Submodularity and Big Data

Jeffrey A. Bilmes

Professor
Departments of Electrical Engineering
& Computer Science and Engineering
University of Washington, Seattle
http://melodi.ee.washington.edu/~bilmes

Friday, May 3rd, 2013

Outline

1. Big Data - The Good, The Bad, and The Ugly

2. Submodularity: Generalized Independence

3. Document Summarization

4. Speech Summarization

5. General Summarization

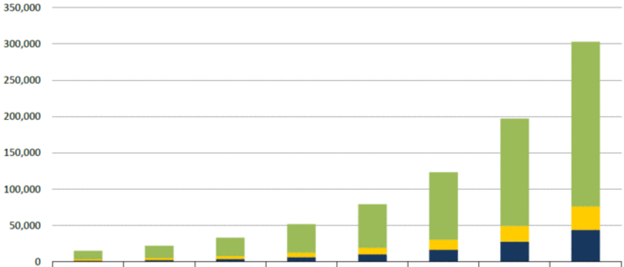
Outline

1. Big Data - The Good, The Bad, and The Ugly
2. Submodularity: Generalized Independence
3. Document Summarization
4. Speech Summarization
5. General Summarization

Big Data

- Big Data is Big and Getting Bigger.

Total Archived Capacity, by Content Type, Worldwide, 2008-2015 (Petabytes)



	2008	2009	2010	2011	2012	2013	2014	2015
🟩 Unstructured	11,430	16,737	25,127	39,237	59,600	92,536	147,885	226,716
🟨 Database	1,952	2,782	4,065	6,179	9,140	13,824	21,532	32,188
🟦 E-mail	1,652	2,552	4,025	6,575	10,411	16,796	27,817	44,091

Source: Enterprise Strategy Group, 2010.

- Every day, we create 2.5 quintillion bytes (2.5 billion gigabytes) of data (source: IBM).
- 90% of the world's data has been created in the last two years.

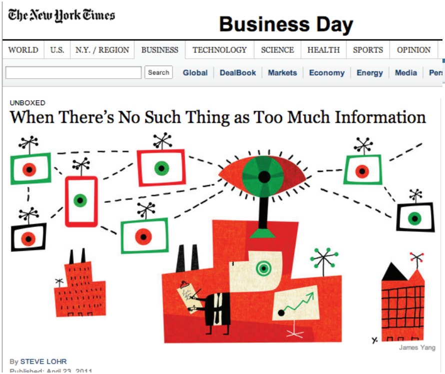
Big Data

Big Data: The Good

- "More is Different", P.W. Anderson, 1972, "The ability to reduce everything to simple fundamental laws does not imply the ability to start from those laws and reconstruct the universe."
- Bigger is not just bigger — bigger is different.
- Emergent properties: the whole is not just more but very different than the sum of its parts:
    - H2O molecules: small (*n*-body), medium (water), large (global weather systems), all w/o changing form of molecular collision events.
    - Neurons: small (neural spike trains, population coding), large (biological intelligence, consciousness)
    - Pixels: small (random dot patterns), large (pointillism, photograph, visual scenes).
    - Sinusoids: small (harmonic series), large (musical textures)
    - People: small (psychology), large (collective intelligence, social choice theory, wisdom of the crowd).

Big Data: The Good

- More is indeed different, and this can be exploited.
- Organizations adopting "data-driven decision making" achieve productivity gains 5 to 6 percent (NYTs, 2/11/2012).



The New York Times

Business Day

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

Search | Global | DealBook | Markets | Economy | Energy | Media | Per

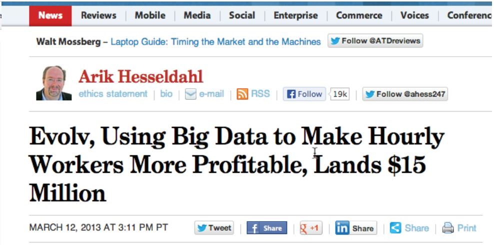
UNBOXED

When There's No Such Thing as Too Much Information

James Yang

By STEVE LOHR

Published: April 23, 2011



News | Reviews | Mobile | Media | Social | Enterprise | Commerce | Voices | Conferenc

Walt Mossberg – Laptop Guide: Timing the Market and the Machines | Follow @ATDreviews

Arik Hesseldahl

ethics statement | bio | e-mail | RSS | Follow 19k | Follow @ahess247

Evolv, Using Big Data to Make Hourly Workers More Profitable, Lands $15 Million

MARCH 12, 2013 AT 3:11 PM PT | Tweet | Share | +1 | Share | Share | Print

Big Data: The Bad

- Big Data is Very Big ⇒ Information Overload.

Big Data: The Bad

- Big Data is Very Big ⇒ Information Overload.
- Personal information overload, even in 1970.

> *In a situation of information overload, inputs exceed the decision maker's capacity to assimilate and act on the information as well as his/her ability to evaluate every alternative. (J. Walker, 1971)*

Big Data: The Bad

- Big Data is Very Big ⇒ Information Overload.
- Personal information overload, even in 1970.

> *In a situation of information overload, inputs exceed the decision maker's capacity to assimilate and act on the information as well as his/her ability to evaluate every alternative. (J. Walker, 1971)*

- Today, there is too much data even for computers to process — data now growing much faster than single core compute ability.

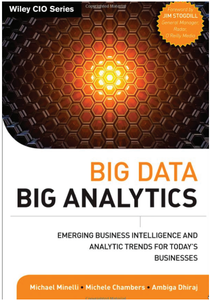
Big Data: The Bad

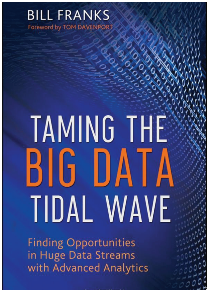
- Big Data is Very Big ⇒ Information Overload.
- Personal information overload, even in 1970.

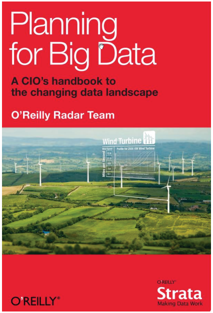
> *In a situation of information overload, inputs exceed the decision maker's capacity to assimilate and act on the information as well as his/her ability to evaluate every alternative. (J. Walker, 1971)*

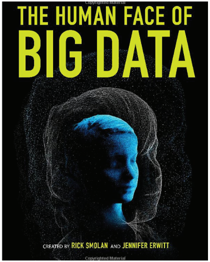
- Today, there is too much data even for computers to process — data now growing much faster than single core compute ability.
- Big data — addressing the new computational challenges:
    1. systems programming,
    2. parallel and distributed computing,
    3. efficient databases.

Many descriptions of big data


Wiley CIO Series
Foreword by JIM STOGDILL General Manager Radar, O'Reilly Media
BIG DATA
BIG ANALYTICS
EMERGING BUSINESS INTELLIGENCE AND ANALYTIC TRENDS FOR TODAY'S BUSINESSES
Michael Minelli • Michele Chambers • Ambiga Dhiraj


BILL FRANKS
Foreword by TOM DAVENPORT
TAMING THE BIG DATA TIDAL WAVE
Finding Opportunities in Huge Data Streams with Advanced Analytics

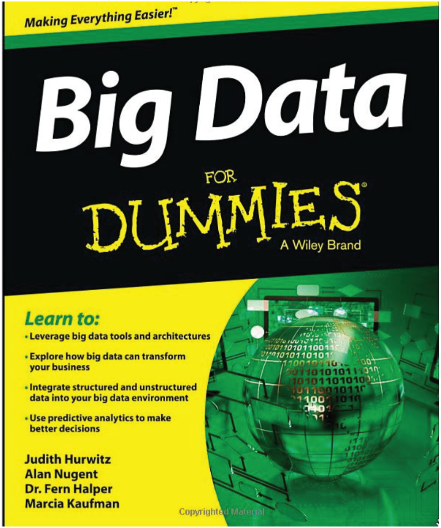

Planning for Big Data
A CIO's handbook to the changing data landscape
O'Reilly Radar Team
Wind Turbine
O'REILLY
O'REILLY Strata Making Data Work


