Counterfatual Explanation of the TCGA

Côme Delecourt, Blaise Hanczar

Feb-July 2025

SUMMARY

We test here two methods (DICE and brute force) to give local explanations of cancer transcriptomics through counterfactual explanations (CFX). For this we have trained an MLP that classifies patients that have cancer (class 1) or don't (class 0). A CFX consists for a given sick patient to find one or many sane ”twins” which corresponds to answering the following question : what minimal and realistic changes would lead the original candidate to change class. The goal of this study is to find valid counterfactuals (CFs) that are close to the factual, plausible in the data distribution and sparse for actionnability. We then assess our results according to these three criteria : proximity, plausibility, sparsity. The chosen methods and sets of parameters don't allow for sparse and plausible counterfactuals as there is an inherent tradeoff.

Key words: Counterfactual Expanation - TCGA - DICE - Brute Force

1 DATASET AND MLP CLASSIFIER

The TCGA is a dataset of 19887 dimensions composed of 9349 samples, 10% of which are class 0 which makes classes uneven. This a high dimensional dataset that is imbalanced with few samples compared to dimensions. We split it into a 70/10/20 train/test/validation batches to train the MLP which is built with three hidden layers : 19887 - 2000 - 200 - 20 - 1 (Around 40 million parameters) with binary classifiaction and the following characteristics :
BatchNorm, Dropout = 0.02,
SGD with momentum = 0.5,
Learning rate = 0.0003,
L1 penalized loss with lambda = 0.00001,
Batch Size = 64,
Epochs = 300,
Leaky ReLU with negative slope 0.01
Accuracy = 98.56
This lets us with a validation batch of 1756 samples.
To separate class 0 and 1 samples based on distribution, we compute 10-NN values for a point and compare it to the distribution of 10-NN in class 0 and class 1 (see figure 1).

A preliminary analysis shows that there is no easy way to get an intuition of the topolgy of the distribution of the data. For instance, we compute the cosine simillarity of the vectors that links each class 0 point to the closest to class 1 point of any two pair (which appeared to always come from the same patient), and order thel by L1 distance to make any kind of curvature appear (figure 2). The result is just high amplitude noise with a positive offset. We also try to take a look at the distribution of values per feature over the whole TCGA (see figure 3), and also class 0 and 1, and we compute the Wasserstein L1 distance fo a given feature's distribution to the average distribution of class 1, class 0 or the dataset. If we plot features according to decreasing class feature Wasserstein distance, it appears that around the 2400 features that are the most out of distribuution for class 0 are much closer to the average distribution of class 1, which could imply that there are around 2400 genes that differenciate sane from sick people.

2 PROVIDING COUNTERFACTUAL EXPLANATIONS AND METHOD CHOICES

Many methods are available for CFX, some using regular optimization like DICE that we chose, or reguar brute force (BF). These are a lot of other methods resorting to mixed integer programming, decision trees, or deep learning generative methods with VAEs or GANs. Given the dataset, GANs could also have been a reasonable choice wich could be further explored.
DICE is a gradient descent optimization method for which we minimize the following loss :

$$Loss = \frac{1}{n}\sum_{i=1}^{n} hingeloss(y'_i) + \frac{\lambda_1}{n}\sum_{i=1}^{n} ||x'_i - x||_1 + \lambda_2 Div(x'_1, \_, x'_n)$$

The loss has two parameters, and the Div metric is computed as determinantal point process. we prune invalid CFs.
On the other hand, we compute a score for BF :

$$Score = \lambda_1(y' - y) - \lambda_2 \frac{||x' - x||_1}{std(L_1)} - \lambda_3 Plaus(x')$$

This score is composed of a classification improvement term, a proximity term, and a plausibility term which can be computed as the probablity to be in the 10-NN distribution (approximated as gaussians), or given through the output score of an OSVM. We keep the cf with the best score for each sample.

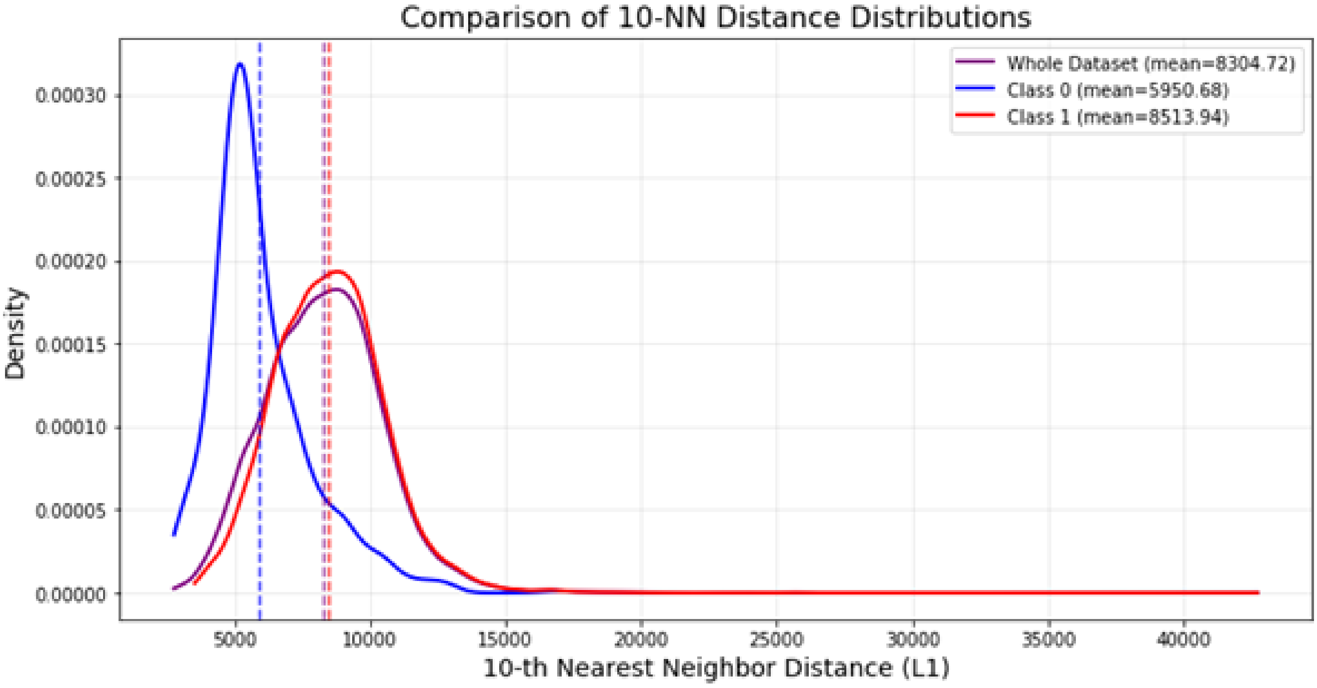

Figure 1. Class 0 and class 1 10-NN distributions

3 EVALUATION METHODS

The counterfactuals are evaluated with validity, proximity, sparsity, sparsity count, diversity, sparse diversity, 10-NN distance, 10-NN dataset probability, 10-NN class 0 probabilty.Specifically, here are the formulas to compute validity, proximity, sparsity, diversity and sparse diversity. We give here the formulas followed by an explanation :

$$Validity = \frac{1}{N}\sum_{i=1}^{N}\left| I\left[y' > 0.5\right] - I\left[y > 0.5\right]\right|$$

Which counts the proportion af samples that have changed class.

$$Proximity_i = \frac{1}{d}\sum_{j=1}^{d}\frac{\left|x'_{ij} - x_{ij}\right|}{MAD_j}$$

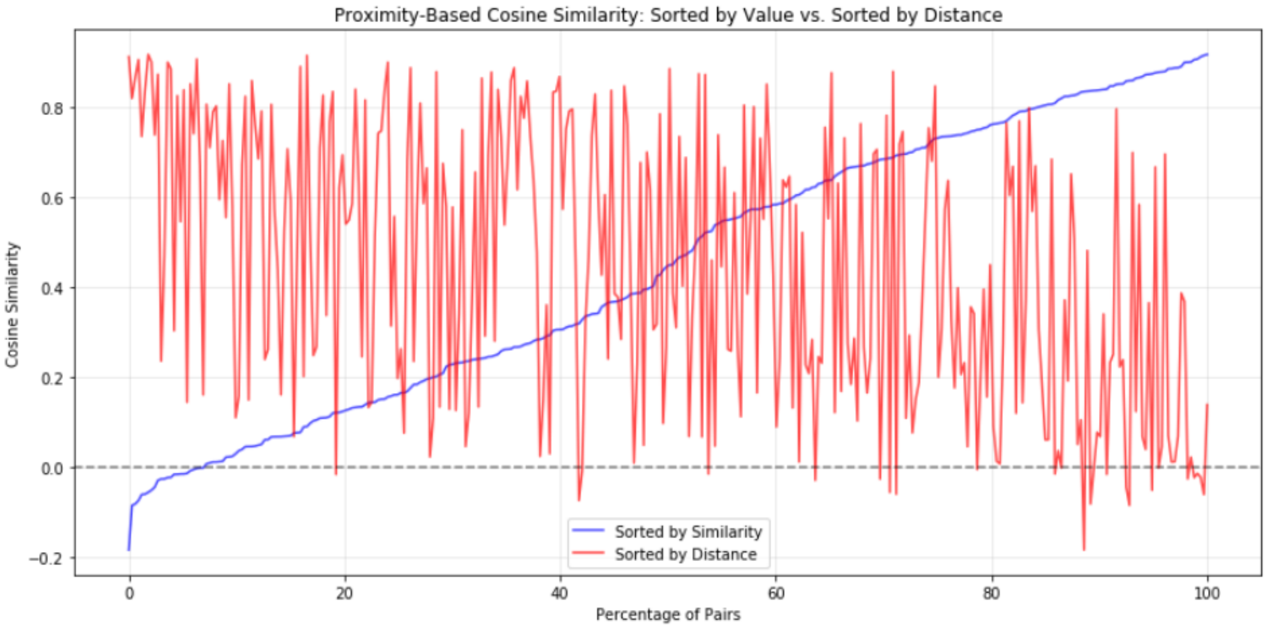

Figure 2. Cosine proximity of pairs of vectors liking class 0 points to the closest class 1 point, ordered by distance

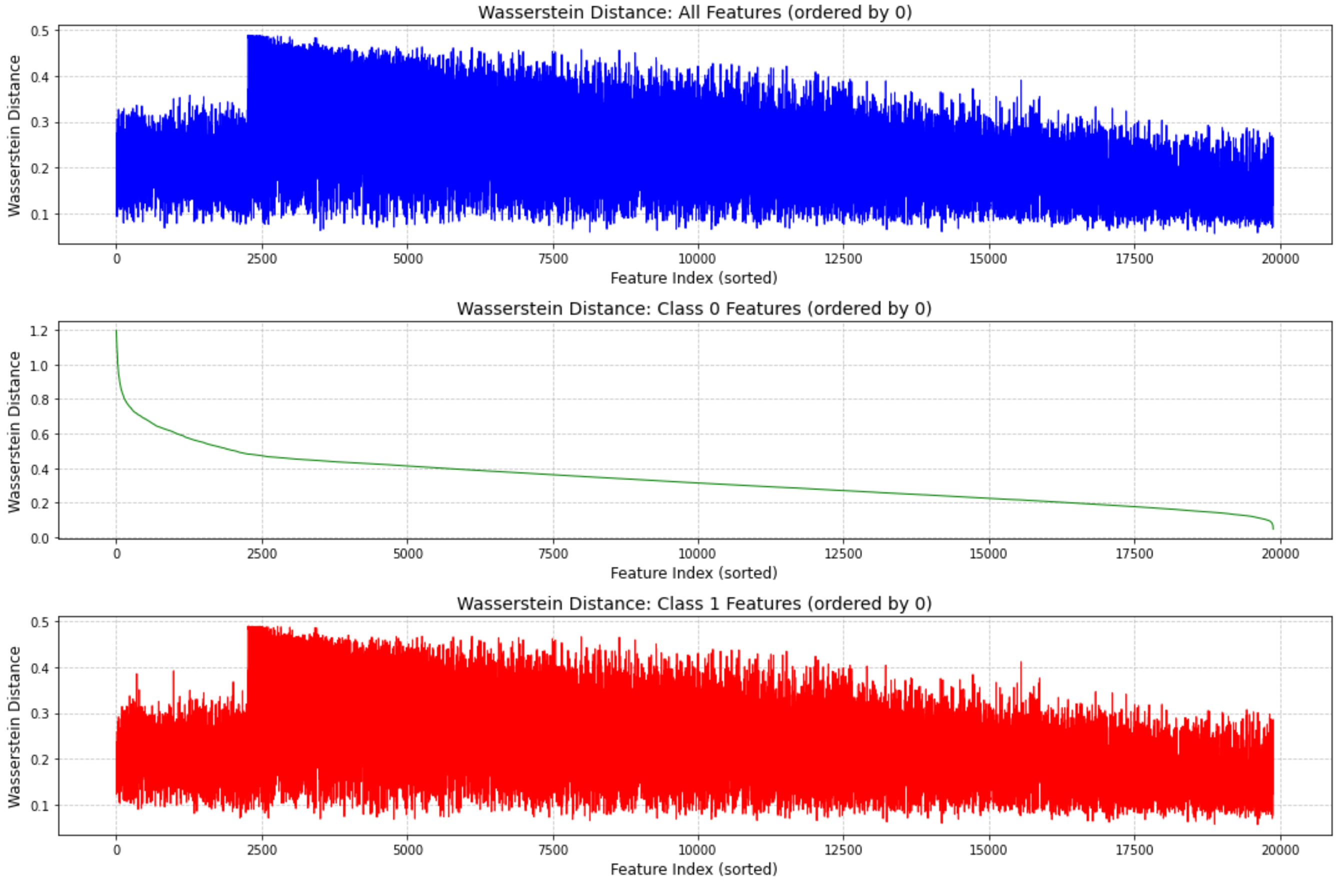

Figure 3. Wasserstein distances of feature distribution to the average feature distribution, for class 0, class 1, and both

For a given counterfactual we compute the distance to the original weighted by feature MAD.

$$Sparsity_i = 1 - \frac{1}{d}\sum_{j=1}^{d} I\left[\left|x'_{ij} - x_{ij}\right| > \epsilon\right]$$

Sparsity is the proportion of features that have not changed further than a certain threshold epsilon.

$$Diversity_i = \frac{2}{k(k-1)} \sum_{1 \le m < n \le k} \left\|x'_m - x'_n\right\|_1$$

When multiple counterfactuals are generated for a one factual, we sum the pairs of L1 ditances between different counterfactuals.

$$IoU_{mn} = \frac{|S_m \cap S_n|}{|S_m \cup S_n|}$$

We define the intersection over the union of the feature values that have changer for two counterfactuals, which is used to define next the sparse diversity of a group of counterfactuals :

$$SparseDiversity_i = 1 - \frac{2}{k(k-1)} \sum_{1 \le m < n \le k} IoU_{mn}$$

The sparse diversity is then computed as the sum over the pairs of IoU$_{mn} forthecounterfactualsassociatedtoafactualpoint.P\left(|X-\mu| > |x-\mu|\right)$ The plausibility is defined as the probability to be outside of the distribution, which is the area between the counterfactual's value and its symetric to the mean distribution of 10-NN of class 0 points assuming it follows a bell curve (figure 1 shows that it is a realistuc assumption).

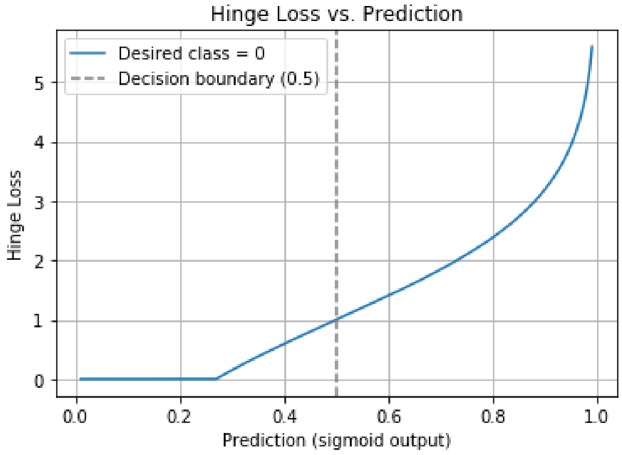

Figure 4. Default hinge loss function

4 DICE

The Dice method is a gradient descent optimization method that generates multiple counterfactuals. We test it generating 3, 6, 10 and 16 CFs. The loss encourages plausibilty through classifiaction

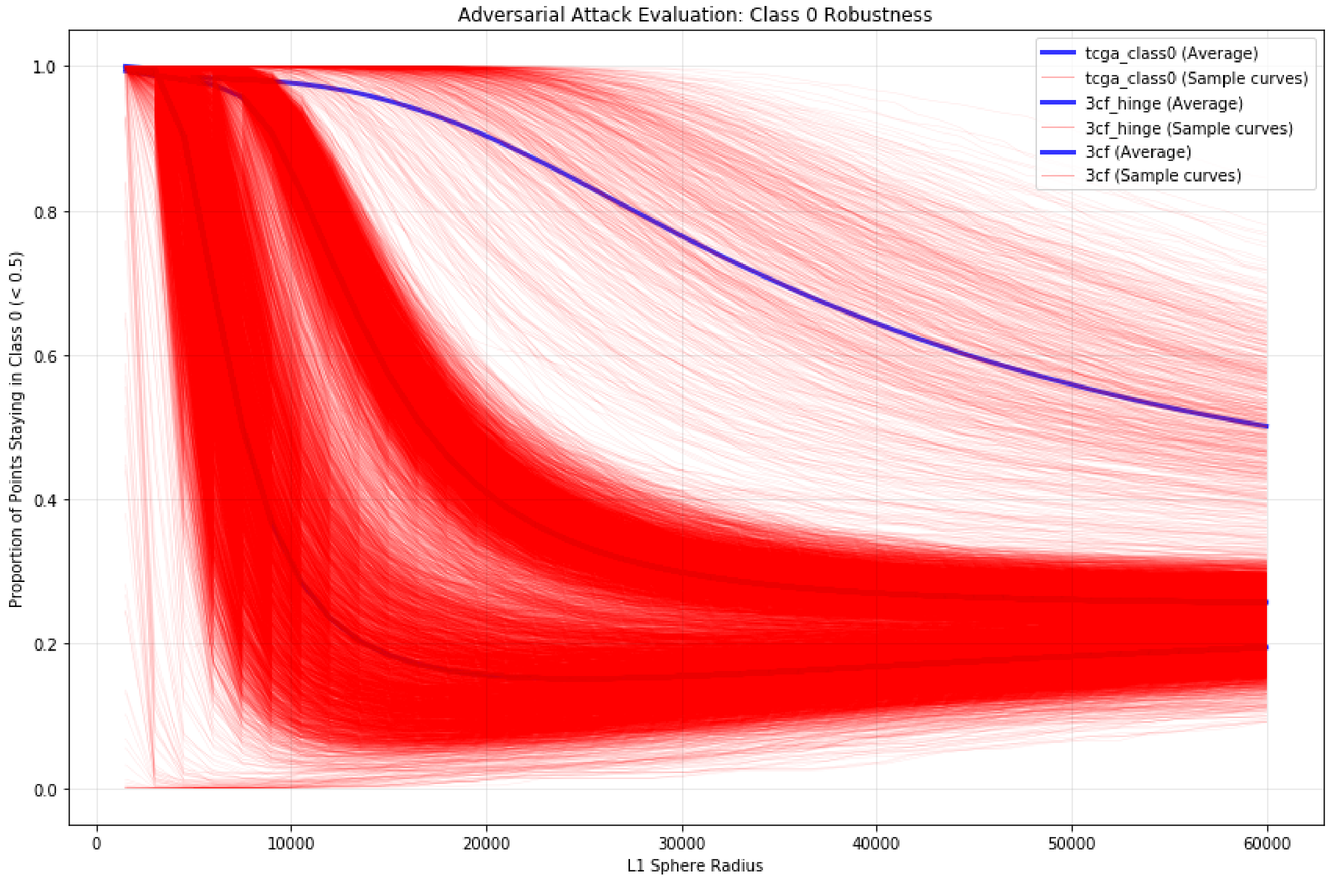

Figure 5. Aversarial attack curve clusters for 3 CFs, from top to bottom curve : for class 0, the modified hinge CFs, the normal hinge CFs

value, diversity, and proximity to the factual. The plausibility is influenced by the parametrization of the hinge loss (figure 4) which by default becomes 0 when under 0.27. This added to the fact that the dataset is very unbalanced which tends to pull the decision boundary of the MLP toward the class 1 samples, causes the produced counterfactuals that are not nearly classified as 0.0 to be unrealistic. This is shown by two metrics : the average 10-NN is higher than class 0 points of the dataset (which is significant since average class 1 points have higher 10-NN distance and also the DICE counterfactuals are even out of distribution of class 1 points). The second metric is adversarial attacks (see figure 5), for which the proportion of valid attacks increases around 5 times sooner than for class 0 points of the TCGA. The tradeoff is of course overall good proximity. The also is a theoretical proximity / diversity tradeoff, which appears to be quite weak in the range of tested values considering the results. We indeed test the DICE method for varrying values of proximity and diversity independently, and also vary the number of produced counterfactual per entry. The algoriithm inherently produces valid counterfactuals, so we don't give validity plots, however we provide a plot of the number of valid counterfactuals against proximity, deiversity, and both at the same time. These differents characteristics outiled appear in the analysis conducted in figures 6 to 11 for proximity and sparsity. And as validity seems to have more immportance, we test the impact of higher values in fugures 12 to 17. Finally in figures 18 to 22, we test the same metrics after applying the post hoc sparsity algorithm. The algrithm has been written to test ranking features for reversion ranking by percentile (original DICE method), or MLP gradient (heuristic to agregate the distance to the boundary and importance at the same time), by descending or ascending order. Descending order should be inefficient but it assesses the efficiency of the ranking method and as it appears it is not as usefull on this dataset. In figure 23, we show for 3 CFs the validity given the number of features reverted, and it has a signmoïd shape centered at 719 features. In figure 24, we try to see a correlation between the number of changed features versus the L1 distance and plausibility, but no cleat trend appears.

We furthermore apply a post-hoc sparsity algorithm on the counterfactuals to enhance sparsity. The feature are reverted in order of increasing MAD importance (more exactly, the value of the p-th percentile of the MAD, see the DICE paper for further explanation). This furthermore worsens plausibilty as it brings the values closer to the boundary, and improves proximity. It also happens to increase diversity. The general outcome is that the default DICE method with sparsity enhancement averages 600 changed features which is very far from the desired 1,2,3 changes. Tweaking the hinge loss (we make it non zero from 0.05 which favour counterfactual classifaction values closer to 0) makes the values far more plausible, and thy prove muche more robust to adversarial attacks (around 4 times). When sparsity enhanced, these counterfactuals average 400 changed features which also is an improvement. The DICE method shows no surprising behaviour but still gives low values on pausibility.

