Ensemble Metrics And Models For Density-Based Clustering

Thomas Charlon, Harvard Medical School
Tianxi Cai, Harvard Medical School

Abstract

Unsupervised clustering methods usually rely on distance-based metrics such as the between-within ratio, the Dunn index, the average silhouette width, etc. The **fpc** package provides several distance-based metrics and depending on the use case some may be preferred to others, *e.g.* the Dunn index will favor a minimal distance between all clusters, while the between-within ratio will favor an average ratio between the size of the clusters and the distance between all clusters. In some applications however, it is unclear to the analyst which metric should be preferred, especially in research and knowledge discovery applications in which the clusters are yet to be characterized, understood, and investigated. Thus many real-world clustering studies will include several distance-based metrics, although one challenge is that they are hardly comparable.

Ensemble models is a broad term that defines models that merge results from different clustering models. The merging method can vary a lot, and one of the most common is to have models each produce a vote for the best result, then sum up the votes and select the most voted result. Thus the results are compared by summing their ranks in each distance-based metric, rather than the actual value of the metric. There are several ways to implement this voting mechanism and several considerations come into play here: should each clustering model vote just for the best result or should all results be ranked ? How should ties be managed ? Should similar metrics be incorporated or should each metric be measuring something very different (*e.g.* the between-within ratio and the average silhouette width will often vote for the same model while the Dunn index produces very different results) ? The OPTICS k-Xi density-based clustering pipeline computes several distance-based metrics but up to now did not implement any way of merging together their results and making use of ensemble models. It was up to the user to select the most relevant metric or to merge them together. Recently, we applied the pipeline on natural language processing embeddings and the choice of the relevant metric revealed complex and the clusters were unstable. To alleviate this, we implemented ensemble metrics and models in the OPTICS k-Xi pipeline.

The ensemble metrics enabled to increase significantly the stability of the clusters and the reproducibility of the results on evaluation datasets. Additionally, we implemented the selection of the cosine distance and a parameter to set a maximum cluster size. One of the limitations of using distance-base metrics with hierarchical density-based clustering is that since some clusters may be nested within others, the minimal distance between all clusters becomes null, and many distance-based metrics rely on this concept of cluster separation. We observed that the main issue came from when points with high OPTICS distance are clustered together in one large cluster that surrounds all others, and that most often having a few small nested clusters would not impact the metrics significantly, and thus we implemented a restriction on the maximum size of a cluster. Another solution that could be investigated in the future would be to incorporate the hierarchical structure within the distance metrics. This vignette will introduce the natural language processing dataset we were investigating and showcase and introduce the new developments of the **opticskxi** package, with a particular focus on ensemble metrics and models.

Keywords: Density-based clustering, hierarchical clustering, distance-based metrics, ensemble models.

1. Introduction

1.1. Center for Suicide Research and Prevention

The dataset we investigated is the result of applying word2vec on a large corpora of 1,700 mental health related scientific publications, and to be more specific, suicide related.

At the CELEHS laboratory we are part of the Center for Suicide Research and Prevention and are working on building suicide risk prediction models for clinicians to enable them to follow-up in particular with at-risk patients, in particular in two populations of interest: military veterans and teenagers. For that objective, we analyze electronic health records (EHRs) of patients which contain codified data and unstructured text data. Each time a patient visits a hospital, an EHR will be filled with information such as the diagnoses performed or the medications prescribed, which corresponds to the codified data, and with notes and comments from the clinicians justifying why they performed a diagnosis or prescribed a medication, which corresponds to the unstructured text data.

One analysis often performed is to measure the co-occurrence between features and apply pointwise mutual information and singular value decomposition. This enables to highlight which features often appear together in the data, e.g. side-effects of drugs. While the analysis of codified data is well-grounded and has been applied in hospitals for many years, the analysis of unstructured text data has recently gained more traction and results point to the usefulness of incorporating it. One of the challenges is the large number of features: while in codified data we have a specific set of diagnoses and medications, in unstructured text data we have a much larger vocabulary, especially when considering combinations of several words (*i.e.* n-grams). To analyse unstructured text, biomedical studies usually rely on ontologies which define lists of words or n-grams that were curated by clinicians and are relevant to analyze, as the Unified Medical Language System. However even then, the number of features remains large and calculating co-occurrence matrices is computationally challenging, as the number of concepts found in EHR datasets can be up to 100,000, making the computation of the co-occurrence matrices infeasible in practice. Thus analyses usually also rely on a first step of identifying which project-specific concepts are relevant to analyze, and then select only those to compute co-occurrence matrices from EHRs, *e.g.* up to 20,000.

One objective of the work described in this vignette was to build such a suicide-specific dictionary of concepts.

