fit 10-cluster solutions to a 10-cluster synthetic dataset (taken from Handl and Knowles (2005) and generated according to the procedure from Online Supplement F). Because the novelty of our framework lies in the incorporation of cluster disparity, we set $\lambda = 1$ to ensure that disparity is fully prioritized.

We re-run PSO with combinations of particles P, epochs E, and random seeds M, across five random 'outer seeds' (i.e., independent runs with different sets of M seeds), and compare the top five solutions from each PSO run using the adjusted Rand index (these top five solutions are selected according to the procedure in Section 4.1.2, or are simply the five highest-scoring solutions for single-seed PSO).

Table 11 presents the ARI scores and training reward across the various PSO solutions. Results suggest that our multi-seed optimization procedure substantially improves the stability of the highest-scoring cluster solutions. Notably, most of the stability gains are achieved when using just three random seeds (compared with a single seed), with small stability improvements when going from three to five seeds. We proceed with $M = 3$ seeds, $P = 10$ particles, and $E = 25$ epochs for our main analysis.

Table 11 Stability analysis results

Solution	Particles (P), Epochs (E)	ARI	Training reward
Standard PSO	$P = 25, E = 25$	0.400	3.55
3-seed PSO	$P = 10, E = 25$	0.822	3.30
3-seed PSO	$P = 15, E = 12$	0.834	3.245
5-seed PSO	$P = 10, E = 12$	0.866	3.26

Appendix H: Experimental setup and model hyperparameters

For all analyses, the initial predictive model trained before SD-PSO is an Explainable Boosted Machine (EBM, Nori et al. 2019), from the *interpret* Python package. Each EBM allows marginal effects only (no interactions), a max of 50 leaf bins per feature, and a minimum of 50 samples per leaf.

To initialize particles' $K \times J$ cluster centroid matrices, we generate ten initial solutions: nine K-means solutions, plus one random uniform initialization. We then randomly initialize each particle with one of these solutions. We run SD-PSO with the parameter values of $\omega = 0.729$ and $\theta_1 = \theta_2 = 1.49$ Van den Bergh and Engelbrecht (2006). SD-PSO fits cluster-specific residuals using cubic ridge regression, with an α penalty chosen such that the average absolute coefficient value in a null model (i.e., a model fitting EBM residuals to X, which should be uncorrelated) is reduced by half compared with an unpenalized model. We test several α values and choose $\alpha = 10,000$ for our synthetic datasets and $\alpha = 25$ for our Parkinson's dataset.

We run SD-PSO on a training set of $N = 10,000$ observations, then select the top $T = 5$ particle solutions (out of $M \times P = 30$ candidate solutions) with ARI scores above the $q = 0.8$ quantile (described in Section 4.1.2). We then evaluate these $T = 5$ candidate solutions on a held-out validation set of $10,000$ observations. When evaluating a cluster solution, we do not fit separate EBMs to each cluster, but instead use a LitBoost model (from Hjort et al. (2024)) which only permits main effects and feature-by-cluster interactions. The resulting model is then used to calculate the solution's validation SD index value. The highest-scoring solution is selected as the final SD-PSO solution, which is then evaluated on the held-out test set.

Appendix I: PPMI acknowledgment

Data used in the preparation of this article were obtained on September 18, 2021 from the Parkinson's Progression Markers Initiative (PPMI) database (https://www.ppmi-info.org/access-data-specimens/download-data), RRID:SCR_006431. For up-to-date information on the study, visit http://www.ppmi-info.org. PPMI – a public-private partnership – is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners, including 4D Pharma, Abbvie, AcureX, Allergan, Amathus Therapeutics, Aligning Science Across Parkinson's, AskBio, Avid Radiopharmaceuticals, BIAL, BioArctic, Biogen, Biohaven, BioLegend, BlueRock Therapeutics, Bristol-Myers Squibb, Calico Labs, Capsida Biotherapeutics, Celgene, Cerevel Therapeutics, Coave Therapeutics, DaCapo Brainscience, Denali, Edmond J. Safra Foundation, Eli Lilly, Gain Therapeutics, GE HealthCare, Genentech, GSK, Golub Capital, Handl Therapeutics, Insitro, Jazz Pharmaceuticals, Johnson & Johnson Innovative Medicine, Lundbeck, Merck, Meso Scale Discovery, Mission Therapeutics, Neurocrine Biosciences, Neuron23, Neuropore, Pfizer, Piramal, Prevail Therapeutics, Roche, Sanofi, Servier, Sun Pharma Advanced Research Company, Takeda, Teva, UCB, Vanqua Bio, Verily, Voyager Therapeutics, the Weston Family Foundation and Yumanity Therapeutics.

References

Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5):1189–1232, URL http://dx.doi.org/10.1214/aos/1013203451.

Klusowski JM (2021) Universal consistency of decision trees for high dimensional additive models. *arXiv preprint arXiv:2104.13881* .

Maulik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. *Pattern recognition* 33(9):1455–1465.

Scornet E, Biau G, Vert JP (2015) Consistency of random forests. *The Annals of Statistics* 43(4):1716–1741, URL http://dx.doi.org/10.1214/15-AOS1321.

Van den Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajectories. *Information sciences* 176(8):937–971.