THE HUMAN FACE OF BIG DATA
CREATED BY RICK SMOLAN AND JENNIFER ERWITT

BIG DATA
A REVOLUTION THAT WILL TRANSFORM HOW WE LIVE, WORK, AND THINK
VIKTOR MAYER-SCHÖNBERGER
KENNETH CUKIER

"Read this book or miss out on the opportunity to differentiate and grow your business by leveraging big data."
Big Data Governance
An Emerging Imperative
SUNIL SOARES

Big data is big in many ways



Making Everything Easier!™

Big Data

FOR DUMMIES®

A Wiley Brand

Learn to:

- Leverage big data tools and architectures
- Explore how big data can transform your business
- Integrate structured and unstructured data into your big data environment
- Use predictive analytics to make better decisions

Judith Hurwitz
Alan Nugent
Dr. Fern Halper
Marcia Kaufman

Big Data: The Bad

Big data is being over-hyped

- The New Yorker article “Steamrolled by Big Data,” by Gary Marcus, 4/3/2013.
- New York Times, “What You'll Do Next,” by David Brooks,
- New Republic, “What Big Data Will Never Explain,” by Leon Wieseltier.

Although, nice rebuttal by Andrew McAfee, “Pundits: Stop Sounding Ignorant About Data”, April 2013, Harvard Business Review.

Big Data: The Ugly

- Often, Big Data ⇒ Big Redundancy

I STARTED A TASK FORCE TO ELIMINATE REDUNDANCIES IN OUR INTERNAL PROCESSES.
REALLY? I'M DOING THE SAME THING.
Dilbert.com DilbertCartoonist@gmail.com
4-2-11 ©2011 Scott Adams, Inc. Dist. by Universal Uclick

Big Data: The Ugly

- Often, Big Data ⇒ Big Redundancy

I STARTED A TASK FORCE TO ELIMINATE REDUNDANCIES IN OUR INTERNAL PROCESSES.
REALLY? I'M DOING THE SAME THING.
Dilbert.com DilbertCartoonist@gmail.com
4-2-11 ©2011 Scott Adams, Inc. Dist. by Universal Uclick

- Information Overload

Big Data: The Ugly

- Often, Big Data ⇒ Big Redundancy

I STARTED A TASK FORCE TO ELIMINATE REDUNDANCIES IN OUR INTERNAL PROCESSES.
REALLY? I'M DOING THE SAME THING.

Dilbert.com DilbertCartoonist@gmail.com

4-2-11 ©2011 Scott Adams, Inc. Dist. by Universal Uclick

- ~~Information Overload~~ ⇒ Sample Overload

Big Data: The Ugly

- Often, Big Data ⇒ Big Redundancy

I STARTED A TASK FORCE TO ELIMINATE REDUNDANCIES IN OUR INTERNAL PROCESSES.
REALLY? I'M DOING THE SAME THING.
DilbertCartoonist@gmail.com
Dilbert.com
4-21-11 © 2011 Scott Adams, Inc. Dist. by Universal Uclick

- ~~Information Overload~~ ⇒ Sample Overload
- More samples (e.g., documents, photos, web pages, sentences, utterances, & facebook friends) than necessary to represent an amount of information.

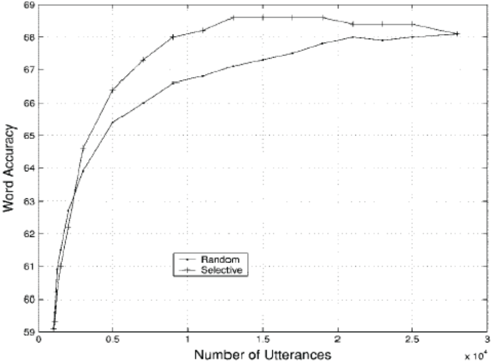
Big Data in Machine Learning

Training and test set sizes are getting big, which is great!

Big Data in Machine Learning

Training and test set sizes are getting big, which is great! However:

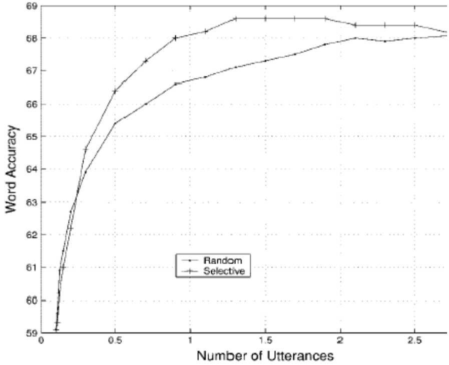
Automatic Speech Recognition
(Riccardi & Hakkani-Tür, 2005)



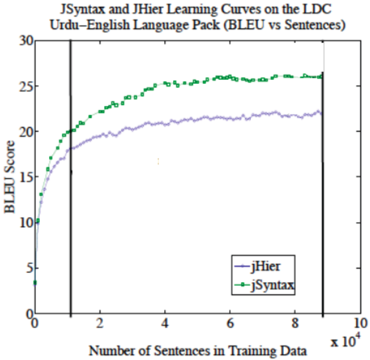
Big Data in Machine Learning

Training and test set sizes are getting big, which is great! However:

Automatic Speech Recognition
(Riccardi & Hakkani-Tür, 2005)



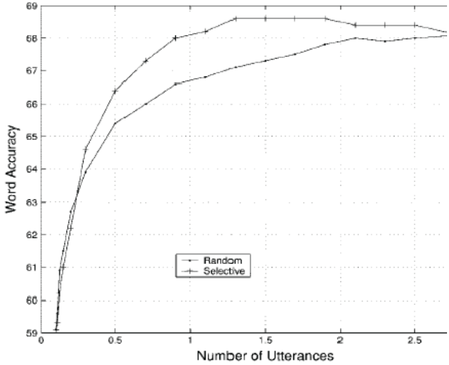
Machine Translation (MT)
(Callison-Burch & Bloodgood, 2010)



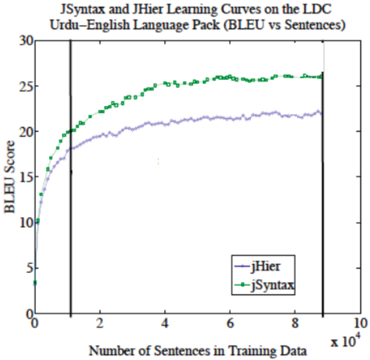
Big Data in Machine Learning

Training and test set sizes are getting big, which is great! However:

Automatic Speech Recognition
(Riccardi & Hakkani-Tür, 2005)



Machine Translation (MT)
(Callison-Burch & Bloodgood, 2010)



Diminishing Returns: the more you have, the less valuable is anything you don't have. Bad for complex systems (e.g., deep models, SVMs).

Big Data: Summary

Big data is different, emergent properties, a higher-level gestalt.

Big data is big, it requires enormous compute to process that much data.

Big data is overabundant, more samples than needed.

There is No Data Like Core Data

Can statistical predictions be cost effective using small data?

There is No Data Like Core Data

Can statistical predictions be cost effective using small data?

Goal: Extract core data from big data. Core data has the properties:

1. Retain information (i.e., decisions based on analysis of original must be close to analysis of core)
2. Same codebook as source. E.g., for text, reduced form of documents still use grammatical human-language sentences. For pictures, still a set of images. For training data, a subset.
3. Need not reconstruct original (need not decompress based on summary to reconstruct original).
4. Small core data should be small enough to significantly reduce cost.

Core data can be like a summary, cliff-notes, extractive summarization.

The Data Subset Selection Problem (DSSS)

- We have big finite set V of size $n = |V|$ of elements.
- Select a small subset $A \subseteq V$ usable as a surrogate for V.
- Ultimate Goal: any question asked based on V can be accurately answered based only on A.
- Ex: Extractive Document Summarization: V is a set of sentences from multiple documents.
- Ex: Automatic speech recognition (ASR): V is a set of training samples (e.g., a large set of acoustic utterances for ASR).

Outline

1. Big Data - The Good, The Bad, and The Ugly
2. Submodularity: Generalized Independence
3. Document Summarization
4. Speech Summarization
5. General Summarization

Quality Functions and Costs

- A quality function $f : 2^V \to \mathbb{R}_+$.
- Given $A \subseteq V$, the value $f(A)$ measures the "quality" of or information within A (how good we measure A to be).
- Moreover, we might have a cost associated with each $v \in V$ measured by a cost function $c : V \to \mathbb{R}_+$.
- Example: $c(v)$ might be the length, or number of words, pixels, or complexity of element v, and $c(A) = \sum_{a \in A} c(a)$.
- If $c(v) = 1$ for all $v \in V$, then $c(A) = |A|$, the size of set A.

Approaches to DSSS Optimization

- Two sensible optimization strategies:
- The "best within a given budget b" approach:

$$A^* \in \operatorname*{argmax}_{A \subseteq V : c(A) \leq b} f(A) \tag{1}$$

- The "least costly with a quality guarantee" approach:

$$A^* \in \operatorname*{argmin}_{A : f(A) \geq \alpha} c(A) \tag{2}$$

Approaches to DSSS

- Without making further assumptions about f, the two optimization problems have exponential cost, even to approximate with any degree of quality assurance, independent of the P$\neq$ NP question.

Approaches to DSSS

- Without making further assumptions about f, the two optimization problems have exponential cost, even to approximate with any degree of quality assurance, independent of the P$\neq$ NP question.
- When f is monotone non-decreasing submodular, then the problems can be solved using the simple greedy algorithm with constant factor guarantees.

Maximization of Non-Decreasing Submodular Functions

- The problem is in general NP-hard (reduction from max-cut).
- Nemhauser et. al. (1978) states that for normalized ($f(\emptyset) = 0$) monotone submodular functions can be maximized, under a cardinality constraint, using a simple greedy algorithm.
- Starting with $S_0 = \emptyset$, we repeat until $|S_i| = k$:

$$S_{i+1} = S_i \cup \left\{ \operatorname*{argmax}_{v \in V \setminus S_i} f(S_i \cup \{v\}) \right\} \tag{3}$$

- Has guarantee $f(S_i) \geq (1 - 1/e) \max_{|S| \leq i} f(S) \approx 0.63 f(S)$.
- Depending on the "curvature" of f, this bound is much better still (better as f becomes less curved).
- Feige (1998): can't be improved. Unless $P = NP$, no polynomial time algorithm can do better than $(1 - 1/e + \epsilon)$ for any $\epsilon > 0$.
- Minoux (1977). Accelerated greedy algorithm significant speeds up process $O(n \log n)$ w. no further approximation needed.

Submodular and Polymatroid Functions

- A function f is submodular if $\forall A, B \subseteq V$

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{4}$$

- Equivalently, a function f is submodular if it satisfies diminishing returns: for all $A \subseteq B$ and $v \notin B$,

$$f(A + v) - f(A) \geq f(B + v) - f(B) \tag{5}$$

  In words: the value of v diminishes as the context in which it is considered grows.

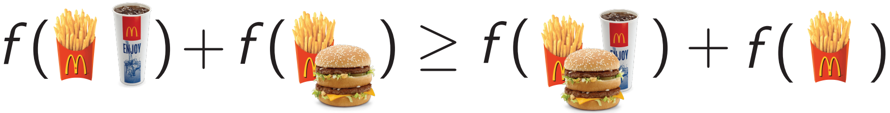
- If a function is also normalized ($f(\emptyset) = 0$), and monotone non-decreasing ($f(A) \leq f(B)$ whenever $A \subseteq B$), then the function is said to be a "polymatroid"

Ex. Submodular: Consumer Costs of Living

- Costs to a consumer are submodular. For example:

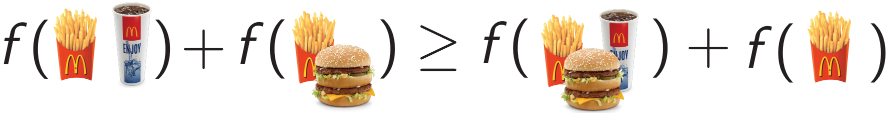
Ex. Submodular: Consumer Costs of Living

- Costs to a consumer are submodular. For example:

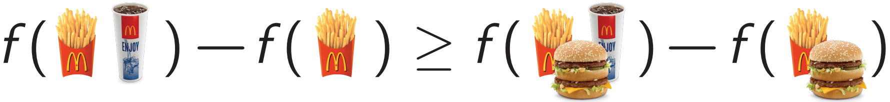


Ex. Submodular: Consumer Costs of Living

- Costs to a consumer are submodular. For example:



- When seen as diminishing returns:



Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \left(\begin{array}{c} | \\ x_1 \\ | \end{array} \right. & \begin{array}{c} | \\ x_2 \\ | \end{array} & \begin{array}{c} | \\ x_3 \\ | \end{array} & \begin{array}{c} | \\ x_4 \\ | \end{array} & \begin{array}{c} | \\ x_5 \\ | \end{array} & \begin{array}{c} | \\ x_6 \\ | \end{array} & \begin{array}{c} | \\ x_7 \\ | \end{array} & \left. \begin{array}{c} | \\ x_8 \\ | \end{array} \right) \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$
\begin{array}{cc}
 & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} &
\begin{pmatrix}
0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\
0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\
0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 5
\end{pmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{pmatrix}
| & | & | & | & | & | & | & | \\
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\
| & | & | & | & | & | & | & |
\end{pmatrix}
\end{array}
$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{cc} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{cc} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} \vert & \vert & \vert & \vert & \vert & \vert & \vert & \vert \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ \vert & \vert & \vert & \vert & \vert & \vert & \vert & \vert \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1,2,3,4,5,6,7,8\}$.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1,2,3\}$, $B = \{3,4,5\}$, $C = \{6,7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{cc} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1,2,3,4,5,6,7,8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix}$$

- Let $A = \{1,2,3\}$, $B = \{3,4,5\}$, $C = \{6,7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \left(\begin{matrix} | \\ x_1 \\ | \end{matrix} \right. & \begin{matrix} | \\ x_2 \\ | \end{matrix} & \begin{matrix} | \\ x_3 \\ | \end{matrix} & \begin{matrix} | \\ x_4 \\ | \end{matrix} & \begin{matrix} | \\ x_5 \\ | \end{matrix} & \begin{matrix} | \\ x_6 \\ | \end{matrix} & \begin{matrix} | \\ x_7 \\ | \end{matrix} & \left. \begin{matrix} | \\ x_8 \\ | \end{matrix} \right) \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c}
\\
1 \\ 2 \\ 3 \\ 4
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{pmatrix}
0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\
0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\
0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 5
\end{pmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\begin{pmatrix}
| & | & | & | & | & | & | & | \\
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\
| & | & | & | & | & | & | & |
\end{pmatrix}
\end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \left(\begin{matrix} | \\ x_1 \\ | \end{matrix} \right. & \begin{matrix} | \\ x_2 \\ | \end{matrix} & \begin{matrix} | \\ x_3 \\ | \end{matrix} & \begin{matrix} | \\ x_4 \\ | \end{matrix} & \begin{matrix} | \\ x_5 \\ | \end{matrix} & \begin{matrix} | \\ x_6 \\ | \end{matrix} & \begin{matrix} | \\ x_7 \\ | \end{matrix} & \left. \begin{matrix} | \\ x_8 \\ | \end{matrix} \right) \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \end{array}\begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{cc} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \left(\begin{matrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{matrix} \right) \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4 × 8 matrix, so $V = \{1,2,3,4,5,6,7,8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{array}$$

- Let $A = \{1,2,3\}$, $B = \{3,4,5\}$, $C = \{6,7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{cc} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1 < r(C) = 2$.