1.2. Word2vec embeddings

Word2vec has emerged in the mid 2010s as a powerful method for natural language processing and for discovering similarities between concepts in unstructured text (Mikolov 2013). The co-occurrence matrix method mentioned above is closely related (Levy and Goldberg 2014). Recent developments in large language models have significantly built upon word2vec and have largely replaced the use of word2vec in real-world applications, however they require extensive computation capabilities. In a related study, we show that much can be still done by interpreting carefully word2vec embeddings, and that the methodology is directly applicable to co-occurrence matrices and LLM-based embeddings. We focus on training a model based on a project-specific corpora and leverage the computationally cost-efficient capabilities of word2vec. Furthermore, much research is devoted to combining results from pre-trained LLM models with study-specific trained models, and consider this research to provide tools and insights towards that goal. Finally, the interpretation of results from LLMs also becomes more complex, and the use of word2vec and co-occurrence methods alleviate this. Overall, it is our opinion that analysis of such methods may still be adapted depending on the applications. While LLMs have outperformed word2vec on most benchmarks, clinical and biomedical studies still mostly rely on simple regression models that provide the robustness and generalizability that clinical interpretation and clinically-actionable knowledge requires.

A broader objective of this work was to extract insights from this corpora to understand the landscape of knowledge contained in those publications, to be able to build efficient suicide risk models.

1.3. Natural language processing embeddings

Dataset creation

Europe PMC maintains a FTP site enabling to download millions of open-access publications, which is introduced in the vignette of the **tidypmc** package. After downloading the publications locally, suicide-relevant publications were selected by searching for the pattern 'suicid' (to match both 'suicide' and 'suicidal') in lowercase-transformed titles. 8,000 publications were identified, and we took a random sample of 1,700 (20%) as a training dataset.

The full-texts of the 1,700 suicide-related publications were then pre-processed by removing some sections deemed irrelevant (*e.g.* supplementary data), transforming to lowercase, and removing partly the punctuation (but keeping *e.g.* dashes).

Word2Vec embeddings

The word2vec embeddings were performed with the **text2vec** package. Once we applied word2vec on the pre-processed input text, we obtained an embedding matrix of *e.g.* 30,000 words in rows, with as 100 columns (the embedding dimension parameter given to the word2vec method). So *e.g.* a word will be represented by a list of 100 numbers in the columns, and each word will have a row. This is our embedding matrix and we want to discover groups of words, *i.e.* related concepts, within the matrix.

1.4. Manual evidence of clusters

Single word queries

Using the embedding matrix, one can select a word, e.g. "medication", fetch the corresponding vector representation, and compare it to all other word representations and return the 50 closest matches. In this dataset, "medication" will return terms "antipsychotic" and "antidepressant", which are examples of mental health medications. This is one example of a group of related concepts, which we call clusters. Similarly, the word "therapy" will return "cbt" and "dbt", acronyms of cognitive behavioral therapy and dialectical behavior therapy.

Vector operations

Embedding models are also well suited to perform vector operations. Two well-known examples illustrate these: on an embedding matrix trained on general text, one can take the embedding vector of the word "king", subtract the one of "man", and add the one of "woman", to produce a list of closest matches (e.g. with the cosine distance) which will include the word "queen", indicating the capability of the model to understand language semantics. The other well-known example is "Paris" - "France" + "Germany" = "Berlin". The second example is particularly interesting because the top matches will also include other capitals as "Moscow", thus suggesting the evidence of a "capitals subspace".

In our dataset, as military veterans are one of our populations of interest, we are particularly interested in post-traumatic stress disorder (PTSD), and arguably, PTSD may be considered to have common symptoms with autism specturm disorder (ASD). By crafting a hypothesis such as "PTSD is to veterans what ASD is to children" and following the patterns of the well-known examples, the vector operation would become: "PTSD" - "veterans" + "ASD" = "children" ?; or alternatively: "PTSD" - "veterans" + "children" = "ASD" ? We found that these operations produced strikingly well-discriminated clusters. The first produced a list of socio-economic statuses (e.g. nationalities, professions, age-related descriptions as "adult", "children") and the second a list of psychiatric diagnoses (e.g. "borderline personality disorder" (BPD) and "major depressive disorder" (MDD).

Our manual exploration thus enabled us to find list of concepts that seemingly constituted clusters. Our next aim was to investigate if we were able to identify these clusters using automated methods, investigate their inter-relations, and try to further discover addtional clusters. For that purpoose, we chose to apply the OPTICS k-Xi density-based clustering pipeline, and chose to visualize the results with sparse coding graphs of nearest neighbors.

