Generalised advanced supervised PCA

Contents

[1 General formula 1](#_Toc70342329)

[2 Memory optimisation 2](#_Toc70342330)

[2.1 Calculation of diagonal elements 2](#_Toc70342331)

[2.2 Calculation of required matrix 2](#_Toc70342332)

[3 Comparison of figures 5](#_Toc70342333)

# General formula

This method is generalisation of advanced supervised PCA to the case of individual attraction of points of each class and individual repulsion of each pair of classes.

Let us consider dimensional space with objects (observations, records) . The attribute number of all records is vector , all attributes of the object is vector . Value of attribute for object is . We also assumed that dataset is centralised: .

Let us have several classes , where is number of classes. Object belongs to class if .

We are interested in finding the linear -dimensional manifold (subspace because of centralisation of ) such that projection points of one class are as close as possible, and projections of points of different classes are as far as possible. Let us denote projection of onto target manifold as

where where symbol “” means transposed matrix, is Kronecker delta:

Let us consider attraction of projections of points of class as and repulsion of points of classes and as .

This means that we want to maximise the following square form:

|  |  |
| --- | --- |
|  | (1) |

Maximisation of (1) can be achieved through search of eigenvectors of matrix , where is Laplacian matrix with elements:

|  |  |
| --- | --- |
|  | (2) |

# Memory optimisation

Matrix cannot be formed in normal size of memory. To avoid such necessity we can consider calculation of and matrix .

For simplicity we consider the case that data matrix is sorted and classes can be described as where

## Calculation of diagonal elements

Let us calculate where . In this case row of matrix will contains elements , , element and one diagonal element equal to minus sum of other elements. This means that

## Calculation of required matrix

Let us consider for simplicity three classes case with cases in each.

Matrix can be presented in the form

where is matrix with off-diagonal elements and diagonal terms , and is matrix with all elements . We also know that .

Data matrix can be represented in the following format

Then we have the following structure:

Let us calculate the product of and :

Now let us completed calculation:

Now let us recall that matrix are all ones matrix with some multiplier for . Let us denote all ones matrix with rows and comulns as

|  |  |
| --- | --- |
|  | (3) |

Let us consider calculation of .

where is mean value of attribute in class . Now let us complete calculation:

Finally we can write:

|  |  |
| --- | --- |
|  | (4) |

For we can write

|  |  |
| --- | --- |
|  | (5) |

Now let us consider diagonal block . It is is matrix with off-diagonal elements and diagonal terms . It is clear that can be presented as weighted sum of all one matrix and identity matrix:

where is identity matrix.

The first summand multiplied by can be rewritten through (4) as

|  |  |
| --- | --- |
|  | (6) |

The second summand multiplied by can be rewritten as

Finally is

|  |  |
| --- | --- |
|  | (7) |

Formulae (3), (5), and (7) allows calculate matrix A with usage of matrix and fragments of data matrix. This procedure does not require huge Laplacian matrix.

# Comparison of figures

The comparison of Advanced supervised PCA and Generalised supervised PCA.

|  |  |
| --- | --- |
| Generalised supervised PCA | Advanced supervised PCA |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |