COT6505 Computational Methods

- Instructor: Dr. Mubarak Shah, shah@cs.ucf.edu, office:238 CSB, http://www.cs.ucf.edu/class/cap6411
- Office Hours:
  - 2PM to 3PM Mon, 4PM-5PM Tu, 3PM-4PM Thurs
- Grading
  - Mid term 25%, Final 35%, assignments (homework, programs),40%
- Text Book
  - Numerical Optimization, Jorge Nocedal and S. Wright, Springer, 1999.
- Other suggested Books
  - Numerical Analysis, Burden and Faires, PWS Kent.
  - Numerical Recipes in C, W. Press et al, Cambridge University press.

Optimization

- People optimize
  - Stocks
  - Job
  - exam
- Convert qualitative description into quantitative function
  - Objective function
  - Variables
  - constraints

Examples

- Transportation problem
- Camera Pose estimation in Computer Vision
- Chess playing
- Robot path planning
- Computing the optimal shape of an automobile or aircraft
- Controlling a chemical process or a mechanical device to optimize or meet standards of robustness

Optimization

- Minima, maxima or zero of a function
- Local minima vs global minima

Optimization Problems

- Single variable
- Multiple variables
- Linear
- Non-linear
- Constraint optimization

$$\min (x_1 - 2)^2 + (x_2 - 1)^2$$
$$\text{subject to } \{x_1^2 - x_2 \leq 0, x_1 + x_2 \leq 2$$

- Unconstraint optimization

Desirable Properties

- Robustness
- Efficiency
- Accuracy

Solution

- Iterative solution $X^0, X^1, X^2, \dots X^n$
  - Initial estimate
  - Convergence $X^n \approx X^{n-1}$

    $X^n \approx P$
    - Linear
    - Super linear
    - Quadratic

Numerical Optimization

- Computation of
  - derivatives,
  - gradient,
  - Jacobian,
  - Hessian
- Analytical derivatives not possible
- Numerical derivatives, finite difference
- Solution of a linear system (Inverse of a matrix)

- Derivative

$$f'(x) = \frac{df}{dx}, \quad x \text{ is a scalar}$$

- Gradient

$$\nabla f(x_1, x_2, \ldots, x_n) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)$$

- Jacobian

$$F(x_1, x_2, \ldots, x_n) = f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots, f_m(x_1, x_2, \ldots, x_n)$$

$$J(F) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

- Hessian

$$f(x_1, x_2, \dots, x_n)$$

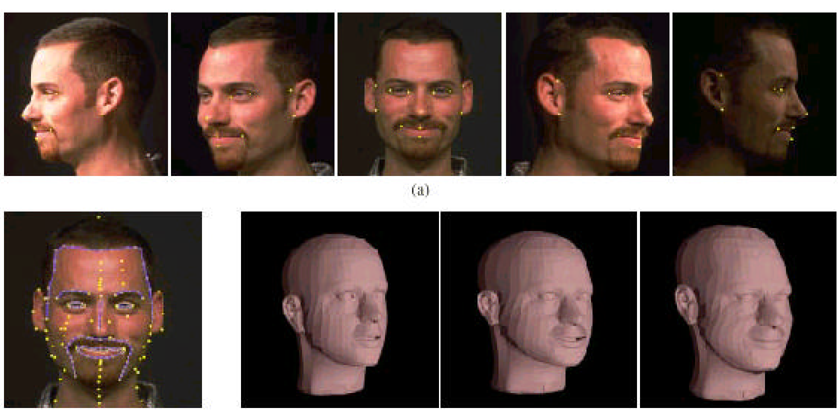
$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Optimization Methods

- Gradient Descent
- Conjugate Gradient
- Newton
- Quasi Newton
- Levenberg Marquadet

Real world Examples

Synthesizing Realistic Facial Expressions



Synthesizing Realistic Facial Expressions

$$x_i'^k = s_k \frac{\mathbf{r}_x^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_X^k}{1 + h^k \mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}}}$$

$$y_i'^k = s_k \frac{\mathbf{r}_y^{\mathbf{k}} \mathbf{p}_{\mathbf{i}} + T_Y^k}{1 + h^k \mathbf{r}_z^{\mathbf{k}} \mathbf{p}_{\mathbf{i}}} \quad w_i^k = (1 + h^k (\mathbf{r}_z^k . \mathbf{p}_i))^{-1}$$

$$w_i^k (x_i'^k + x_i'^k h^k (\mathbf{r}_{\mathbf{z}}^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}}) - s^k (\mathbf{r}_X^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}} + T_X^k)) = 0$$

$$w_i^k (y_i'^k + y_i'^k h^k (\mathbf{r}_{\mathbf{z}}^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}}) - s^k (\mathbf{r}_Y^{\mathbf{k}} . \mathbf{p}_{\mathbf{i}} + T_Y^k)) = 0$$