2. Density-based clustering with ensemble metrics

We first compiled a list of terms based on our manual exploration for which we expected that many would fall into well-separated clusters. We chose 23 vector operations, each combining additions and/or subtractractions from 1 to 4 words as the ones mentioned above, and for each of them return the 50 closest words. This produced a list of 831 unique words, and we subset our embedding matrix to those words and call OPTICS k-Xi on it.

2.1. Individual metrics

To demonstrate the advantages of ensemble metrics, we first show the limitations we encounter when using single metrics and compare the results with the ensemble metrics approach. We start by calling the OPTICS k-Xi pipeline with the newly implemented parameters (*metrics_dist*, *max_size_ratio*, *n_min_clusters*) and plot the default metric (average silhouette width) (Figure 1).

```
R>   library('opticskxi')
R>   data('m_psych_embeds')
R>   df_params = expand.grid(n_xi = 8:15, pts = c(15, 20, 25, 30),
+                               dist = "cosine", dim_red = "ICA",
+                               n_dimred_comp = c(10, 15, 20, 25))
R>   df_kxi = opticskxi_pipeline(m_psych_embeds, df_params,
+                                    metrics_dist = 'cosine',
+                                    max_size_ratio = 0.15, n_min_clusters = 5)
R>   plot(gtable_kxi_profiles(df_kxi))
```

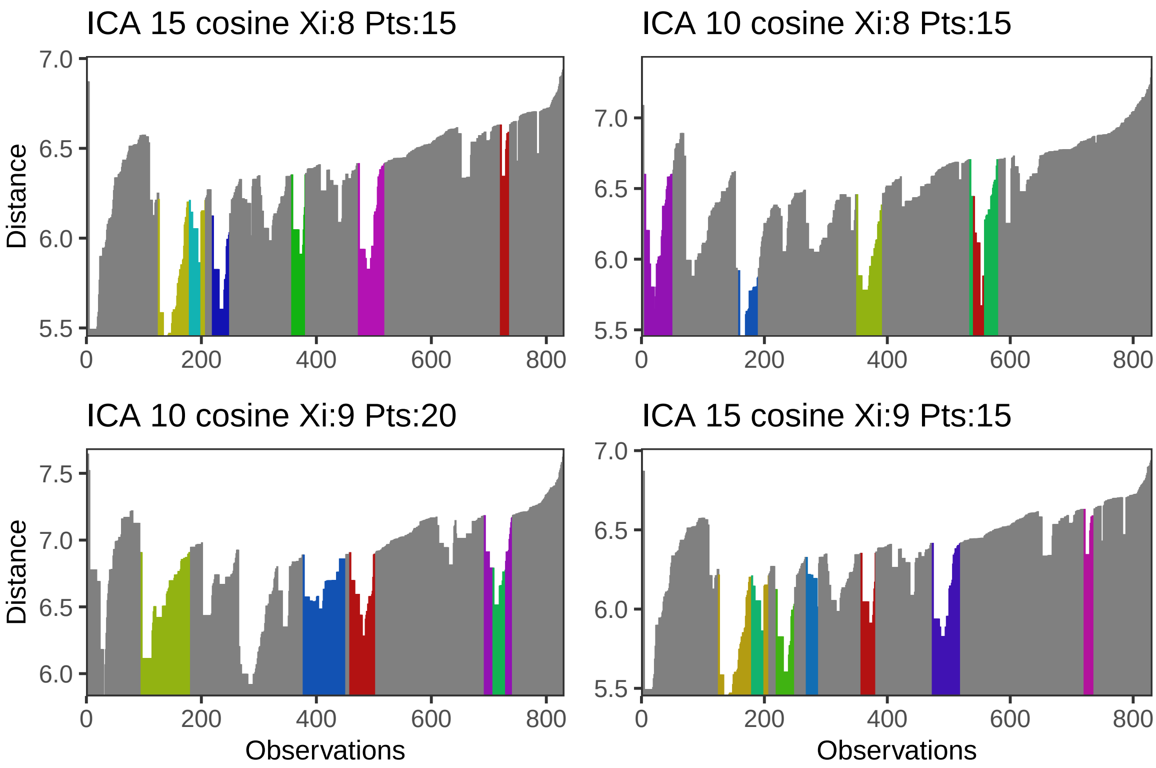

Figure 1: OPTICS k-Xi best 4 models for average silhouette width, ordered by columns then rows. The OPTICS distance profiles for the top models are quite different. The top profile (upper left) slightly ressembles the characteristic logarithmic profile of mediocre models indicating every points are very distant from each other and the clustering model is not optimal. Although some clusters were found, they are very small.

