Analytics and Visualization of Big Data

Fadel M. Megahed

Lecture 10: Similarity of Sets (LSH)

AUBURN UNIVERSITY
SAMUEL GINN
COLLEGE OF ENGINEERING

Department of Industrial and Systems Engineering

Spring 13

Outline for Topics Covered in Chapter 03 (3.1 → 3.4)

Applications of Near-Neighbor Search

- Jaccard Similarity of Sets
- Similarity of Documents
- Collaborative Filtering

Locality-Sensitive Hashing

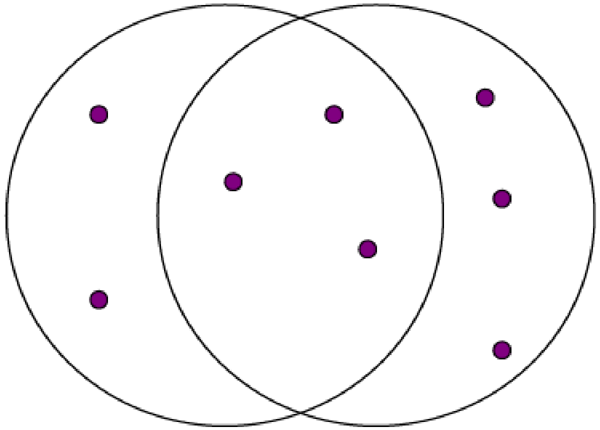
- Shingling
- Minhashing
- LSH for Minhash Signatures
- Combining the Techniques

How do we Define Similarity?

- Typically, we want to have items that have common features → we use this to say there are similar ☺

- The **Jaccard Similarity** of two sets is:
  - $Sim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$

- The **Jaccard Distance** between sets is 1 minus their Jaccard similarity: $d(C_1, C_2) = 1 - |C_1 \cap C_2| / |C_1 \cup C_2|$



3 in intersection
8 in union
Jaccard similarity= 3/8
Jaccard distance = 5/8

How does this Concept Relate to “Big Data” Analytics?

- **Goal:** Finding textually similar documents in a collection of news articles or web pages
  - Character-level similarity vs. similar meaning?
- Two levels of **similarity**:
  - **Exactness:** Easy, character-by-character comparison
  - **Near duplicates:** More involved; topic of today’s class
- Typical applications in **Big Data Analytics**:
  - Plagiarism detection
  - Articles from the same source
  - Collaborative filtering

Outline for Topics Covered in Chapter 03 (3.1 → 3.4)

Applications of Near-Neighbor Search

- Jaccard Similarity of Sets
- Similarity of Documents
- Collaborative Filtering

Locality-Sensitive Hashing

- Shingling
- Minhashing
- LSH for Minhash Signatures
- Combining the Techniques

Problem Description for Finding Similar Documents

Problem Statement:

- Given a large number (N in the millions or billions) of text documents, find pairs that are "near duplicates"

