Solutions 12: Dimension reduction

BIO-369

Prof. Anne-Florence Bitbol
EPFL

1 Breast cancer cell features

a) See Jupyter notebook.
In `data`, there are 569 rows and 30 columns. Each row corresponds to a patient and each column to a feature. Consistently, there are 30 elements in `features`, and 569 elements in `labels`. The features are quite diverse and do not have the same units (for instance radius and area clearly do not, although they are related, and many others are more different, e.g. smoothness, compactness etc.). Given this heterogeneity of the features, comparing the overall scale of different features does not make much sense. Thus, it seems to be a good idea to standardize each column of the data, so that its mean is zero and its standard deviation is one.

b) See Jupyter notebook.
There are apparent blocks in the correlation matrix, and in particular, some blocks appear to have very high correlation values, close to 1.
Focusing on the first feature (feature 0), by looking at the first row, we find that five other features have a correlation coefficient larger than 0.9 with this first feature. Looking at their names, feature 0 is 'mean radius', while those to which it is highly correlated are 'mean perimeter', 'mean area', 'worst radius', 'worst perimeter' and 'worst area'. This makes sense because the area and perimeter can be directly computed from the radius in the simple case where the nucleus has a spherical shape and a circular section in the field of view. Furthermore, if the shape is almost spherical/circular, then the worst radius and the mean radius are going to be close to one another, and more generally they are expected to be positively correlated. These 6 features are highly dependent and correlated.
For the second feature, 'mean texture', we observe that it has a high correlation only with 'mean texture'. Again, this makes sense, these are two measures of texture. It is also interesting to note that texture and radius are not highly correlated.
Given these observations, the data does not have 30 independent features or dimensions, as some features are highly dependent and correlated.

c) See Jupyter notebook.
Each normalized eigenvalue of the correlation matrix represents the fractions of the variance explained by the direction associated to its eigenvector. Here the first one explains 44% of the variance, the second one 19%, etc. The cumulative sum gives us the fraction of the variance explained by including all directions up to the eigenvector associated to this eigenvalue (63% if we take the first and second ones).
Here we can see that starting from the third eigenvalue, directions carry less than 10% of the variance each, and that with 6 or 7 directions, 90% of the variance of the data is explained. This confirms that the effective dimension of the data is substantially smaller than 30, consistently with our observation that some features are highly dependent on others.

d) See Jupyter notebook.
As expected from this rescaling (standardization) of the data, we find that the first column of the scaled data has mean 0 and standard deviation 1.
In the plot where the dimension of the data has been reduced to 2 by PCA, the two types of cell samples occupy different regions of space, as seen by the red and blue markers, but the border between the two areas is quite extended and dense, and here are some a few red markers in the region occupied mainly by blue ones. Furthermore, we do not have two clusters isolated from one

another. Thus this representation of these measures can potentially help to make a diagnosis but an important uncertainty will remain.

e) See Jupyter notebook.

f) See Jupyter notebook.
In light of our answers to question b), we should choose the pair of features (0,1) rather than (0,2) if we wish to best describe the data with just two features. Indeed features 0 and 1 are weakly correlated and carry different types of information, while 0 and 2 are strongly correlated and essentially carry twice the same information. Retaining just this pair of features, the separation between the two types of cells looks worse compared to the PCA case (more red markers in the blue region), but the separation is not terribly bad either. Hence, it is better to focus on the first two PCA directions, but here, the difference is not huge.

g) See Jupyter notebook.
In 3 dimensions we can see that points are spread in the PCA space but that they are not in the original feature space, which is due to the high correlations between features 0 and 2. PCA allows to systematically control the specific dimensions to add.

h) See Jupyter notebook.
Here again, we can see a rather good separation between blue and red markers, but there are still some red markers in the blue zone. A new point compared to before is that now there seem to be two separated "clusters" in the data, with one of them composed essentially of red markers (except 3 blue ones), which could be helpful to improve diagnosis. But we also note that there is a substantial number of red markers in the blue "cluster" (on its left side).

i) See Jupyter notebook.
Here again, we can see a rather good separation between blue and red markers, but there are still some red markers in the blue zone. The separation between two "clusters" in the data is less clear than with UMAP but clearer than with PCA.

2 Single-cell RNA sequencing and cell types

a) See Jupyter notebook.
In `mrna_data`, there are 19,972 rows and 3005 columns. Each row corresponds to a gene and each column to a cell. Consistently, there are 3005 elements in `mrna_labels`, which give the type of each cell. We should be careful about this data format because here rows are variables and columns are observations of all the variables. The most standard data format for dimension reduction is the opposite one.

b) See Jupyter notebook.

c) See Jupyter notebook.
Before applying PCA (or UMAP or t-SNE), we need to transpose the transformed data, because of the non-standard rows and columns explained in a).
In the two-dimensional representation of the data from PCA, we can see some separation between cell types (e.g. oligodendrocytes are rather well separated from the rest) but there are lots of overlap overall. Besides, each cell type seems to yield a rather elongated and noisy cloud of points.

d) See Jupyter notebook.
With UMAP, we observe that all the cell types now form quite well separated clusters, and that there is essentially one cluster per cell type. There are a few exceptions (markers of one color in a cluster of another column). Here UMAP gives a much more interpretable picture than PCA and it looks like this plot could be used to label unknown cells with a reasonably good accuracy. We also see that a larger-scale structure seems to emerge with three groups of cell types closer and thus more similar within their group than to others (pyramidal CA1 and SS1, and interneurons on the one hand, astrocytes, endothelial-mural and microglia on the other, and oligodendrocytes alone). Looking back at PCA we can see some indication of such a grouping but it was far less clear. From this analysis, this grouping remains speculative.

e) See Jupyter notebook.
With t-SNE, results are rather similar overall to UMAP, with rather well-defined clusters for each cell type, and the same potential groupings that emerge. However some aspects are less clear than with UMAP, in particular a few cells are completely outside of clusters, which did not occur with UMAP, and the separation between clusters is less neat. Furthermore, quite strikingly, interneurons now have two clusters instead of one, and they are separated by the pyramidal SS cluster, with one of them being very close to it. We also observed a rather mixed cluster in this area, which also existed with UMAP.
Overall, for this dataset, the two non-linear methods perform much better than PCA in terms of providing an interpretable visualization of the data.