Sidenote : We tested different methods for the dice post-hoc

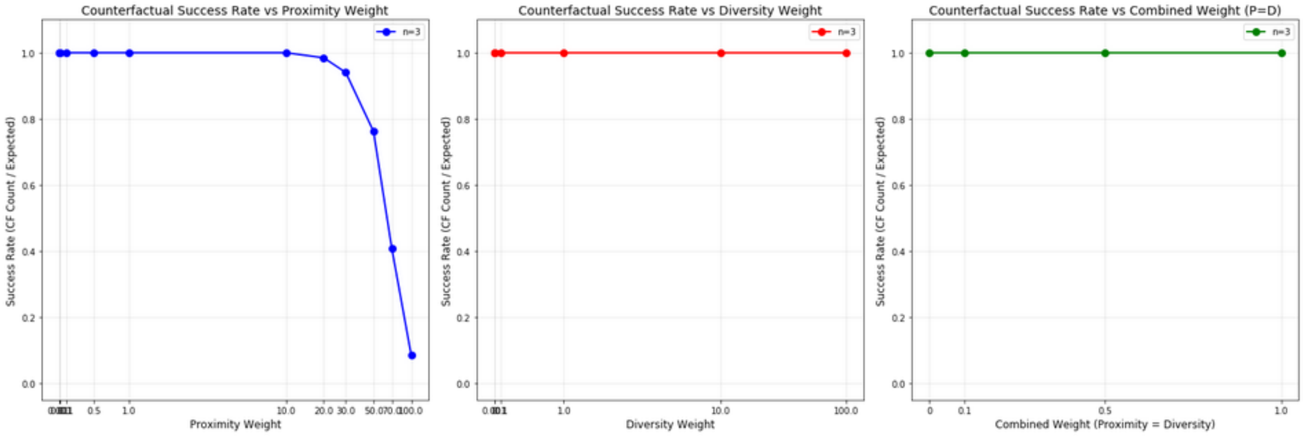

Figure 6. Validity against proximity, diversity and both

sparsity ranking. They show to have low impact on the sparsity. Reverting features by decreasing importance shift the average final changed feature count by approximately 10, and reverting them randomly (we try to revert each feature once) had better results by a small margin, though more testing should be done to verify these statements.