Example: Rank function of a matrix

Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 2 & 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 3 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{array} = \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \left(\begin{matrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{matrix} \right) \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3$, $r(B) = 3$, $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4$, $r(A \cap B) = 1$ $< r(C) = 2.$

Example: Rank function of a matrix

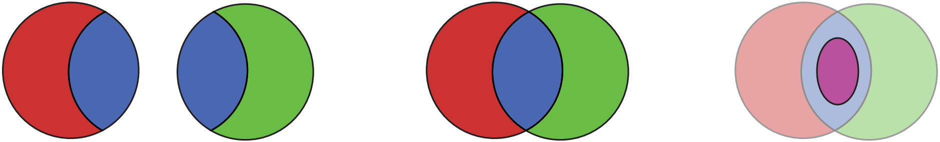
Consider the following 4×8 matrix, so $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

$$\begin{array}{cc} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 2 & 2 & 3 & 0 & 1 & 3 & 1 \\ 0 & 3 & 0 & 4 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} | & | & | & | & | & | & | & | \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ | & | & | & | & | & | & | & | \end{pmatrix} \end{array}$$

- Let $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $C = \{6, 7\}$, $A_r = \{1\}$, $B_r = \{5\}$.
- Then $r(A) = 3,$ $r(B) = 3,$ $r(C) = 2$.
- $r(A \cup C) = 3$, $r(B \cup C) = 3$.
- $r(A \cup A_r) = 3$, $r(B \cup B_r) = 3$, $r(A \cup B_r) = 4$, $r(B \cup A_r) = 4$.
- $r(A \cup B) = 4,$ $r(A \cap B) = 1$ $< r(C) = 2$.
- $6 =$ $r(A) + r(B) > r(A \cup B) + r(A \cap B)$ $= 5$

The Venn and Art of Submodularity

$$\underbrace{r(A)+r(B)}_{=r(A_r)+2r(C)+r(B_r)} \geq \underbrace{r(A\cup B)}_{=r(A_r)+r(C)+r(B_r)} + \underbrace{r(A\cap B)}_{=r(A\cap B)}$$



Rank and Big Data

- Suppose we have a vector space where vectors represent data items (e.g., photos, documents, etc.), and span within the vector space represents "information"
- Let the set V represent the set of "big data".
- If we can find a subset $A \subseteq V$ such that $r(A) = r(V)$ then items A span everything, and anything else not in A is irrelevant given A.
- Rank of a matrix (even in high dimensions) is not ideal for the big-data problem.

Shannon Information: Entropy

- Given $V = \{1, 2, \dots, n\}$ and random variables $X_1, X_2, \dots, X_n$.
- For a given subset $\{a_1, a_2, \dots, a_k\} = A \subset V$ define the function

$$f(A) = H(X_{a_1}, X_{a_2}, \dots, X_{a_k}) \triangleq H(X_A) \tag{6}$$

- $f(V)$ is the amount of Shannon information in all variables.
- Given A such that $f(A) = f(V)$ then, then random variables X_A contain all the information. Any random variable not in A is a deterministic function of those in A.
- The entropy function is also submodular:

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{7}$$

- Captures not only notion of dependence/independence but also partial dependence/independence.
- Entropy is hard to evaluate in practice (requires a joint distribution).

Polymatroids: Generalized Dependence

- All submodular functions express “abstract independence”

Polymatroids: Generalized Dependence

- All submodular functions express "abstract independence"
- Within some combinatorial geometry defined by a polymatroid function, there is a notion of "independence" , i.e., $A \perp\!\!\!\perp B$:

$$f(A \cup B) = f(A) + f(B), \tag{8}$$

Polymatroids: Generalized Dependence

- All submodular functions express "abstract independence"
- Within some combinatorial geometry defined by a polymatroid function, there is a notion of "independence" , i.e., $A \perp\!\!\!\perp B$:

$$f(A \cup B) = f(A) + f(B), \tag{8}$$

- and a notion of "conditional independence" , i.e., $A \perp\!\!\!\perp B | C$:

$$f(A \cup B \cup C) + f(C) = f(A \cup C) + f(B \cup C) \tag{9}$$

Polymatroids: Generalized Dependence

- All submodular functions express "abstract independence"
- Within some combinatorial geometry defined by a polymatroid function, there is a notion of "independence" , i.e., $A \perp\!\!\!\perp B$:

$$f(A \cup B) = f(A) + f(B), \tag{8}$$

- and a notion of "conditional independence" , i.e., $A \perp\!\!\!\perp B | C$:

$$f(A \cup B \cup C) + f(C) = f(A \cup C) + f(B \cup C) \tag{9}$$

- and a notion of "dependence" (conditioning reduces valuation):

$$f(A|B) \triangleq f(A \cup B) - f(B) < f(A), \tag{10}$$

Polymatroids: Generalized Dependence

- All submodular functions express "abstract independence"
- Within some combinatorial geometry defined by a polymatroid function, there is a notion of "independence" , i.e., $A \perp\!\!\!\perp B$:

$$f(A \cup B) = f(A) + f(B), \tag{8}$$

- and a notion of "conditional independence" , i.e., $A \perp\!\!\!\perp B | C$:

$$f(A \cup B \cup C) + f(C) = f(A \cup C) + f(B \cup C) \tag{9}$$

- and a notion of "dependence" (conditioning reduces valuation):

$$f(A|B) \triangleq f(A \cup B) - f(B) < f(A), \tag{10}$$

- and a notion of "conditional mutual information

$$I_f(A; B|C) \triangleq f(A \cup C) + f(B \cup C) - f(A \cup B \cup C) - f(C) \geq 0$$

Polymatroids: Generalized Dependence

- All submodular functions express "abstract independence"
- Within some combinatorial geometry defined by a polymatroid function, there is a notion of "independence" , i.e., $A \perp\!\!\!\perp B$:

$$f(A \cup B) = f(A) + f(B), \tag{8}$$

- and a notion of "conditional independence" , i.e., $A \perp\!\!\!\perp B | C$:

$$f(A \cup B \cup C) + f(C) = f(A \cup C) + f(B \cup C) \tag{9}$$

- and a notion of "dependence" (conditioning reduces valuation):

$$f(A|B) \triangleq f(A \cup B) - f(B) < f(A), \tag{10}$$

- and a notion of "conditional mutual information

$$I_f(A; B|C) \triangleq f(A \cup C) + f(B \cup C) - f(A \cup B \cup C) - f(C) \geq 0$$

- and a notion of "information amongst a collection of sets"

$$I_f(S_1; S_2; \ldots ; S_k) = \sum_{i=1}^{k} f(S_k) - f(S_1 \cup S_2 \cup \cdots \cup S_k) \tag{11}$$

Generalized Abstract Independence: Submodularity

- Many set functions are submodular, including:
  - Graph cut functions, hypergraph cut functions
  - Facility location functions (in operations research)
  - Value of information
  - Joint Active Learning/Semi-Supervised Learning
  - Social network influence, value of a friend.
  - Many energy functions in graphical models and probabilistic reasoning
  - Coverage functions (set cover, sensor placement, . . .)
  - Economies of scale, network externalities, manufacturing costs
- Submodular functions can be minimized in polynomial time!
- As previously mentioned, submodular functions can be constant-factor approximately optimized in low-order polynomial time.

Submodular approaches to Big Data Subset Selection

- We are given a big data set V
- Approach: find a good polymatroid function $f : 2^V \to \mathbb{R}_+$ that represents information in V.

1) Heuristic: build f by hand, hoping that f is a good proxy for the information within V. Acknowledge that f is a surrogate objective, guarantees are only in terms of f.
2) More recent approach: attempt to learn f in some fashion based on training data.

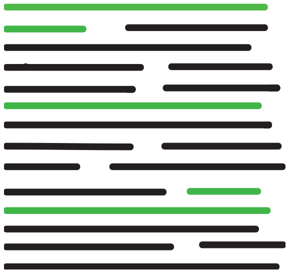
- We report on both kinds of results next.

Outline

1. Big Data - The Good, The Bad, and The Ugly
2. Submodularity: Generalized Independence
3. Document Summarization
4. Speech Summarization
5. General Summarization

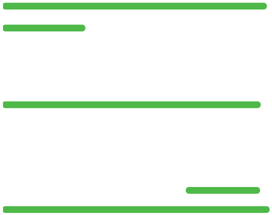
Extractive Document Summarization

- We extract sentences (green) as a summary of the full document



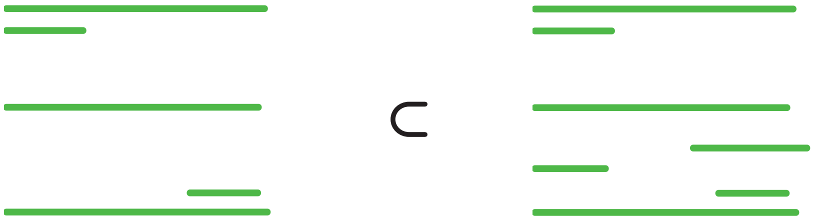
Extractive Document Summarization

- We extract sentences (green) as a summary of the full document



Extractive Document Summarization

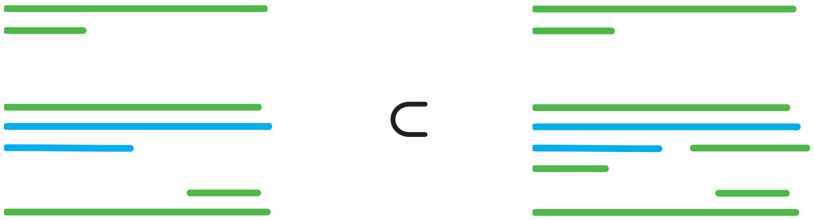
- We extract sentences (green) as a summary of the full document



- The summary on the left is a subset of the summary on the right.

Extractive Document Summarization

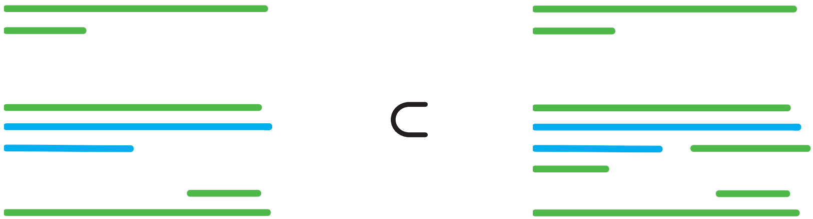
- We extract sentences (green) as a summary of the full document



- The summary on the left is a subset of the summary on the right.
- Consider adding a new (blue) sentence to each of the two summaries.

Extractive Document Summarization

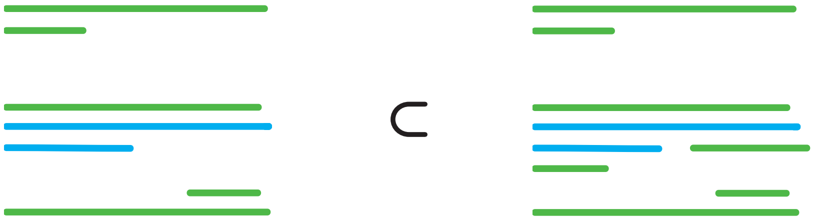
- We extract sentences (green) as a summary of the full document



- The summary on the left is a subset of the summary on the right.
- Consider adding a new (blue) sentence to each of the two summaries.
- The marginal (incremental) benefit of adding the new (blue) sentence to the smaller (left) summary is no more than the marginal benefit of adding the new sentence to the larger (right) summary.

Extractive Document Summarization

- We extract sentences (green) as a summary of the full document



- The summary on the left is a subset of the summary on the right.
- Consider adding a new (blue) sentence to each of the two summaries.
- The marginal (incremental) benefit of adding the new (blue) sentence to the smaller (left) summary is no more than the marginal benefit of adding the new sentence to the larger (right) summary.
- diminishing returns ⇔ submodularity

Problem setup

- The ground set V corresponds to all the sentences in a document.
- Extractive document summarization: select a small subset $S \subseteq V$ that accurately represents the entirety (ground set V).
- The summary S^* must be (budget) length-limited.

$$c(S^*) = \sum_{i \in S^*} c_i \leq b \tag{12}$$

- A set function $f : 2^V \to \mathbb{R}$ measures the quality of the summary S,

Problem (Document Summarization Optimization Problem)

$$S^* \in \operatorname*{argmax}_{S \subseteq V} f(S) \textit{ subject to: } \sum_{i \in S} c_i \leq b. \tag{13}$$

A Practical Algorithm for Large-Scale Summarization

When f is both **monotone** and **submodular**:

- Greedy algorithm with partial enumeration (Sviridenko, 2004), theoretical guarantee of near-optimal solution, but not practical.
- A greedy algorithm (Lin and Bilmes, 2010): near-optimal with theoretical guarantee $(1 - 1/\sqrt{e})$, and practical/scalable!
    - Choose next element with largest ratio of gain over **scaled** cost:

$$k \leftarrow \underset{i \in U}{\operatorname{argmax}} \frac{f(G \cup \{i\}) - f(G)}{(c_i)^r}. \tag{14}$$

    - Scalability: the argmax above can be solved by $O(\log n)$ calls of f, thanks to submodularity
    - Integer linear programming (ILP) takes 17 hours vs. greedy which takes < 1 second!!

The General Form of Our Submodular Functions

- Two properties of a good summary: relevance and non-redundancy.
- The redundancy penalty is usually what violates monotonicity.
- Our approach: we positively **reward diversity** instead of negatively penalizing redundancy:

Definition (The general form of our submodular functions)

$$f(S) = \mathcal{L}(S) + \lambda \mathcal{R}(S)$$

- $\mathcal{L}(S)$ measures the coverage (or fidelity) of summary set S to the document.
- $\mathcal{R}(S)$ rewards diversity in S.
- $\lambda \geq 0$ is a trade-off coefficient.

Coverage function

Coverage Function

$$\mathcal{L}(S) = \sum_{i \in V} \min \{\mathcal{C}_i(S), \alpha\, \mathcal{C}_i(V)\}$$

- $\mathcal{C}_i$ measures how well i is covered by S.
- One simple possible $\mathcal{C}_i$ (that we use) is:

$$\mathcal{C}_i(S) = \sum_{j \in S} w_{i,j},$$

  where $w_{i,j} \geq 0$ measures the similarity between i and j.
- With this C_i, $\mathcal{L}(S)$ is monotone submodular, as required.

Diversity reward function

Diversity Reward Function

$$\mathcal{R}(S) = \sum_{i=1}^{K} \sqrt{\sum_{j \in P_i \cap S} r_j}.$$

- $P_i, i = 1, \cdots K$ is a partition of the ground set V
- $r_j \geq 0$: **singleton reward** of j, which represents the importance of j to the summary.
- square root over the sum of rewards of sentences belong to the same partition (diminishing returns).
- $\mathcal{R}(S)$ is monotone submodular as well.

Diversity Reward Function Mixtures

Alternatively, we can utilize multiple partitions/clusterings, produce a diversity reward function for each one, and mix them together.

Multi-resolution Diversity Reward

$$\mathcal{R}(S) = \lambda_1 \sum_{i=1}^{K_1} \sqrt{\sum_{j \in P_i^{(1)} \cap S} r_j} + \lambda_2 \sum_{i=1}^{K_2} \sqrt{\sum_{j \in P_i^{(2)} \cap S} r_j} + \cdots$$

DUC Evaluations

- DUC (Document Understanding Conference) data `http://duc.nist.gov/`
- Standard Evaluation of extractive document summarization managed by NIST in the years 2004-2007.
- Tasks are both query independent (DUC '04) and query dependent summarization (DUC '05-'07), which is more like web search.
- Standard measure of evaluation performance is the ROUGE measure.
- ROUGE is based on a collection of human generated summaries, so the ROUGE measure can be only used to evaluate a summary.

