CS665: Advanced Data Mining

Lecture#17: SVD-3

U Kang
KAIST

Outline

- ☐ **SVD Properties**
- ☐ Query feedback
- ☐ Conclusion

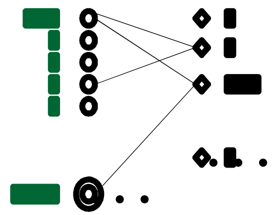
SVD - Other properties - summary

- can produce orthogonal basis (obvious)
- can solve over- and under-determined linear pr oblems (see C(1) property)
- can compute ‘fixed points’ (= ‘steady state pro b. in Markov chains’) (see C(4) property)

Properties – sneak preview!

IMPORTANT!

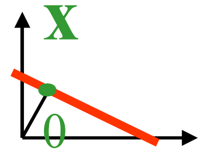
A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \, \mathbf{\Lambda}_{[r \times r]} \, \mathbf{V}^{\mathbf{T}}_{[r \times m]}$



B(5): $(\mathbf{A}^{T} \mathbf{A})^{k} \, \mathbf{v}' \sim (\text{constant}) \, \mathbf{v}_{1}$

C(1): $\mathbf{A}_{[n \times m]} \, \mathbf{x}_{[m \times 1]} = \mathbf{b}_{[n \times 1]}$

then, $\mathbf{x}_0 = \mathbf{V} \, \mathbf{\Lambda}^{(-1)} \, \mathbf{U}^{T} \, \mathbf{b}$: shortest, actual or least-squa res solution



C(4): $\mathbf{A}^{\mathbf{T}} \mathbf{A} \, \mathbf{v}_{1} = \lambda_{1}^{2} \, \mathbf{v}_{1}$

SVD -outline of properties

- (A): obvious
- (B): less obvious
- (C): least obvious (and most powerful!)

Properties - by defn.:

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

A(1): $\mathbf{U}^T_{[r \times n]} \mathbf{U}_{[n \times r]} = \mathbf{I}_{[r \times r]}$ (identity matrix)

A(2): $\mathbf{V}^T_{[r \times n]} \mathbf{V}_{[n \times r]} = \mathbf{I}_{[r \times r]}$

A(3): $\mathbf{\Lambda}^k = \mathrm{diag}(\lambda_1^k, \lambda_2^k, \ldots \lambda_r^k)$ (k: ANY real nu mber)

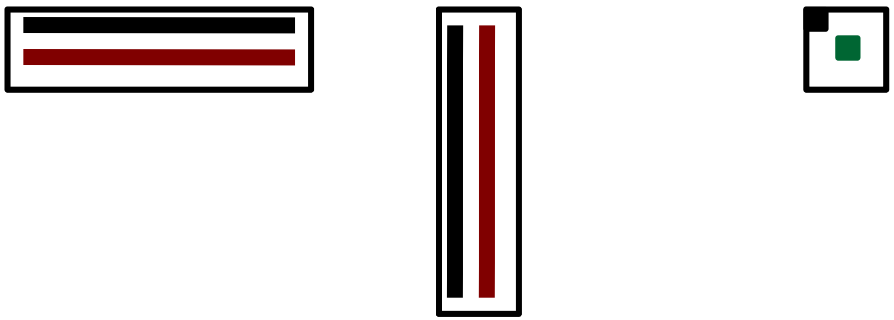
A(4): $\mathbf{A}^{\mathbf{T}} = \mathbf{V} \mathbf{\Lambda} \mathbf{U}^{\mathbf{T}}$

Reminder: ‘column orthonormal’

- $\mathbf{V}^{\mathrm{T}} \quad \mathbf{V} \quad = \quad \mathbf{I}_{[\mathbf{r}\ \mathbf{x}\ \mathbf{r}]}$

v1

$\mathbf{v}_2$



$\mathbf{v_1}^{\mathrm{T}} \times \mathbf{v_1} = 1$

$\mathbf{v_1}^{\mathrm{T}} \times \mathbf{v_2} = 0$

Less obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

B(1): $\mathbf{A}_{[n \times m]} (\mathbf{A}^{T})_{[m \times n]} = ??$

Less obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

B(1): $\mathbf{A}_{[n \times m]} (\mathbf{A}^{T})_{[m \times n]} = \mathbf{U} \mathbf{\Lambda}^{2} \mathbf{U}^{T}$

symmetric; Intuition?

Less obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V^T}_{[r \times m]}$

B(1): $\mathbf{A}_{[n \times m]} (\mathbf{A}^T)_{[m \times n]} = \mathbf{U} \mathbf{\Lambda}^2 \mathbf{U}^T$

symmetric; Intuition?

'document-to-document' similarity matrix

B(2): symmetrically, for '**V**'

$(\mathbf{A}^T)_{[m \times n]} \mathbf{A}_{[n \times m]} = \mathbf{V} \mathbf{\Lambda}^2 \mathbf{V}^T$

Intuition?

Less obvious properties

A: term-to-term similarity matrix

B(3): $((\mathbf{A}^T)_{[m \times n]} \mathbf{A}_{[n \times m]})^k = \mathbf{V} \, \mathbf{\Lambda}^{2k} \, \mathbf{V}^T$

and

B(4): $(\mathbf{A}^T \mathbf{A})^k \sim \mathbf{v}_1 \, \lambda_1^{2k} \, \mathbf{v}_1^T$ for k>>1

where

$\mathbf{v}_1$: [m x 1] first column (singular-vector) of **V**

λ_1: strongest singular value

Proof of (B4)?

Less obvious properties

B(4): $(\mathbf{A}^T \mathbf{A})^k \sim \mathbf{v}_1 \lambda_1^{2k} \mathbf{v}_1^T$ for k>>1

$\boxed{\text{B(5): } (\mathbf{A}^T \mathbf{A})^k \mathbf{v'} \sim (\text{constant})\, \mathbf{v}_1}$

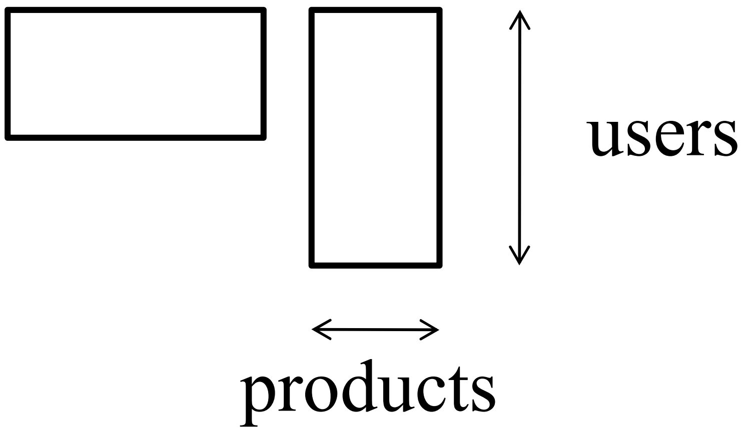
ie., for (almost) any **v'**, it converges to a vector parallel to $\mathbf{v}_1$

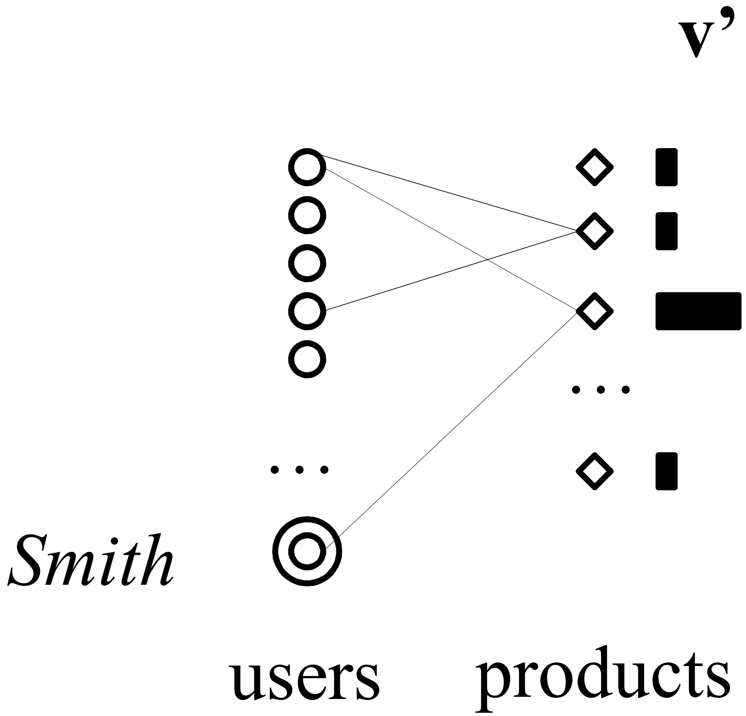
Thus, useful to compute first singular vector/value (as well as the next ones, too...)

Proof of (B5)?

