Numerical Optimization

Class Administrivia

Motivated Examples

Lectures

- This class is offered in English
- Most of lectures will be given using blackboard. You need to take note by yourself.
  - Except today's lecture
- No textbook, but will list some reading materials.
  - List of reference books is posted online.
  - List of websites is also posted.
- TA:
- Office hour: Wednesday 10:00-12:00

Grading

- 40% homework (every two weeks)
- 20% midterm (tentative date: 4/19)
- 30% final exam/project (tentative date: 6/21)
  - If your midterm does not pass a threshold, you need to take the final exam.
  - If your midterm passes a threshold, you can choose to do project or to take the final exam.
  - Project will be research orientated, including paper reading and implementation.
- 10% attendance

Topics will be covered

1. Introduction
2. One-dimensional optimization
3. Unconstrained optimization
4. Least square problem
5. Constrained optimization
6. Global optimization or/and discrete optimization

- Details is listed on the class website:
  - http://www.cs.nthu.edu.tw/~cherung/teaching/2010cs5321/index.html

Your goals of this class

- Learning some basic optimization concepts and methods, as well as their limitations.
  - This will help a lot for you to read papers and to do researches.
- Knowing the numerical techniques behind the numerical optimization methods.
  - This can help you to verify the equations and to derive new methods.

Motivated examples

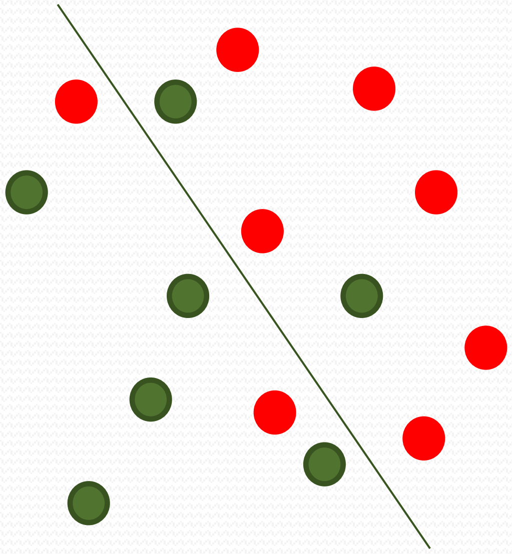
- Ex1: Classification in pattern recognition (SVM)
- Ex2: Floorplanning in VLSI design
- Ex3: Network component analysis in system biology
- Ex4: Localization problem in wireless network
- Ex5: Topology mapping problem in high-performance computing
- Ex6: Job interview problem for one variable optimization

Ex1: Classification

- Given two sets A={a1,a2,…,ak} and B={b1,b2,…,bm} in $\mathbb{R}^n$. Find a line to separate them such that the "error" is minimized.
- The line equation is $\phi(a) = \langle v, a \rangle - \gamma$
- The errors are

$$e_-(a_i, v, \gamma) = \begin{cases} 0 & \text{if } \langle v, a_i \rangle - \gamma \geq 0 \\ \gamma - \langle v, a_i \rangle & \text{if } \langle v, a_i \rangle - \gamma < 0 \end{cases}$$

$$e_+(b_i, v, \gamma) = \begin{cases} 0 & \text{if } \langle v, b_i \rangle - \gamma \leq 0 \\ \langle v, b_i \rangle - \gamma & \text{if } \langle v, b_i \rangle - \gamma > 0 \end{cases}$$



Andrzej P. Ruszczyński, "Nonlinear optimization", chap 1, 2006

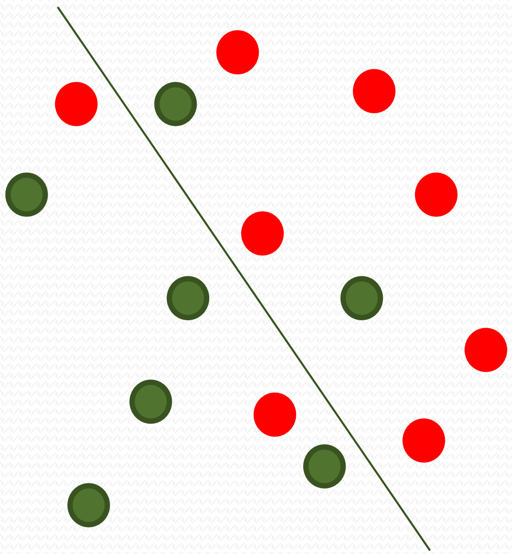
Classification - constrains

- Constrains
  - If v=0 and γ=0, errors are zero
  - Let ||v||=1
- Problem:

$$\min_{v,\gamma} \sum_{i=1}^{k} e_{-}(a_i, v, \gamma) + \sum_{j=1}^{m} e_{+}(b_j, v, \gamma)$$

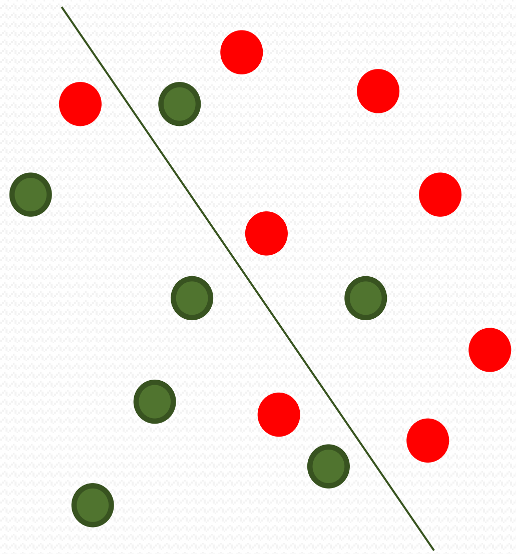
$$\text{s.t.} \quad \|v\| = 1$$

Piecewise linear function



Classification – Slack variables

- Let $e_{-}(a_i, v, \gamma) = s_i, e_{+}(b_j, v, \gamma) = z_j$
  - Slack variables
- Problem:



$$\min_{v,\gamma,s,z} \sum_{i=1}^{k} s_i + \sum_{j=1}^{m} z_j$$

$$\text{s.t.} \quad \|v\| = 1$$

$$\langle v, a_i \rangle - \gamma + s_i \geq 0, \qquad i = 1,\ldots,k$$

$$\langle v, b_j \rangle - \gamma + z_j \leq 0, \qquad j = 1,\ldots,m$$

Classification (SVM)

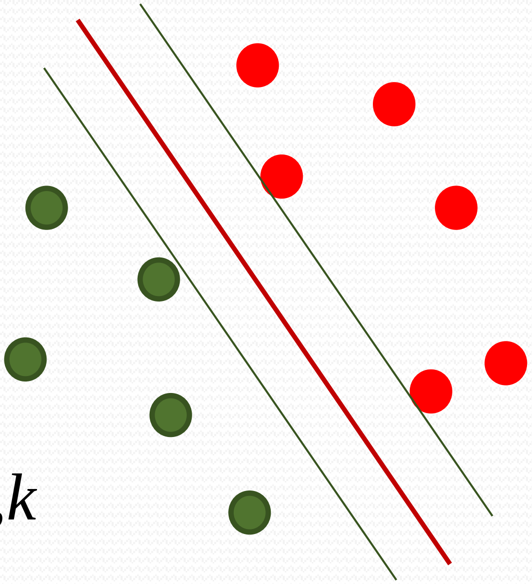
- Adding buffers
- Problem:

$$\min_{v,\gamma,s,z} \sum_{i=1}^{k} s_i + \sum_{j=1}^{m} z_j$$

$$\text{s.t.} \quad \|v\| = 1$$

$$\langle v, a_i \rangle - \gamma + s_i \geq 1, \qquad i = 1,\ldots,k$$

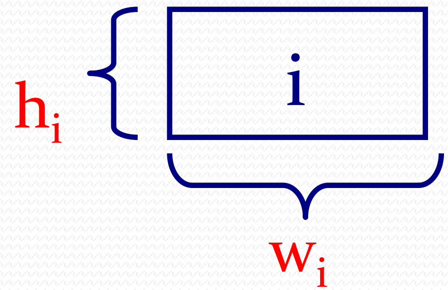
$$\langle v, b_j \rangle - \gamma + z_j \leq -1, \qquad j = 1,\ldots,m$$