NIST's ROUGE-N evaluation function

While NIST's ROUGE-N recall score is the standard evaluation measure, it turns out also to be submodular:

$$f_{\text{ROUGE-N}}(S) \triangleq \frac{\sum_{i=1}^{K} \sum_{e \in R_i} \min(c_e(S), r_{e,i})}{\sum_{i=1}^{K} \sum_{e \in R_i} r_{e,i}},$$

where

- S is the candidate summary (a set of sentences extracted from the ground set V)
- $c_e : 2^V \to \mathbb{Z}_+$ is the number of times an n-gram e occurs in summary S, clearly a modular function for each e.
- R_i is the set of n-grams contained in the reference summary i (given K reference summaries).
- and $r_{e,i}$ is the number of times n-gram e occurs in reference summary i.
- ROUGE-N is of course unavailable to optimize directly.

Generic Summarization

Rouge-1: higher is better

- DUC-04: generic summarization

Table : ROUGE-1 recall (R) and F-measure (F) results (%) on DUC-04. DUC-03 was used as development set.

DUC-04	R	F
$\mathcal{L}_1(S)$	39.03	38.65
$\mathcal{R}_1(S)$	38.23	37.81
$\mathcal{L}_1(S) + \lambda\mathcal{R}_1(S)$	**39.35**	**38.90**
Takamura and Okumura (2009)	38.50	-
Wang et al. (2009)	39.07	-
Lin and Bilmes (2010)	-	38.39
Best system in DUC-04 (peer 65)	38.28	37.94

DUC-05 results

Rouge-2: higher is better

Table : ROUGE-2 recall (R) and F-measure (F) results (%)

	R	F
$\mathcal{L}_1(S) + \lambda \mathcal{R}_Q(S)$	7.82	7.72
$\mathcal{L}_1(S) + \sum_{\kappa=1}^{3} \lambda_\kappa \mathcal{R}_{Q,\kappa}(S)$	**8.19**	**8.13**
Daumé III and Marcu (2006)	6.98	-
Wei et al. (2010)	8.02	-
Best system in DUC-05 (peer 15)	7.44	7.43

- DUC-06 was used as training set for the objective function with single diversity reward.
- DUC-06 and 07 were used as training sets for the objective function with multi-resolution diversity reward

DUC-06 results

Rouge-2: higher is better

Table : ROUGE-2 recall (R) and F-measure (F) results (%)

	R	F
$\mathcal{L}_1(S) + \lambda \mathcal{R}_Q(S)$	9.75	9.77
$\mathcal{L}_1(S) + \sum_{\kappa=1}^{3} \lambda_\kappa \mathcal{R}_{Q,\kappa}(S)$	**9.81**	**9.82**
Celikyilmaz and Hakkani-tür (2010)	9.10	-
Shen and Li (2010)	9.30	-
Best system in DUC-06 (peer 24)	9.51	9.51

- DUC-05 was used as training set for the objective function with single diversity reward.
- DUC-05 and 07 were used as training sets for the objective function with multi-resolution diversity reward

DUC-07 results

Rouge-2: higher is better

Table : ROUGE-2 recall (R) and F-measure (F) results (%)

	R	F
$\mathcal{L}_1(S) + \lambda \mathcal{R}_Q(S)$	12.18	12.13
$\mathcal{L}_1(S) + \sum_{\kappa=1}^{3} \lambda_\kappa \mathcal{R}_{Q,\kappa}(S)$	**12.38**	**12.33**
Toutanova et al. (2007)	11.89	11.89
Haghighi and Vanderwende (2009)	11.80	-
Celikyilmaz and Hakkani-tür (2010)	11.40	-
Best system in DUC-07 (peer 15), using web search	**12.45**	12.29

- DUC-05 was used as training set for the objective function with single diversity reward.
- DUC-05 and 06 were used as training sets for the objective function with multi-resolution diversity reward.

Max-Margin Learning of Submodular Mixtures

- Learning submodular functions is hard, but we have recently developed a method to learn mixtures of submodular components.
- I.e., We consider hypothesis functions with the following form

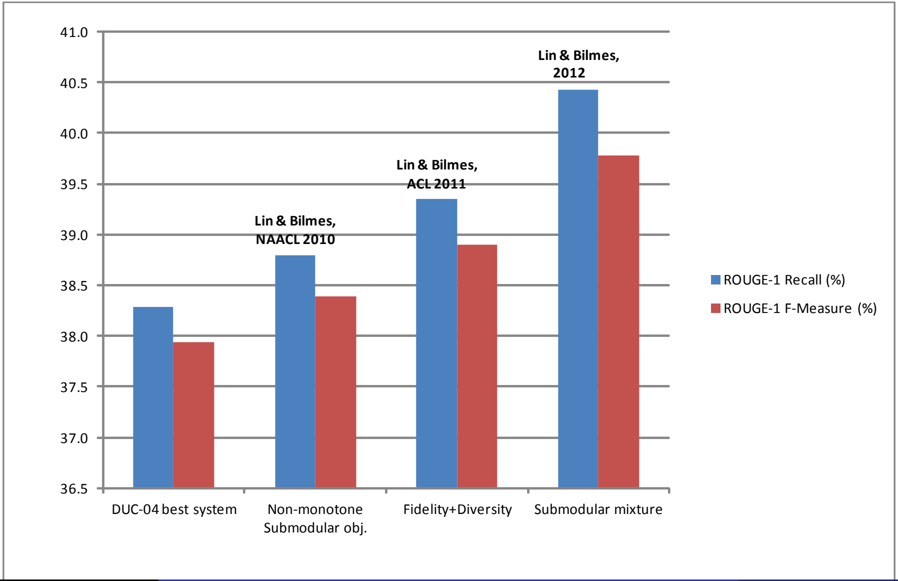
$$h(x; w) = \operatorname*{argmax}_{y \in \mathcal{Y}_x} s(x, y)$$
$$= \operatorname*{argmax}_{y \in \mathcal{Y}_x} \sum_i w_i f_i(x, y).$$

where $w \geq 0$ and $f_i : \mathcal{X} \times \mathcal{Y}_{\mathbf{x}} \to \mathbb{R}$ is submodular on $\mathcal{Y}_{\mathbf{x}}$ for a given $x \in \mathcal{X}$.

- We learn the mixture coefficients using a submodular loss (related to rouge) and a large-margin learning objective.
- Submodular components consist of the types above, with addition of facility location like components.