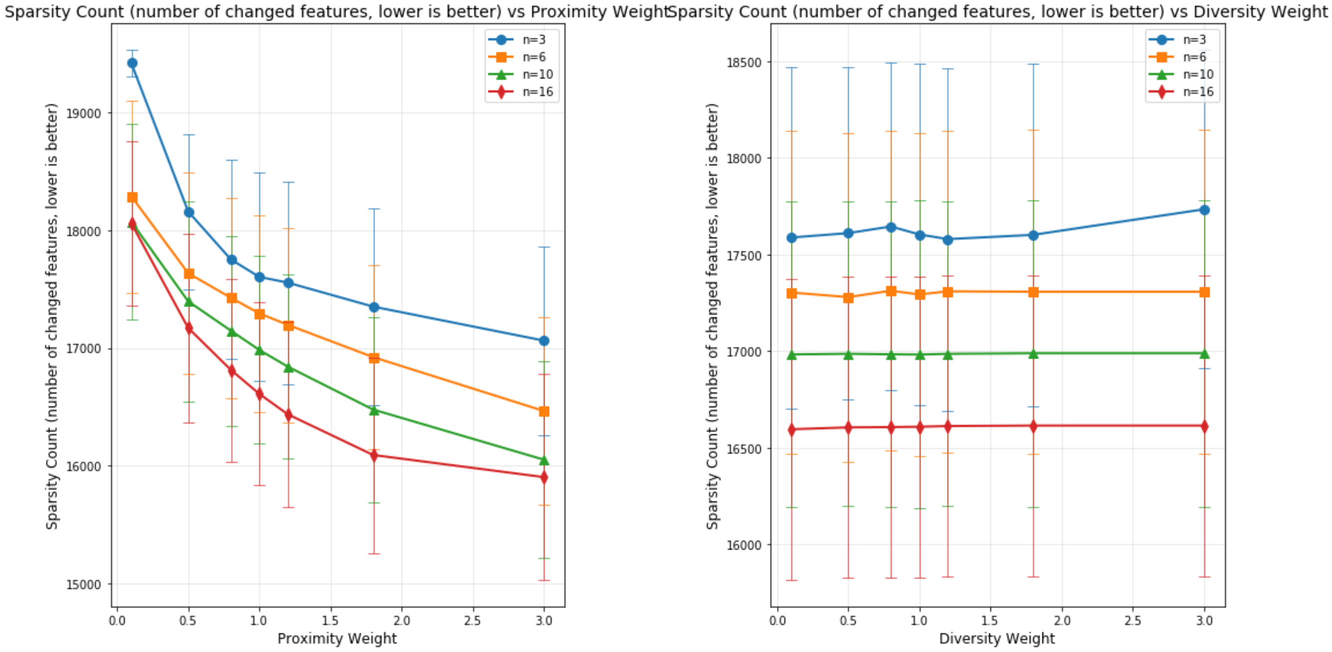

Figure 7. Sparsity against proximity and diversity

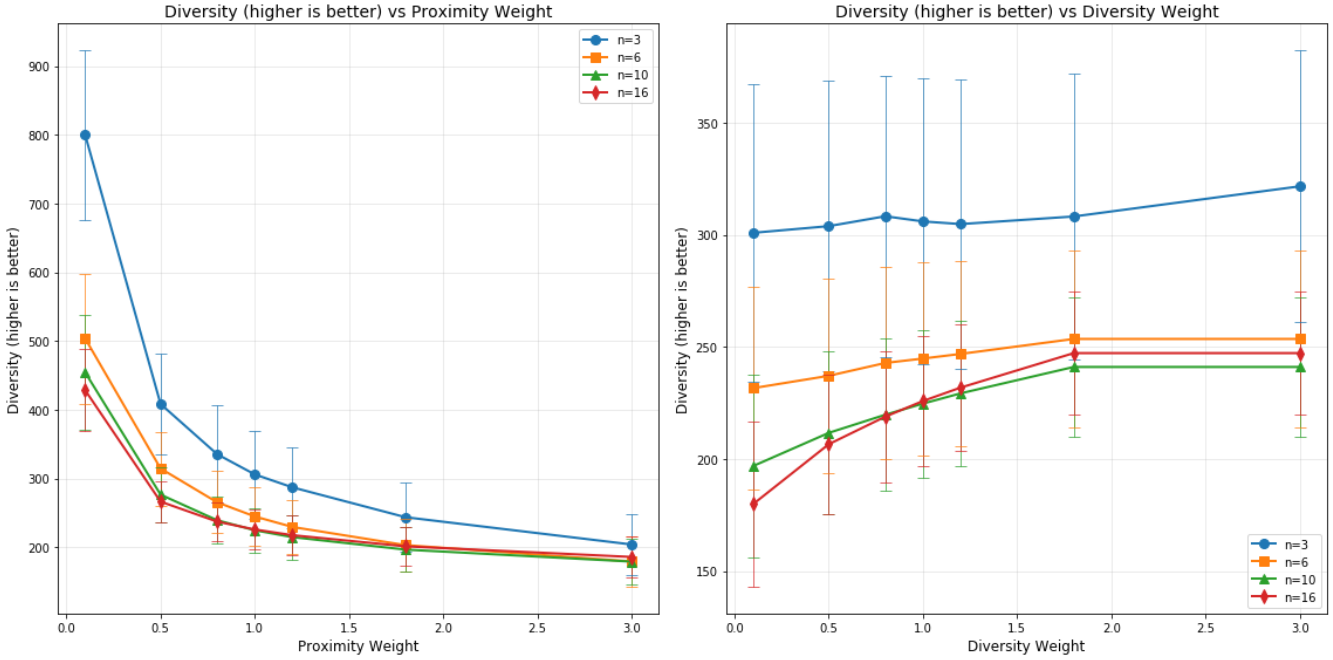

Figure 8. Diversity against proximity and diversity

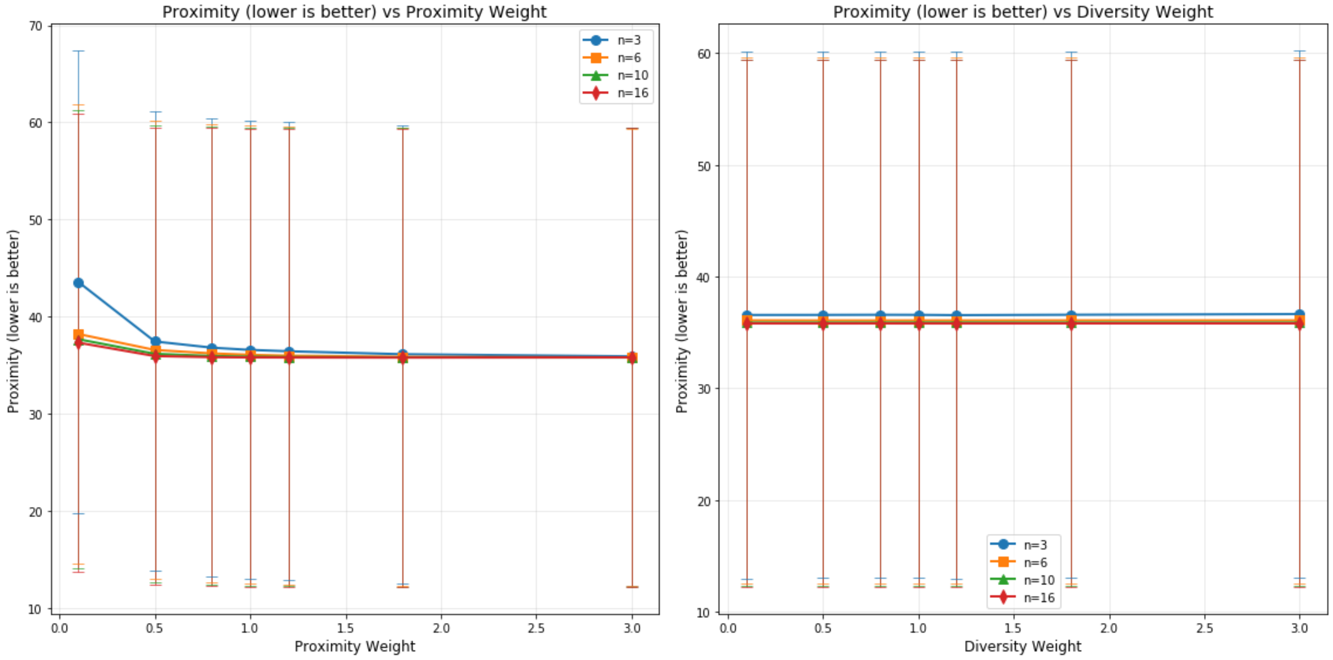

Figure 9. Proximity against proximity and diversity

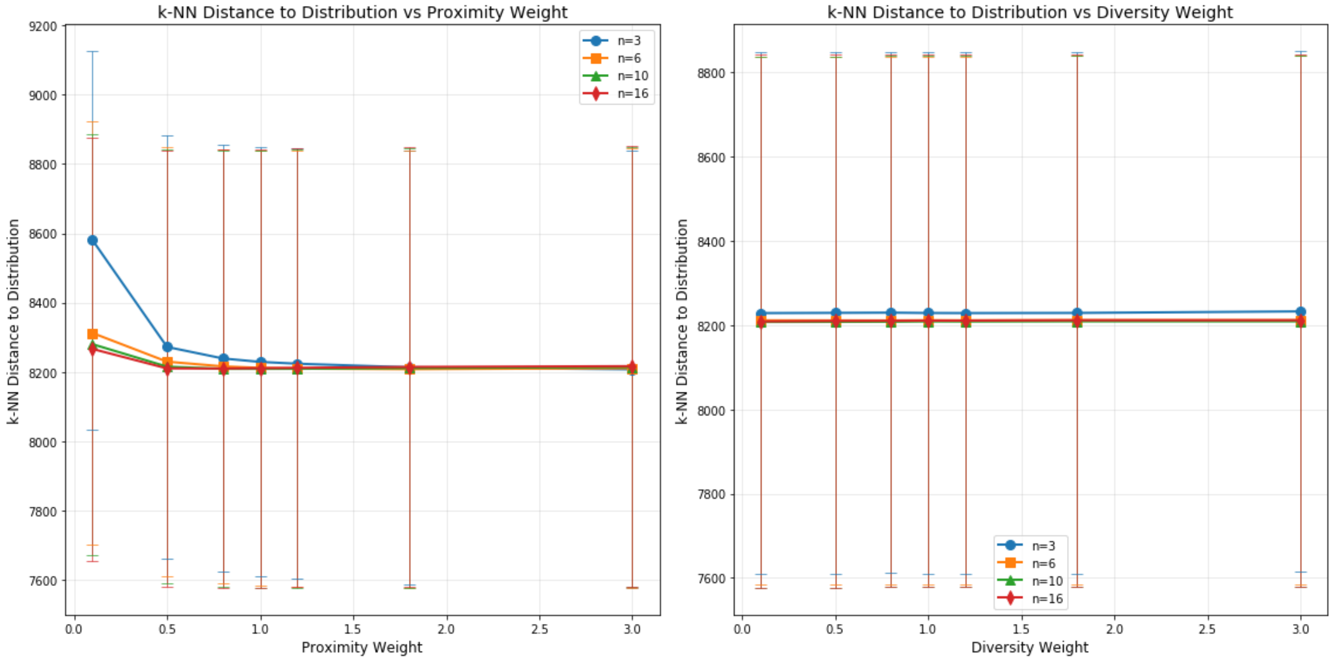

Figure 10. Average 10-NN distance to the data distribution against proximity and diversity

