TANA15 Numerical Linear Algebra

Fredrik Berntsson (fredrik.berntsson@liu.se)

Computer Exercise 4. The SVD and Applications

1 General information

The programming assignments are intended to take more than four hours to complete. It is therefore important to be well prepared so that the computer sessions are used effectively. The assignment consists of a mixture of theoretical exercises and practical programming.

A few of the exercises are based on custom Matlab programs or datasets that are provided in order to save time. These can be downloaded from the internet.

2 Data compression

The singular value decomposition of a $m \times n$ matrix A is

$$A = U\Sigma V^T = \sum_{k=1}^{r} \sigma_k u_k v_k^T,$$

where r is the rank of A. Let

$$A_p = \sum_{k=1}^{p} \sigma_k u_k v_k^T, \qquad p \leq r.$$

Then A_p is the best approximation to A from the class of rank p matrices.

Exercise 2.1 Show that $\mathrm{rank}(A_p) = p$ and that

$$|A - A_p\|_2 = \sigma_{p+1}. \qquad \square$$

Any digital image, with mn pixels, can be viewed as an $m \times n$ matrix A. Thus we can approximate the the image by an expansion of the first p singular components, i.e. $A \approx A_p$. If $p < r$ we save storage. This can be useful if the image is to be transmitted over an electronic network.

Exercise 2.2 Suppose we approximate A by A_p. How much memory is saved? Write down an expression that depend on m, n, and p. $\square$

Copy the file *Goose.mat* from the course library. You can view the original image by typing:

```
>> load Goose.mat
>> colormap('gray');
>> imshow(GooseBW);
```

Next we want to compute, and display, a low rank approximation of the image `A=GooseBW`. Type

```
>> [U,S,V]=svd(A); ,
>> p=10 , Ap=U(:,1:p)*S(1:p,1:p)*V(:,1:p)';
>> imshow(Ap);
```

and a rank $p = 10$ image is computed and displayed.

Exercise 2.3 Experiment and find a value of p that gives a good approximation. How much memory can you save? $\square$

Exercise 2.4 Display the singular values of the matrix A by typing `semilogy(diag(S))`. The singular values tend to zero rapidly and the approximation $A \approx A_p$ improves.

The singular value σ_1 is special. What does A_1 look like?

The singular values are a measure of the "energy" of the different rank one components that make up the original image. How many components are needed to capture 25, 50, and 75%, of the total energy? Type

```
>> s=diag(S); p=10; sum(s(2:p))/sum(s(2:end))
```

to compute the energy included in the rank p approximation. $\square$

3 Circle fitting

We want to fit a circle to m given points (x_i, y_i). Any circle can be represented as,

$$F(u) = a(x^2 + y^2) + b_1 x + b_2 y + c = 0.$$

Here $u = (a, b_1, b_2, c)^T$ are the unknown parameters. By inserting the data points (x_i, y_i) into the expression for $F(u)$ we get a linear system $Bu = 0$, where $B \in \mathbb{R}^{m\times 4}$. In order to obtain a non–trivial solutuon we put a constraint $\|u\|_2 = 1$ on the solution, and thus solve the over determined system,

$$\min_u \|Bu\|_2, \quad \text{subject to} \quad \|u\|_2 = 1.$$

Hint The Matlab code `CircleData` can be used to create test data.

Exercise 3.1 Do the following

1. Show that the center $z = (z_1, z_2)^T$ and radius r of the circle can be written,

$$z = (z_1, z_2)^T = -(\frac{b_1}{2a}, \frac{b_2}{2a})^T, \qquad r = \sqrt{\frac{b_1^2 + b_2^2}{4a^2} - \frac{c}{a}}.$$

2. Give the expression for the matrix B.

3. Show that the solution of the minization problem is given by the right singular vector v associated with the smallest singular value of B.

4. Use Matlabs `svd` to find the best circle fit given this data set. Display the results. $\square$

4 Automatic Character Recognition

Often in applications you study objects that occur in various types, or classes. One is then intressted in determining the particular type, or class, of a certain unknown object. This is called the *Classification problem* and is used in many technical applications. Examples include mail filters where a particular email is either junk or it is not. Another application is the automatic sorting of mail done by the postal services. Each envelope is photographed and the postal code identified. The individual digits each has a type, i.e. one of the characters 0,1,. . . ,9. An automatic algorithm is then used to identify each digit; and the mail can be sent to the correct post office for distribution.

In this exercise we will study the problem of automatically reading handwritten digits. We are given a number of 16×16 pixel bitmap images that were all scanned from actual mail sent through the US Postal Service. Each image represents one handwritten digit. The images are divided into two sets. First there is a *Reference set* which is used to create the digit recognition algorithm. Second there is the *Test set* that is used to evaluate the algorithm.

Exercise 4.1 The images are available as a file `DataSet.mat`. Load the data into Matlab. There are two matrices `RefSet` and `TestSet` that contain the reference digits and the test digits respectively. Each column in the matrices represents one indivudual digit.

Use the program `DisplayDigit` to look at a couple of images. The vectors `RefAns` and `TestAns` contain the actual digit that the images are supposed to represent. So that `RefAns(k)` tells what digit is represented by the image `RefSet(:,k)`. $\square$

The *Reference set* is used to create the character recognition algorithm. The idea is to identify common traits for the different classes, i.e. the different digits. The basic idea is that we split the reference set into smaller subsets that collect digits of a specific type. Then we implement a function that measures how much a given unknown digit has with the digits in the different subsets.

The character recognition algorithm will be implemented using the following steps. Suppose that the matrix `Rj` collects all digits from the reference set that represent the digit j. We then compute the SVD,

```
>> [Uj,Sj,Vj]=svd(Rj);
```

Most of the digits in the reference set look very similar. There are a few different styles of writing but mostly all the handwritten digits are only small variations of others. This means that the collection of digits of a certain type can be well approximayed by a low rank approximation. The following idea can be used to compute the distance from an unknown digit D to one of the subspaces that collect digits of a single type.

- Approximate the space span(R_j) by a subspace span$(Uj(:, 1:k)$ for a fixed number k.
- Compute the orthogonal distance from D to the linear subspace span$(Uj(:, 1:k))$.

Exercise 4.2 Implement a function

```
>> d = DistanceFromSubspace( Uj(:,1:k) , TestDigit );
```

that computes the distance as detailed above. $\square$.

Hint: The orthogonal distance between a vector and a linear subspace can be formulated as a least squares problem.

Exercise 4.3 Write a Matlab function `ClassifyDigit` That uses the rank k subspaces created above, and uses the orthogonal distance from an unknown digit to the respektive subspaces to classify an unkown digit from the *test set*, i.e. a function

```
>> Type = ClassifyDigit( S , TheSubspaces );
```

Hint You need 10 different matrices; each with a subspace corresponding to one of the digits 0-9. Its convinient to let the input parameter `TheSubSpaces` be a `cell`-array and each cell contain one of the matrices $U_j(:, 1:k)$. Read the documentation about cell arrays. $\square$

Exercise 4.4 Use the `ClassifyDigit` function to determine the type of all the unknown digits in the matrix `TestSet`. Experiment to find a good value for the dimension k of the subspaces used. How many digits from the test set are classified correctly? $\square$

Remark The test set was collected from actual mail. But its not realistic in the sense that many of the "easy" cases were removed thus making this test set much more difficult than what you'd expect from the application. This is to make the difference in performance bigger between different algorithms.