- This is called supporting vector machine (SVM)

Ex2: Floorplanning

- An early stage of physical design that determines module positions, shapes, and orientations
- Module i:
  - Width and height: wi and hi
  - Area: wi·hi
  - Aspect ratio: hi / wi

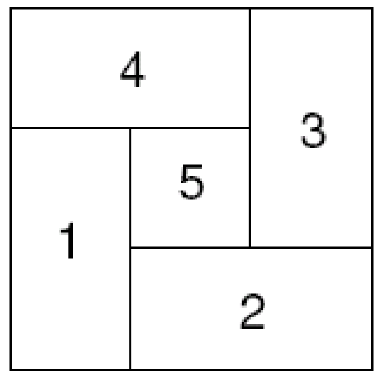


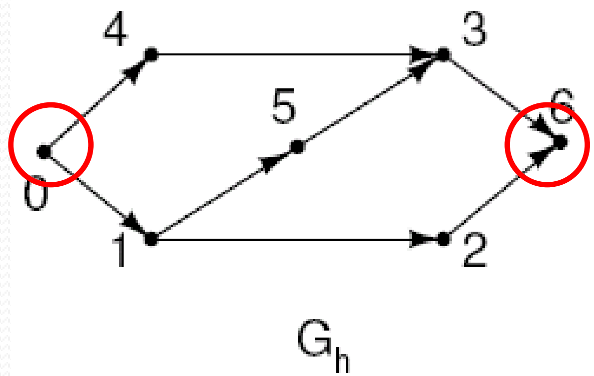
- Goal: Find width wi, height Ai/wi, and lower left corner coordinates xi and yi, i = 1, …, n, such that the area is minimized

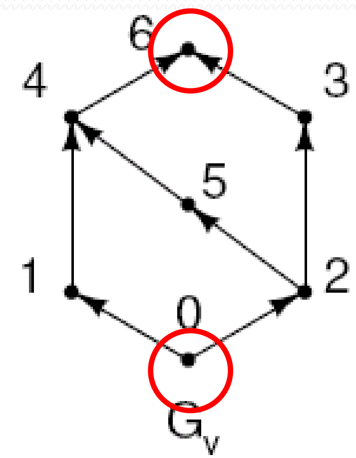
Chuan Lin, Hai Zhou, Chris Chu, 2006
"A revisit to floorplan optimization by Lagrangian relaxation"

Constraint graph

- Horizontal constraint graph Gh of n modules
  - Edge (i,j)   Gh if module j is to the right of module i
- Vertical constraint graph Gv of n modules
  - Edge (i,j)   Gv if module j is above module i







Problem formulation (P_{area})

PROBLEM 1 (MINIMUM AREA FLOORPLANNING).

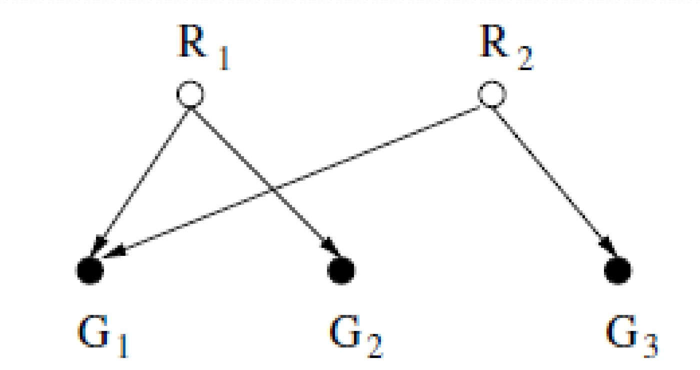
$$
\begin{aligned}
P_{area}: \quad & \text{Minimize} \quad (x_{n+1} - x_0)(y_{n+1} - y_0) \\
& \text{subject to} \quad x_j \geq x_i + w_i \quad , \forall (i,j) \in G_h \\
& \qquad y_j \geq y_i + A_i / w_i \quad , \forall (i,j) \in G_v \\
& \qquad L_i \leq w_i \leq U_i \quad , \forall 1 \leq i \leq n
\end{aligned}
$$

$$
L_i = \sqrt{A_i / r_i^{max}} \quad \text{and} \quad U_i = \sqrt{A_i / r_i^{min}}
$$