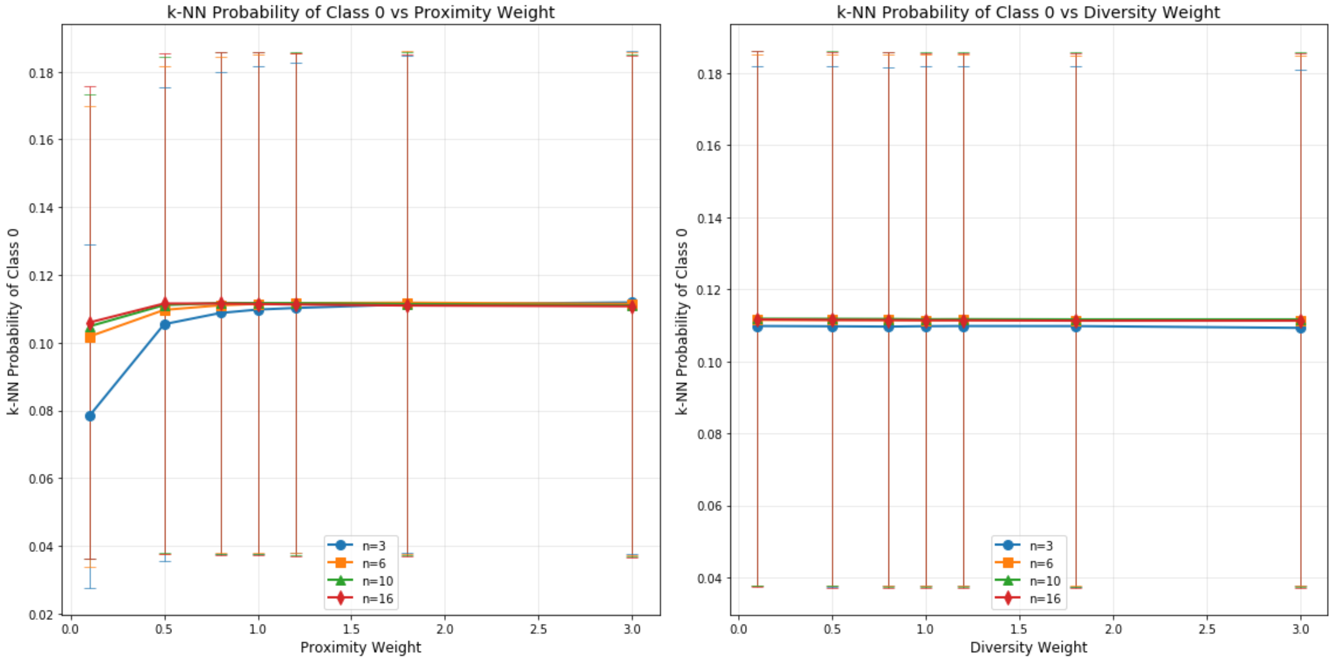

Figure 11. Average 10-NN plausibilty in class 0 against proximity and diversity

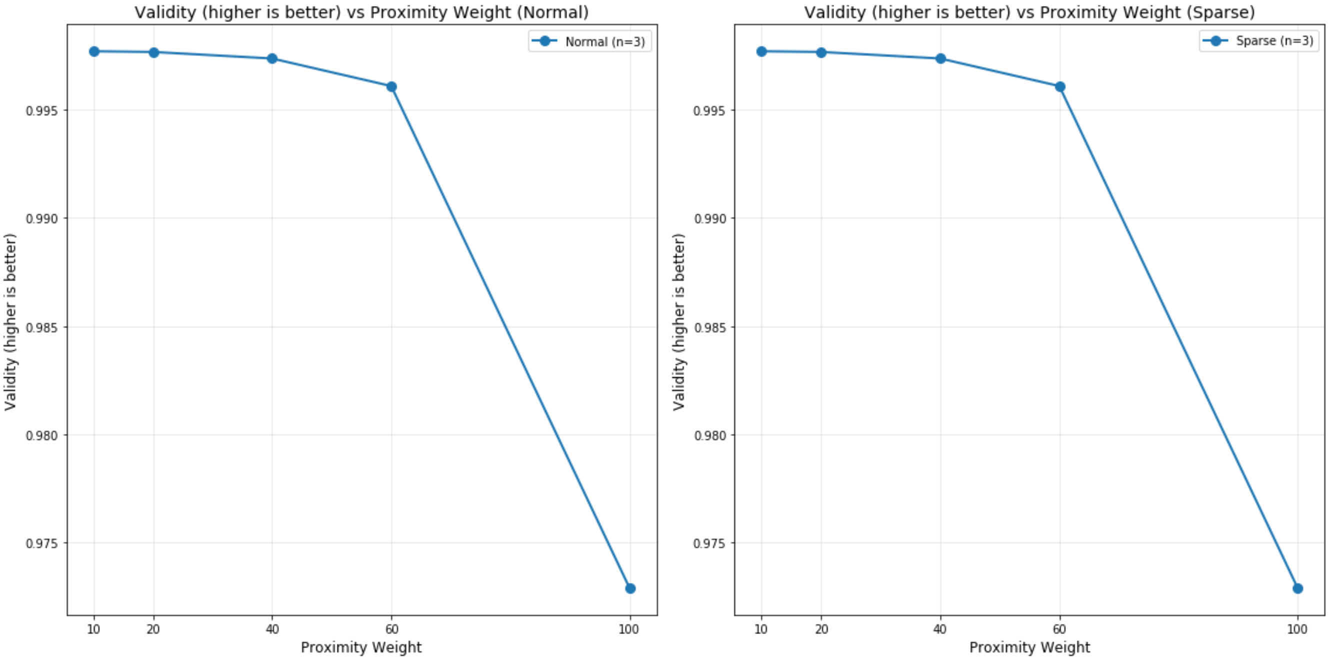

Figure 12. Validity against proximity for normal and sparse CFs

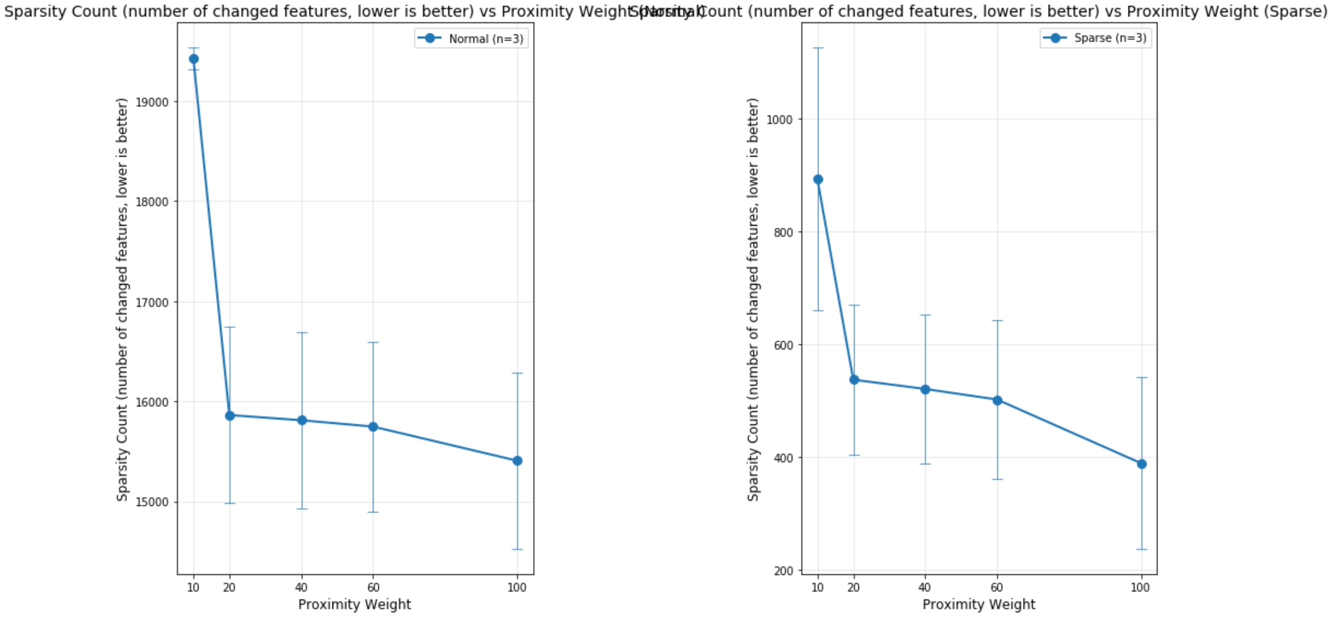

Figure 13. Sparsity against proximity for normal and sparse CFs

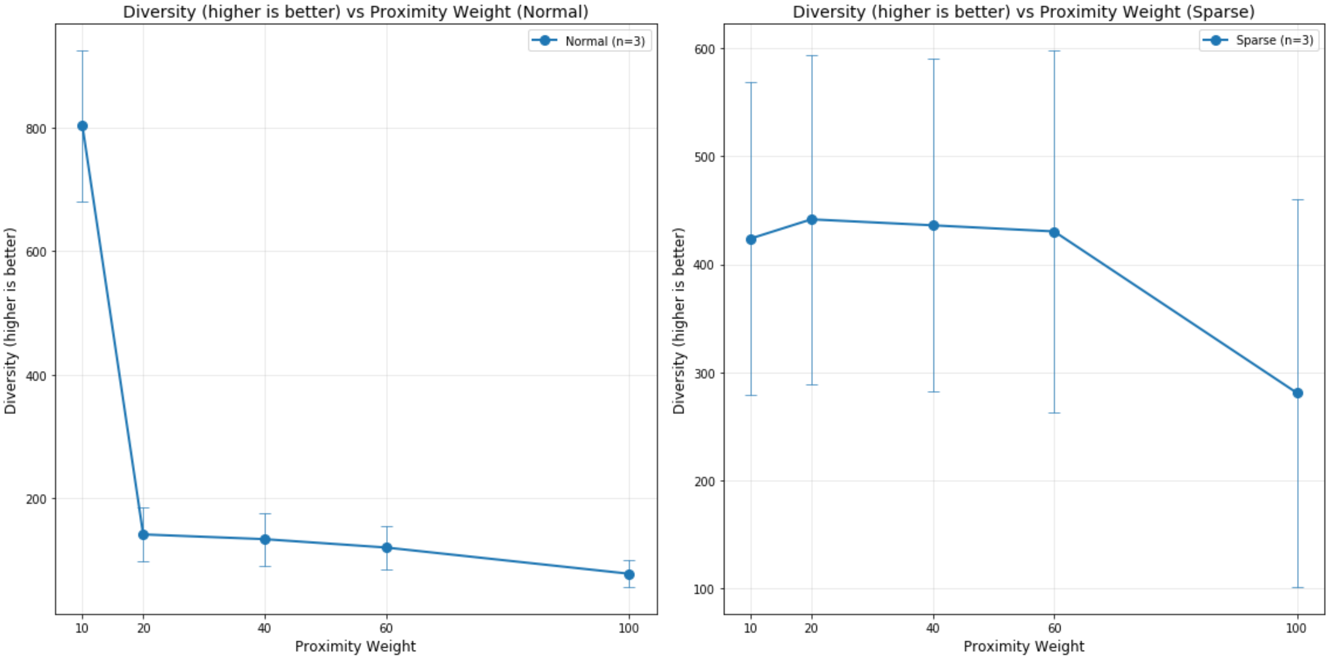

Figure 14. Diversity against proximity for normal and sparse CFs

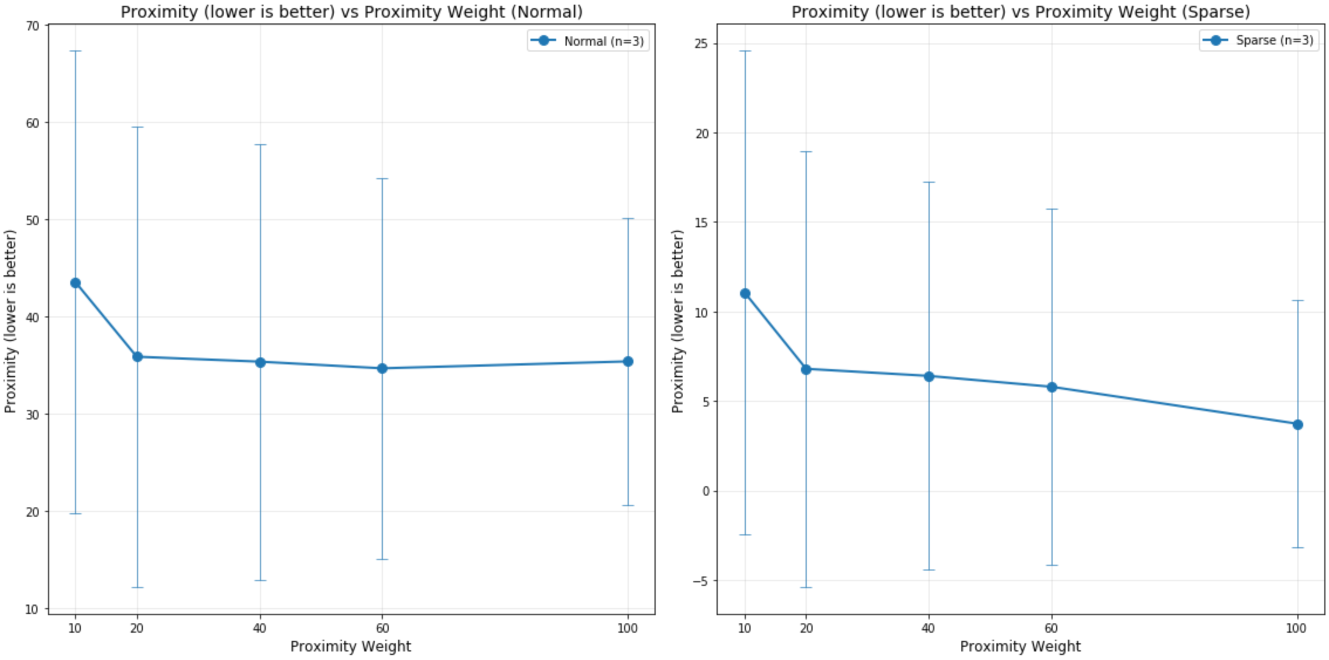

Figure 15. Proximity against proximity for normal and sparse CFs

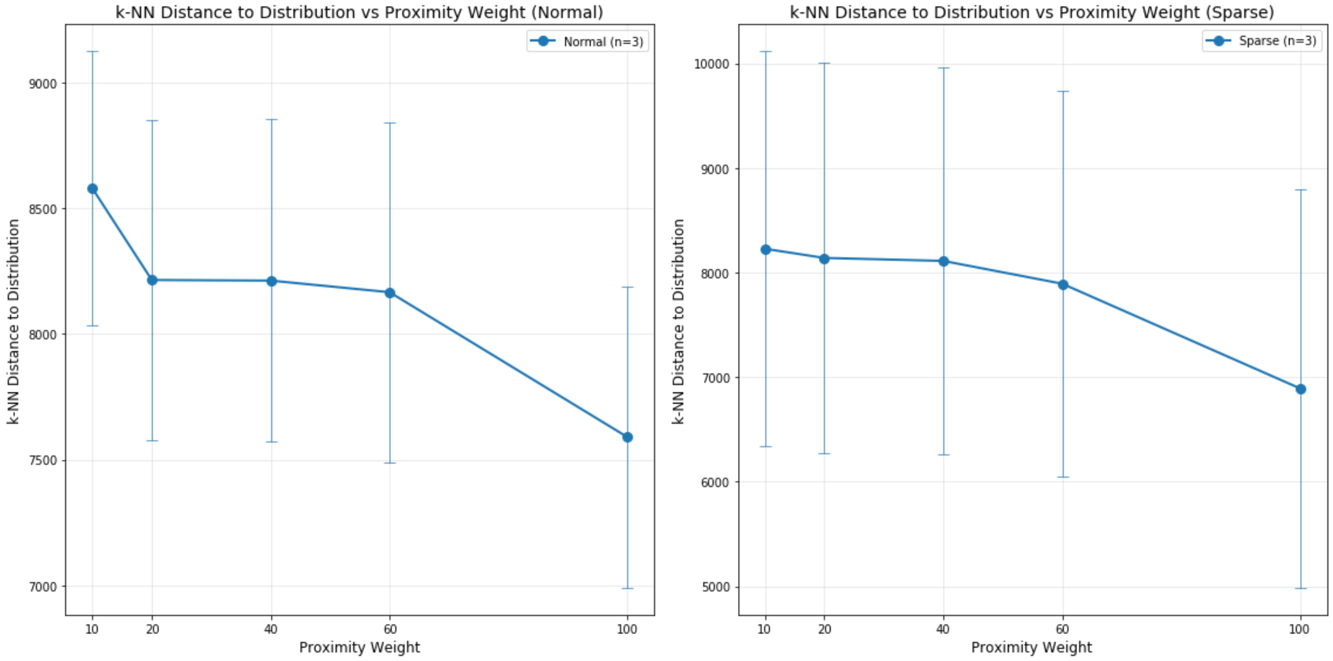

Figure 16. 10-NN average distance to ditribution against proximity for normal and sparse CFs

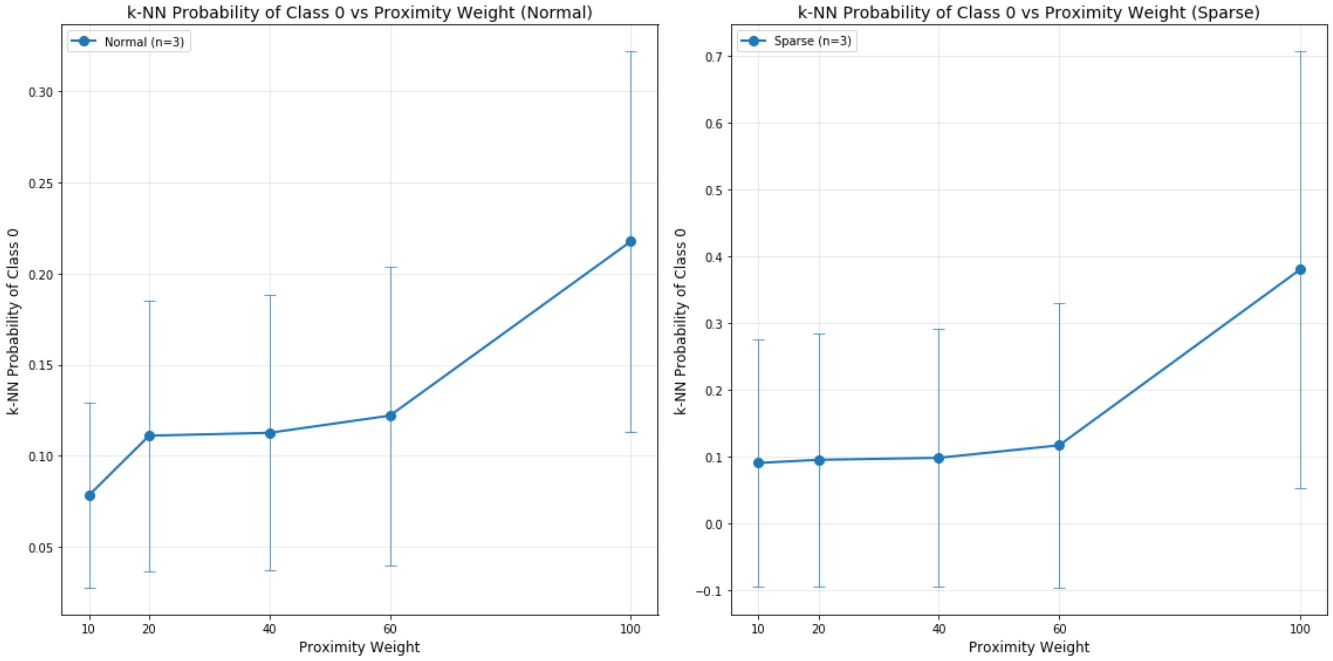

Figure 17. Average 10-NN plausibilty in class 0 against proximity for normal and sparse CFs

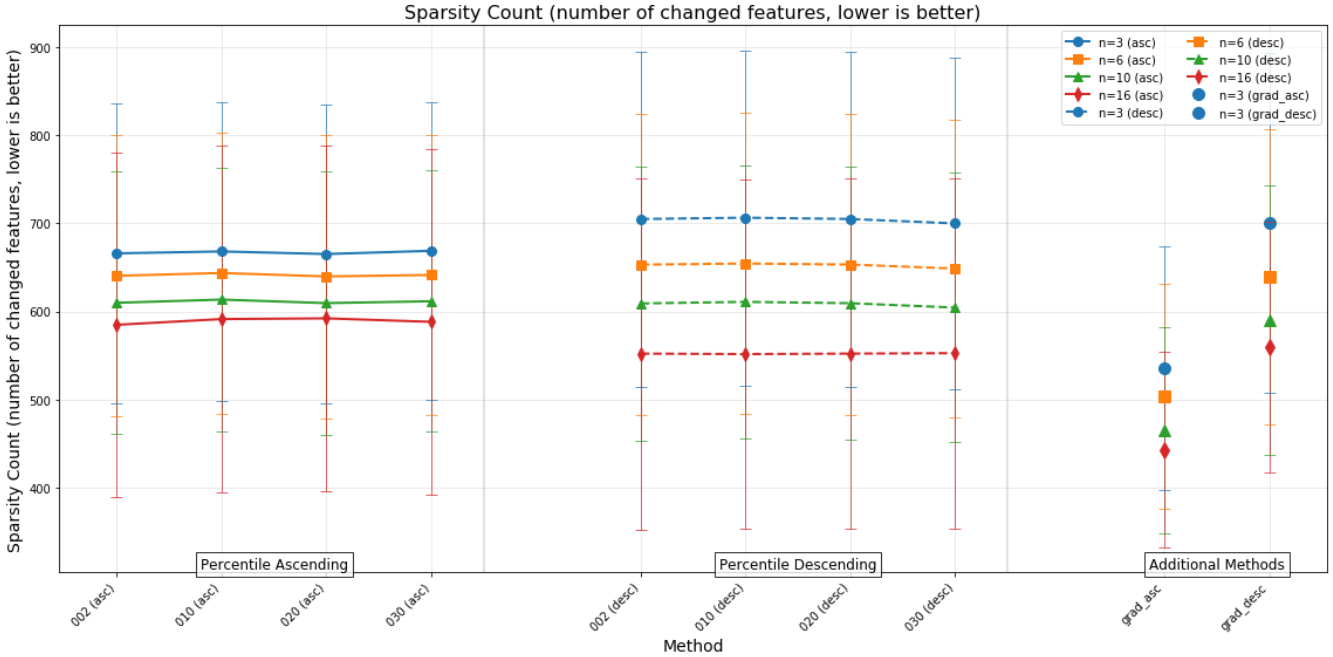

Figure 18. Sparsity against proximity for sparse CFs for percentile and gradient feature ordering

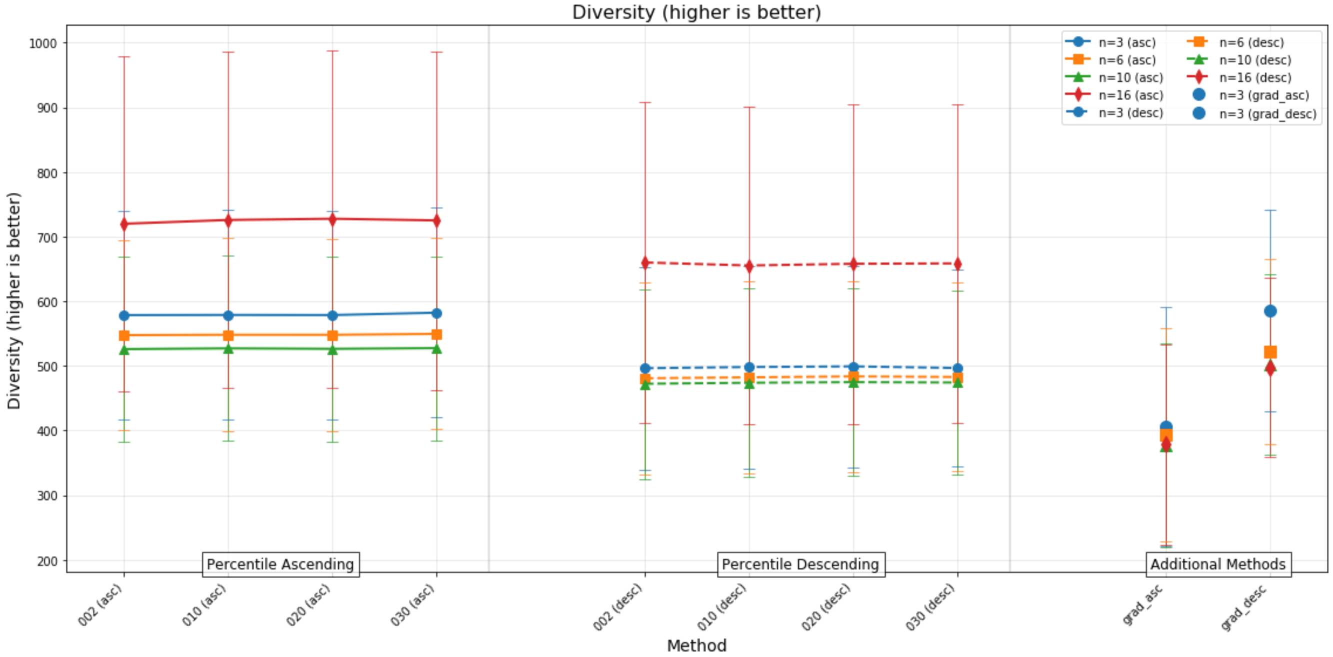

Figure 19. Diversity against proximity for sparse CFs for percentile and gradient feature ordering

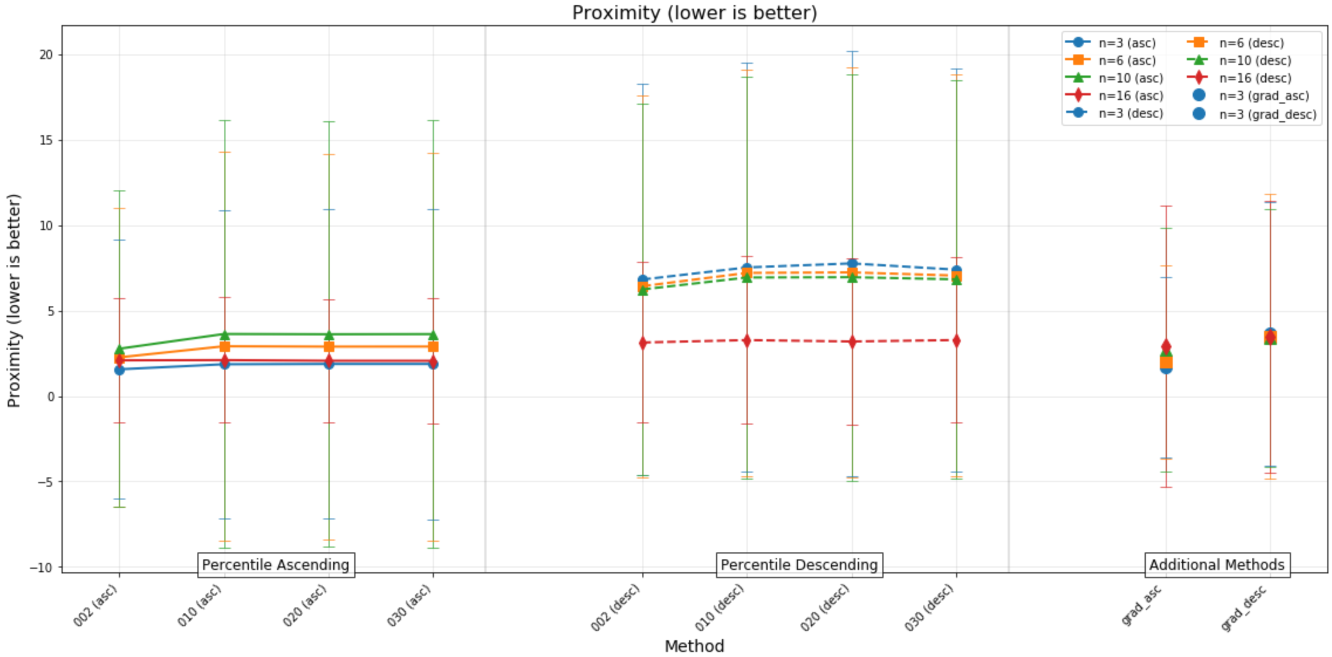

Figure 20. Proximity against proximity for sparse CFs for percentile and gradient feature ordering

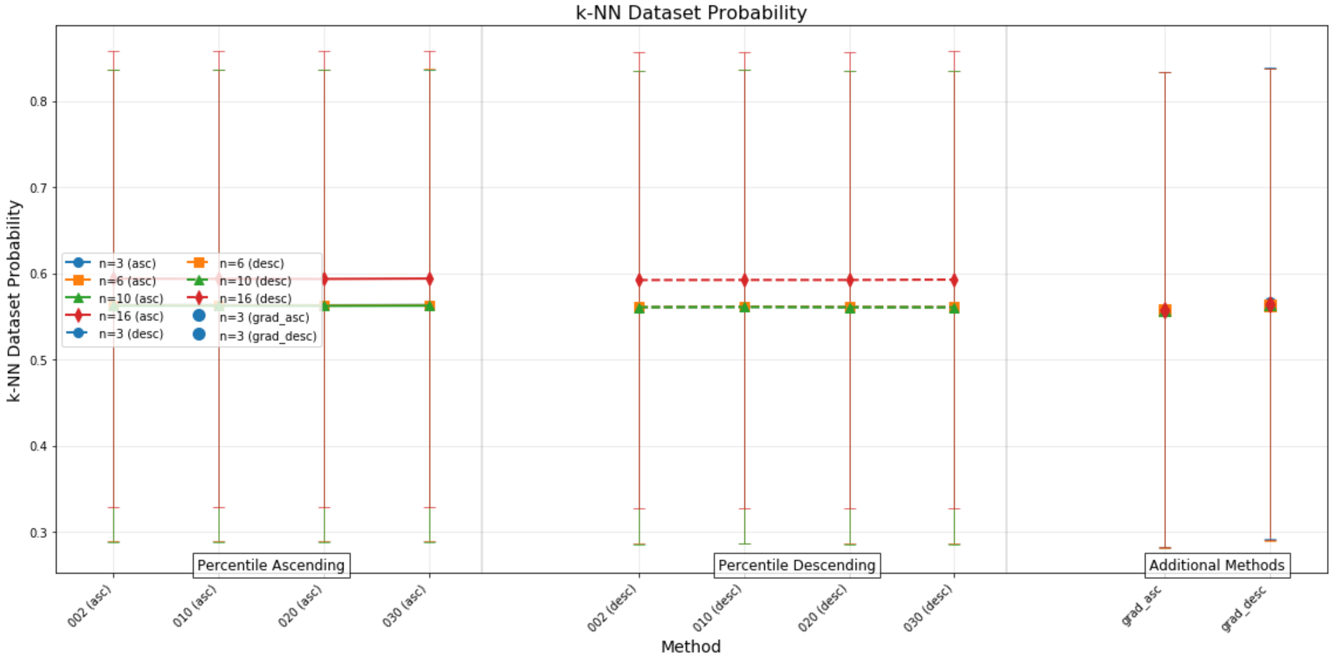

Figure 21. 10-NN average distance to ditribution against proximity for sparse CFs for percentile and gradient feature ordering