Let's now have a look to two other common metrics: the between-within ratio (Figure 2) and the Dunn index (Figure 3).

```
R>   plot(gtable_kxi_profiles(df_kxi, metric = 'bw.ratio'))
```

```
R>   plot(gtable_kxi_profiles(df_kxi, metric = 'dunn'))
```

We can also plot the metrics values (Figure 4).

```
R>   plot(ggplot_kxi_metrics(df_kxi, n = 15,
+                               metric = c("avg.silwidth", "bw.ratio", "dunn")))
```

2.2. Ensemble metrics and models

The ensemble metrics and models have been developed as two nested modules with their own sets of parameters.

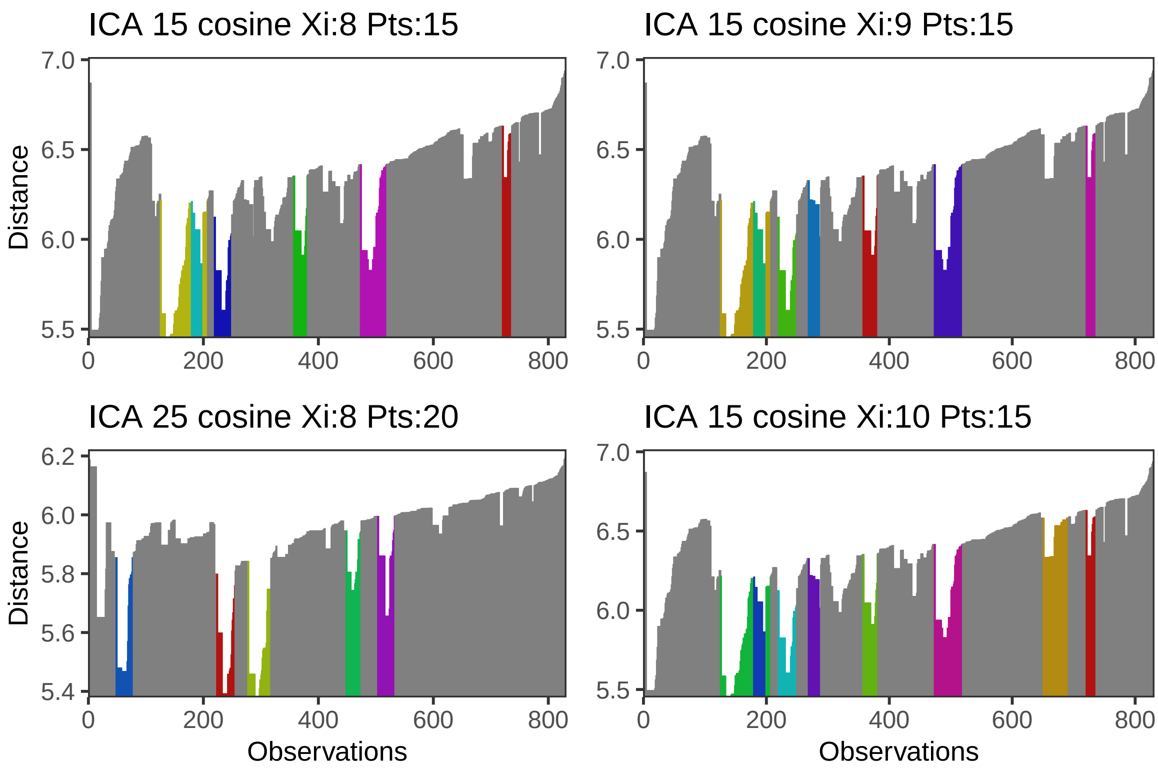

Figure 2: OPTICS k-Xi best 4 models for between-within ratio, ordered by columns then rows. The top model is the same as for average silhouette width, which is expected since the metrics are similar.

Summing of thresholded ranks

The function *ensemble_metrics* is the most inner one and will rank the metrics pre-computed by the OPTICS k-Xi pipeline. Here the parameters are:

- *n_top* Threshold of number of models to rank
- *df_params* The models dataframe, output of $opticskxi_pipeline$
- *metrics* Names of metrics to use, *NULL* for all (by default 8)
- *metrics_exclude* Names of metrics to exclude
- *n_models* Number of best models to return

Several approaches can be taken to sum the ranks of the models. To focus on the best models, we choose to rank only the top models for each metrics and set all other to 0, instead of *e.g.* summing the ranks of all models over all metrics. This behavior is controlled by the *n_top* parameter

In a second step, we sum the ranks and return only the top models, and this is controlled by the *n_models* parameter. The output is a list of the rankings matrix, for quality control purposes, and the selected models' parameters data frame (Table 1).