Property (B5)

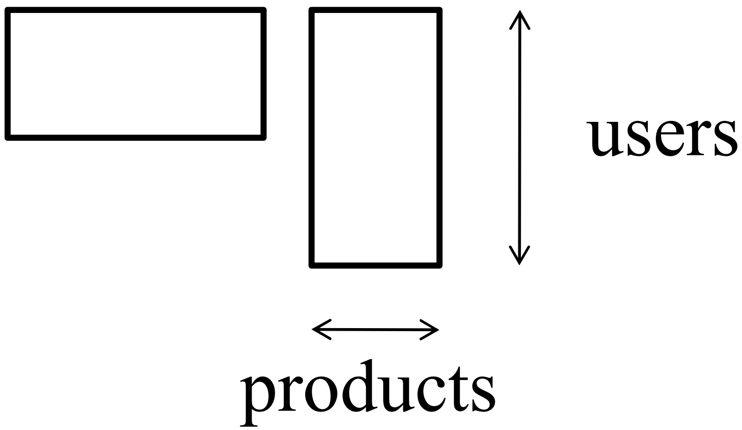
- Intuition:
  - $(\mathbf{A}^T \mathbf{A})\ \mathbf{v}'$
  - $(\mathbf{A}^T \mathbf{A})^k\ \mathbf{v}'$

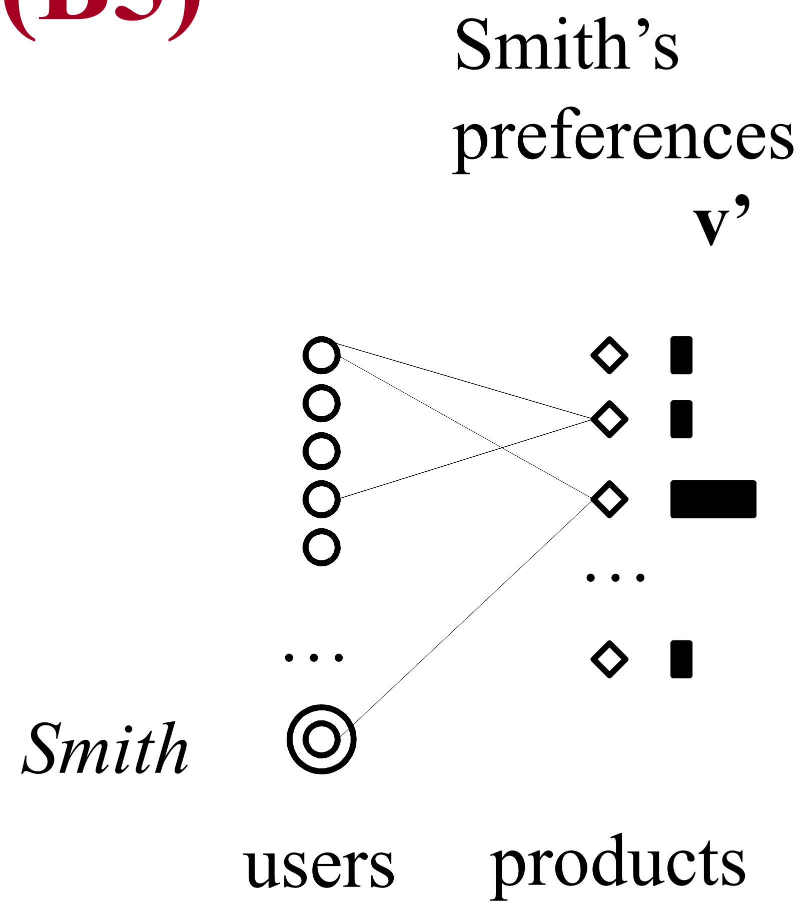




Property (B5)

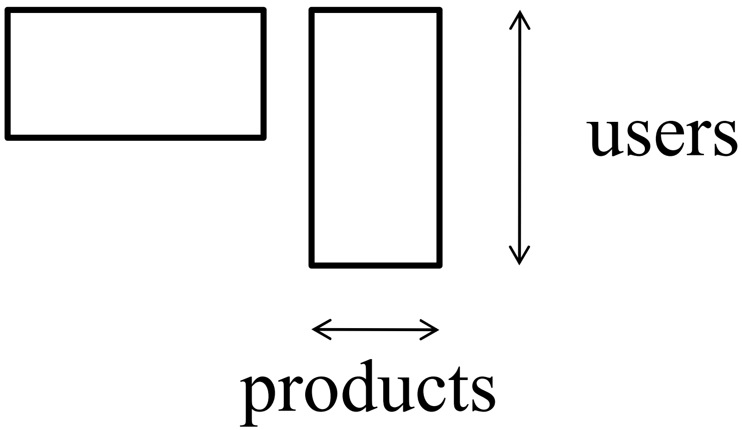
- Intuition:
  - $(\mathbf{A}^T \mathbf{A})\ \mathbf{v'}$
  - $(\mathbf{A}^T \mathbf{A})^k\ \mathbf{v'}$

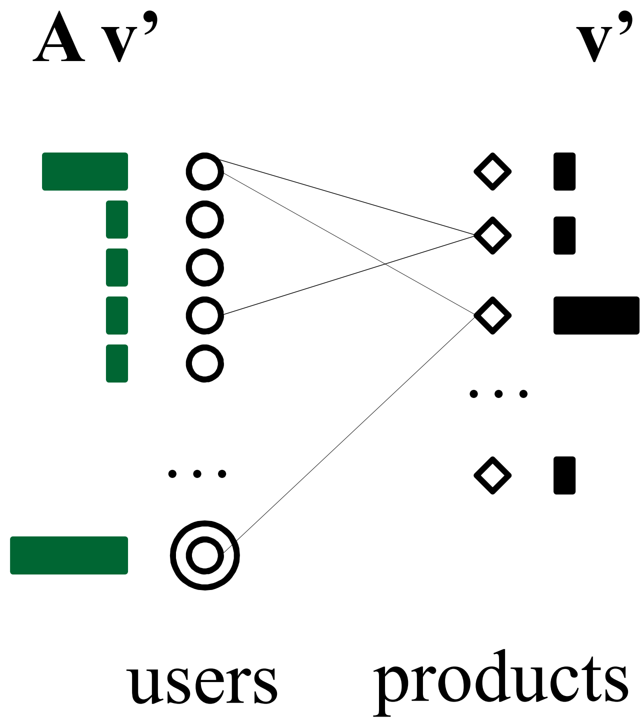




Property (B5)

- Intuition:
  - $(\mathbf{A}^T \mathbf{A})\, \mathbf{v'}$
  - $(\mathbf{A}^T \mathbf{A})^k\, \mathbf{v'}$

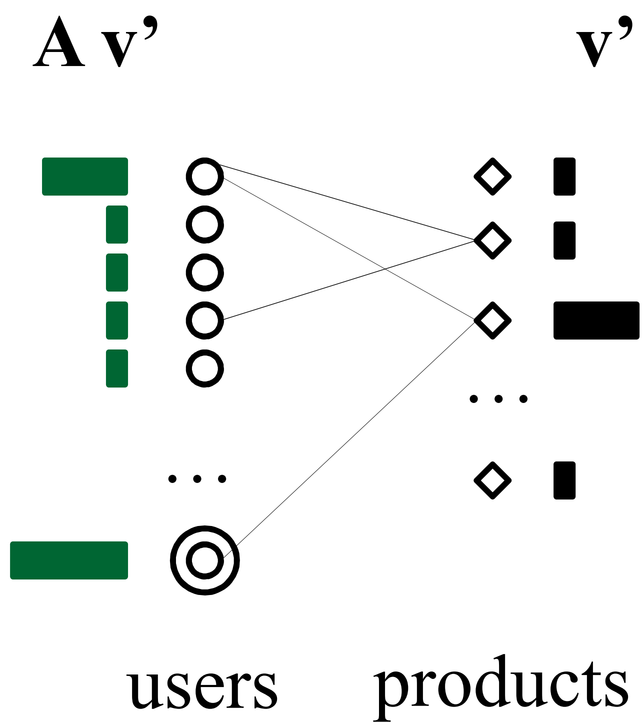


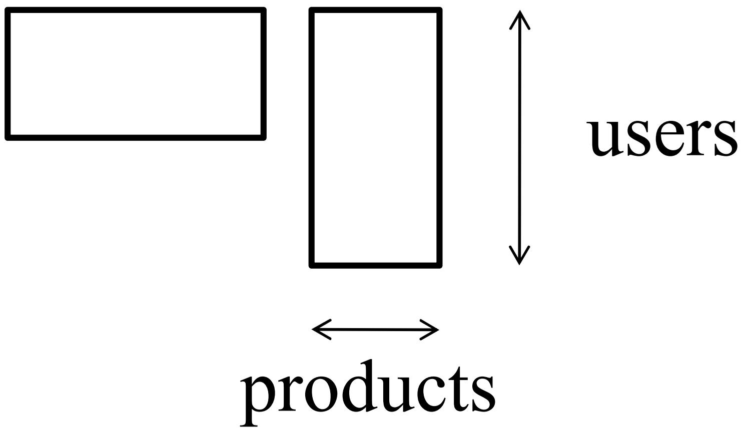


Property (B5)

- Intuition:
  - $(\mathbf{A}^T \mathbf{A})\ \mathbf{v'}$
  - $(\mathbf{A}^T \mathbf{A})^k\ \mathbf{v'}$

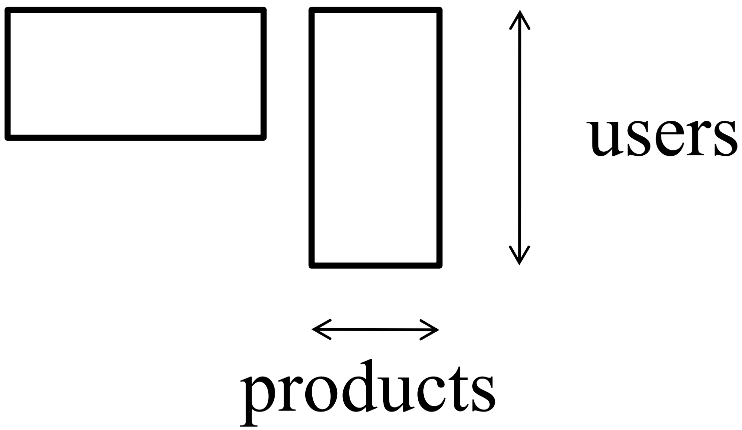
similarities to Smith

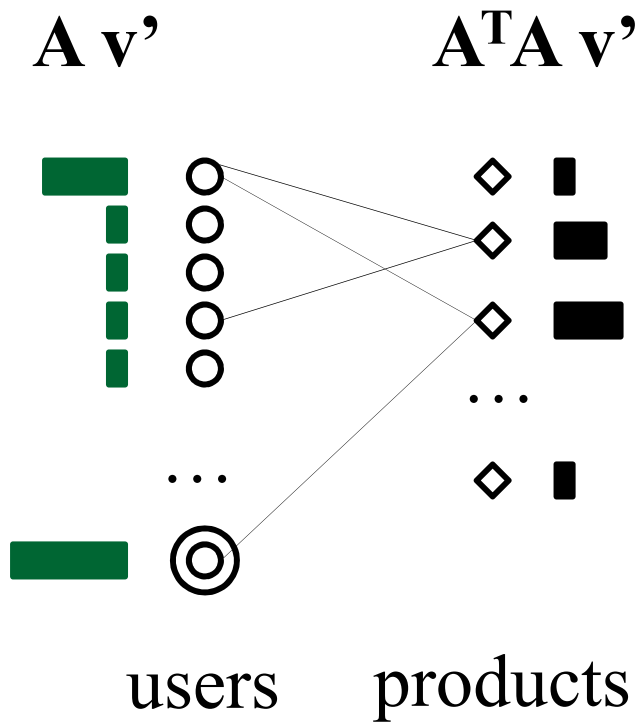




Property (B5)