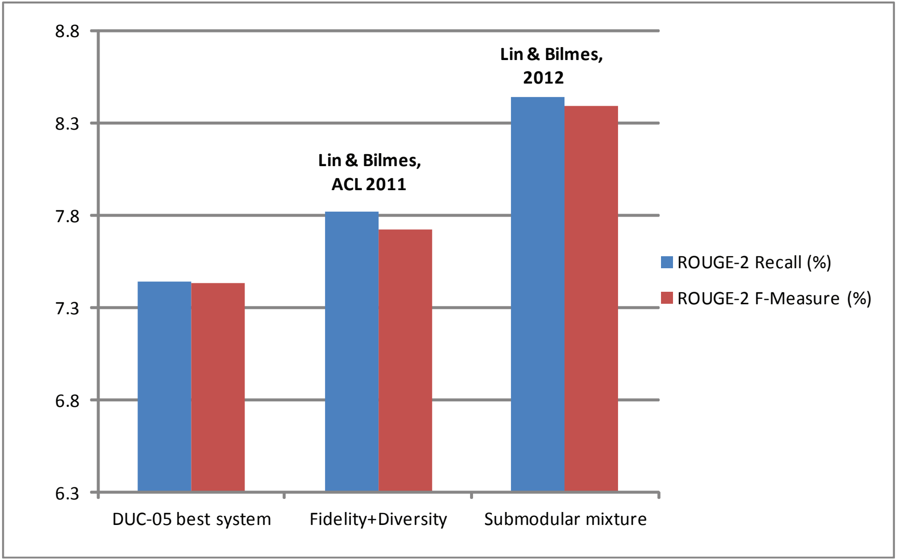
DUC-04 Results

Rouge-1: higher is better



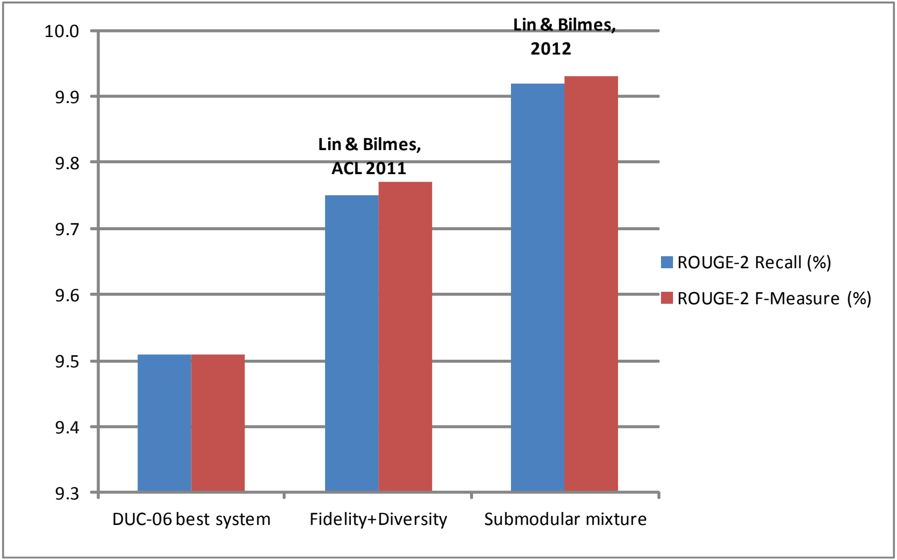
DUC-05 Results

Rouge-2: higher is better



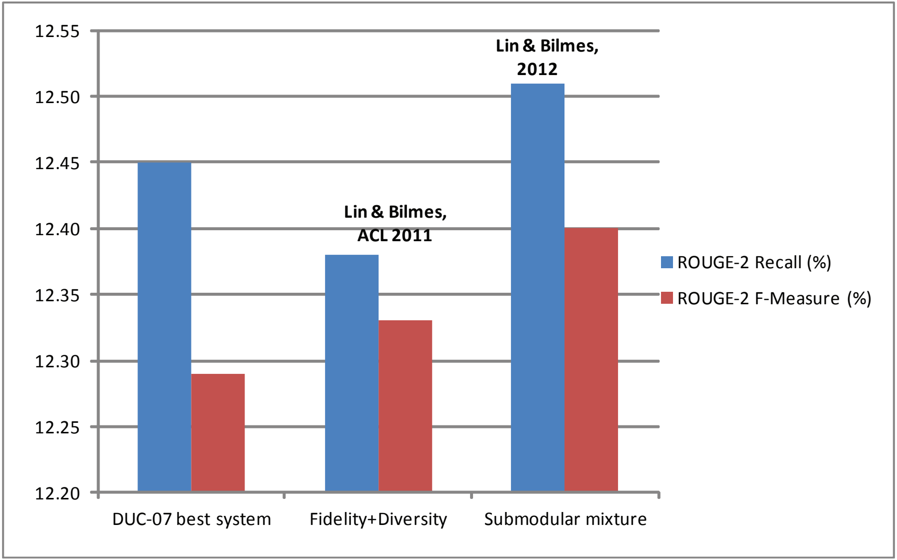
DUC-06 Results

Rouge-2: higher is better



DUC-07 Results

Rouge-2: higher is better



Outline

1. Big Data - The Good, The Bad, and The Ugly

2. Submodularity: Generalized Independence

3. Document Summarization

4. Speech Summarization

5. General Summarization

Speech Subset Selection: Two Forms

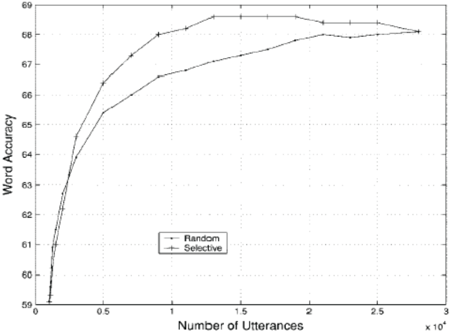
1. Corpus Summarization: Given a large set of speech utterances $V = \{v_1, v_2, \ldots, v_n\}$, choose a small subset $A \subseteq V$ that is representative of V.
   - Summary must be representative (and answer any query accurately) relative to the whole.
   - Goal: training on summary should result in same performance as training on whole.

Speech Subset Selection: Two Forms

1. Corpus Summarization: Given a large set of speech utterances $V = \{v_1, v_2, \ldots, v_n\}$, choose a small subset $A \subseteq V$ that is representative of V.
   - Summary must be representative (and answer any query accurately) relative to the whole.
   - Goal: training on summary should result in same performance as training on whole.
2. Corpus Selection: Given a large set of speech utterances $V = \{v_1, v_2, \ldots, v_n\}$, choose a good subset $A \subseteq V$ that limits the vocabulary size, say $\Gamma(A)$.
   - Ex: large amount of acoustics while limit complexity of language model

Corpus Summarization: motivation

- Large vocabulary speech recognition training is both resource (disk, memory) and time consuming.
- Particularly acute with recent models (Deep Neural Networks) which can take weeks to train a single configuration of a single system.
- Training (and test) data sets are redundant. From: Riccardi & Hakkani-T ur, 2005, although this kind of curve is typical.



- Why waste time/resources on information you already know?

Instantiating a Submodular Function: Some Choices

- Polymatroid: non-negative monotone non-decreasing submodular
- Facility location:

$$f(A) = \sum_{i \in V} \max_{j \in A} w_{ij} \tag{15}$$

- Saturated graph cut:

$$f(A) = \sum_{i \in V} \min(C_i(A), \alpha C_i(A)) \tag{16}$$

  where $C_i(A) = \sum_{j \in A} w_{ij}$.

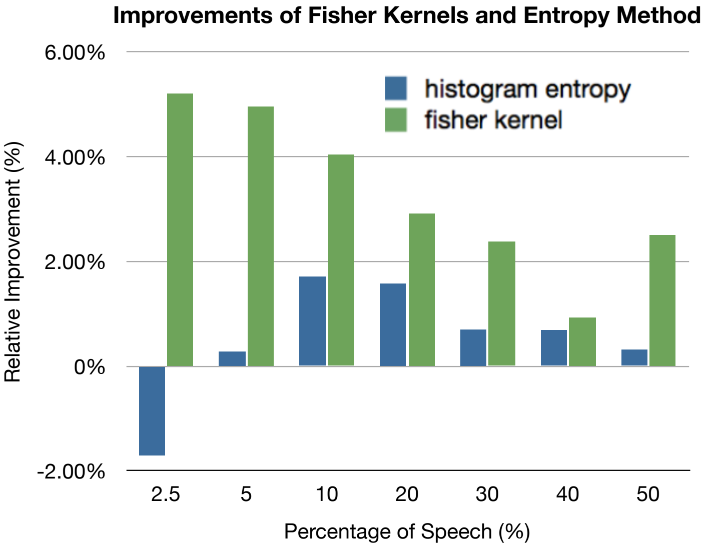
- Diversity function:

$$f(A) = \sum_{k=1}^{K} \sqrt{m(A \cap V_k)} \tag{17}$$

  for partition $V = V_1 \cup \cdots \cup V_k$.