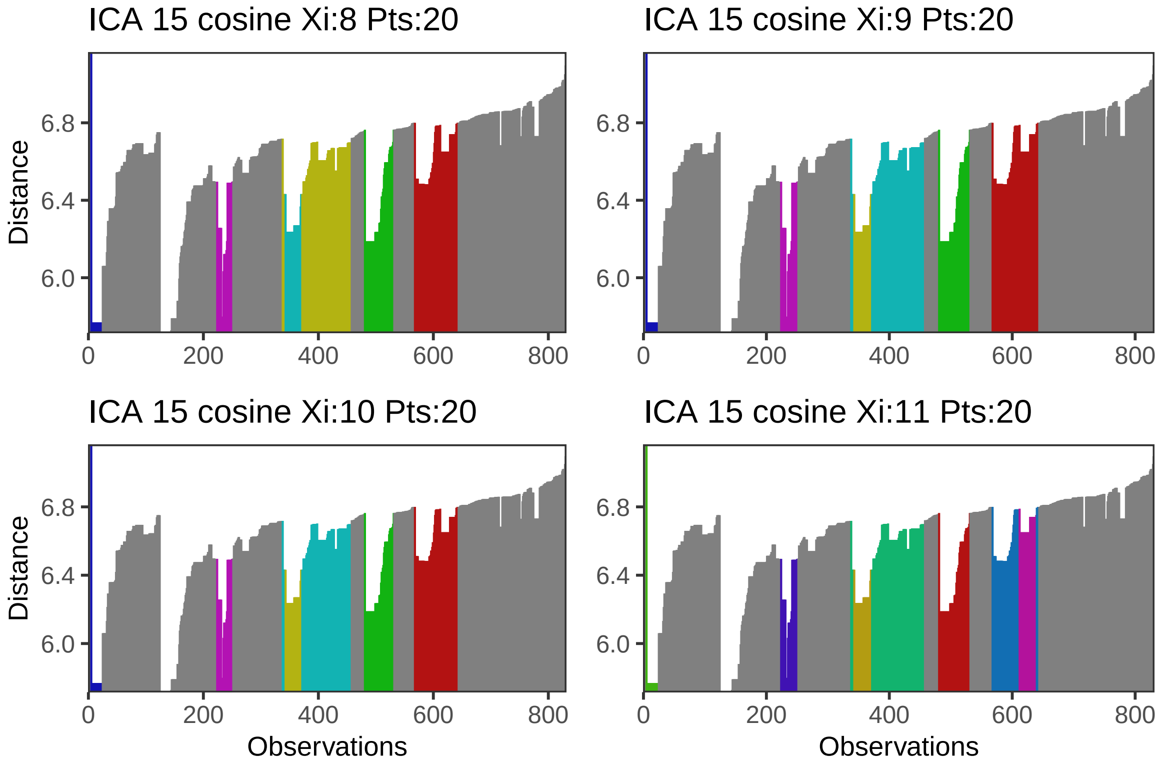

Figure 3: OPTICS k-Xi best 4 models for Dunn index, ordered by columns then rows. However, the second model might be more interesting to investigate, as we obtain more clusters and more points are clusters in total.

```
R>   ensemble_metrics(n_top = 50, df_params = df_kxi)[[1]] %>%
+      print_vignette_table('Ensemble')
```

Bootstrapping on several rank thresholds

The outer function is *ensemble_models* and is meant to be used on metrics objects computed with several different values of *n_top*. Above we have set $n_top = 50$, here we use 10%, 20%, and 50% of the number of models tested (Figure 5).

```
R>   df_ensemble_kxi = ensemble_models(df_kxi, n_models = 4,
+                                      model_subsample = c(0.1, 0.2, 0.5))
R>   plot(gtable_kxi_profiles(df_ensemble_kxi))
```

We can also visualize the clustering with dimensionality reduction. We performed the OPTICS k-Xi pipeline with independent component analysis (ICA) and the best model used 10 components. In contrast with principal component analysis (PCA), ICA does not order components and the results on less components will not be a subset of the components in higher dimensions. Still, for ease of visualization and summarization, here we display the clustering results obtained with 10 components on an ICA performed with only 4 components (Figure 6).