- Intuition:
  - $(A^T A)\ \mathbf{v}'$
  - $(A^T A)^k\ \mathbf{v}'$





Property (B5)

- Intuition:
  - $(\mathbf{A}^T \mathbf{A})\ \mathbf{v'}$ what Smith's 'friends' like
  - $(\mathbf{A}^T \mathbf{A})^k\ \mathbf{v'}$ what k-step-away-friends like

(ie., after *k* steps, we get what everybody likes, and Smith's initial opinions don't count)

Less obvious properties - repeated:

$$A(0):\ \mathbf{A}_{[n \times m]} = \mathbf{U}_{[\,n \times r\,]}\, \mathbf{\Lambda}_{[\,r \times r\,]}\, \mathbf{V}^{\mathbf{T}}_{[\,r \times m]}$$

$$B(1):\ \mathbf{A}_{[n \times m]}\, (\mathbf{A}^{T})_{[m \times n]} = \mathbf{U}\, \mathbf{\Lambda}^{2}\, \mathbf{U}^{T}$$

$$B(2):\ (\mathbf{A}^{T})_{[m \times n]}\, \mathbf{A}_{[n \times m]} = \mathbf{V}\, \mathbf{\Lambda}^{2}\, \mathbf{V}^{T}$$

$$B(3):\ (\,(\mathbf{A}^{T})_{[m \times n]}\, \mathbf{A}_{[n \times m]}\,)^{k} = \mathbf{V}\, \mathbf{\Lambda}^{2k}\, \mathbf{V}^{T}$$

$$B(4):\ (\mathbf{A}^{T}\mathbf{A}\,)^{k} \sim v_1\, \lambda_1^{2k}\, v_1^{T}$$

$$B(5):\ (\mathbf{A}^{T}\mathbf{A}\,)^{k}\, \mathbf{v'} \sim (\text{constant})\, \mathbf{v}_1$$

Least obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

C(1): $\mathbf{A}_{[n \times m]} \mathbf{x}_{[m \times 1]} = \mathbf{b}_{[n \times 1]}$

let $\mathbf{x}_0 = \mathbf{V} \mathbf{\Lambda}^{(-1)} \mathbf{U}^T \mathbf{b}$

if under-specified, $\mathbf{x}_0$ gives 'shortest' solution

if over-specified, it gives the 'solution' with the smallest least squares error

KAIST Computer Science

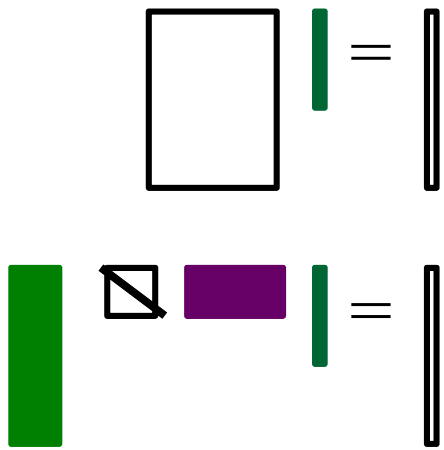
Least obvious properties

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \, \mathbf{\Lambda}_{[r \times r]} \, \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

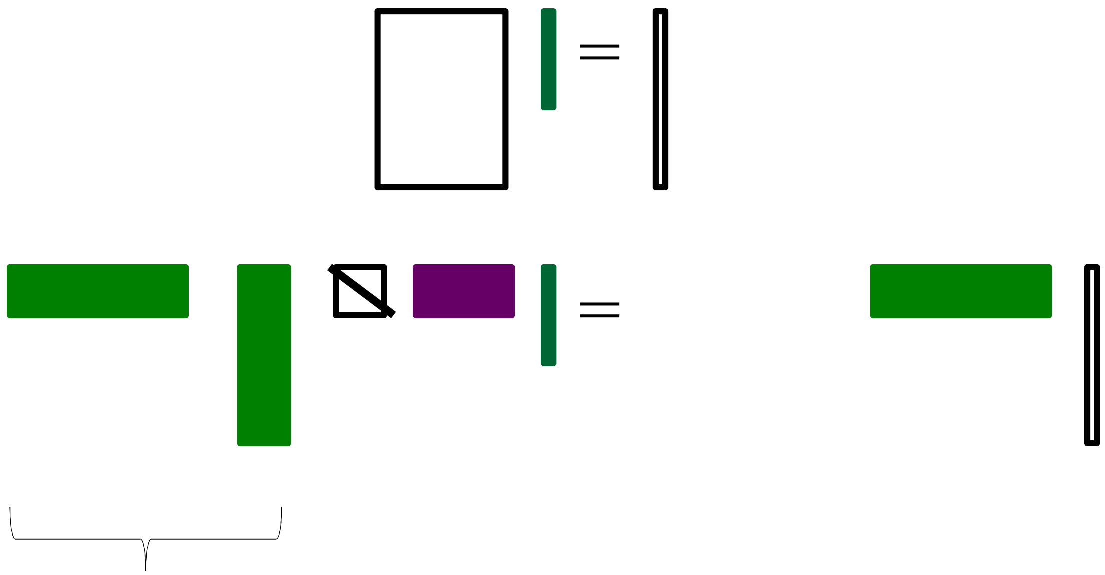
C(1): $\mathbf{A}_{[n \times m]} \, \mathbf{x}_{[m \times 1]} = \mathbf{b}_{[n \times 1]}$

let $\mathbf{x}_0 = \mathbf{V} \, \mathbf{\Lambda}^{(-1)} \, \mathbf{U}^{\mathrm{T}} \, \mathbf{b}$

Slowly:

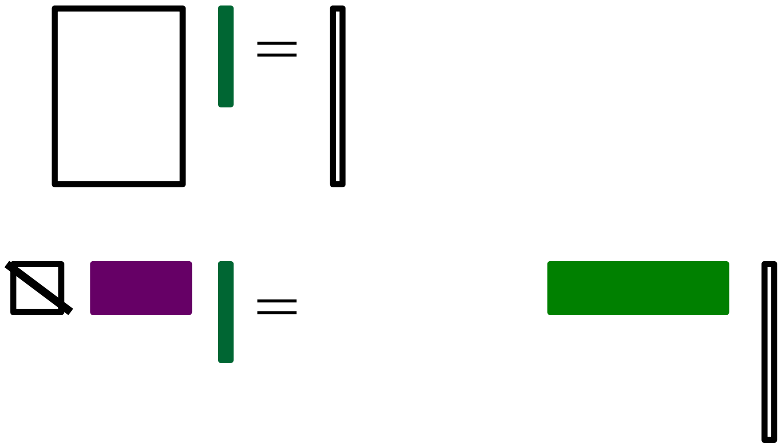


Slowly:

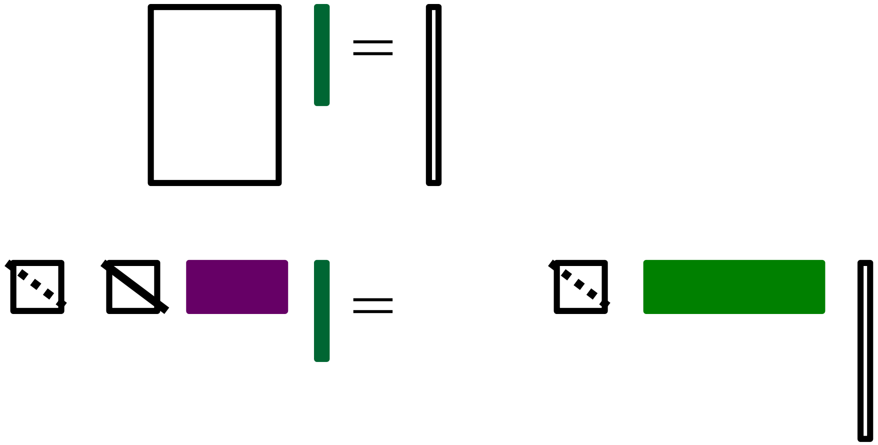


Identity
U: column-orthonormal

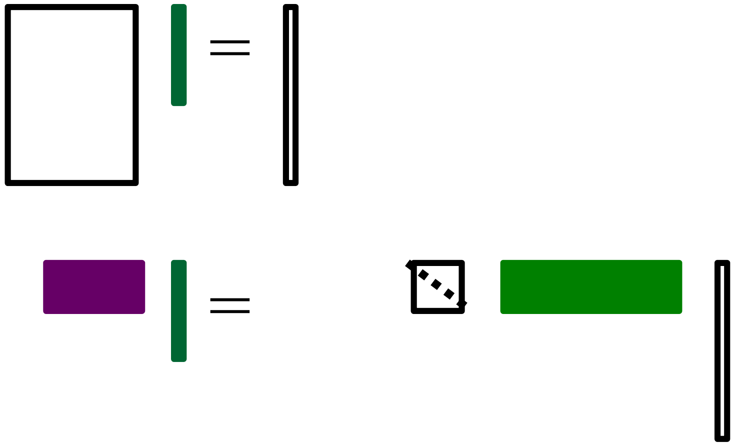
Slowly:



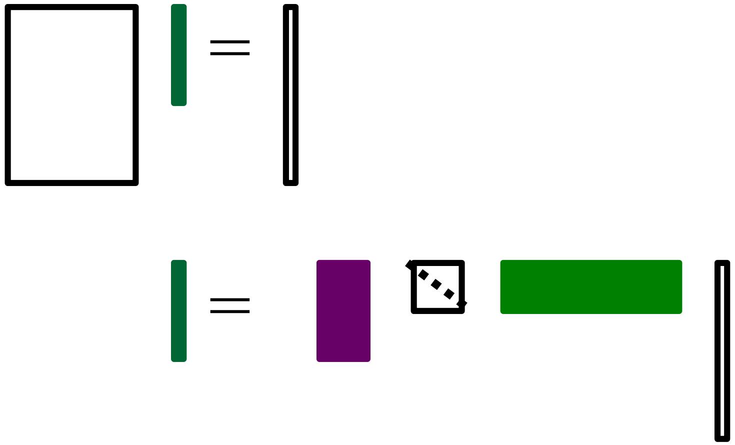
Slowly:



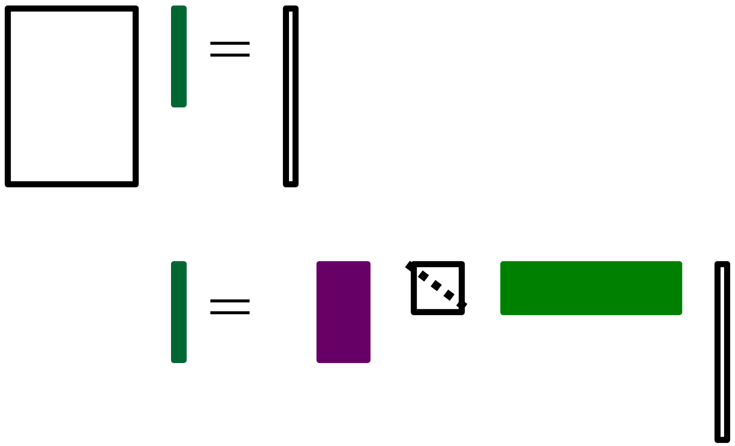
Slowly:



Slowly:



Slowly:

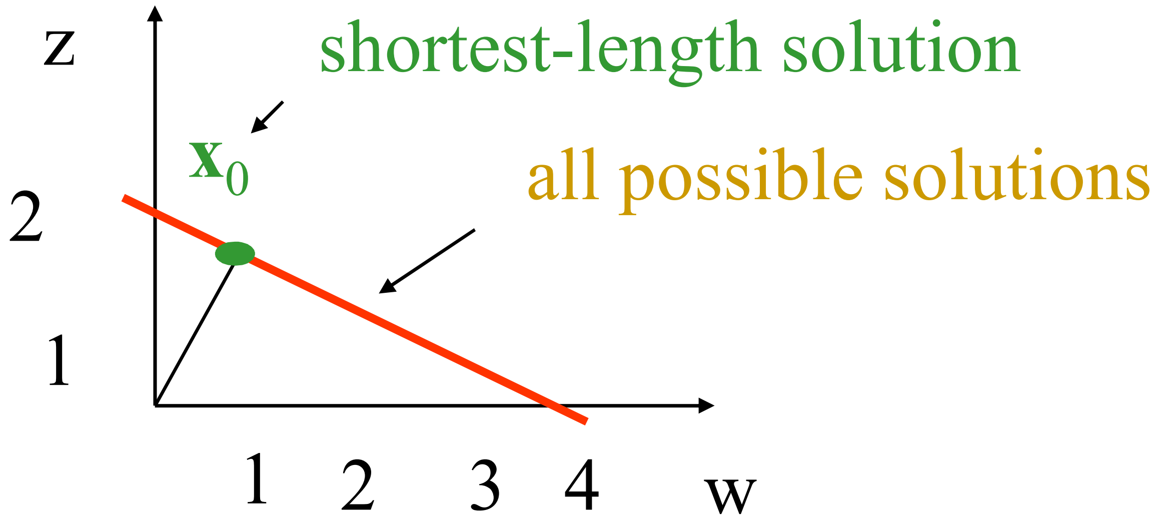


$$\mathbf{x} \quad \mathbf{V} \quad \Lambda^{-1} \quad \mathbf{U}^{T} \quad \mathbf{b}$$

Least obvious properties

Illustration: under-specified, eg

$[1\ 2]\ [w\ z]^T = 4$ (ie, $1\,w + 2\,z = 4$)



Exercise

Verify formula:

$\mathbf{A} = [1\ 2] \qquad \mathbf{b} = [4]$

$\mathbf{A} = \mathbf{U}\ \boldsymbol{\Lambda}\ \mathbf{V}^{T}$

$\mathbf{U} = ??$

$\boldsymbol{\Lambda} = ??$

$\mathbf{V} = ??$

$\mathbf{x_0} = \mathbf{V}\ \boldsymbol{\Lambda}^{(-1)}\ \mathbf{U}^{T}\ \mathbf{b}$

Exercise

Verify formula:

$\mathbf{A} = [1\ 2] \qquad \mathbf{b} = [4]$

$\mathbf{A} = \mathbf{U}\ \boldsymbol{\Lambda}\ \mathbf{V}^T$

$\mathbf{U} = [1]$

$\boldsymbol{\Lambda} = [\ \mathrm{sqrt}(5)\ ]$

$\mathbf{V} = [\ 1/\mathrm{sqrt}(5) \qquad 2/\mathrm{sqrt}(5)\ ]^T$

$\mathbf{x_0} = \mathbf{V}\ \boldsymbol{\Lambda}^{(-1)}\ \mathbf{U}^T\ \mathbf{b}$

Verify formula:

Exercise

$\mathbf{A} = [1\ 2] \quad \mathbf{b} = [4]$

$\mathbf{A} = \mathbf{U}\, \boldsymbol{\Lambda}\, \mathbf{V}^T$

$\mathbf{U} = [1]$

$\Lambda = [\ \mathrm{sqrt}(5)\ ]$

$\mathbf{V} = [\ 1/\mathrm{sqrt}(5) \quad 2/\mathrm{sqrt}(5)\ ]^T$

$\mathbf{x_0} = \mathbf{V}\, \boldsymbol{\Lambda}^{(-1)}\ \mathbf{U}^T\, \mathbf{b} = [\ 1/5 \quad 2/5]^T\ [4]$

$= [4/5 \quad 8/5]^T:\ w = 4/5,\ z = 8/5$

KAIST Computer Science

Exercise

Verify formula:

Show that w= 4/5, z = 8/5 is

(a) A solution to 1*w + 2*z = 4 and

(b) Minimal (wrt Euclidean norm)

Exercise

Verify formula:

Show that w= 4/5, z = 8/5 is

(a) A solution to 1*w + 2*z = 4 and

A: easy

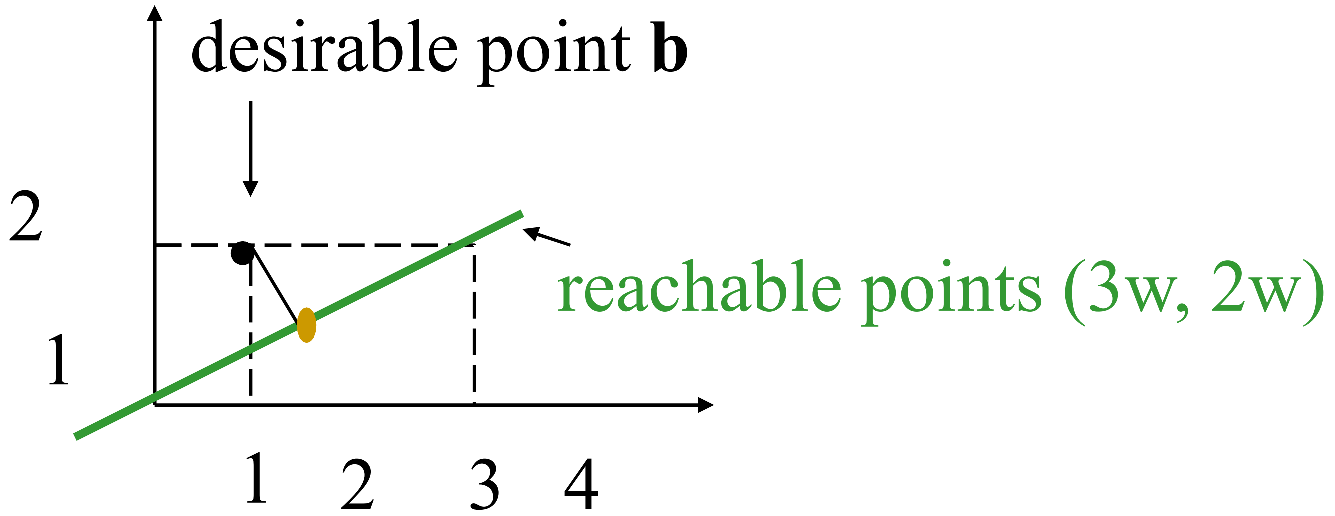
(b) Minimal (wrt Euclidean norm)

A: [4/5 8/5] is perpenticular to [2 -1]

Least obvious properties – cont'd

Illustration: over-specified, eg

$[3\ 2]^T\ [w] = [1\ 2]^T$ (ie, $3\,w = 1$; $2\,w = 2$)



Exercise

Verify formula:

$\mathbf{A} = [3\ 2]^T \qquad \mathbf{b} = [\ 1\ \ 2]^T$

$\mathbf{A} = \mathbf{U}\ \mathbf{\Lambda}\ \mathbf{V}^T$

$\mathbf{U} = ??$

$\mathbf{\Lambda} = ??$

$\mathbf{V} = ??$

$\mathbf{x_0} = \mathbf{V}\ \mathbf{\Lambda}^{(-1)}\ \mathbf{U}^T\ \mathbf{b}$

Exercise

Verify formula:

$\mathbf{A} = [3\ 2]^T \qquad \mathbf{b} = [\ 1\ \ 2]^T$

$\mathbf{A} = \mathbf{U}\ \mathbf{\Lambda}\ \mathbf{V}^T$

$\mathbf{U} = [\ 3/\mathrm{sqrt}(13) \quad 2/\mathrm{sqrt}(13)\ ]^T$

$\mathbf{\Lambda} = [\ \mathrm{sqrt}(13)\ ]$

$\mathbf{V} = [\ 1\ ]$

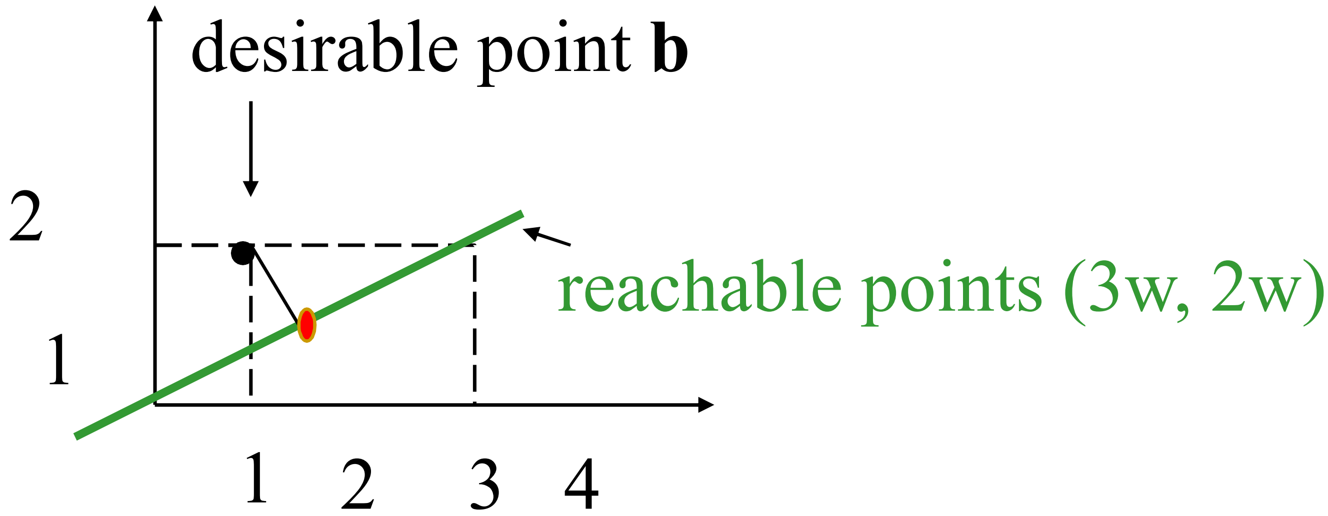
$\mathbf{x_0} = \mathbf{V}\ \mathbf{\Lambda}^{(-1)}\ \mathbf{U}^T\ \mathbf{b} = [\ 7/13\ ]$

Exercise

Verify formula:

$[3\ 2]^T\ \ [7/13] = [1\ 2]^T$

$[21/13\ \ 14/13\ ]^T$ -> 'red point'

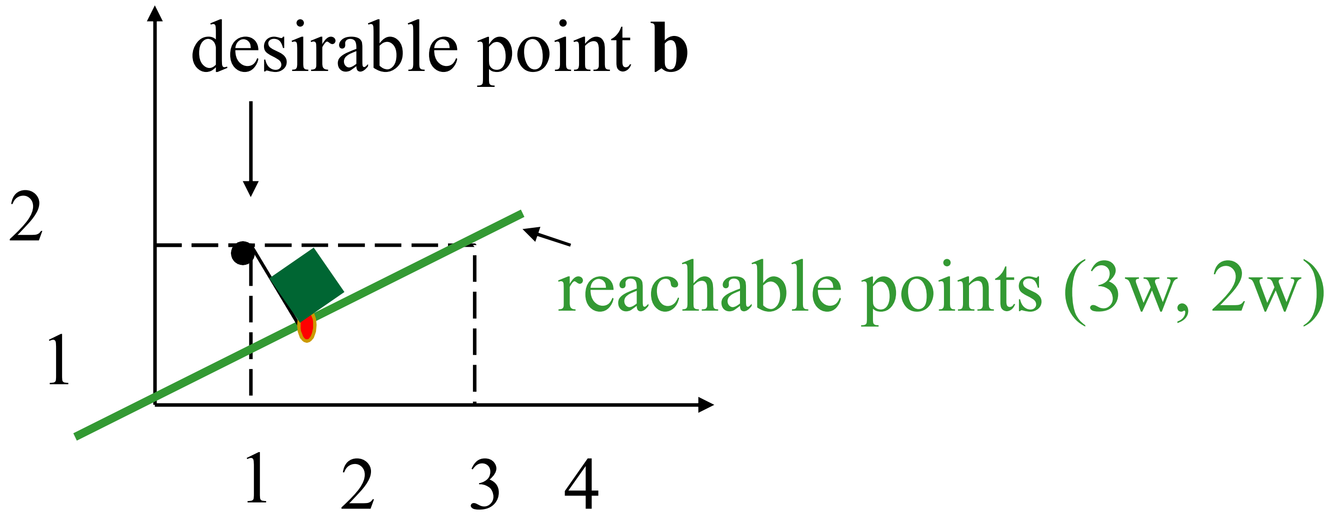


Exercise

Verify formula:

$[3\ 2]^T\ \ [7/13] = [1\ 2]^T$

$[21/13\ \ 14/13\ ]^T$ -> 'red point' - perpenticular?

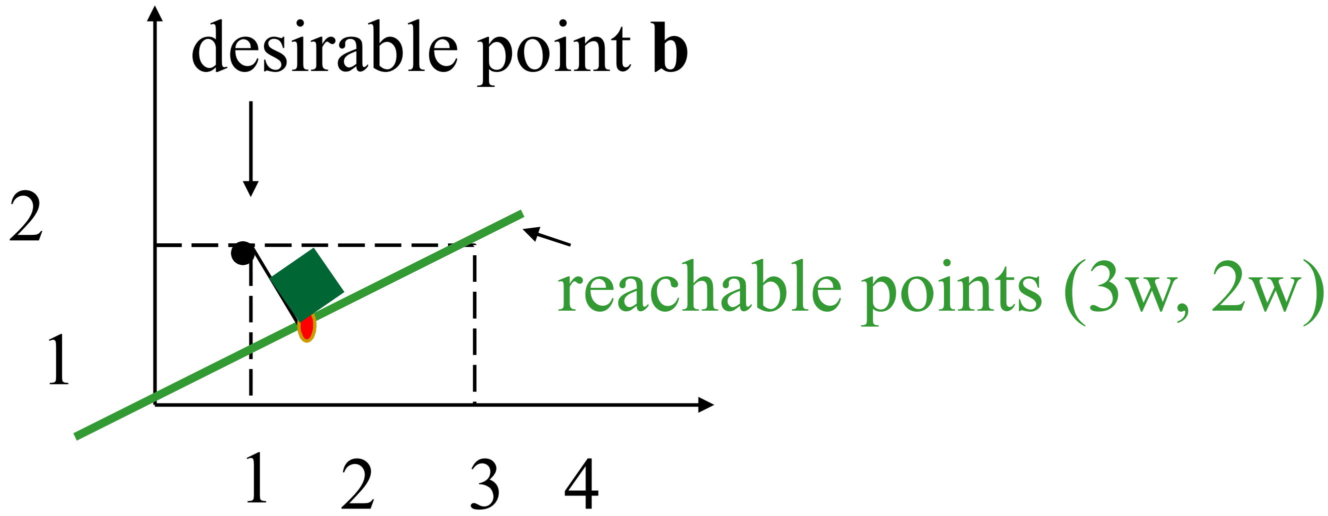


Verify formula:

Exercise

A: [3 2] . ([1 2] – [21/13   14/13]) =

[3 2] . [-8/13   12/13] = [3 2] . [-2   3] = 0



Least obvious properties - cont'd

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

C(2): $\mathbf{A}_{[n \times m]} \mathbf{v}_{\mathbf{1}\,[m \times 1]} = \lambda_{\mathbf{1}} \mathbf{u}_{\mathbf{1}\,[n \times 1]}$

where $\mathbf{v_1}$, $\mathbf{u_1}$ the first (column) vectors of **V**, **U**. ($\mathbf{v_1}$ == right-singular-vector)

C(3): symmetrically: $\mathbf{u_1^T A} = \lambda_1 \mathbf{v_1^T}$

$\mathbf{u_1}$ == left-singular-vector

Therefore:

Least obvious properties - cont'd

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

$\boxed{\text{C(4): } \mathbf{A}^{\mathbf{T}} \mathbf{A}\, \mathbf{v}_1 = \lambda_1^{2}\, \mathbf{v}_1}$

(**fixed point** - the dfn of eigenvector for a symmetric matrix)

Least obvious properties - altogeth er

A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$

C(1): $\mathbf{A}_{[n \times m]} \mathbf{x}_{[m \times 1]} = \mathbf{b}_{[n \times 1]}$

then, $\mathbf{x}_0 = \mathbf{V} \mathbf{\Lambda}^{(-1)} \mathbf{U}^{T} \mathbf{b}$**:** shortest, actual or least-squa res solution

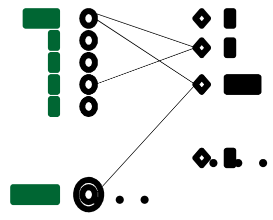
C(2): $\mathbf{A}_{[n \times m]} \mathbf{v}_{\mathbf{1}\,[m \times 1]} = \lambda_{\mathbf{1}} \mathbf{u}_{\mathbf{1}\,[n \times 1]}$

C(3): $\mathbf{u}_1^{\mathbf{T}} \mathbf{A} = \lambda_1 \mathbf{v}_1^{\mathbf{T}}$

C(4): $\mathbf{A}^{\mathbf{T}} \mathbf{A} \mathbf{v}_1 = \lambda_1^2 \mathbf{v}_1$

