Ranking using Multivariate Prognostic Data

Fang Cao and Pinghan Wu

Advisors: Kobi Abayomi & Nagi Gebraeel

Georgia Institute of Technology
The H. Milton Stewart School of Industrial and Systems Engineering

Outline

- Introduction
  - Problem Setting
  - Literature Review
- Top-k identification
  - Using Database Ranking Algorithm
  - Unique challenges
  - Statistical approach based on CS ranking algorithm
  - Base-case simulation result
- Future work

Problem Setting

- We consider a fleet of identical units (say trucks, airplanes, locomotives, ships, etc.) where each unit consists of several critical components (hydraulics, avionics, engine, etc.)
  - A critical component is a component whose failure constitutes a catastrophic failure of the entire system.
  - The degradation of each critical component can be monitored using sensor technology
  - Sensor signals can be used to predict the remaining useful lifetimes of critical components, a process referred to as prognostics.

Significance

- **Objective**
  - **To identify the Best/Worst subset (Top-k) of units through a ranking procedure, which relies on Prognostic information that is synthesized from real-time sensor signals.**

Introduction

- **Ranking** of data is an important topic in the field of database queries in Computer Science and generally used to identify top search results (e.g. Google, Yahoo etc.)
- Due to the large size of computer databases, smart algorithms are needed to avoid mining the entire database to obtain the TOP Matches, i.e. Top-k matches/results.
- The Threshold Algorithm (TA) is an example of such Ranking Algorithms.

Literature review

Top-k related Ranking Algorithms

R. Fagin(1999)

S. Nepal, M.V. Ramakrishna(1999)

K. C. Chang and S. Hwang(2002)

R. Fagin, A. Lotem, and M. Naor(2003)

P. P. Bonissone and A. Varma(2005)

Degradation and Prognostic models

Lu and Meeker (1993)

Whitmore (1995)

Yang and Yang (1998)

Lu *et al.* (2001)

N. Gebraeel *et al.* (2005)

N. Gebraeel(2010)

Threshold Algorithm – A CS example

10 houses (R1-R10), 4 attributes, scores sorted for each attribute

Distance	***Price***	***Size***	***Community***
R8(0.95)	R10(1.00)	R3(0.95)	R5(1.00)
R2(0.90)	R3(0.95)	R10(0.80)	R7(0.95)
R5(0.85)	R7(0.85)	R4(0.70)	R8(0.90)
R3(0.80)	R8(0.80)	R8(0.65)	R2(0.85)
R7(0.75)	R5(0.75)	R7(0.60)	R4(0.80)
R9(0.70)	R2(0.65)	R2(0.55)	R3(0.70)
R4(0.65)	R6(0.60)	R9(0.50)	R1(0.65)
R1(0.60)	R1(0.50)	R5(0.45)	R9(0.55)
R10(0.55)	R4(0.40)	R6(0.40)	R6(0.45)
R6(0.50)	R9(0.30)	R1(0.30)	R10(0.30)

Goal:

Find the Top-3 houses that have the best/ closest match to our search criteria

Threshold Algorithm – A CS example

Scoring Function → **Average**

Distance	*Price*	*Size*	*Community*
R8(0.95)	R10(1.00)	R3(0.95)	R5(1.00)
R2(0.90)	R3(0.95)	R10(0.80)	R7(0.95)
R5(0.85)	R7(0.85)	R4(0.70)	R8(0.90)
R3(0.80)	R8(0.80)	R8(0.65)	R2(0.85)
R7(0.75)			
R9(0.70)			
R4			
R1			
R1			
R6			

Threshold Value
3.90/4
3.60/4
3.30/4
3.10/4

Top 3 Houses

R3(3.40/4)
R8(3.30/4)
R7(3.15/4)
R5(3.05/4)
R2(2.95/4)
R10(2.65/4)
R4(2.55/4)

You ... ar... wil... hig... ls, the ... s an Upper ... bsequent ... res

Are you sure that these are the top-3?

YES

Unique Challenges in our Problem Setting

Computer Science setting	Prognostic setting
Items are ranked based on score of each individual **attribute**	**Units** must be ranked based on score of each individual **component**
Scores are **deterministic** (values)	Scores are mean/median of **RLD**s
Scores are **fixed** for a relatively long period of time	Scores are updated based on **real-time** signals from sensors

Unique Challenges in our Problem Setting

Computer Science setting	Prognostic setting
Computational complexity related to **size** of database	Computational complexity is related to **updating frequency**, i.e. calculating RLD of each component, and partially size of database
Distribution of scores are **not** considered	**Distribution** of scores can be utilized to **reduce** search steps

Computational Challenges

1. Each time a sensor signal is observed from a given component, it is used to update its RLD.
2. Thus, the score of each critical component in every unit may change with each sensor observation.
3. With each update, the overall score of the units will change, hence the Top-k results will also change.

- **As a result, for real-time applications, we need fast ranking algorithms that can identify the Top-k units quickly.**

Statistical Approach in Top-k Ranking

- We are interested in making probabilistic statements about the constituents of the Top-k at any point before the stopping criterion of the TA algorithm is met.
  - The stopping criterion is met at the row where the **threshold value** is no greater than the **minimum overall score** of current top-k list
- We would like to the investigate how to leverage correlation among components to reduce search steps needed to identify Top-k

Top-k - statistical approach

- *n* units, *m* components
- X_{ij} – score for the j^{th} component of i^{th} unit
- $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ iid and $\mathbf{X}_i$ has cdf $F_{\mathbf{X}_i}$
- $X_{(r)j}$ – r^{th} order statistic for component (column) j
- $\mathbf{X}_{(r)} = [X_{(r)1}, X_{(r)2}, \ldots, X_{(r)m}]$ is the r^{th} row
- g(.) – scoring function

Unit	X_1	X_2	…	X_m	Threshold
(n)					$g(\mathbf{X}_{(n)})$
⋮					⋮
(r)			⋮		$g(\mathbf{X}_{(r)})$
⋮					⋮
(1)					$g(\mathbf{X}_{(1)})$

Distribution for $X_{(r)}$

- Two-component case

$$F_{X_{(r)}}(x_1, x_2) = P(X_{(r)1} \le x_1, X_{(r)2} \le x_2)$$

$$= P(\text{At least } r \text{ of } X_{i1} \le x_1, \text{ at least } r \text{ of } X_{i2} \le x_2)$$

$$= \sum_{l_1, l_2 \ge r} P(\text{Exact } l_1 \text{ of } X_{i1} \le x_1, \text{Exact } l_2 \text{ of } X_{i2} \le x_2)$$

$$= \sum_{\substack{0 \le t_1, t_2, t_3, t_4 \le n \\ t_1 + t_2 + t_3 + t_4 = n \\ r \le l_1, l_2 \le n}} \binom{n}{t_1\ t_2\ t_3\ t_4} p_1^{t_1} p_2^{t_2} p_3^{t_3} p_4^{t_4}$$

X_{i1}	X_{i2}: $\le x_2$	X_{i2}: $> x_2$	Total
$\le x_1$	t_1	t_2	l_1
$> x_1$	t_3	t_4	$n- l_1$
Total	l_2	$n-l_2$	n

Where $p_1 = P(X_{i1} \le x_1, X_{i2} \le x_2)$, $p_2 = P(X_{i1} \le x_1, X_{i2} > x_2)$, $p_3 = P(X_{i1} > x_1, X_{i2} \le x_2)$, $p_4 = P(X_{i1} > x_1, X_{i2} > x_2)$, which can be calculated from F_{X_i}.

Distribution for $X_{(r)}$

- Top row ("best unit"):

$$F_{\mathbf{X}_{(n)}}(x_1, x_2) = p_1^n$$

- Bottom row ("worst unit"):

$$F_{\mathbf{X}_{(1)}}(x_1, x_2) = 1 - (P_2 + P_4)^n - (P_3 + P_4)^n + p_4^n$$

Distribution for $X_{(r)}$

- m-component case

$$F_{X_{(r)}}(x_1, x_2, \dots, x_m)$$

$$= P(X_{(r)1} \le x_1, X_{(r)2} \le x_2, \dots, X_{(r)m} \le x_m)$$

$$= \sum_{\substack{0 \le t_1, t_2, \dots, t_{2^m} \le n \\ \sum t_i = n,\, r \le l_1, l_2, \dots, l_m \le n}} \binom{n}{t_1\ t_2\ \dots\ t_{2^m}} p_1^{t_1} p_2^{t_2} \dots p_{2^m}^{t_{2^m}}$$

which can be implemented through programming.

Distribution of $g(X_{(r)})$

- Recall that g(.) is the scoring function, and $g(\mathbf{X}_{(r)})$ is the threshold value for the r^{th} row.
- When we use minimum as our scoring function, i.e.

$g(\mathbf{X}_{(r)}) = \min(X_{(r)1}, X_{(r)2},\ldots, X_{(r)m})$, we have

$$G_r(s) = P(g(\mathbf{X}_{(r)}) \le s)$$

$$= P(X_{(r)1} \le s, \text{or } X_{(r)2} \le s, \ldots, \text{or } X_{(r)m} \le s)$$

$$= 1 - P(X_{(r)1} > s,\ X_{(r)2} > s, \ldots,\ X_{(r)m} > s)$$

$$= 1 - \bar{F}_{\mathbf{X}_{(r)}}(s, s, \ldots, s)$$

Top-k - statistical approach

- F_{X_i} – Joint distribution for the scores of components based on sensor signals
  - Two candidate distribution: ME and MIG
- Multivariate Exponential Distribution
  - Widely applied in lifetime modeling, reliability, etc.
  - Several multivariate versions of such distribution proposed
  - Able to model dependence across components

Base-case simulation study

- We adopt bivariate exponential distribution proposed by Marshall & Olkin in our simulation study
- Experiment parameters

n – number of machines

ρ – correlation coefficient

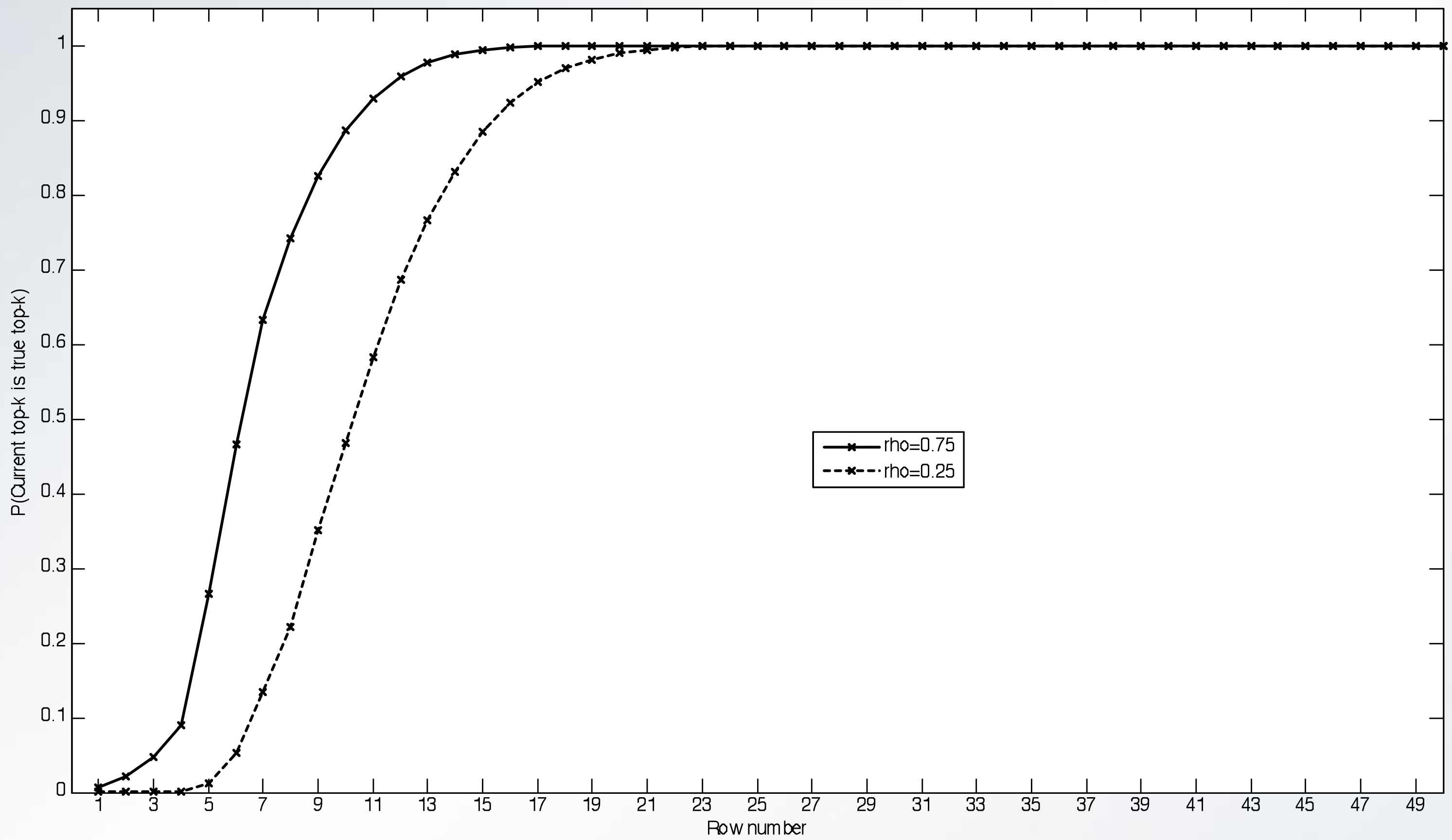
k – size of top-k list

Base-case simulation study

- Output
  - At each row, the probability that the stopping criterion of Threshold Algorithm is met at that row is calculated, i.e.

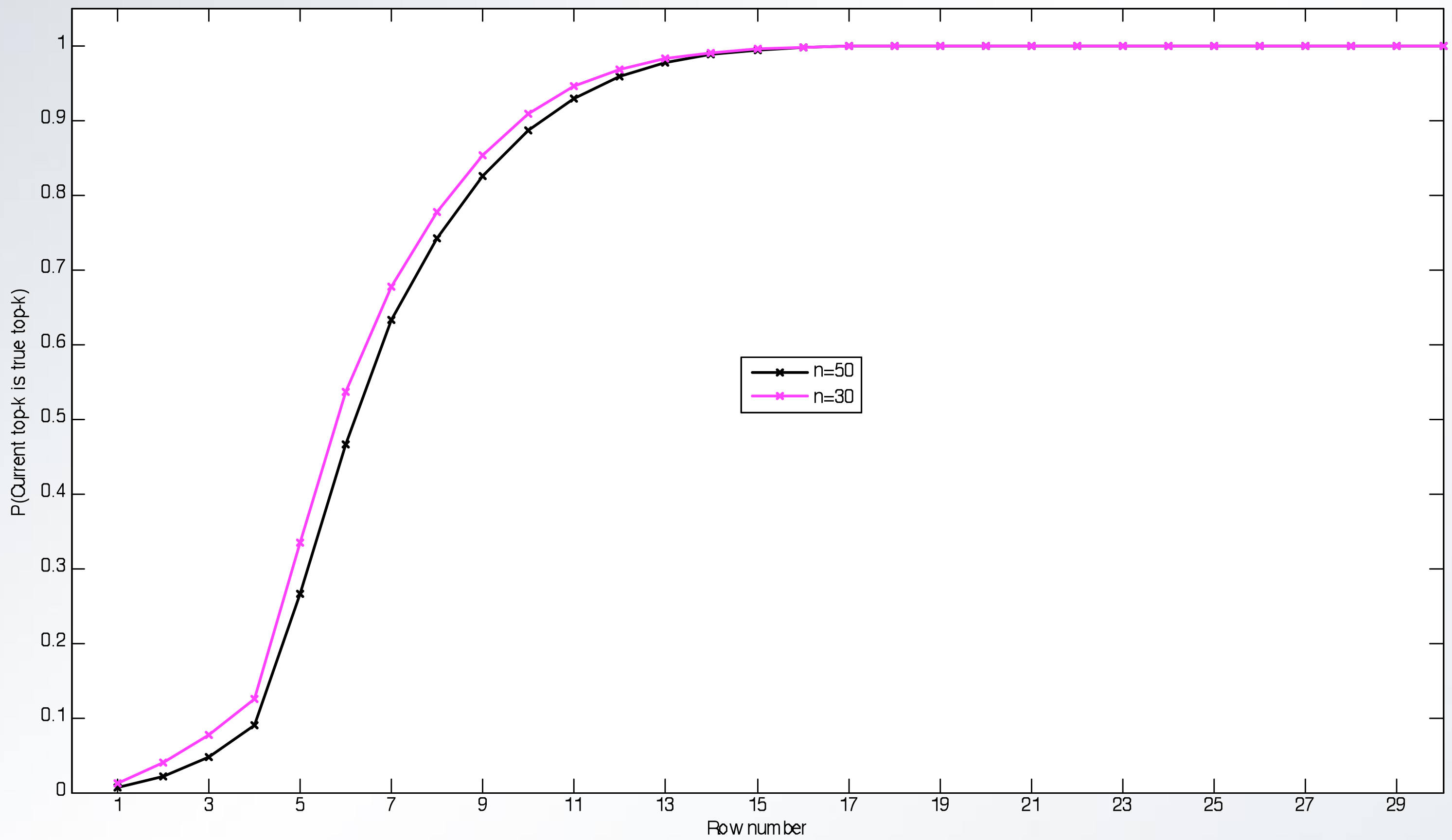
    $P(g(\mathbf{X}_{(r)}) \leq T_r))$, where $g(\mathbf{X}_{(r)})$ is the threshold value and T_r is the minimum overall score of current top-k list.

Base-case simulation study



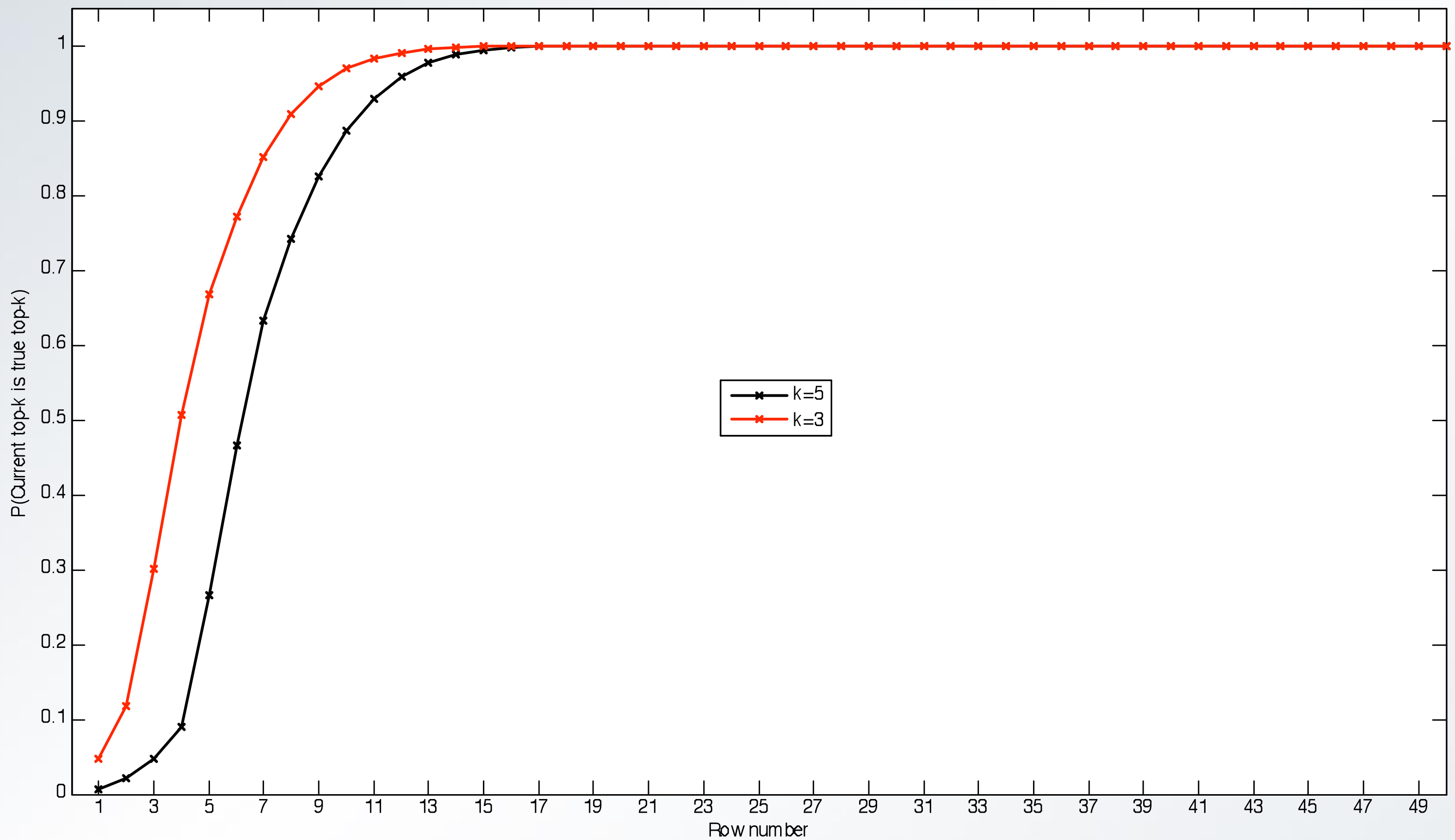
n = 50, k=5, ρ= 0.75 vs 0.25

Base-case simulation study



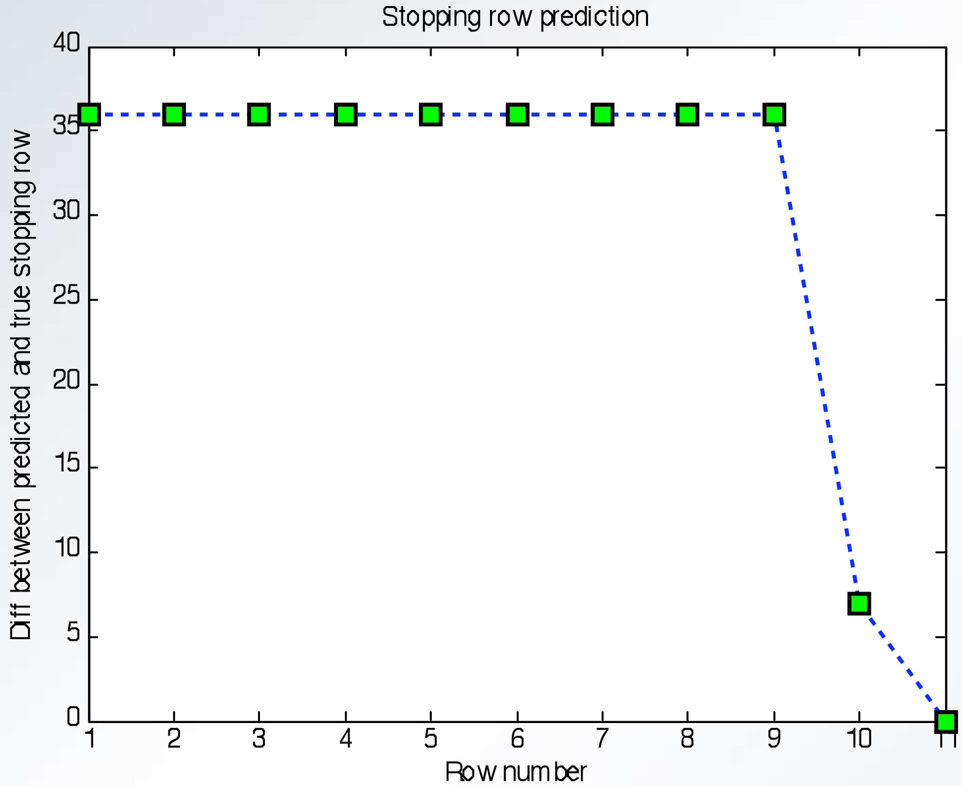
k=5, ρ= 0.75, n = 50 vs 30

Base-case simulation study



n=50, ρ= 0.75, k = 5 vs 3

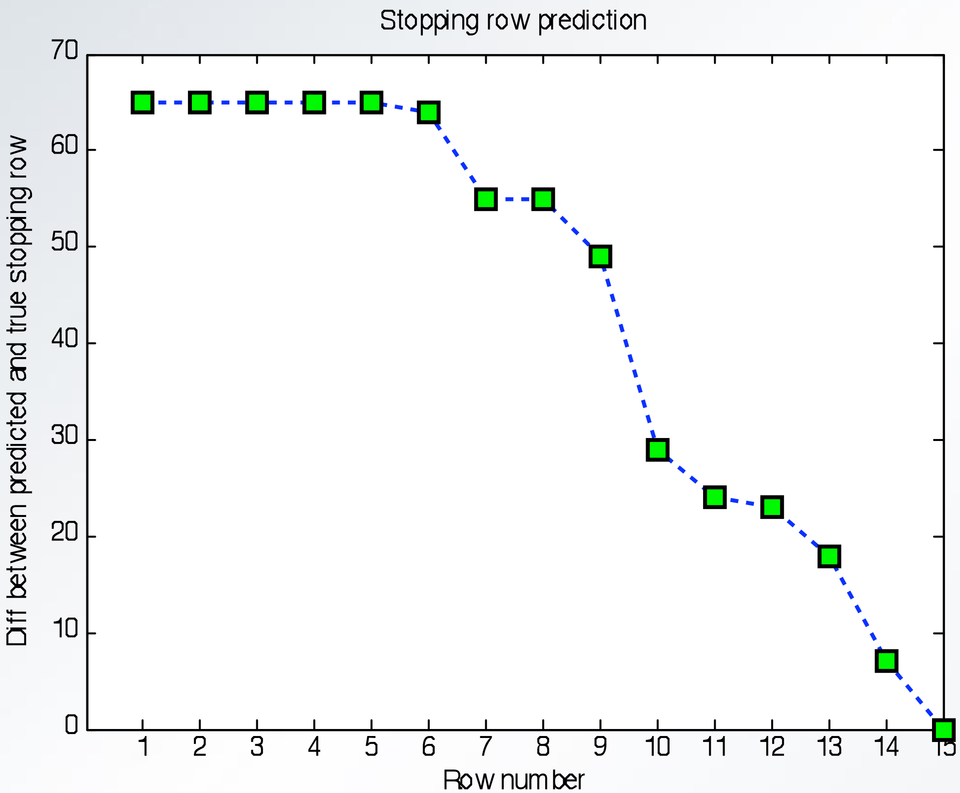
Simulation study – stopping row prediction



$\rho = 0.9$

n = 100, k = 10, 1-α = 0.95

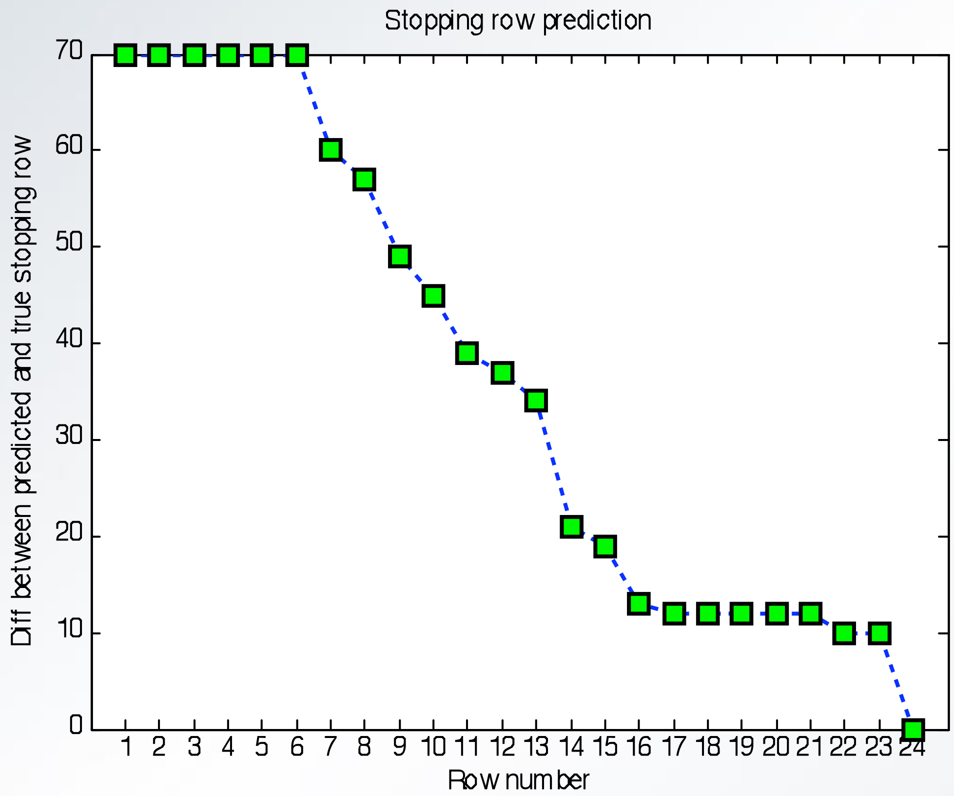
Simulation study – stopping row prediction



ρ = 0.5

n = 100, k = 10, 1-α = 0.95

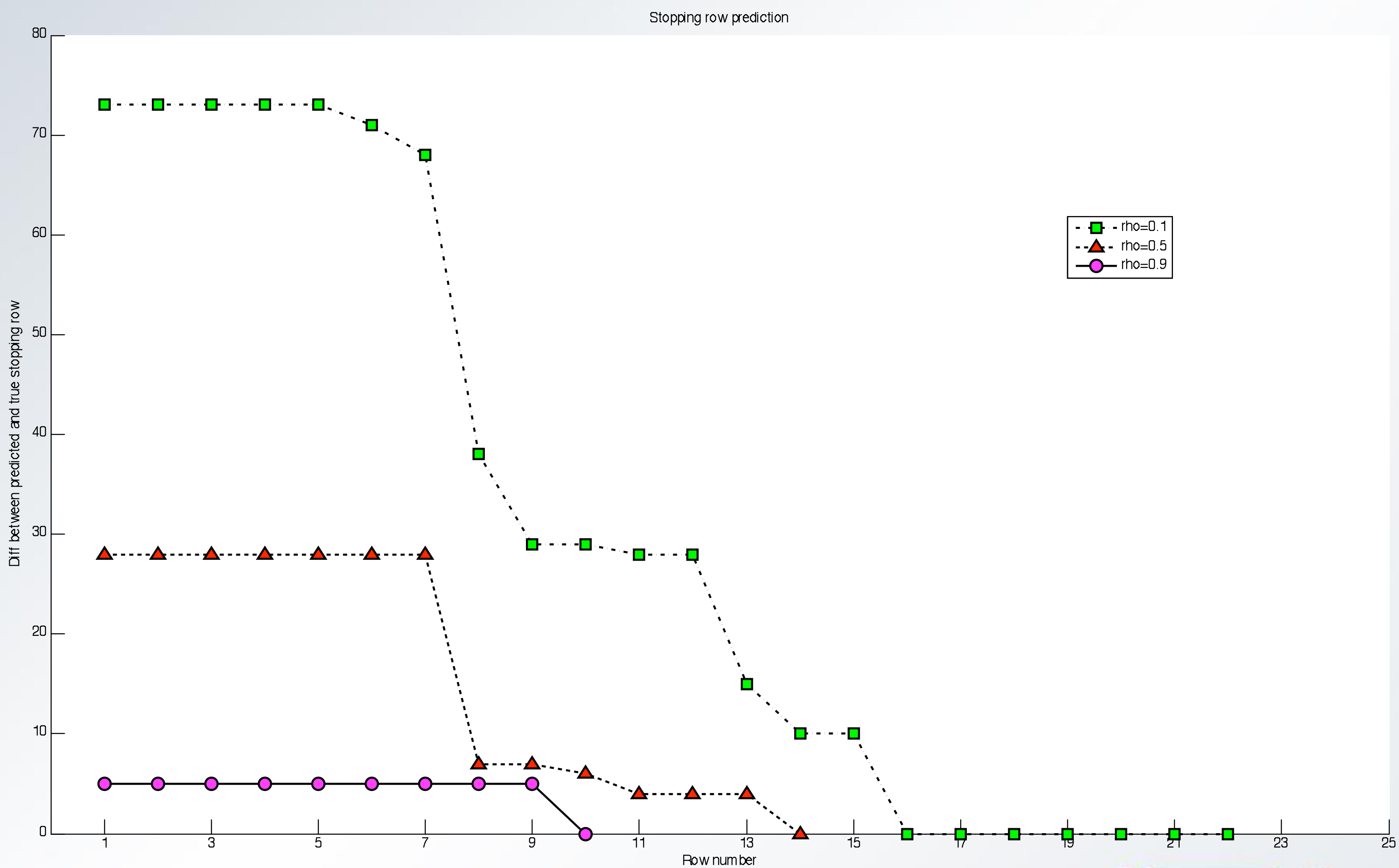
Simulation study – stopping row prediction



$\rho = 0.1$

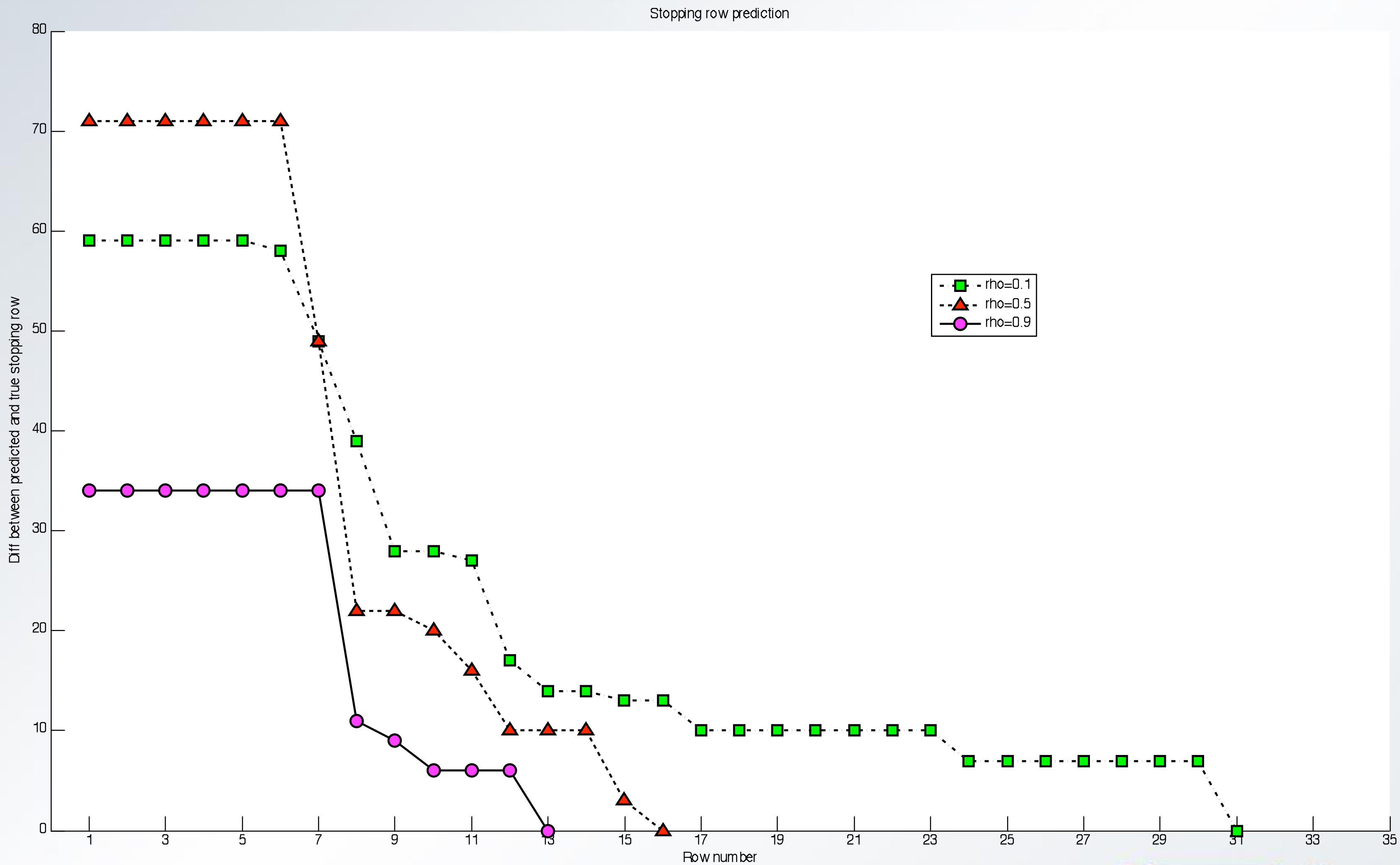
n = 100, k = 10, 1-α = 0.95

Simulation study – stopping row prediction



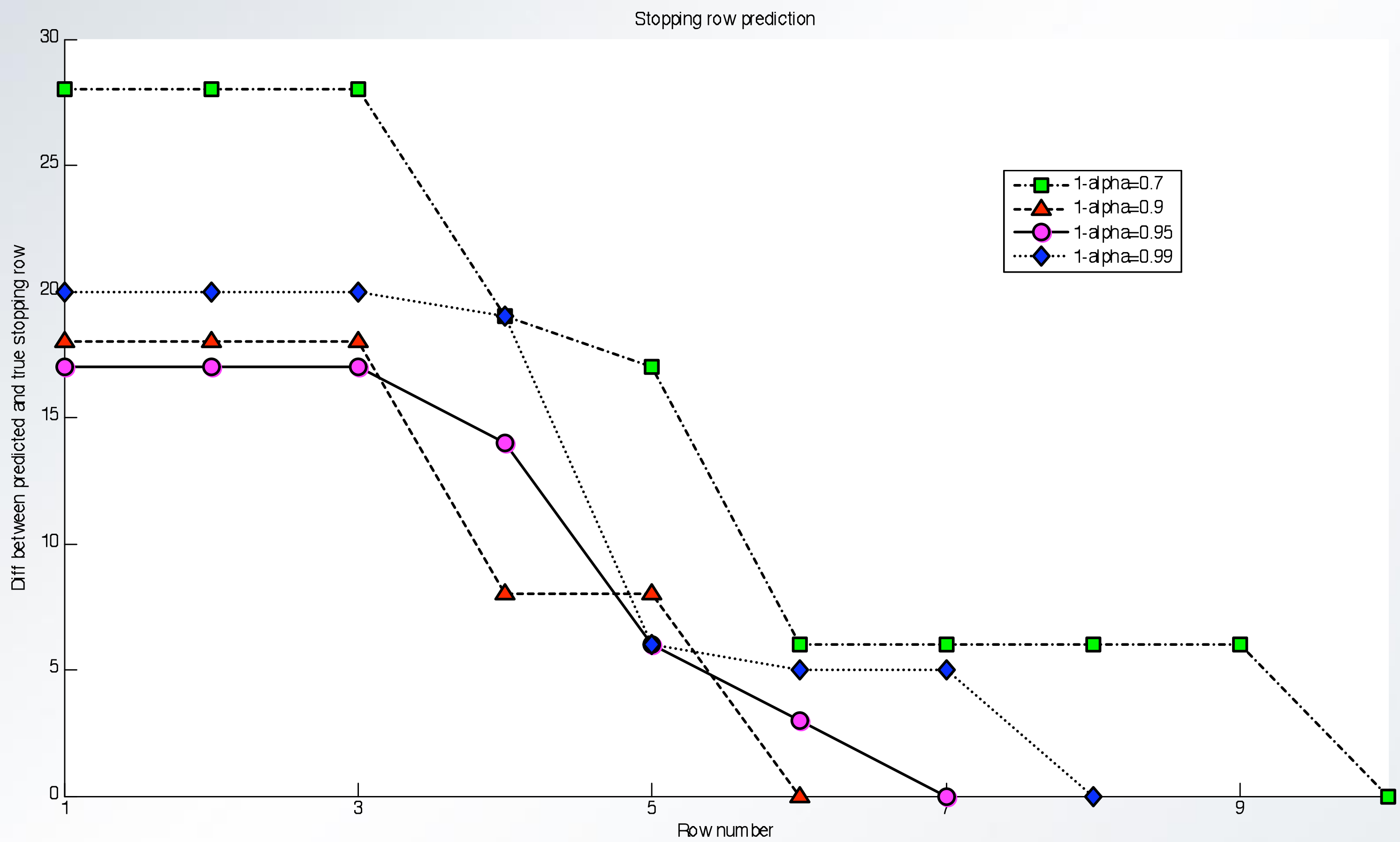
n = 100, k = 10, 1-α= 0.95

Simulation study – stopping row prediction



n = 100, k = 10, 1-α= 0.95

Simulation study – stopping row prediction



n = 50, k = 5, ρ= 0.5

Top-k - statistical approach

- Current work:

  Investigating the Multivariate Inverse Gaussian Case

  – Degradation signals are modeled using Brownian motion-based models

  – First passage time, and hence the RLD of a component is Inverse-Gaussian

  – Correlation structure of components modeled

Minimal sensor probing

- Statistical approach - future work
  - Consider the cost of probing/communicating with a sensor, i.e., data acquisition.
  - Minimize the number of sensor probes necessary to identify the top-k
  - Further exploit the any interdependencies among the degradation of components in the same unit to make probe decisions

Thank you!