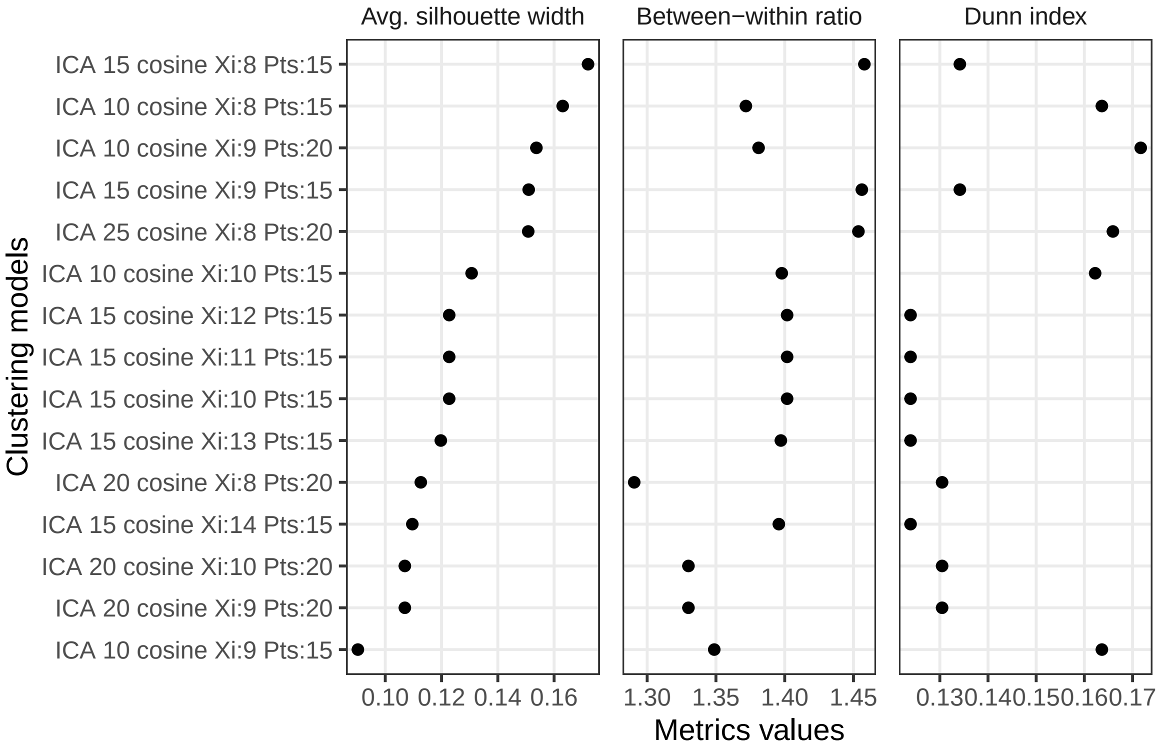

Figure 4: Numeric values of metrics for the top models by average silhouette width. Average silhouette width and between-within ratio have mostly similar results, while the Dunn index results are very different.

```
R>   df_ica = fortify_ica(m_psych_embeds, n.comp = 4,
+                          sup_vars = data.frame(Clusters = df_ensemble_kxi$clusters[[1]]))
R>   ggpairs(df_ica, 'Clusters', ellipses = TRUE, axes = 1:4) %>% grid::grid.draw()
```

For comparison, here are the clusters that would've been obtained using the model with best Dunn index metric (Figure 7).

```
R>   best_kxi <- get_best_kxi(df_kxi, metric = 'dunn')
R>   fortify_ica(m_psych_embeds, n.comp = 4,
+                 sup_vars = data.frame(Clusters = best_kxi$clusters)) %>%
+      ggpairs('Clusters', ellipses = TRUE, axes = 1:4) %>% grid::grid.draw()
```

3. Conclusions

This vignette showcased and demonstrated the use of ensemble metrics and models in the **opticskxi** package. While here the user could've manually investigated a few models and chosen the most appropriate one manually, these methods were also particularly implemented to enable chaining of several passes of OPTICS k-Xi density-based clustering, thus necessitating an automated way of choosing the best clustering models.

avg.silwidth	bw.ratio	ch	pearsongamma	dunn	dunn2	entropy	widestgap	sindex
0	22	33	23	0	0	11	31	49
29	0	25	22	7	0	50	0	40
0	50	50	0	37	4	24	8	0
37	0	7	42	0	34	19	32	3
22	32	8	32	0	41	0	0	43
9	0	42	18	0	38	0	49	24
26	30	26	24	0	16	3	28	46
17	9	0	31	31	45	27	43	0
20	6	20	29	34	48	0	46	0
34	31	35	0	33	47	0	45	0

Table 1: In the first slot of the object returned by *ensemble_metrics*, we can investigate which metric voted for each model. Here we have the 10 models with highest sum of metrics ranks thresholded to 50. The top ensemble model was not the top model for any metric (otherwise the rank value would be 50), but was in the top 10 for 4 metrics (rank greater than 40), and was outside of the top 50 for 3 metrics (rank value set to 0).

Further details on the analysis of this dataset within the CSRP and the larger NLP method which makes use of OPTICS k-Xi ensemble metrics and chains several passes will be published in another NLP dedicated package. Curious readers can follow me on social media to keep updated or send me an e-mail for further details.