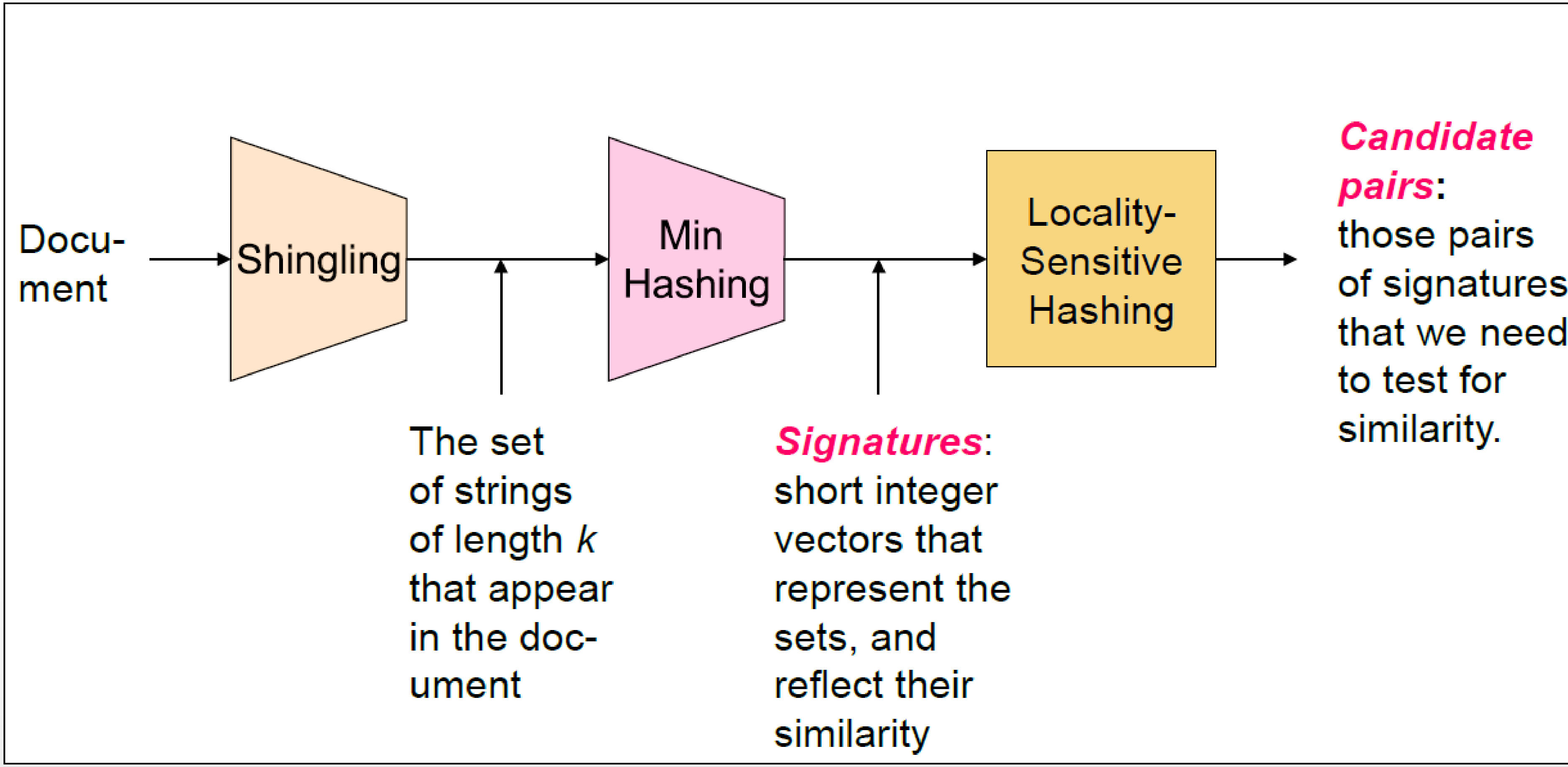
Issues:

- Many small pieces of one doc can appear out of order in another
- Too many docs to compare all pairs
- Docs are so large or so many that they cannot fit in main memory

The Step-by-Step Guideline for Finding Similar Documents

- **Shingling**: Convert documents, emails, etc., to sets of short strings that appear within it
- **Minhashing**: Convert large sets to short signatures, while preserving similarity
- **Locality-sensitive hashing**: Focus on pairs of signatures likely to be from similar documents

The Step-by-Step Guideline for Finding Similar Documents



Source: Slide Adapted Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

Shingling – What is Shingling?

- A ***k*-shingle** (or ***k*-gram**) for a document is a sequence of *k* tokens that appears within the document
  - Tokens can be characters, words or something else, depending on application
  - Assume tokens = characters for reading the book examples

- **Example:** k=2; D_1= abcab
  - Set of 2-shingles: S(D1)={ab, bc, ca}
  - **Option:** Shingles as a bag

- **Represent a doc by the set of hash values of its *k*-shingles**

How do we pick *k*?

Similarity Metric for Shingles

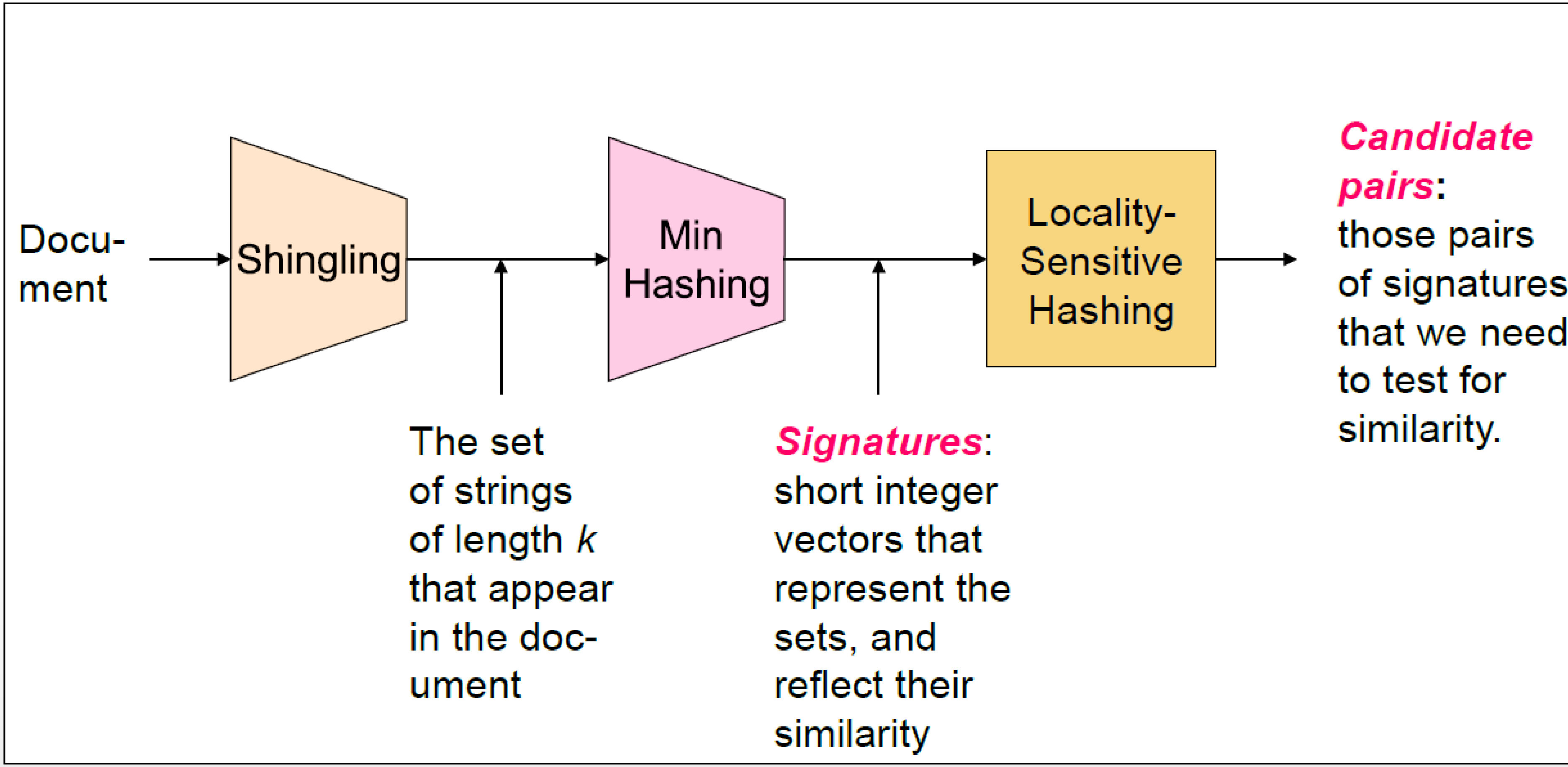
- Document D_1 = set of k-shingles $C_1=S(D_1)$
- Equivalently, each document is a 0/1 vector in the space of *k*-shingles
  - Each unique shingle is a dimension
  - Vectors are very sparse
- A natural similarity measure is the **Jaccard similarity:**
  - $Sim(D1, D2) = |C1 \cap C2| / |C1 \cup C2|$
- **Assumption:** Documents that have lots of shingles in common have similar text, even if the text appears in different order

Motivation for Minhash/LSH

- **Suppose we need to find near-duplicate documents among N=1 million documents**

- Naïvely, we'd have to compute pairwise Jaccard similarites for every pair of docs

The Step-by-Step Guideline for Finding Similar Documents



Source: Slide Adapted Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

A Side Note: The Characteristic Matrix

- Properties of the Matrix:
  - Rows = elements of the universal set
  - Columns = sets
- 1 if and only if the token is a member of the set
- Column similarity is the Jaccard similarity of the sets of their rows with 1
- Typical matrix is sparse

A Side Note: The Characteristic Matrix – Exercise

- Suppose that we have the following sets:
  - Universal set {a, b, c, d, e}.
  - S1 = {a, d}, S2 = {c}, S3 = {b, d, e}, and S4 = {a, c, d}.

- What is the *characteristic matrix* for this problem? ☺

Outline: Finding Similar Columns

- **So far:**
  - Documents → Sets of shingles
  - Represent sets as boolean vectors in a matrix
- **Next Goal: Find similar columns**
- **Approach:**

1. **Signatures of columns:** small summaries of columns
2. **Examine pairs of signatures** to find similar columns –**Essential property:** Similarities of signatures & columns are related
3. **Optional:** check that columns with similar sigs. are really similar

- **Warnings:**
  - Comparing all pairs may take too long: **job for LSH/Minhash**

Minhashing

- Imagine the rows of the boolean matrix permuted under **random permutation** π

- Define a "hash" function $h_\pi(C)$ = the number of the first (in the permuted order π) row in which column C has value 1:

$$h_\pi (C) = min\ \pi (C)$$

- Use several (e.g., 100) independent hash functions to create a signature of a column

Minhashing – An Example

Suppose we pick the order of rows *beadc*. This permutation defines a minhash fn h that maps sets to rows. Compute the minhash value for all S according to h (i.e. for S_1, S_2, S_3, and S_4).

$Element$	S_1	S_2	S_3	S_4
a	1	0	0	1
b	0	0	1	0
c	0	1	0	1
d	1	0	1	1
e	0	0	1	0

Note that: It is typical to replace the letters naming the rows by integers 0, 1, 2, etc.

Minhashing and Jaccard Similarity – A Surprising Property

- There is a remarkable connection between minhashing and Jaccard similarity of the sets that are minhashed.
  - **The probability that the minhash function for a random permutation of rows produces the same value for two sets equals the Jaccard similarity of those sets.**

- To see why, check p. 80 in the book ☺

Minhash Signatures – An Example

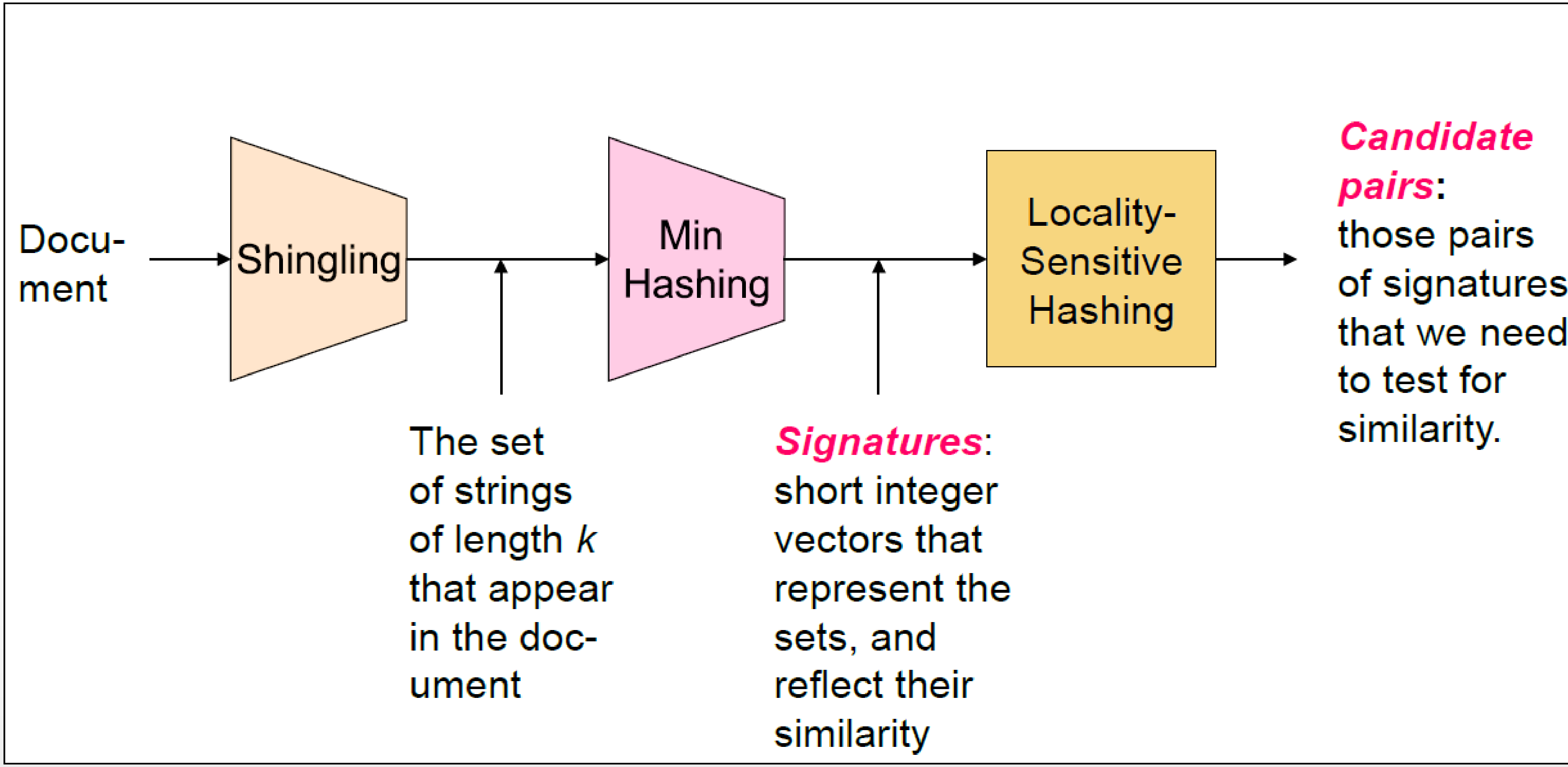
Using the new indices, let us return the signature matrix using these two hash functions ☺

Row	S_1	S_2	S_3	S_4	$x+1 \mod 5$	$3x+1 \mod 5$
0	1	0	0	1	1	1
1	0	0	1	0	2	4
2	0	1	0	1	3	2
3	1	0	1	1	4	0
4	0	0	1	0	0	3

Similarity for Signatures

- We know: $\mathbf{Pr}[h_\pi(C1) = h_\pi(C2)] = sim(C1, C2)$
- Now generalize to multiple hash functions
- **The similarity of two signatures is the fraction of the hash functions in which they agree**
- Note: Because of the minhash property, the similarity of columns is the same as the expected similarity of their signatures

The Step-by-Step Guideline for Finding Similar Documents



Source: Slide Adapted Jure Leskovic, Stanford CS246, Lecture Notes, see http://cs246.stanford.edu

LSH – General Idea

- **Goal:** Find documents with Jaccard similarity at least *s* (for some similarity threshold, e.g., *s*=0.8)

- **LSH –** General idea: Use a function f(x,y) that tells whether *x* and *y* is a **candidate pair***:* a pair of elements whose similarity must be evaluated

- **For minhash matrices:**
  - Hash columns of signature matrix *M* to many buckets
  - Each pair of documents that hashes into the same bucket is a candidate pair

LSH – General Idea

- Pick a similarity threshold *s*, a fraction < 1
- Columns *x* and *y* of M are a **candidate pair** if their signatures agree on at least fraction *s* of their rows:

  $M(i, x) = M(i, y)$ for at least frac. *s* values of *i*
- Note: We expect documents *x* and y to have the same similarity as their signatures (see previous slides)

Analytics and Visualization of Big Data

Fadel M. Megahed

Lecture 10: Similarity of Sets (LSH)

AUBURN UNIVERSITY
SAMUEL GINN
COLLEGE OF ENGINEERING

Department of Industrial and Systems Engineering

Spring 13