Project 2: Clustering

MVE441

Ivan Flensburg, Filip Westberg, Victor Brun

May 4, 2022

Data:

- Data has no missing values
- $\approx$ 20.500 features
- $\approx$ 800 samples
- Unimodal?

Data exploration:

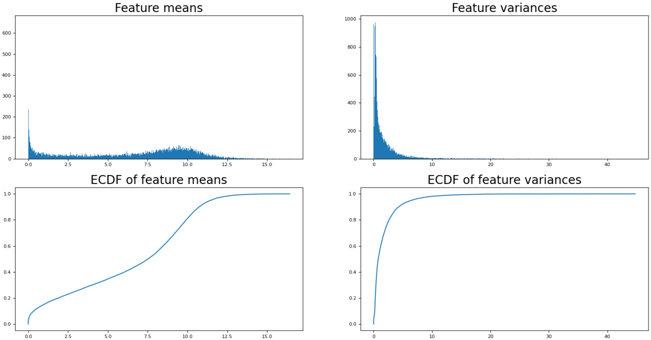

Figure: Histograms[2] and ECDF:s for sample means and variances. Feature variances while in a large span, are very dense close to zero - i.e pseudo-constant features

[1] No. of bins : 1000

Data exploration cont'd: variance filtering

A variance threshold of 2.2 leaves 5071 features remaining

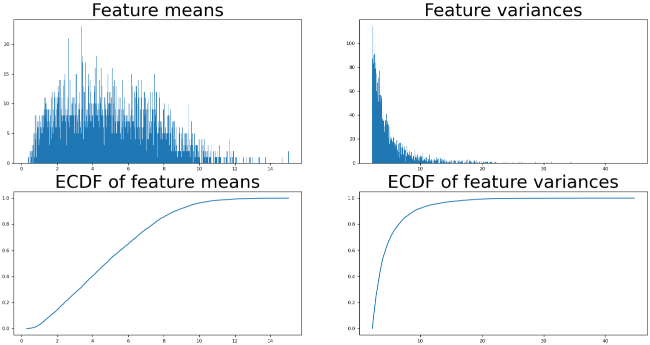

Figure: Note the disappearance of features with zero means, still wide ranges means standardisation[4]is of interest

[2] Centering and scaling - we tested several scalers but settled on a MinMax.

PCA:

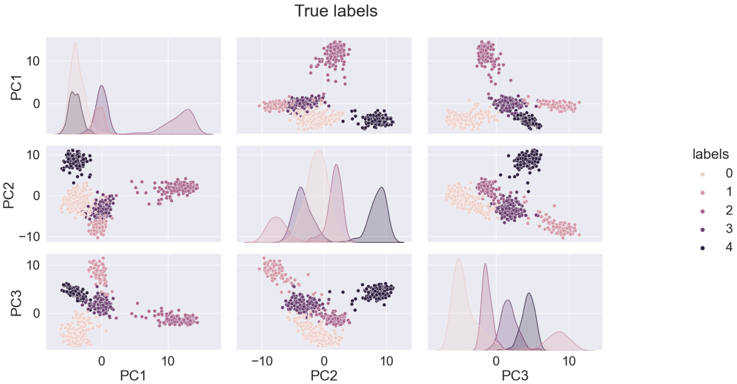

Figure: Pairplots 5 of the leading three eigenvectors, colored according to the true labels

PCA cont'd:

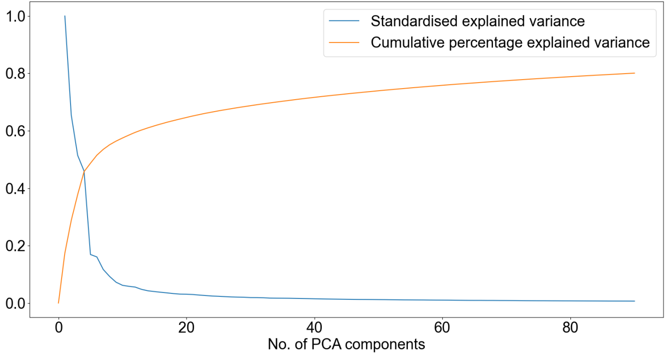

Figure: Screeplot of 90 components with 80% of explained variance (EV), with ≈ 4900 components dismissed. We "standardise" by dividing by the largest eigenvalue.