Properties - conclusions

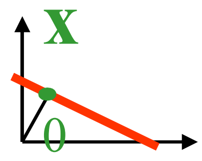
A(0): $\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \mathbf{\Lambda}_{[r \times r]} \mathbf{V}^{\mathbf{T}}_{[r \times m]}$



B(5): $(\mathbf{A}^{T} \mathbf{A})^{k} \mathbf{v'} \sim (\text{constant})\ \mathbf{v}_{1}$

C(1): $\mathbf{A}_{[n \times m]} \mathbf{x}_{[m \times 1]} = \mathbf{b}_{[n \times 1]}$

then, $\mathbf{x}_0 = \mathbf{V} \mathbf{\Lambda}^{(-1)} \mathbf{U}^{T} \mathbf{b}$**:** shortest, actual or least-squares solution



C(4): $\mathbf{A}^{\mathbf{T}} \mathbf{A}\, \mathbf{v}_{\mathbf{1}} = \lambda_{\mathbf{1}}^{\mathbf{2}}\, \mathbf{v}_{\mathbf{1}}$

Outline

- ☑ SVD Properties
- ☐ **Query feedback**
- ☐ Conclusion

Query feedbacks

[Chen & Roussopoulos, sigmod 94]

Sample problem:

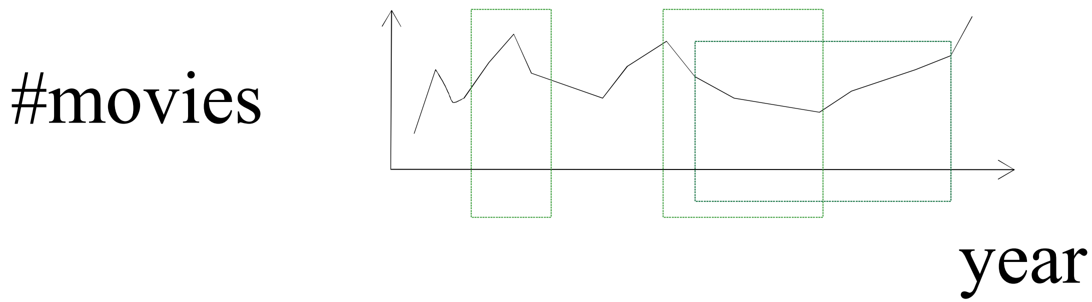
estimate selectivities (e.g., '*how many movies were made between 1940 and 1945*?')

for query optimization,

LEARNING from the query results so far!!

Query feedbacks

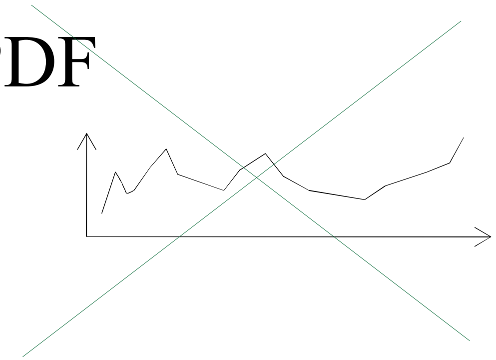
- Given: past queries and their results
  - #movies(1925,1935) = 52
  - #movies(1948, 1990) = 123
  - …
  - And a new query, say #movies(1979,1980)?
- Give your best estimate



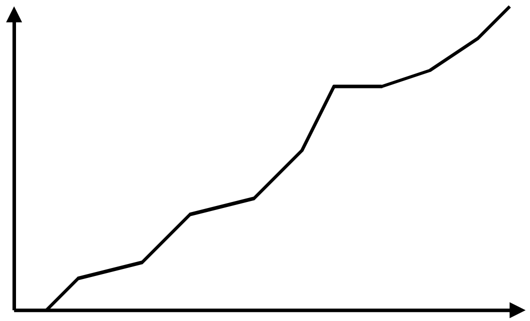
Query feedbacks

Idea #1: consider a function for the CDF (cummulat ive distr. function), eg., 6-th degree polynomial (o r splines, or anything else)

PDF



count, so far



year

Query feedbacks

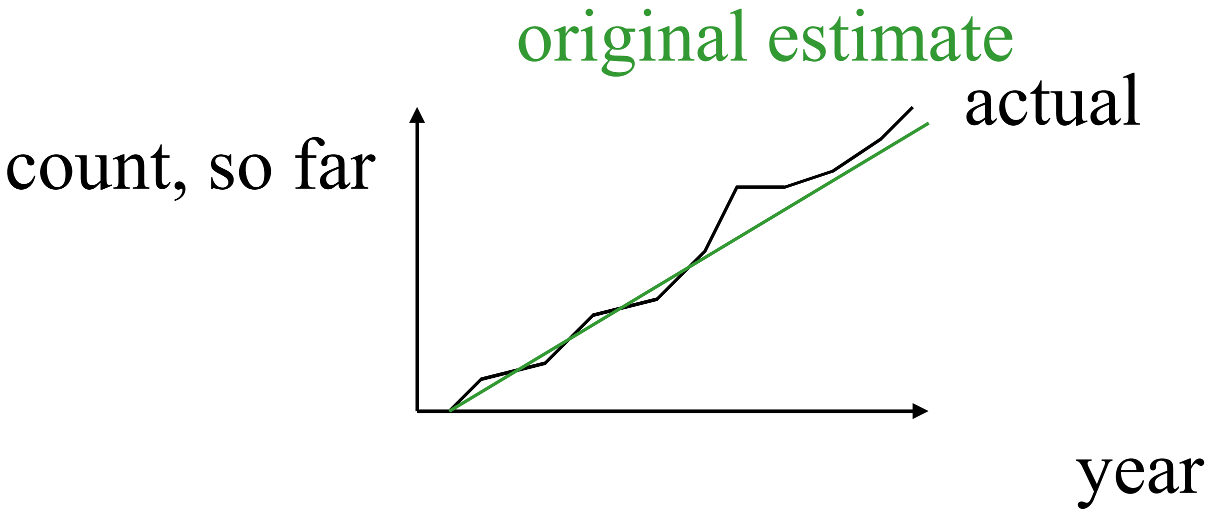
For example

F(x) = # movies made until year 'x'

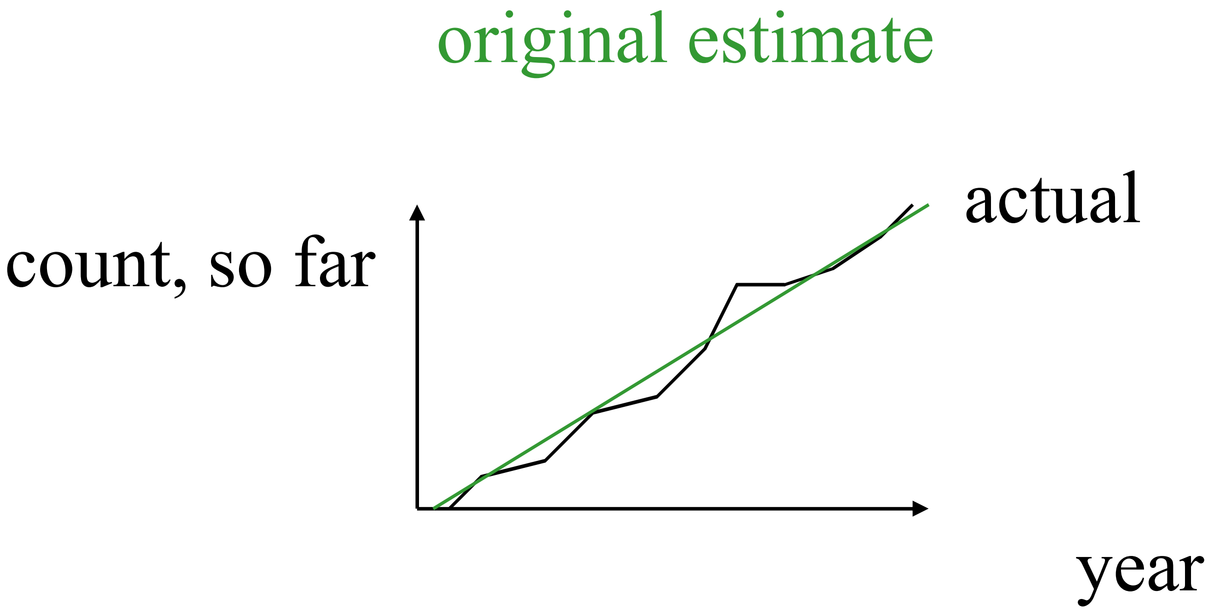
$$= a_1 + a_2 * x + a_3 * x^2 + \dots a_7 * x^6$$

Query feedbacks

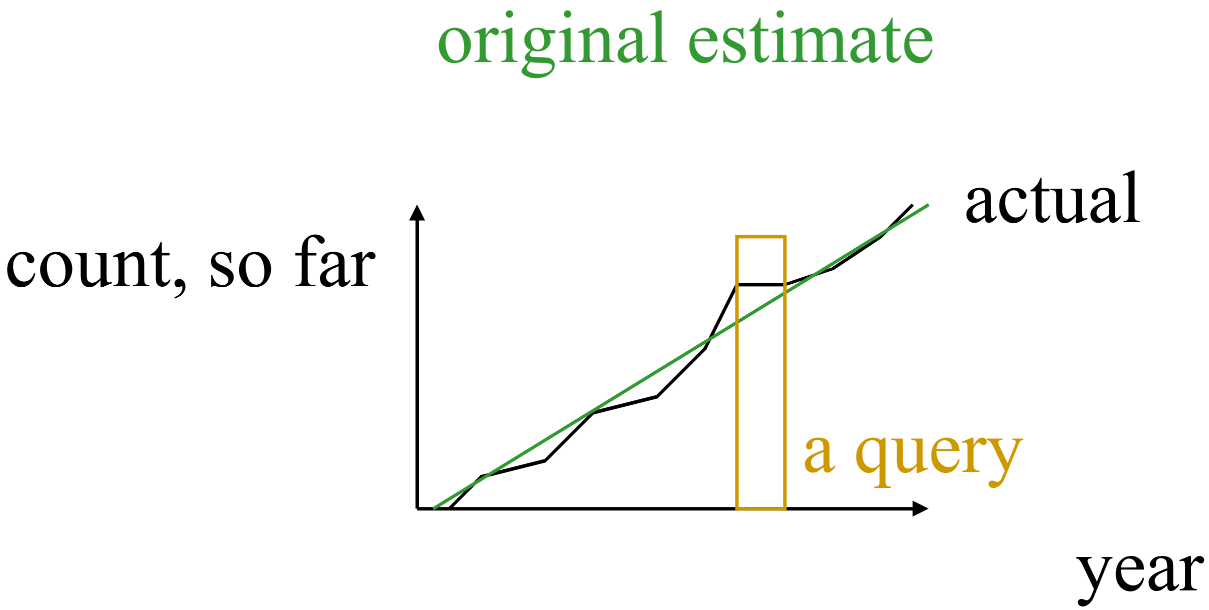
GREAT idea #2: adapt your model, as you see the a ctual counts of the actual queries