4. Acknowledgements

This work was funded by the CSRP (https://csrp.mgh.harvard.edu/) and the CELEHS laboratory at Harvard Medical School led by Prof. Tianxi Cai (https://celehs.hms.harvard.edu/), and was initiated while I was working as a consultant for the non-profit research organization Parse Health (https://parse-health.org/).

References

Levy O, Goldberg Y (2014). "Neural word embedding as implicit matrix factorization." *Advances in neural information processing systems*, **27**.

Mikolov T (2013). "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*, **3781**.

Affiliation:

Thomas Charlon
CELEHS Laboratory
Department of Biomedical Informatics

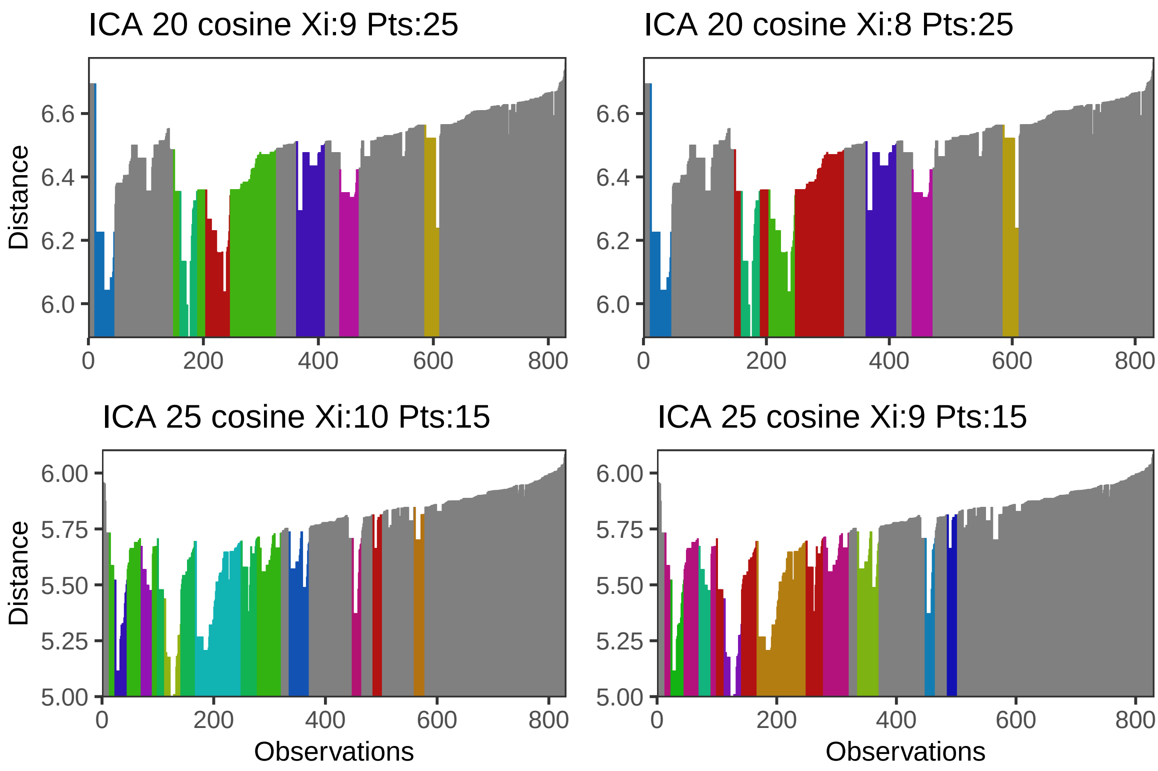

Figure 5: OPTICS k-Xi best 4 models for ensemble metrics, ordered by columns then rows. The best model corresponds to the second best Dunn index model.

Harvard Medical School
10 Shattuck Street, Boston
E-mail: charlon@protonmail.com

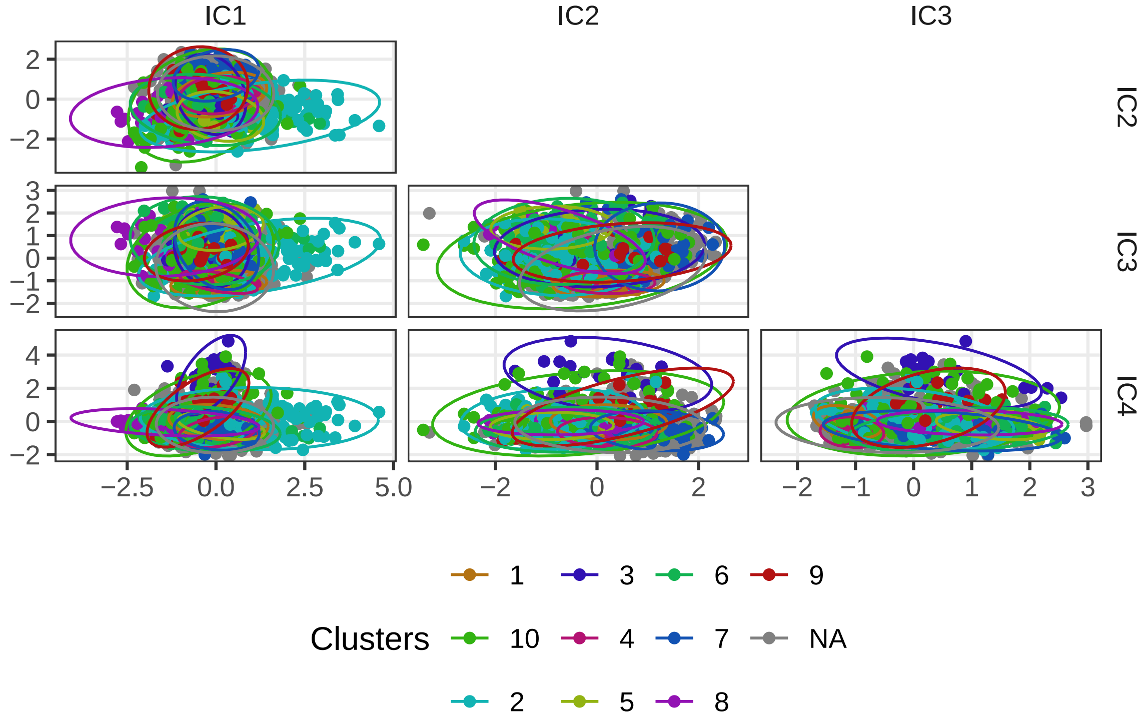

Figure 6: ICA dimensionality reduction, with ensemble clusters mapped. Components 2 and 3 have the less cluster overlap overall.

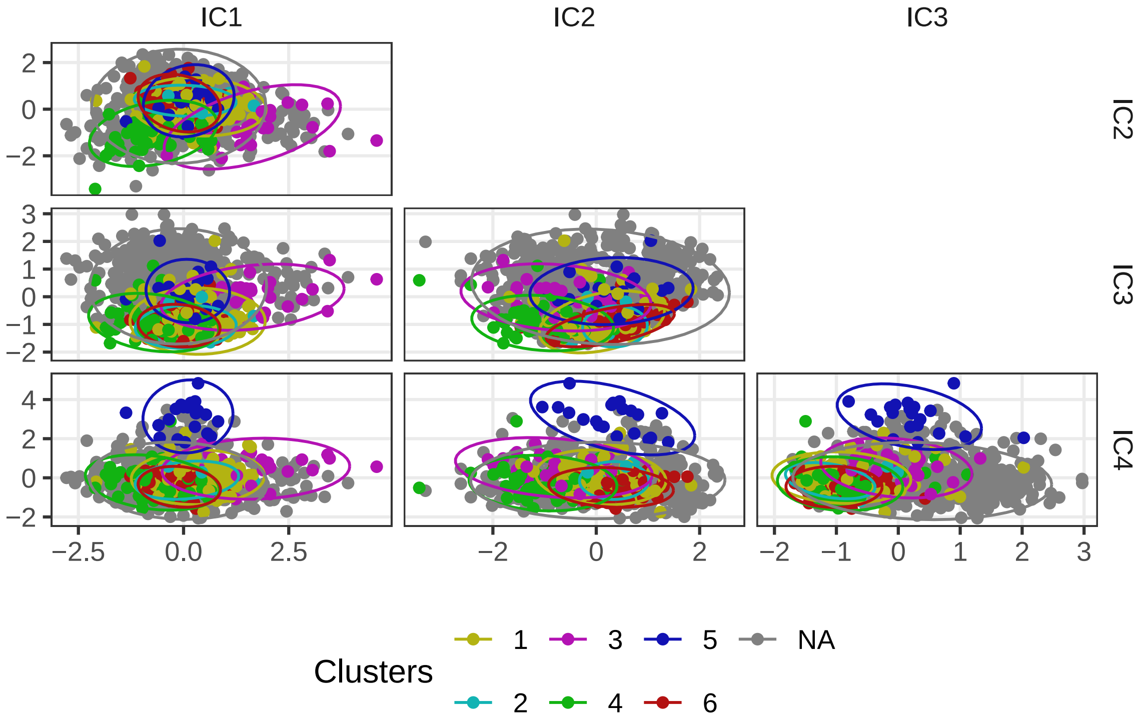

Figure 7: ICA dimensionality reduction, with best Dunn index clusters mapped.