Clustering:

- We use Kmeans, GMM and Agglomerative Hierarchical clustering with Davies-Bouldin, silhouette and Calinski-Harabasz indices
- All these indices have a tendency to score convex clusters higher - though the pairplots show semi-convex clusters
- We estimate from the pairplots that there are ≥ 4 clusters and so run between 3 to 7 clusters.

Clustering cont'd:

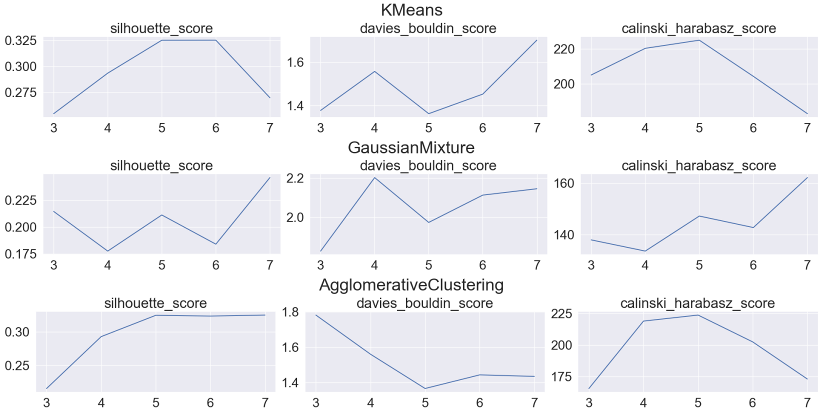

Figure: Silhouette, Davies-Bouldin (DB) and Calinski-Harabasz (CH) metrics for different cluster methods, we're looking for some kind of agreement between them.

Clusters:

Metrics:

- Silhouette: Ranges in $[-1, 1]$, with 1 as optimal
- DB: Metric is relative, with lower values being better (within-cluster scatter/between-cluster scatter)
- CH: Higher values are better (between-cluster dispersion/within-cluster dispersion)

"Optimal" cluster counts:

- KMeans: Leaning towards 5, with 4,6 as possibilities.
- GMM: Disregarding confirmation bias, this seems very inconclusive.
- Agg.Hier. : Here 5 seems "best", with 4,6 competing.

Predicted clusters:

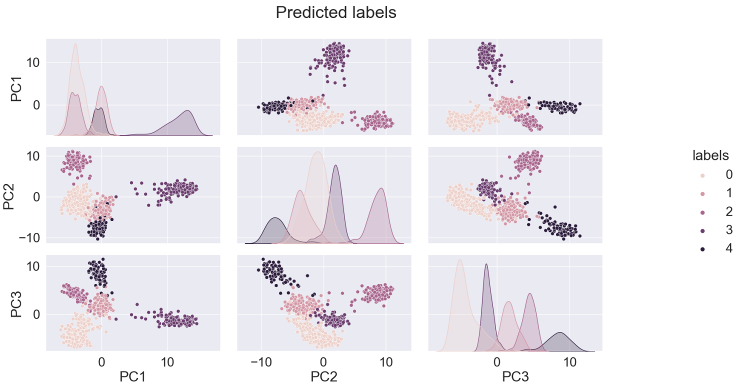

Figure: Predicted labels with agglomerated hierarchical clustering, looking for 5 clusters.

Predicted clusters cont'd:

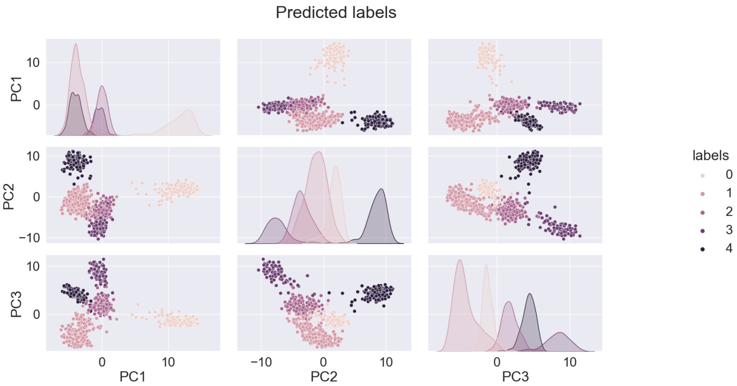

Figure: Predicted labels with Kmeans, looking for 5 clusters.

Predicted clusters cont'd:

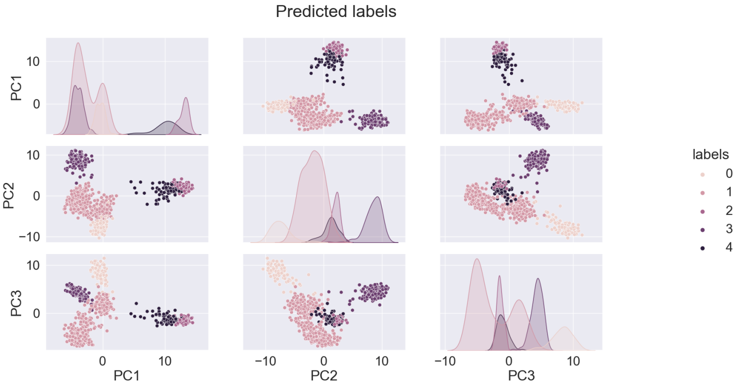

Figure: Predicted labels with GMM, looking for 5 clusters. This seems substantially worse than previous models.

Agreement:

- We quantify the overlap between our predicted clusters and the ground truth using Fawlkes-Mallow (FM) and adjusted Rand score.
- Stochastic models were run with $n\_inits = 1000$, agglomerative clustering was run with average linkage.

	Adj. Rand	FM
Kmeans	0.987	0.990
GMM	0.650	0.767
Agg. Hier.	0.983	0.987

Agreement cont'd:

- In this case then, GMM came off worse than either Kmeans or Agglomerative clustering, with more inconclusive (and worse) internal indices and lower comparative metrics.
- Kmeans and Agglomerative clustering in turn have metrics that are almost suspiciously high, though seemingly agreeing with the pairplots.

Consensus kmeans clustering:

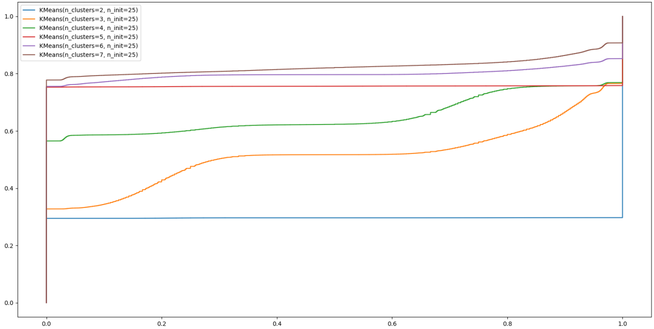

Figure: Consensus edf for kmeans with clusters 2:7, red is 5

Consensus gmm clustering:

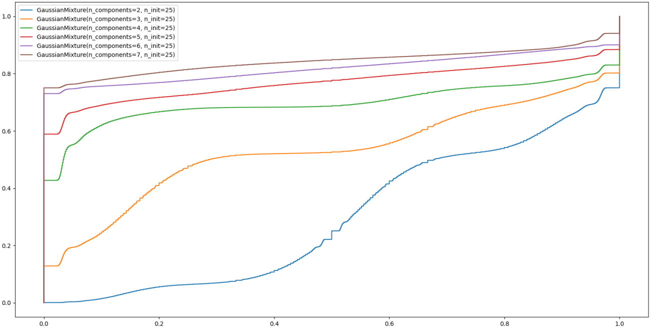

Figure: Consensus edf for GMM with clusters 2:7, red is 5

Consensus agglomerative clustering:

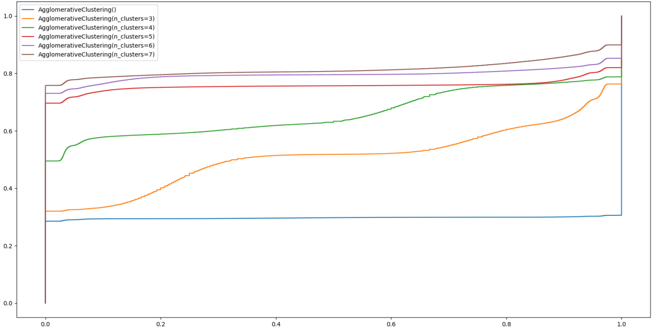

Figure: Consensus edf for hierarchical agglomerative clustering with clusters 2:7, red is 5

Consensus PAC:

Clusters	2	3	4	5	6	7
Kmeans	0.00249	0.438	0.204	0.00560	0.0968	0.130
GMM	0.750	0.674	0.403	0.295	0.170	0.190
Agg.	0.0205	0.443	0.293	0.124	0.122	0.141

Table: Model and no. of clusters vs. PAC for thresholds $(0.01, 0.99)$

- Here we see the relative instability of GMM, consensus is somewhat inconclusive for agglomerative clustering, with clearer results from kmeans.
- Our best guess is 5 or 6 clusters

Feature filtering:

Set up:

- standardised data,
- 50 repeated sub-samples for consensus matrix calculation.

Filters:

- none - fitting model to every available feature,
- variance filtering - fitting model to every feature with variance less than some threshold,
- principal components - fitting model to a specified number of principal components with larges eigenvalues,
- unimodal - fitting model to features having $p \geq 0.05$ in Hartigan's dip test,
- multimodal - fitting model to features having $p < 0.05$ in Hartigan's dip test.

Feature filtering: stability

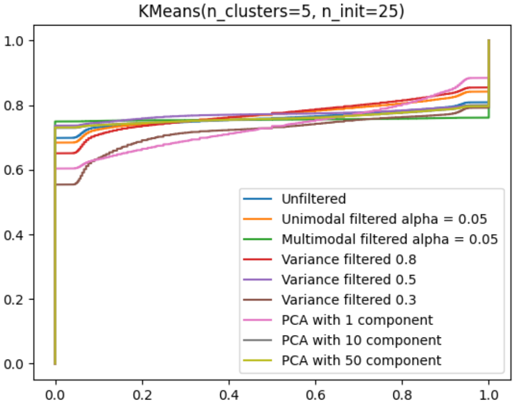

Figure: eCDF plot of the flattened consensus matrix for a K-means model, fitted to data to which different feature filters have been applied.

Feature filtering: performance

Filter	**PAC**	**Sil.**	**DB**	**CH**	**FM**	**AR**
None	0.110	0.155	2.336	75.32	0.990	0.987
Unimodal	0.158	0.143	2.519	71.11	0.816	0.760
Multimodal	0.012	0.162	2.097	90.36	0.979	0.972
$\hat{\sigma}^2 < 0.8$	0.204	0.137	2.583	68.03	0.810	0.752
$\hat{\sigma}^2 < 0.5$	0.062	0.197	2.117	107.0	0.858	0.814
$\hat{\sigma}^2 < 0.3$	0.238	0.093	2.638	40.24	0.920	0.889
1 PC	0.280	0.559	0.560	5178	0.433	0.236
10 PC	0.070	0.389	1.147	275.1	0.987	0.983
50 PC	0.065	0.266	1.640	146.4	0.990	0.987

Table: Several metrics for a K-means model fitted to data to which different feature filters have been applied. Sil. = *Silhouette score*, DB = *Davies-Bouldin score*, CH = *Calinski-Harabasz score*, FM = *Fowlkes-Mallows score*, AR = *Adjusted rand score*.

Conclusions and questions:

- Multimodal filtering is the most stable and it is comparable in performance to the variance filters. They do however seem to perform worse than the PC filters (note: internal indices for 1 PC is not a good measure).
- Evaluation of internal indices requires a more systematized approach - i.e calculating mean/variance, some sort of consensus between indices should be made rigorous?
- PAC