Query feedbacks

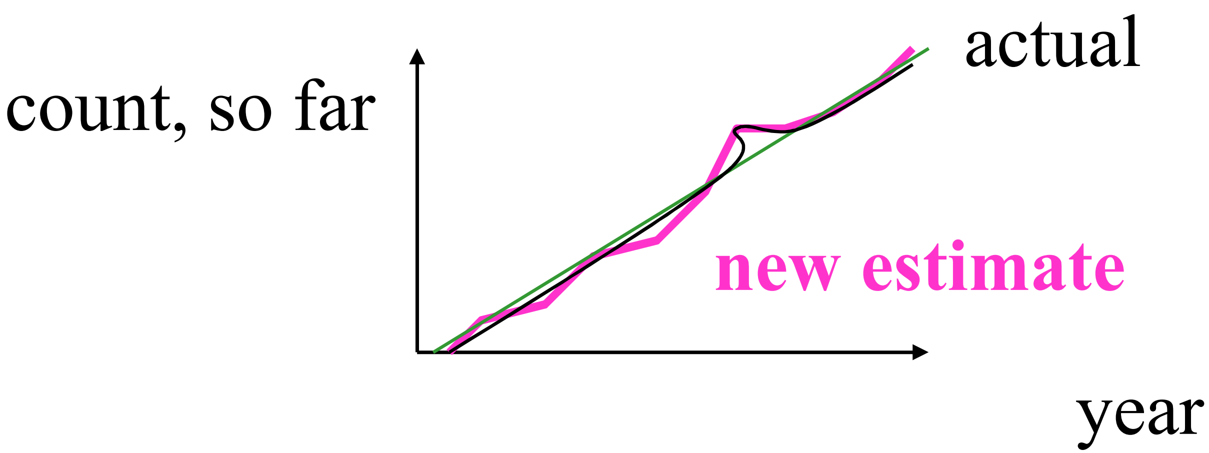


Query feedbacks



Query feedbacks

original estimate



Query feedbacks

Eventually, the problem becomes:

- estimate the parameters a_1, ... a_7 of the model
- to minimize the least squares errors from the real a nswers so far.

Formally:

Query feedbacks

Formally, with *n* queries and 6-th degree polynomials:

$$
\begin{bmatrix}
X11 & X12 & & & X17 \\
 & & & & \\
 & & & & \\
 & & & & \\
Xn1 & Xn2 & & & Xn7
\end{bmatrix}
\begin{bmatrix}
a1 \\ a2 \\ \\ \\ a7
\end{bmatrix}
=
\begin{bmatrix}
b1 \\ b2 \\ \\ \\ bn
\end{bmatrix}
$$

KAIST Computer Science

Query feedbacks

where $x_{i,j}$ such that Sum ($x_{i,j}$ * a_i) = our estimate for the # of movies and b_j: the actual

$$
\begin{bmatrix}
X11 & X12 & & & X17 \\
 & & & & \\
 & & & & \\
 & & & & \\
Xn1 & Xn2 & & & Xn7
\end{bmatrix}
\begin{bmatrix}
a1 \\ a2 \\ \\ \\ a7
\end{bmatrix}
=
\begin{bmatrix}
b1 \\ b2 \\ \\ \\ bn
\end{bmatrix}
$$

Query feedbacks

For example, for query '*find the count of movies during (1920-1932)*':

$a_1 + a_2 * 1932 + a_3 * 1932^2 + \ldots$

-

$(a_1 + a_2 * 1920 + a_3 * 1920^2 + \ldots)$

$$\begin{bmatrix} X11 & X12 & & & X17 \\ & & & & \\ & & & & \\ & & & & \\ Xn1 & Xn2 & & & Xn7 \end{bmatrix} \begin{bmatrix} a1 \\ a2 \\ \\ \\ a7 \end{bmatrix} = \begin{bmatrix} b1 \\ b2 \\ \\ \\ bn \end{bmatrix}$$

Query feedbacks

In matrix form:

$$\mathbf{X} \quad \mathbf{a} = \mathbf{b}$$

$$\begin{array}{l} \text{1st query} \\ \\ \\ \\ \text{n-th query} \end{array} \begin{bmatrix} X11 & X12 & & & X17 \\ & & & & \\ & & & & \\ & & & & \\ Xn1 & Xn2 & & & Xn7 \end{bmatrix} \begin{bmatrix} a1 \\ a2 \\ \\ \\ a7 \end{bmatrix} = \begin{bmatrix} b1 \\ b2 \\ \\ \\ bn \end{bmatrix}$$

Query feedbacks

In matrix form:

$$\mathbf{X}\ \mathbf{a} = \mathbf{b}$$

and the least-squares estimate for **a** is

$$\mathbf{a} = \mathbf{V}\ \mathbf{\Lambda}^{(-1)}\ \mathbf{U}^{\mathbf{T}}\ \mathbf{b}$$

according to property C(1)

(let $\mathbf{X} = \mathbf{U}\ \mathbf{\Lambda}\ \mathbf{V}^{\mathbf{T}}$)

Query feedbacks - enhancements

The solution

$$\mathbf{a} = \mathbf{V}\ \Lambda^{(-1)}\ \mathbf{U}^{\mathbf{T}}\ \mathbf{b}$$

works, but needs expensive SVD each time a new query arrives

GREAT Idea #3: Use 'Recursive Least Squares', to adapt **a** incrementally.

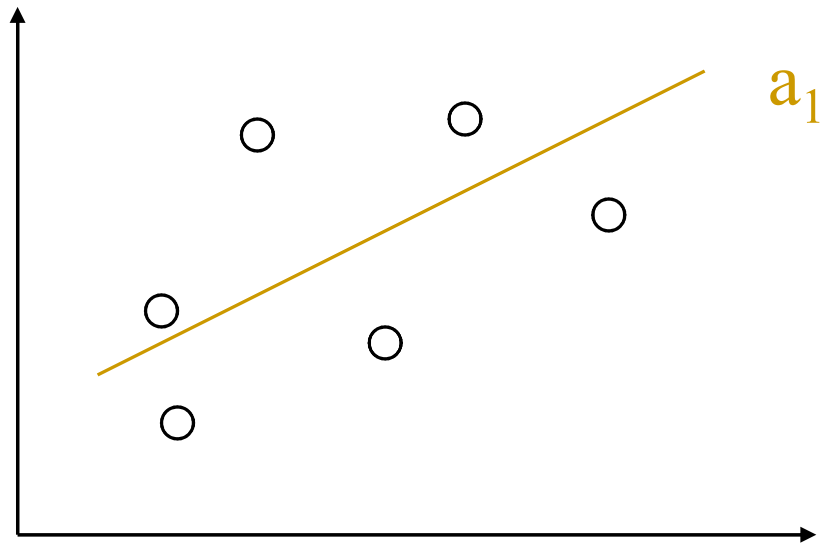
Details: in paper - intuition:

Query feedbacks - enhancements

Intuition:

