Colorado School of Mines

Computer Vision

Professor William Hoff
Dept of Electrical Engineering &Computer Science
http://inside.mines.edu/~whoff/

Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD)

- SVD is a matrix technique that has some important uses in computer vision
- These include:
  - Solving a set of homogeneous linear equations
    - Namely we solve for the vector **x** in the equation **Ax** = **0**
  - Guaranteeing that the entries of a matrix estimated numerically satisfy some given constraints (e.g., orthogonality)
    - For example, we have computed **R** and now want to make sure that it is a valid rotation matrix

Singular Value Decomposition (SVD)

- Any (real) *mxn* matrix **A** can be written as the product of three matrices

  $$\mathbf{A} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$$

  - **U** (*mxm*) and **V**(*nxn*) have columns that are mutually orthogonal unit vectors
  - **D** (*mxn*) is diagonal; its diagonal elements σ_i are called singular values, and $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$

p=min(M,N)

$$\boldsymbol{A}_{M\times N} = \boldsymbol{U}_{M\times P}\boldsymbol{\Sigma}_{P\times P}\boldsymbol{V}^T_{P\times N}$$

$$= \left[\begin{array}{c|c|c} \boldsymbol{u}_0 & \cdots & \boldsymbol{u}_{p-1} \end{array}\right] \begin{bmatrix} \sigma_0 & & \\ & \ddots & \\ & & \sigma_{p-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_0^T \\ \hline \cdots \\ \hline \boldsymbol{v}_{p-1}^T \end{bmatrix},$$

- If only the first *r* singular values are positive, the matrix **A** is of rank *r* and we can drop the last *p-r* columns of **U** and **V**

$$\mathbf{U}^T\mathbf{U} = \mathbf{I},\ \mathbf{V}^T\mathbf{V} = \mathbf{I}$$

$$\mathbf{u}_i \cdot \mathbf{u}_j = \mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij}$$

Some properties of SVD

- We can represent A in terms of the vectors u and v

  $\mathbf{A}\,\mathbf{v}_j = \sigma_j\,\mathbf{u}_j$

- or

$$\mathbf{A} = \sum_{j=0}^{p-1} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

- The vectors $\mathbf{u}_j$ are called the "principal components" of **A**

- Sometimes we want to compute an approximation to **A** using fewer principal components
- If we truncate the expansion, we obtain the best possible least squares approximation[1] to the original matrix **A**

$$\mathbf{A} \approx \sum_{j=0}^{t} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$$

[1] In terms of the Frobenius norm, defined as

$$\|\mathbf{A}\|_F = \sum_{i,j} a_{i,j}^2$$

Some properties of SVD (continued)

- We have

$$\mathbf{A} = \mathbf{U}\ \mathbf{D}\ \mathbf{V}^T$$

- Look at

$$\mathbf{A}\ \mathbf{A}^T = (\mathbf{U}\ \mathbf{D}\ \mathbf{V}^T)\ (\mathbf{U}\ \mathbf{D}\ \mathbf{V}^T)^T = \mathbf{U}\ \mathbf{D}\ \mathbf{V}^T \mathbf{V}\ \mathbf{D}\ \mathbf{U}^T = \mathbf{U}\ \Lambda\ \mathbf{U}^T$$

- where $\lambda_i = \sigma_i^2$

- Multiplying by **U** on the right on each side yields

$$(\ \mathbf{A}\ \mathbf{A}^T\ )\ \mathbf{U} = \mathbf{U}\ \Lambda$$

- or

$$(\mathbf{A}\ \mathbf{A}^T)\ \mathbf{u}_j = \lambda_j\ \mathbf{u}_j$$

- So the columns of **U** are the eigenvectors of $\mathbf{A}\ \mathbf{A}^T$

Some properties of SVD (continued)

- Similarly, we have

  $\mathbf{A} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$

- Look at

  $\mathbf{A}^T\mathbf{A} = (\mathbf{U}\,\mathbf{D}\,\mathbf{V}^T)^T(\mathbf{U}\,\mathbf{D}\,\mathbf{V}^T) = \mathbf{V}\,\mathbf{D}\,\mathbf{U}^T\mathbf{U}\,\mathbf{D}\,\mathbf{V}^T = \mathbf{V}\,\Lambda\,\mathbf{V}^T$

- where $\lambda_i = \sigma_i^2$

- Multiplying by **V** on the right on each side yields

  $(\mathbf{A}^T\mathbf{A})\,\mathbf{V} = \mathbf{V}\,\Lambda$

- or

  $(\mathbf{A}^T\mathbf{A})\,\mathbf{v}_j = \lambda_j\,\mathbf{v}_j$

- So the columns of **V** are the eigenvectors of $\mathbf{A}^T\mathbf{A}$

Application: Solving a System of Homogeneous Equations

- We want to solve a system of m linear equations in n unknowns, of the form $\mathbf{Ax} = 0$
  - Assume $m >= n$-1 and rank($\mathbf{A}$)=n-1
- Any vectors $\mathbf{x}$ that satisfy $\mathbf{Ax} = 0$ are in the "null space" of A
  - $\mathbf{x}$=0 is a solution, but it is not interesting
  - If you find a solution $\mathbf{x}$, then any scaled version of $\mathbf{x}$ is also a solution
- As we will see, these equations can arise when we want to solve for
  - The elements of a camera projection matrix
  - The elements of a homography transform