Results on TIMIT

Fisher Kernel (Submodular) vs. Histogram Entropy (Not-submodular)



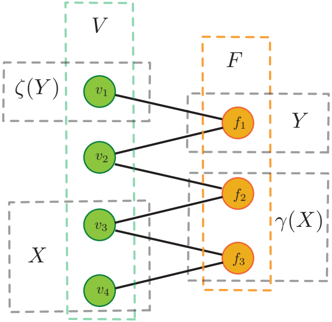
Speech Subset Selection: Two Forms

1. Corpus Summarization: Given a large set of speech utterances $V = \{v_1, v_2, \ldots, v_n\}$, choose a small subset $A \subseteq V$ that is representative of V.
   - Summary must be representative (and answer any query accurately) relative to the whole.
   - Goal: training on summary should result in same performance as training on whole.
2. Corpus Selection: Given a large set of speech utterances $V = \{v_1, v_2, \ldots, v_n\}$, choose a good subset $A \subseteq V$ that limits the vocabulary size, say $\Gamma(A)$.
   - Ex: large amount of acoustics while limit complexity of language model

Corpus Selection: motivation

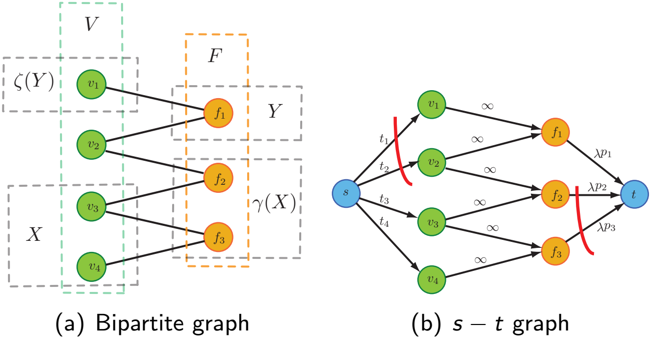
- Machine learning: complexity is often linear in number of samples but polynomial in the number of types of objects.
- Canonical example: speech recognition: adding more training samples is relatively easy, except when the vocabulary expands (e.g., $O(N^3)$ or $O(N^4)$).
- This inhibits rapid turnaround time for novel and expensive surface methods (e.g., deep acoustic modeling in speech recognition).
- Goal: find a way to select a subset of the data while limiting the number of types.

Corpus Selection: description



- Bipartite graph (V, F, E) where V is the set of utterances (sentences) and F is the set of words.
- $\gamma(X)$ are the neighbors of X, $\zeta(Y)$ are the sole neighbors of Y.
- $\Gamma(X) \triangleq w(\gamma(X))$ is submodular, where $w : 2^V \to \mathbb{R}$ is a modular weight function, while $w(\zeta(Y))$ is supermodular.
- King et al. 2005: maximizing $w(\zeta(Y))$ with cardinality constraint, using greedy algorithm. This can do unboundedly poorly.

Fast Parametric Max flow implementation



(a) Bipartite graph (b) $s - t$ graph

Figure : In subfigure (a), $V = \{v_1, v_2, v_3, v_4\}$ and $F = \{f_1, f_2, f_3\}$. For $X = \{v_3, v_4\}$, $\gamma(X) = \{f_2, f_3\}$; for $Y = \{f_1\}$, $\zeta(Y) = \{v_1\}$. In (b), the s-t graph corresponding to $w(V \setminus X) + \lambda\Gamma(X)$.

Results - try to get words with many phones

Corpus D, $\Gamma_2(X) = \sum_{i \in \gamma(X)} p_i = \sum_{i \in \gamma(X)} \frac{c}{q_i}$, where c is a constant, and q_i is the number of phonemes in the pronunciation of word i.

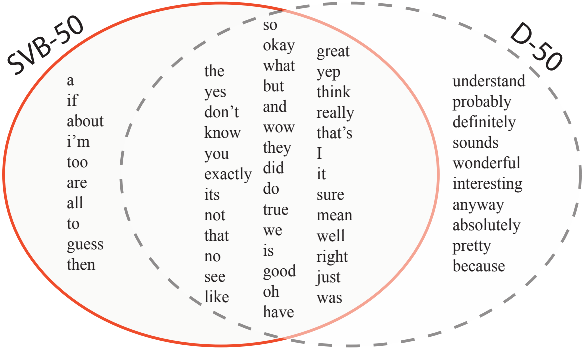


Figure : Venn diagram showing the vocabulary difference between SVitchboard-50 and D-50.

Outline

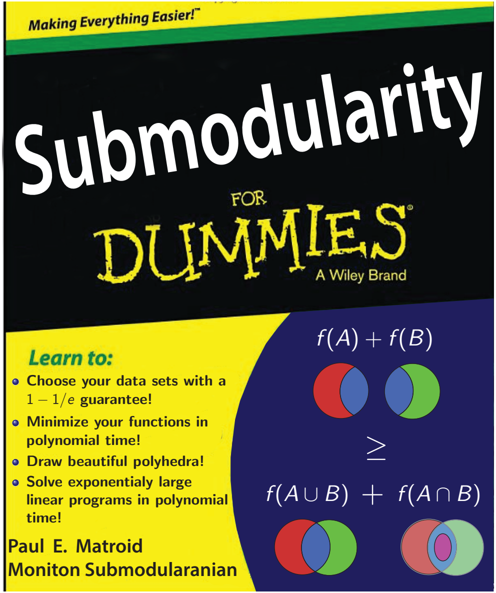
1. Big Data - The Good, The Bad, and The Ugly
2. Submodularity: Generalized Independence
3. Document Summarization
4. Speech Summarization
5. General Summarization

Image Collection & Core Set On Right

References & Notes

- Submodular class notes: http://j.ee.washington.edu/~bilmes/classes/ee596a_fall_2012/
- Submodularity in machine learning: http://melodi.ee.washington.edu/~bilmes/pgs/class_rescat.html#Submodularity
- Hui Lin and Jeff Bilmes. A Class of Submodular Functions for Document Summarization. In The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT-2011), Portland, OR, June 2011. (long paper) http://melodi.ee.washington.edu/~bilmes/pgs/b2hd-lin2011-class-submod-sum.html
- Hui Lin and Jeff Bilmes. Multi-document Summarization via Budgeted Maximization of Submodular Functions. In North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2010), Los Angeles, CA, June 2010. http://melodi.ee.washington.edu/~bilmes/pgs/b2hd-lin2010-submod-sum-nlp.html
- Hui Lin and Jeff Bilmes. Learning Mixtures of Submodular Shells with Application to Document Summarization. In Uncertainty in Artificial Intelligence (UAI), AUAI, Catalina Island, USA, July 2012. http://melodi.ee.washington.edu/~bilmes/pgs/b2hd-hui2012-submodular-shells-summarization.html

The End: Thank you!



Making Everything Easier!™

Submodularity

FOR DUMMIES®

A Wiley Brand

Learn to:

- Choose your data sets with a $1 - 1/e$ guarantee!
- Minimize your functions in polynomial time!
- Draw beautiful polyhedra!
- Solve exponentialy large linear programs in polynomial time!

Paul E. Matroid
Moniton Submodularanian

$$f(A) + f(B)$$

$$\geq$$

$$f(A \cup B) \quad + \quad f(A \cap B)$$

Can Data Compression Help?

- Data is often redundant. When fully non-redundant, it is in compressed form.

Can Data Compression Help?

- Data is often redundant. When fully non-redundant, it is in compressed form.
- Properties of compression:
  1. Lossless compression algorithms typically reconstruct exactly the original data, at the original size. Information about size is preserved.
  2. Lossy compression also often preserves size on reconstruction (e.g., music or video).
  3. Compression often changes the code, uses codewords that are uninterpretable (look like random bit strings) except in the context of a decoder. E.g., a human can't directly read a gzipped `.txt` file.

Can Data Compression Help?

- Data is often redundant. When fully non-redundant, it is in compressed form.
- Properties of compression:
  1. Lossless compression algorithms typically reconstruct exactly the original data, at the original size. Information about size is preserved.
  2. Lossy compression also often preserves size on reconstruction (e.g., music or video).
  3. Compression often changes the code, uses codewords that are uninterpretable (look like random bit strings) except in the context of a decoder. E.g., a human can't directly read a gzipped `.txt` file.
- Hence, compression has undesirable properties in this case.