Ex3:Network component analysis

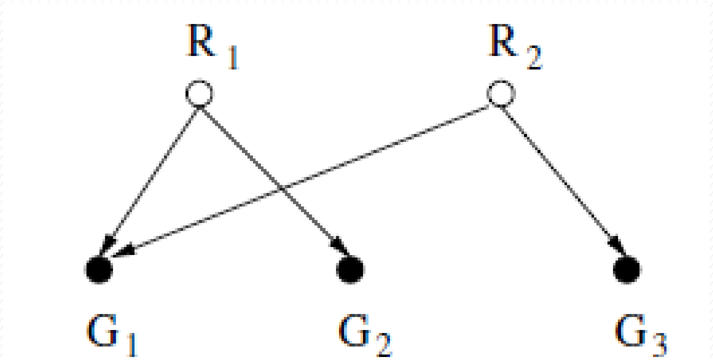
- Network: a bipartite graph (R,G,E)
  - **R**: concentration of active form of regulatory proteins.
  - **G**: gene expressions that are observed through a series of experiments.
  - **E**: the arrows connecting nodes indicate which regulatory proteins have influence on which genes.

James C. Liao, Riccardo Boscolo, Young-Lyeol Yang, Linh My Tran, Chiara Sabatti§, and Vwani P. Roychowdhury, 2003, "Network component analysis: Reconstruction of regulatory signals in biological systems"



Problem formulation

- E = AP, where E,A,P are matrices
  - E(N×M):each row represents a gene; each column represents an experiment
  - A(N×L):each column represents the action of a regulatory protein on all the genes
  - P(L×M) matrix with each row representing the profile of each regulatory protein across experiments.
- E is known; A and P are what we want to compute



Network constrains

- Elements of matrix A need to satisfy the zero patterns that specified by the network
- Problem formulation

$$\min_{A,P} \|E - AP\|$$

$$\text{s.t.} \quad A(i, j) = 0 \quad \text{if } (i,j) \notin \text{edge E}$$

Ex4: Localization problem

For n nodes $X=\{x1,x2,\ldots xn\}$ and m anchor nodes $A=\{a1,a2,\ldots,am\}$, given partial relative distance information between nodes, find the positions of $\{x1,x2,\ldots xn\}$.

Pratik Biswas and Yinyu Ye, 2004 "Semidefinite programming for ad hoc wireless sensor network localization"