Application: Solving a System of Homogeneous Equations (continued)

- The solution **x** is the eigenvector corresponding to the only zero eigenvalue of $\mathbf{A}^T\mathbf{A}$
  - Proof: We want to minimize

$$\|\mathbf{Ax}\|^2 = (\mathbf{Ax})^T \mathbf{Ax} = \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} \qquad \text{subject to} \quad \mathbf{x}^T \mathbf{x} = 1$$

  - Introducing a Lagrange multiplier λ, this is equivalent to minimizing

$$L(\mathbf{x}) = \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - \lambda(\mathbf{x}^T \mathbf{x} - 1)$$

  - Take derivative wrt **x** and set to zero

$$\mathbf{A}^T \mathbf{Ax} - \lambda \mathbf{x} = 0$$

  - Thus, λ is an eigenvalue of $\mathbf{A}^T\mathbf{A}$, and $\mathbf{x} = \mathbf{e}_\lambda$ is the corresponding eigenvector. $L(\mathbf{e}_\lambda) = \lambda$ is minimized at λ=0, so $\mathbf{x} = \mathbf{e}_0$ is the eigenvector corresponding to the zero eigenvalue.

Example

- Let

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

- Find solution **x** to **Ax=0**

$$\mathbf{A}^T\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Eigenvalues and eigenvectors of $\mathbf{A}^T\mathbf{A}$:

$$\lambda_1 = 0, \mathbf{e}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \lambda_2 = 1, \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \lambda_3 = 1, \mathbf{e}_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

So $\mathbf{x}=\mathbf{e}_1$ is the solution. To verify:

$$\mathbf{Ax} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \mathbf{0}$$

So it does work

Solving Homogeneous Equations with SVD

- Given a system of linear equations **Ax** = 0
- Then the solution **x** is the eigenvector corresponding to the only zero eigenvalue of $\mathbf{A}^T\mathbf{A}$
- Equivalently, we can take the SVD of **A**; ie., $\mathbf{A} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$
  - And **x** is the column of **V** corresponding to the zero singular value of **A**
  - (Since the columns are ordered, this is the rightmost column of **V**)

- Example

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\text{Svd}: \quad \mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

So the last column of V is indeed the solution **x**

Solving Homogeneous Equations - Matlab

```
clear all
close all

% Solve the system of equations Ax = 0
A = [ 1  0  0;
      0  1  0 ];

[U,D,V] = svd(A);
x = V(:,end);                    % get last column of V
```

- Output

```
>> U
U =
    1   0
    0   1
>> D
D =
    1   0   0
    0   1   0
```

```
>> V
V =
    1   0   0
    0   1   0
    0   0   1
>> x
x =
    0
    0
    1
```

Another application: Enforcing constraints

- Sometimes you generate a numerical estimate of a matrix **A**
  - The values of **A** are not all independent, but satisfy some algebraic constraints
  - For example, the columns and rows of a rotation matrix should be orthonormal
  - However, the matrix you found, **A'**, does not satisfy the constraints
- SVD can find the closest matrix[1] to **A** that satisfies the constraints exactly
- Procedure:
  - You take the SVD of $\mathbf{A}' = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$
  - Create matrix **D'** with singular values equal to those expected when the constraints are satisfied exactly
  - Then $\mathbf{A} = \mathbf{U}\,\mathbf{D}'\,\mathbf{V}^T$ satisfies the desired constraints by construction

[1] In terms of the Frobenius norm

Example – rotation matrix

- The singular values of R should all be equal to 1 ... we will enforce this

```
clear all
close all

% Make a valid rotation matrix
ax = 0.1;   ay = -0.2;  az = 0.3;   % radians
Rx = [ 1 0 0; 0 cos(ax) -sin(ax); 0 sin(ax) cos(ax)];
Ry = [ cos(ay) 0 sin(ay); 0 1 0; -sin(ay) 0 cos(ay)];
Rz = [ cos(az) -sin(az) 0; sin(az) cos(az) 0; 0 0 1];

R = Rz * Ry * Rx

% Ok, perturb the elements of R a little
Rp = R + 0.01*randn(3,3)

[U,D,V] = svd(Rp);  % Take SVD of Rp

D    % Here is the actual matrix of singular values

% Recover a valid rotation matrix by enforcing constraints
Rc = U * eye(3,3) * V'
```