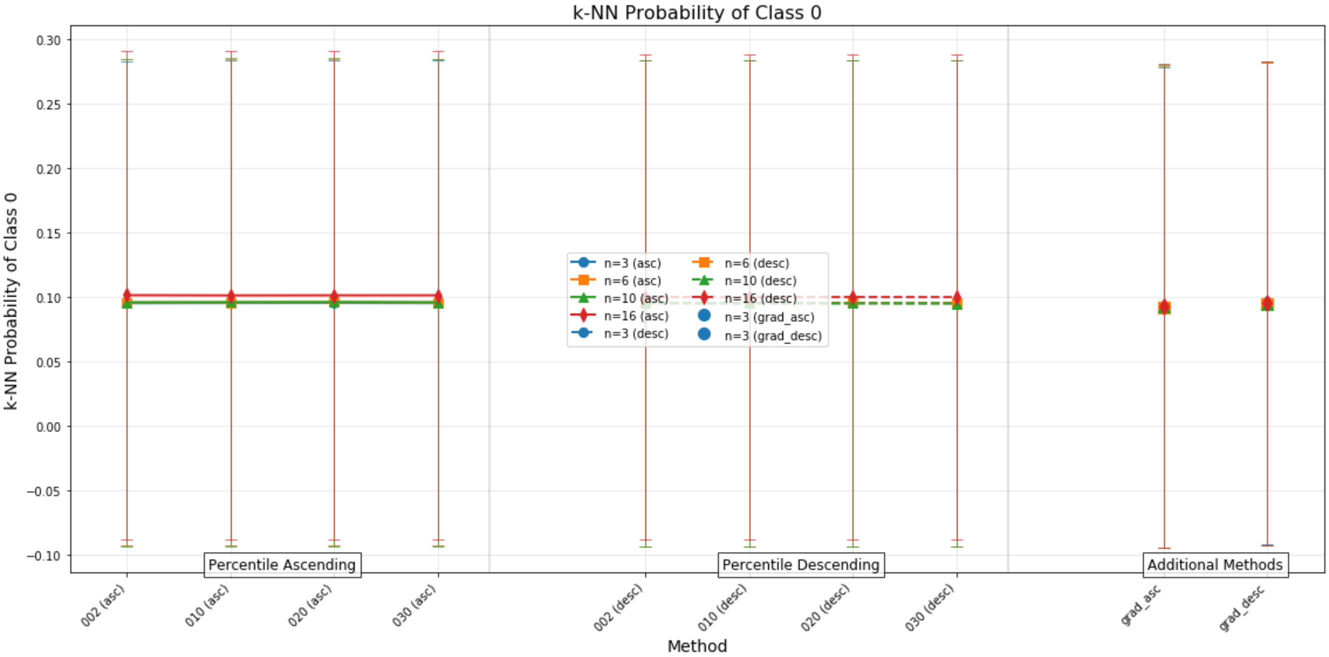

Figure 22. Average 10-NN plausibilty in class 0 against proximity for sparse CFs for percentile and gradient feature ordering

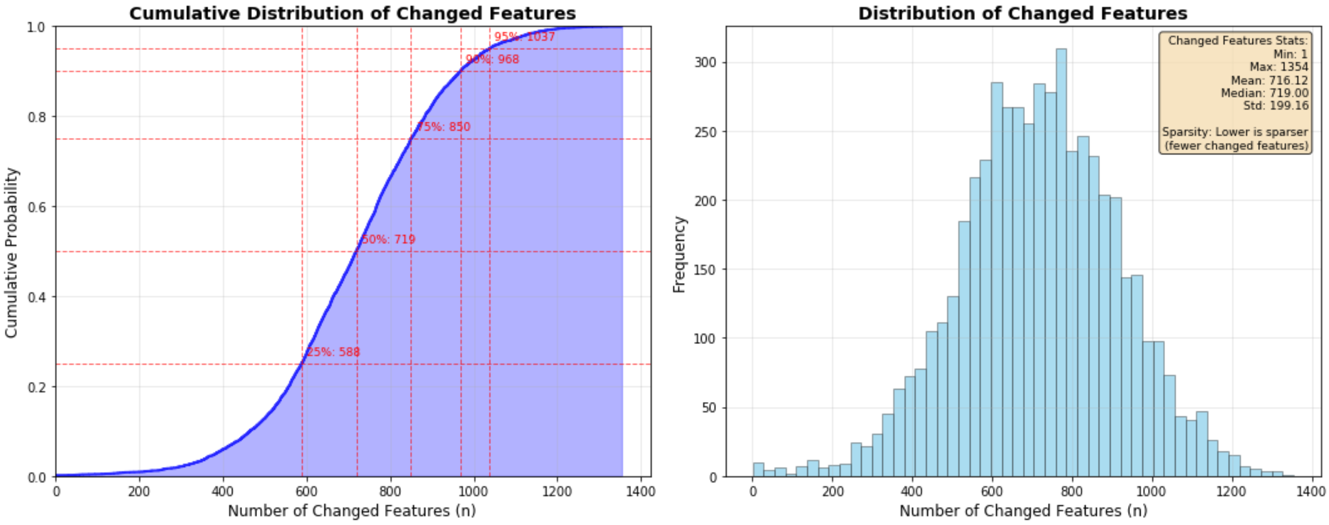

Figure 23. Sparsity against number of changed features, and distribution of changed features for 3 sparse CFs

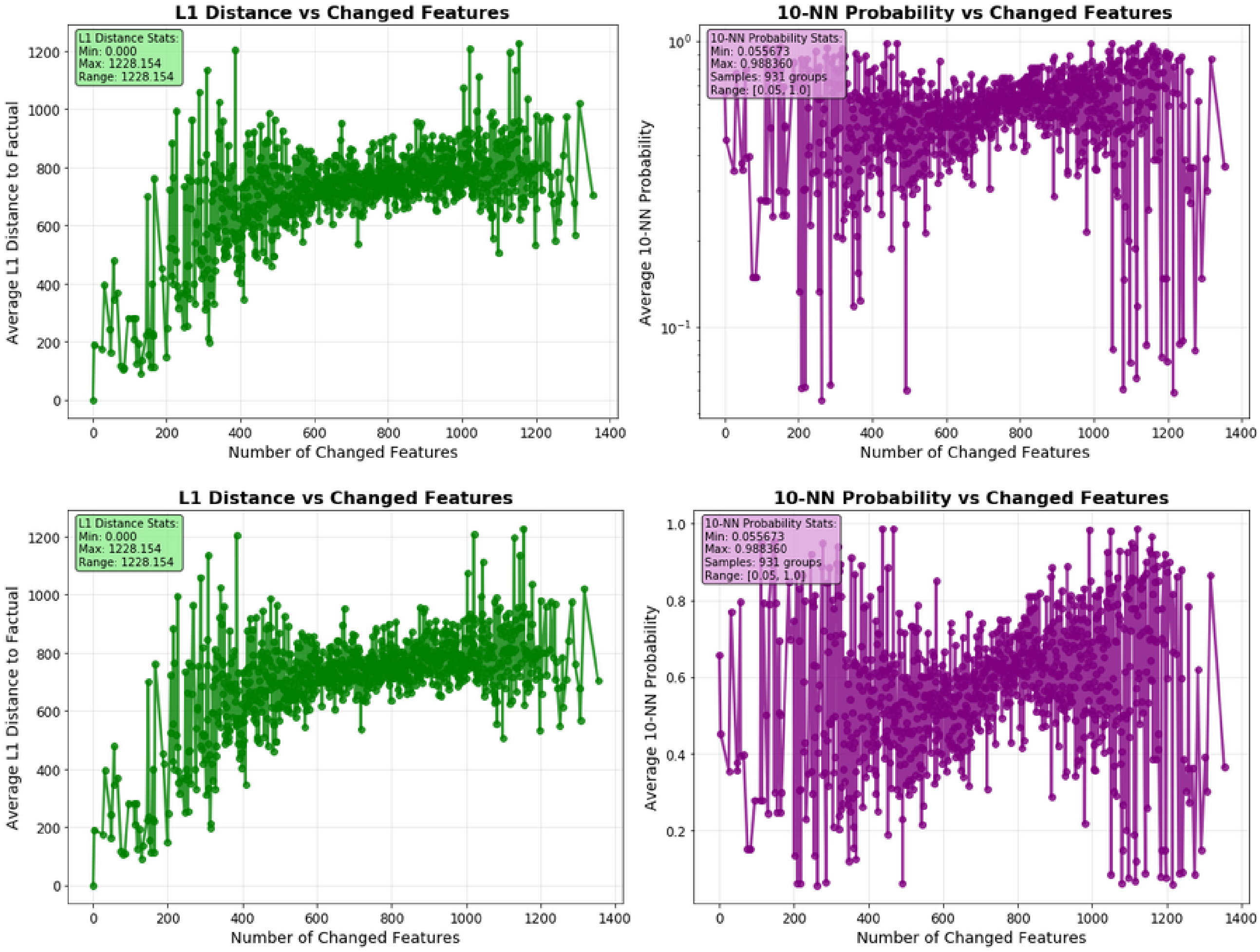

Figure 24. L1 distance to factuals and 10-NN plausibility against number of changed features, with different scales for plausibilty for 3 sparse CFs

5 BRUTE FORCE

Taking an opposed philosophy from the forst approach, we deide to fix sparsity as a contraint through a brute force search. The method is the following : we search in a grid of values in the L1 hypercuboïd that is bound by the min and max values of the class 0 points. That means that for a given vector, supposing we want to vary p features, we chose p features at a time and look at all combinations of values that we sampled between the min and max value recorded for each given feature for the class 0. Each combination is then scored given the rate of classification improvement, proximity and plausibility. Since the dataset has high dimensionality, we are very restricted in this approach and could anly test p=1 and p=2 due to exponential complexity. As expected, the number of valid couterfactuals is very low for this approach. For p=1, we derive a table of the local importance for the modification of one feature (which means testing for this feature's value in the range of class 0 and tracking for the best classification improvement rate). The plausibily for the 4 valid counterfactuals is also very low. For p=2, we already have to make a drastic reduction going to 1756 to 100 samples, and going for a BF heuristic as we chose to cut down the number of combinations. We operate this time on a rectangle made by features, but we keep the top 2 percents of features that explain 20% of the global importance (see figure 25), that we comine with any of the features.

For the 18 cancers present over the 1756 tested class 1 samples, when ranking the top 36 most important features for a given cancer, it appears that there are 154 features that appear at least twice in the set of 18 cancers, at most 9 times and on average 3.4 times. That shows that a given locally "important" feature does not explain cancer globally (figure 26).

On the other hand it appears that among the 154 features that appear more than twice, they all appear between 19 and 35 times in the top 36 features of each cancer, and on average 29.3 times. This shows that thought cancers are pairwise not affected by the same genes, the top 154 genes explain most of the importance of the different cancers (figure 27).

6 RESULTS ANALYSIS

The two methods applied to the TCGA and over the parameters tested show that it is very hard to oversome the sparsity versus plausibility tradeoff, while keeping valid counterfactuals. For instance a loss or score integrating an L1 will increase proximity to factual but decrease plausibilty, while very moderately enhancing sparsity (less than 5 percents for extreme values of the parameter on DICE). Computational requirements fro brute force make the method unrealistic to test for number of features sufficient to be plausible. Any method that induces the use of a latent space or uses dimensionallity reduction may introduce difficulties conscerning sparsisty though it can capture accurate manifold caracteristics, howerver these are still to be explored.