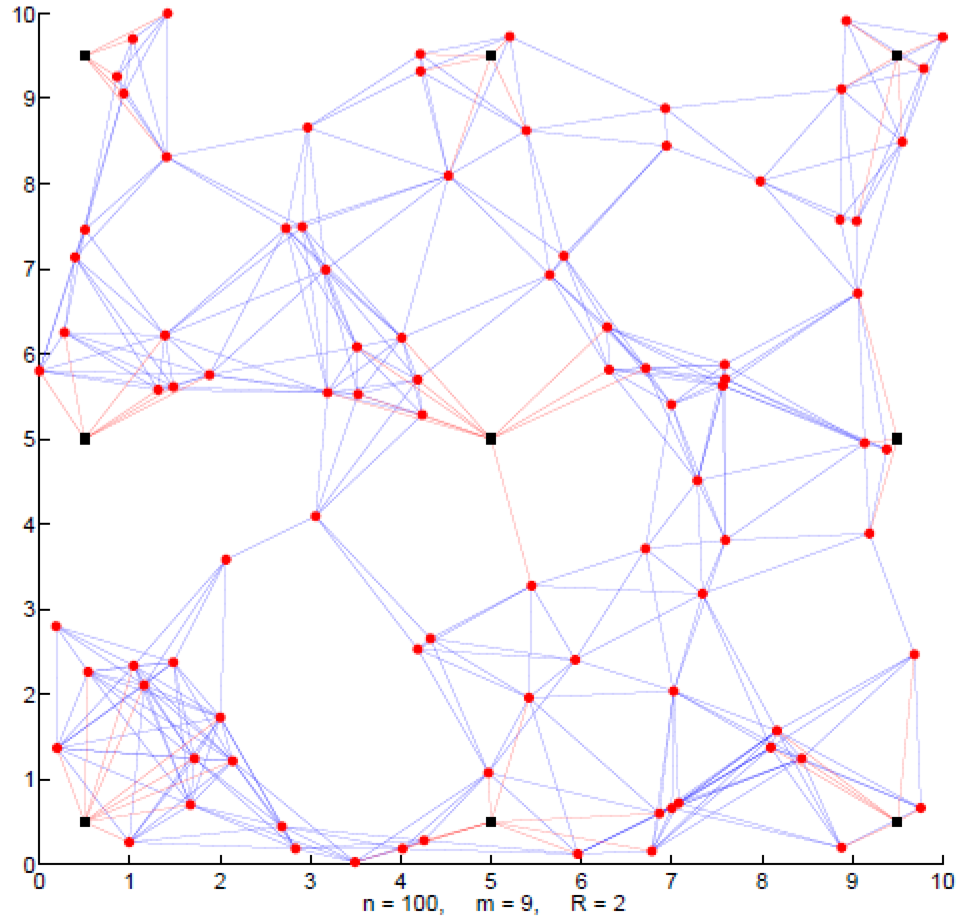


Figure 1.1: Graph of partial EDM with sensors ∘ and anchors ■

Euclidean distance matrix (EDM)

$$D = [d_{ij}] = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} = \begin{bmatrix} 0 & d_{12} & d_{13} \\ d_{12} & 0 & d_{23} \\ d_{13} & d_{23} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 5 \\ 1 & 0 & 4 \\ 5 & 4 & 0 \end{bmatrix}$$

$$d_{ij} = \|x_i - x_j\|_2^2 \triangleq \langle x_i - x_j \, , \, x_i - x_j \rangle$$

- If we know the full EDM, we can find the relative coordinates (up to rotation).
- If we know the full EDM and some anchor nodes, we can identify the absolute coordinates of X.
- EDM is usually noisy. Only maximum likelihood positions is available.

Problem formulation

$$\min_{x_1,\cdots x_n} \sum_{(i,j)\in E} \left| d(x_i, x_j) - d_{ij} \right| + \sum_{(i,j)\in E} \left| d(a_i, x_j) - d_{ij} \right|$$

- Not very robust. Need more constrains
- Distance lower bound

$$\|x_i - x_j\|^2 \geq (\underline{r}_{ij})^2, \; \|a_k - x_j\|^2 \geq (\underline{r}_{kj})^2$$

- Distance upper bound

$$\|x_i - x_j\|^2 \leq (\bar{r}_{ij})^2, \; \|a_k - x_j\|^2 \leq (\bar{r}_{kj})^2$$

Problem formulation

Define: $(u)_- = \max\{0, -u\}$ and $(u)_+ = \max\{0, u\}$

$$\begin{aligned}
\min \quad & \textstyle\sum_{i,j\in N_e,\ i<j} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{k,j\in N_e} (\alpha_{kj}^+ + \alpha_{kj}^-) \\
& + \textstyle\sum_{i,j\in N_l,\ i<j} \beta_{ij}^- + \sum_{k,j\in N_l} \beta_{kj}^- \\
& + \textstyle\sum_{i,j\in N_u,\ i<j} \beta_{ij}^+ + \sum_{k,j\in N_u} \beta_{kj}^+ \\
\text{s.t.} \quad & \|x_i - x_j\|^2 - (\hat{d}_{ij})^2 = \alpha_{ij}^+ - \alpha_{ij}^-,\ \forall\, i,j \in N_e,\ i<j, \\
& \|a_k - x_j\|^2 - (\hat{d}_{kj})^2 = \alpha_{kj}^+ - \alpha_{kj}^-,\ \forall\, k,j \in N_e, \\
& \|x_i - x_j\|^2 - (\underline{r}_{ij})^2 \geq -\beta_{ij}^-,\ \forall\, i,j \in N_l,\ i<j, \\
& \|a_k - x_j\|^2 - (\underline{r}_{kj})^2 \geq -\beta_{kj}^-,\ \forall\, k,j \in N_l, \\
& \|x_i - x_j\|^2 - (\underline{r}_{ij})^2 \leq \beta_{ij}^+,\ \forall\, i,j \in N_u,\ i<j, \\
& \|a_k - x_j\|^2 - (\underline{r}_{kj})^2 \leq \beta_{kj}^+,\ \forall\, k,j \in N_u \\
& \alpha_{ij}^+,\ \alpha_{ij}^-,\ \alpha_{kj}^+,\ \alpha_{kj}^-,\ \beta_{ij}^-,\ \beta_{kj}^-, \beta_{ij}^+,\ \beta_{kj}^+ \geq 0.
\end{aligned}$$

Ex5:Topology mapping problem

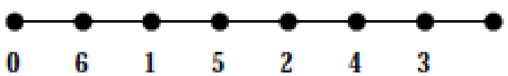
- There are n tasks and n processors. The traffic between tasks is recorded in a matrix A; The communication cost (distance, hops) between processors is recorded in a matrix D.
- Find an assignment P of tasks to processors such that the “communication cost” is minimized.
  - The overall cost

$$C_P = \sum_{i,j=1}^{n} A_{P_i,P_j} D_{i,j},$$

  - The maximum cost

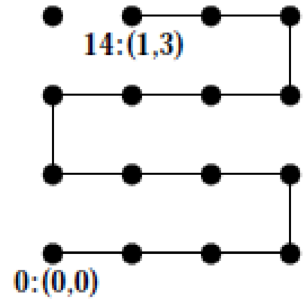
$$C_P = \max_{i,j=1,..n} A_{P_i,P_j} D_{i,j}$$

Examples

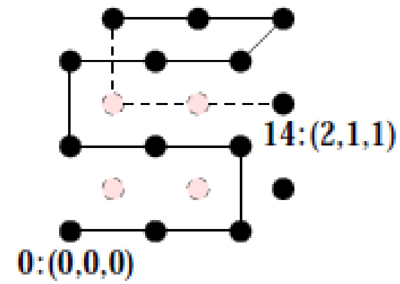


(a) Ring-wrapping

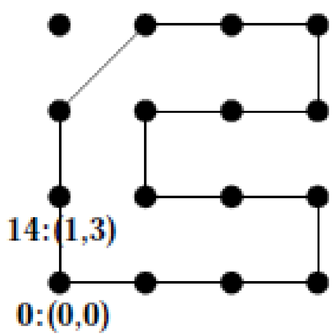
(b) Ring-scattering



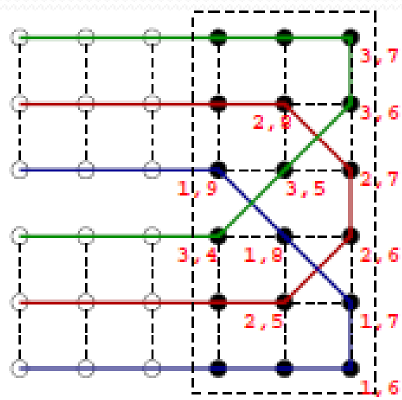
(c) Embed a line to a 2D grid



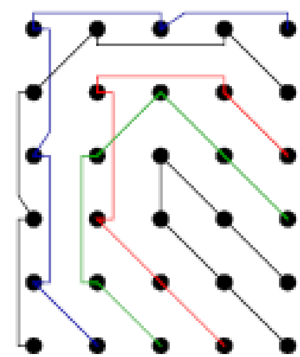
(d) Embed a line to a 3D grid



(e) Embed a ring to a 2D grid



(f) Fold a 12x3 grid to a 6x6 2D



(g) 90 degree turn

Ex6:Job interview problem

- There are *N* interviewees for one position.
- They are interviewed in a random order.
- The result is either accepted or rejected.
- The rejected one cannot be recalled.
- If one is accepted, he/she cannot be replaced.
- The decision to accept or reject is based on the relative ranks of the interviewee so far.
- The object is to select "the best" applicant.