Synthesizing Realistic Facial Expressions

Computing Projective Transformation

$$x' = \frac{a_1 x + a_2 y + b_1}{c_1 x + c_2 y + 1}$$

Projective

$$y' = \frac{a_3 x + a_4 y + b_2}{c_1 x + c_2 y + 1}$$

$$E = \sum [f(x', y') - f(x, y)]^2 = \sum e^2$$

min

Computing Projective Transformation

Motion Vector:

$$\mathbf{m} = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad b_1 \quad b_2 \quad c_1 \quad c_2]^T$$

Video Mosaic

Video Mosaic

Video Mosaic

Preliminaries

Eigen Vectors and Eigen Values

The eigen vector, x, of a matrix *A* is a special vector, with the following property

$$Ax = lx$$ Where ë is called eigen value

To find eigen values of a matrix A first find the roots of:

$$\det(A - lI) = 0$$

Then solve the following linear system for each eigen value to find corresponding eigen vector

$$(A - lI)x = 0$$

Example

$$A = \begin{bmatrix} -1 & 2 & 0 \\ 0 & 3 & 4 \\ 0 & 0 & 7 \end{bmatrix}$$

Eigen Values

$$l_1 = 7,\ l_2 = 3,\ l_3 = -1$$

$$\mathbf{x_1} = \begin{bmatrix} 1 \\ 4 \\ 4 \end{bmatrix},\ \mathbf{x_2} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix},\ \mathbf{x_3} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$ Eigen Vectors

Eigen Values

$$\det(A - lI) = 0$$

$$\det\left(\begin{bmatrix} -1 & 2 & 0 \\ 0 & 3 & 4 \\ 0 & 0 & 7 \end{bmatrix} - l\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) = 0$$

$$\det\left(\begin{bmatrix} -1-l & 2 & 0 \\ 0 & 3-l & 4 \\ 0 & 0 & 7-l \end{bmatrix}\right) = 0$$

$$(-1-l)((3-l)(7-l)-0) = 0$$

$$(-1-l)(3-l)(7-l) = 0$$

$$l = -1, \quad l = 3, \quad l = 7$$

Eigen Vectors

$l = -1$ $\qquad (A - lI)x = 0$

$$\left(\begin{bmatrix} -1 & 2 & 0 \\ 0 & 3 & 4 \\ 0 & 0 & 7 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 2 & 0 \\ 0 & 4 & 4 \\ 0 & 0 & 8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$0 + 2x_2 + 0 = 0$$

$$0 + 4x_2 + 4x_3 = 0$$

$$0 + 0 + 8x_3 = 0$$

$$x_1 = 1, \quad x_2 = 0, \quad x_3 = 0$$

$$\mathbf{x_1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$trace(A) = \sum_{i=1}^{n} A_{ii}$$

$$trace(A) = \sum_{i=1}^{n} l_i \text{ where } l_i \text{ are eigen valu es}$$

$$\det(A) = \prod_{i=1}^{n} l_i$$

$\det A = 0$ if and only if A is singular

$$\det AB = (\det A)(\det B)$$

$$\det A^{-1} = \frac{1}{\det A}$$

$QQ^T = Q^T Q = I$, Q is orthogonal

$$Q^{-1} = Q^T$$

$$\det Q = \det Q^T = \pm 1$$

A symetric n$\times$n matrix is positive definite if
$X^T AX > 0$.

$A = LU$, LU decomposition, L is a Lower triangular, and U is a upper triangular

$A = CU$, QR decompoistion, C is orthonorma l, U is upper trinagular matrix

Norms

$$\| X \|_1 = \sum_{i=1}^{n} | x_i |, \text{ vector norm}$$

$$\| X \|_2 = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}} = (X^T X)^{\frac{1}{2}}, \text{ vector norm}$$

$$\| \mathbf{A} \| = \max_{\|x\|=1} \| \mathbf{A}X \|, \text{ matrix norm}$$

$$\| \mathbf{A} \|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} | a_{ij} |$$

Norms

Condition number

$$k(A) = \| A \| \| A^{-1} \|$$

The matrix A is well-conidtioned if K(A) is close to one and is not well-conditioned, when K(A) is significantly greater than one.

Singular Value Decomposition (SVD)

Theorem: Any m by n matrix A, for which $m \geq n$,can be written as

$$A = O_1 \Sigma O_2$$

mxn mxn nxn nxn

Σ is diagonal

O_1 , O_2 are orthogonal

$O_1^T O_1 = O_2^T \, O_2 = I$