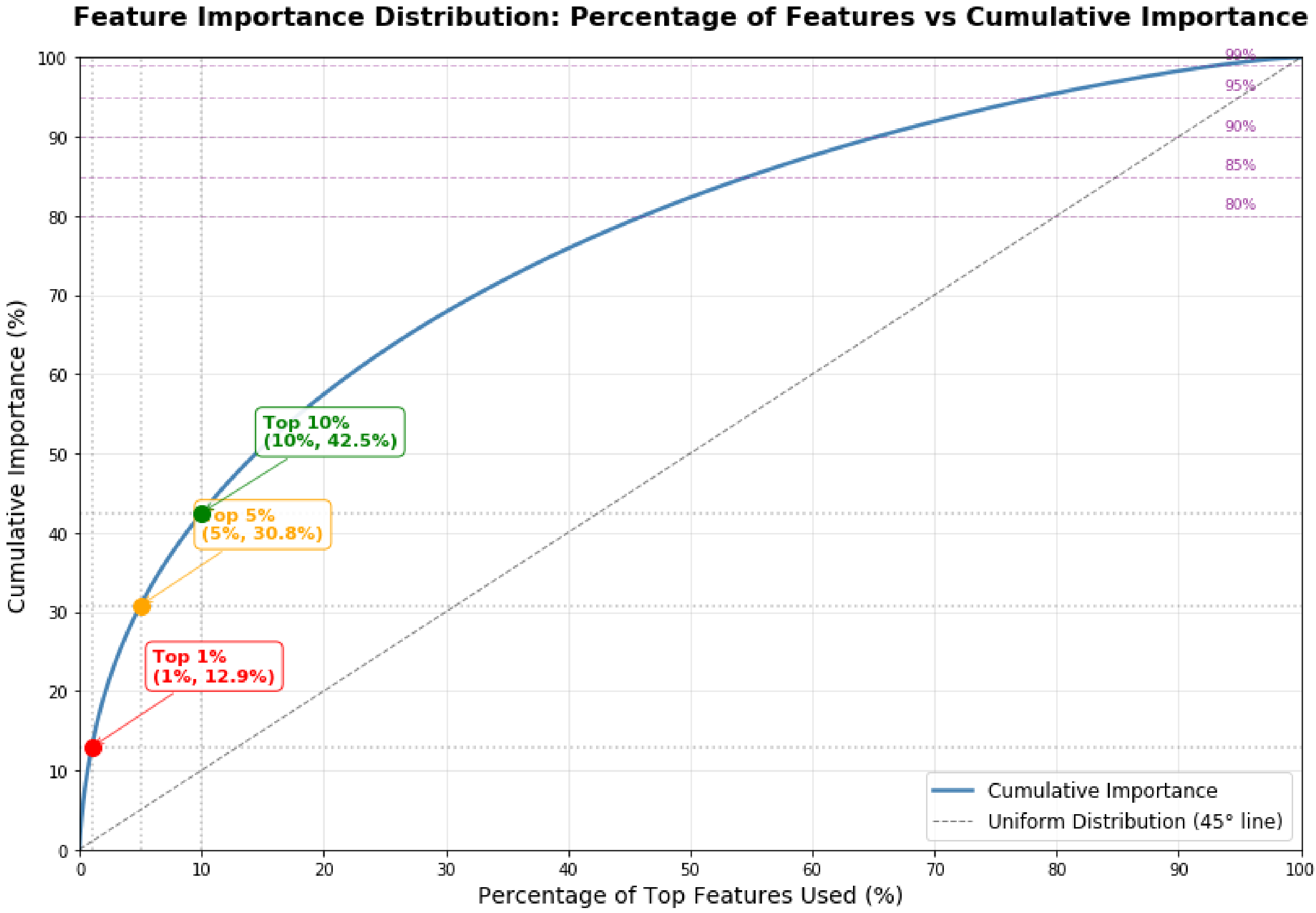

Figure 25. Cumulative importance added by decreasing importance of features.

[t]

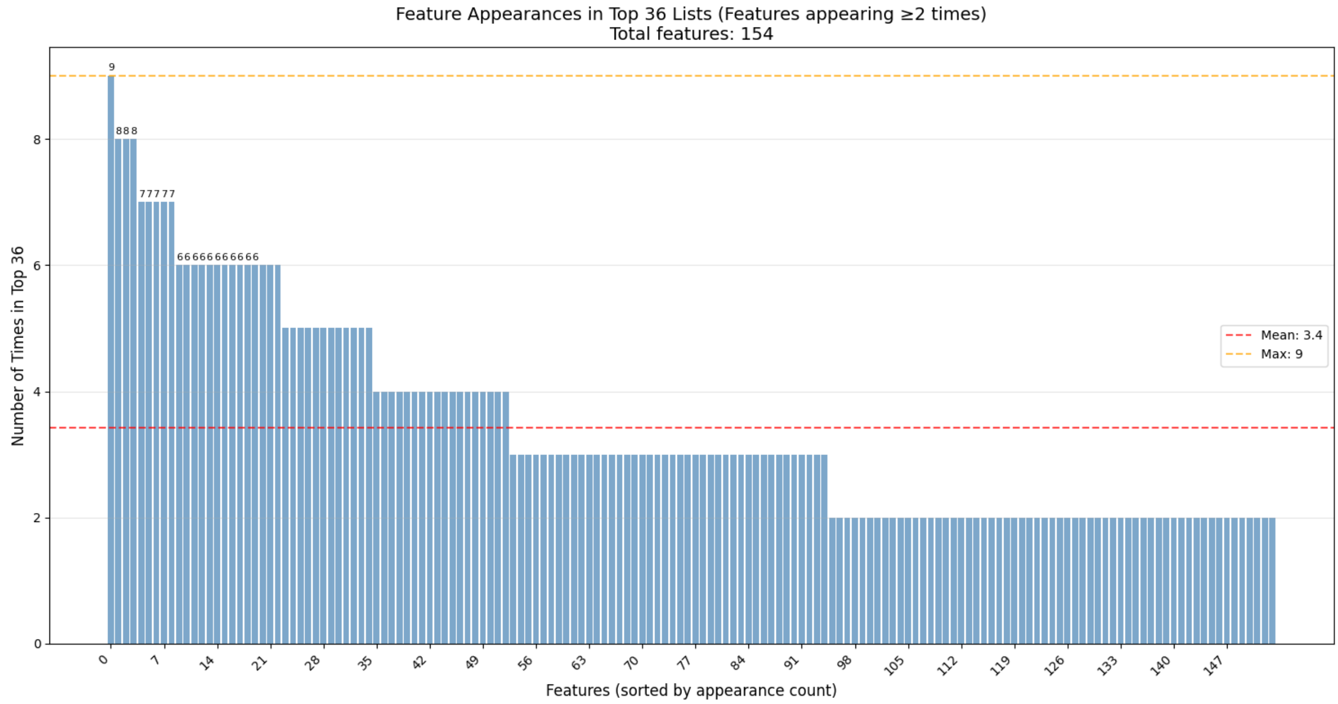

Figure 26. Top 36 features appearance counts in the 18 cancers

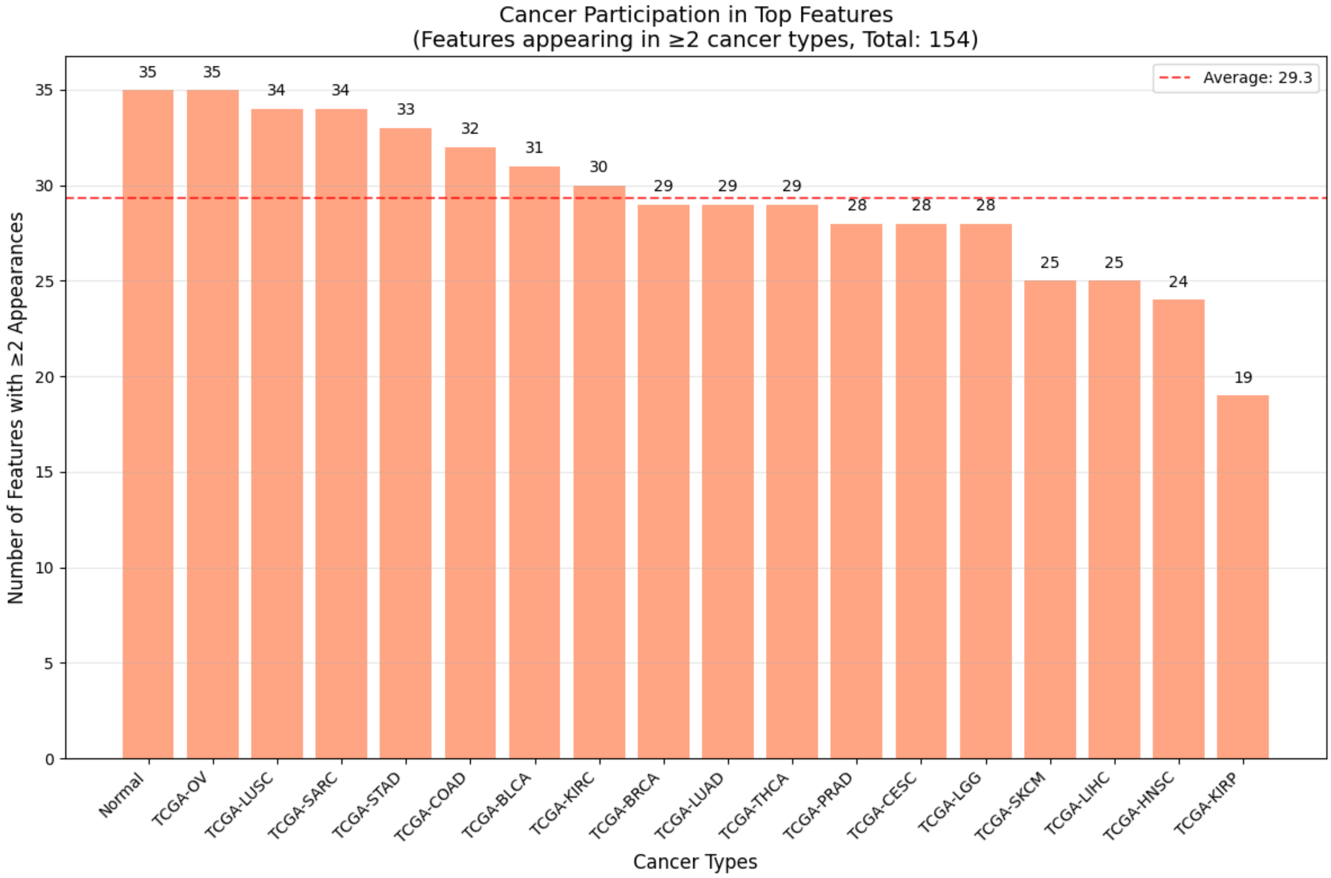

Figure 27. Top 154 features participation count in the cancers

REFERENCES

Pawelczyk, M., Haug, J., Broelemann, K. & Kasneci, G., 2019. Learning Model-Agnostic Counterfactual Explanations for Tabular Data, *arXiv preprint*. https://arxiv.org/pdf/1910.09398

Guyomard, V., Fessant, F., Guyet, T., Bouadi, T. & Termier, A., 2022. VC-Net: A Self-Explaining Model for Realistic Counterfactual Generation, *arXiv preprint*. https://arxiv.org/pdf/2212.10847

Panagiotou, E., Heurich, M., Landgraf, T. & Ntoutsi, E., 2024. TABCF: Counterfactual Explanations for Tabular Data Using a Transformer-Based VAE, *arXiv preprint*. https://arxiv.org/pdf/2410.10463

Madaan, N. & Bedathur, S., 2023. Navigating the Structured What-If Spaces: Counterfactual Generation via Structured Diffusion, *arXiv preprint*. https://arxiv.org/pdf/2312.13616

Van Looveren, A., Klaise, J., Vacanti, G. & Cobb, O., 2021. Conditional Generative Models for Counterfactual Explanations, *arXiv preprint*. https://arxiv.org/pdf/2101.10123

Hellemans, S., Algaba, A., Verboven, S. & Ginis, V., 2025. Flexible Counterfactual Explanations with Generative Models, *arXiv preprint*. https://arxiv.org/pdf/2502.17613

Rustad, A., 2022. tabGAN: A Framework for Utilizing Tabular GAN for Data Synthesizing and Generation of Counterfactual Explanations, *MSc thesis*, Norwegian University of Science and Technology. https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3035146

Barredo-Arrieta, A. & Del Ser, J., 2020. Plausible Counterfactuals: Auditing Deep Learning Classifiers with Realistic Adversarial Examples, *arXiv preprint*. https://arxiv.org/pdf/2003.11323

Mothilal, R. K., Sharma, A. & Tan, C., 2020. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations, *Proceedings of the 2020 FAT* Conference*, pp. 607–617. https://arxiv.org/pdf/1905.07697

Wachter, S., Mittelstadt, B. & Russell, C., 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR, *Harvard Journal of Law & Technology*, **31**(2), 841–887. https://arxiv.org/pdf/1711.00399

Sanderson, J., Mao, H. & Woo, W. L., 2025. DiPACE: Diverse, Plausible and Actionable Counterfactual Explanations, *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART)*, pp. 342–349. https://www.scitepress.org/Papers/2025/132191/132191.pdf