How to make decision?

- The objective function: 1 if the best one is selected, 0 otherwise.
- If accepting too early, you might miss better one later; if too late, the better one could be rejected earlier.
- Basic strategy: passing on the first *k* candidates and then selecting the next "best so far".
  - How to decide the "best" k?

Fixed best candidate

- Suppose the best candidate is at position j
  - The probability is 1/N
- For j to be selected, the best of (j-1) must be in the [1..k].
  - The probability is k/(j-1), given k<j.
- So, the probability for j to be selected is

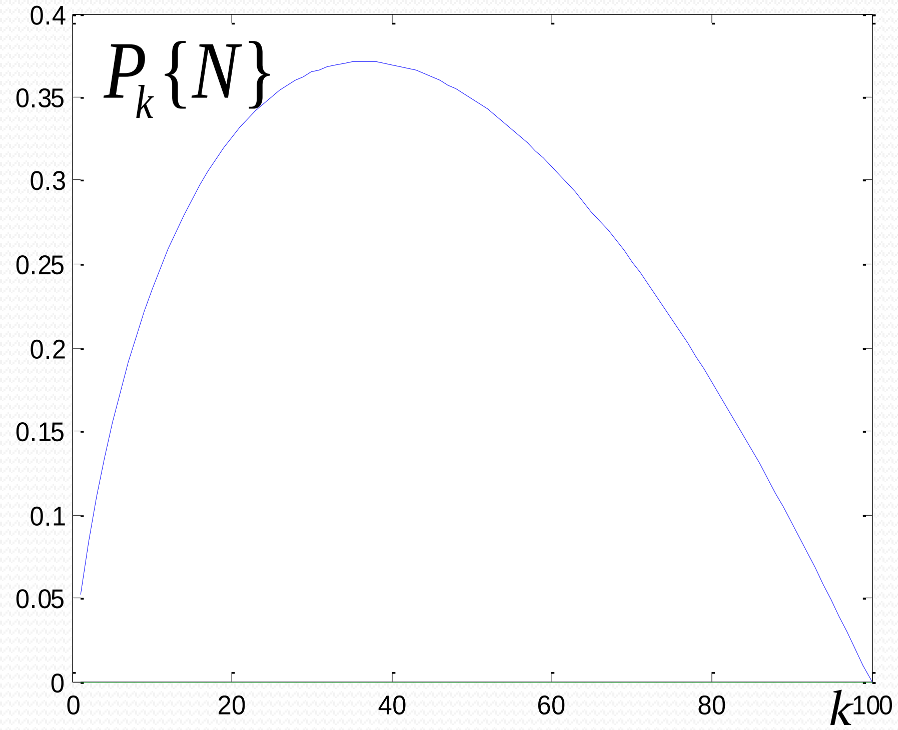
$$P_k\{N, j\} = \frac{1}{N}\left(\frac{k}{j-1}\right)$$

Fixed k

- If we fixed k, and sum up all possible j, j>k

$$P_k\{N\} = \frac{k}{N}\sum_{j=k+1}^{N}\left(\frac{1}{j-1}\right)$$

$$= \frac{k}{N}\left[\sum_{j=1}^{N-1} 1/j - \sum_{j=1}^{k-1} 1/j\right]$$



- Find k to maximize P{N}

For large enough k and N

$$P_k\{N\} = \frac{k}{N}\sum_{j=k+1}^{N}\left(\frac{1}{j-1}\right) = \frac{k}{N}\left[\sum_{j=1}^{N-1} 1/j - \sum_{j=1}^{k-1} 1/j\right]$$

$$\rightarrow \frac{k}{N}\left(\ln N - \ln(k-1)\right) \approx \frac{k}{N}\ln\frac{N}{k}$$

- To find the best strategy, we need to decide what k is?
- P(k) is called "unimodal", which means it has only one maximum x*; for x<x*, f(x) is monotonically increasing, and for x>x*, f(x) is monotonically decreasing.