least squares fit

b

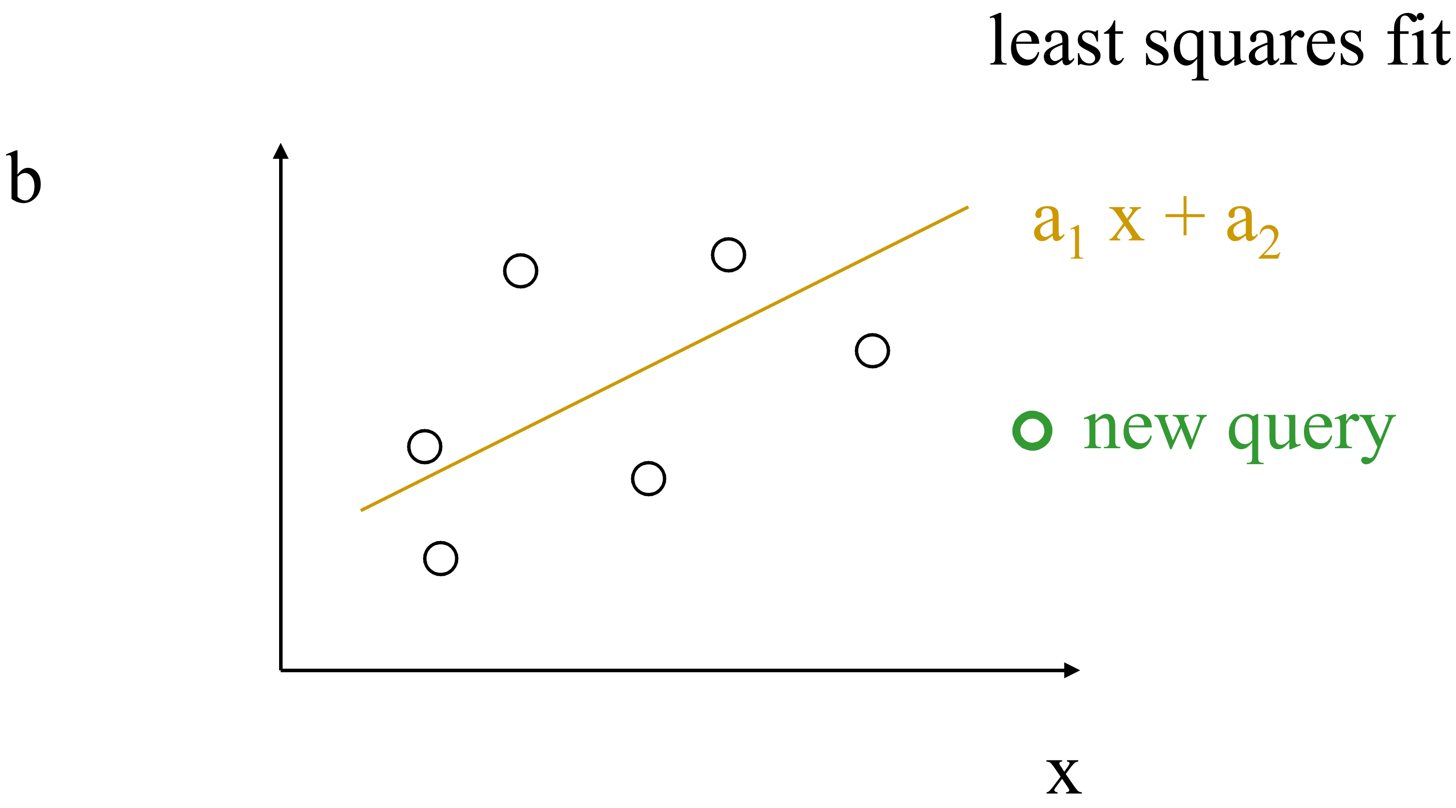


$a_1 x + a_2$

x

Query feedbacks - enhancements

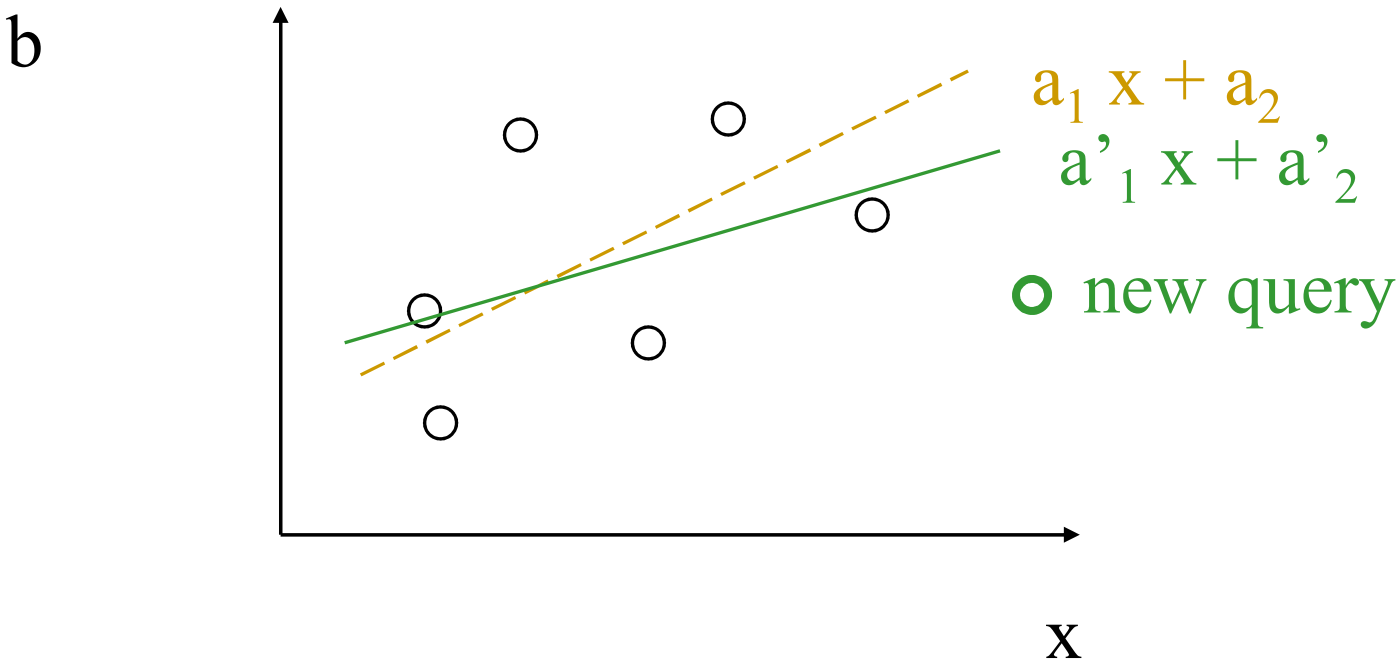
Intuition:



Query feedbacks - enhancements

Intuition:

least squares fit



Query feedbacks - enhancements

the new coefficients can be quickly com puted from the old ones, plus statistics in a (7x7) matrix

(no need to know the details, although th e RLS is a brilliant method)

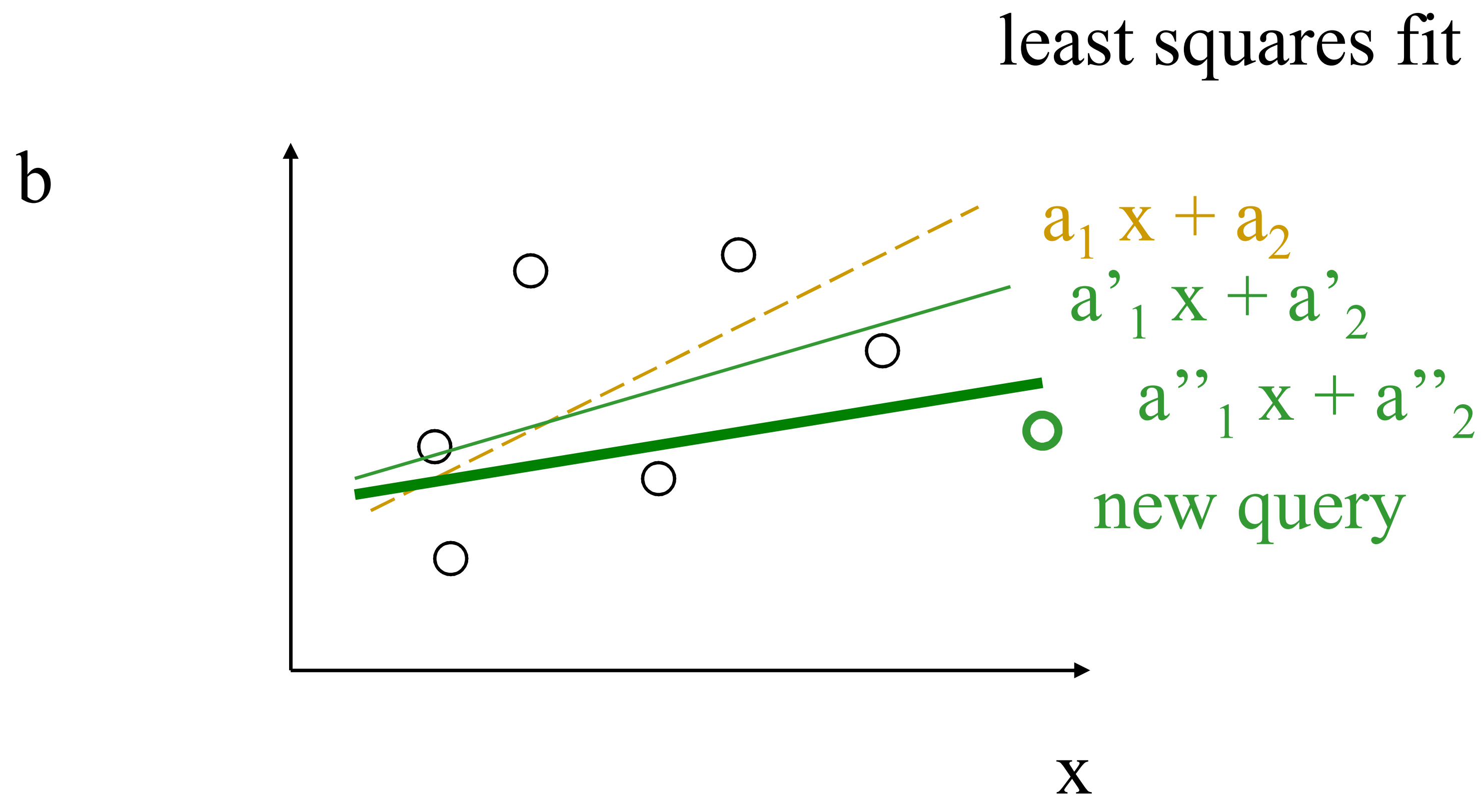
Query feedbacks - enhancements

GREAT idea #4: 'forgetting' factor - we can even down-play the weight of olde r queries, since the data distribution mi ght have changed.

(comes for 'free' with RLS...)

Query feedbacks - enhancements

Intuition:



Query feedbacks - conclusions

SVD helps find the Least Squares solution, to adapt to query feedbacks

(RLS = Recursive Least Squares is a great method to incrementally update least-squares fits)

Outline

- ☑ SVD Properties
- ☑ Query feedback
- ☐ **Conclusion**

Conclusions

- SVD: a **valuable** tool
- given a document-term matrix, it finds ‘concepts’ (LSI)
- ... and can reduce dimensionality (KL)
- ... and can find rules (PCA; RatioRules)

Conclusions cont’d

- ... and can find fixed-points or steady-state probabilities (google/ Kleinberg/ Markov Chains)
- ... and can solve optimally over- and under-constraint linear systems (least squares / query feedbacks)

References

- Chen, C. M. and N. Roussopoulos (May 1994). Adaptive Selectivity Estimation Using Query Feedback. Proc. of the ACM-SIGMOD , Minneapolis, MN.