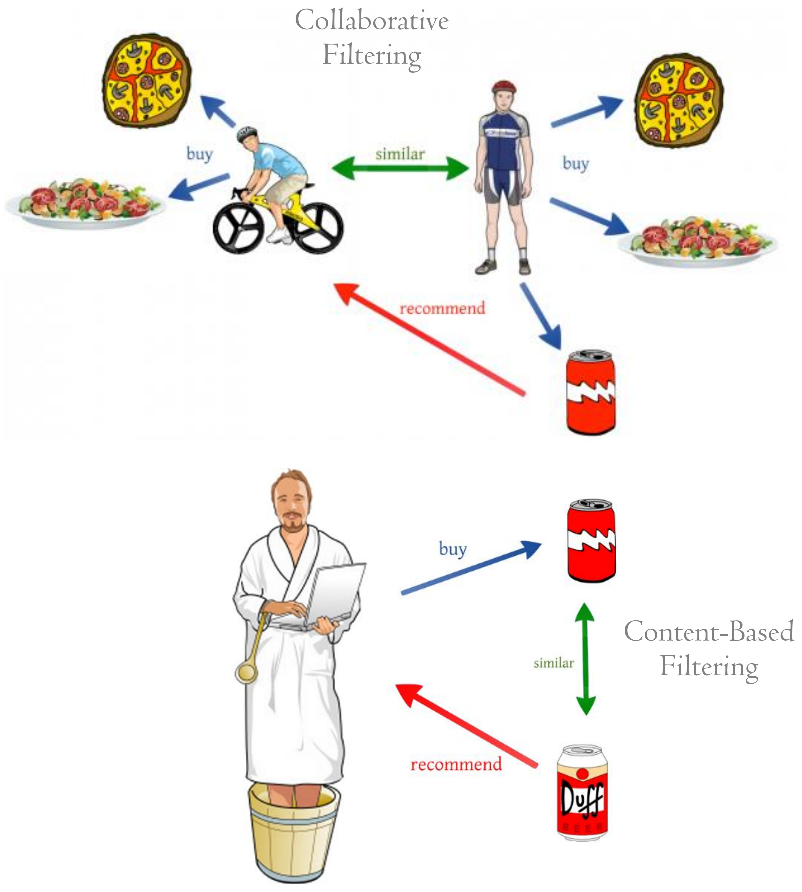
Day 5

凸最適化の応用
Practical Convex Optimization

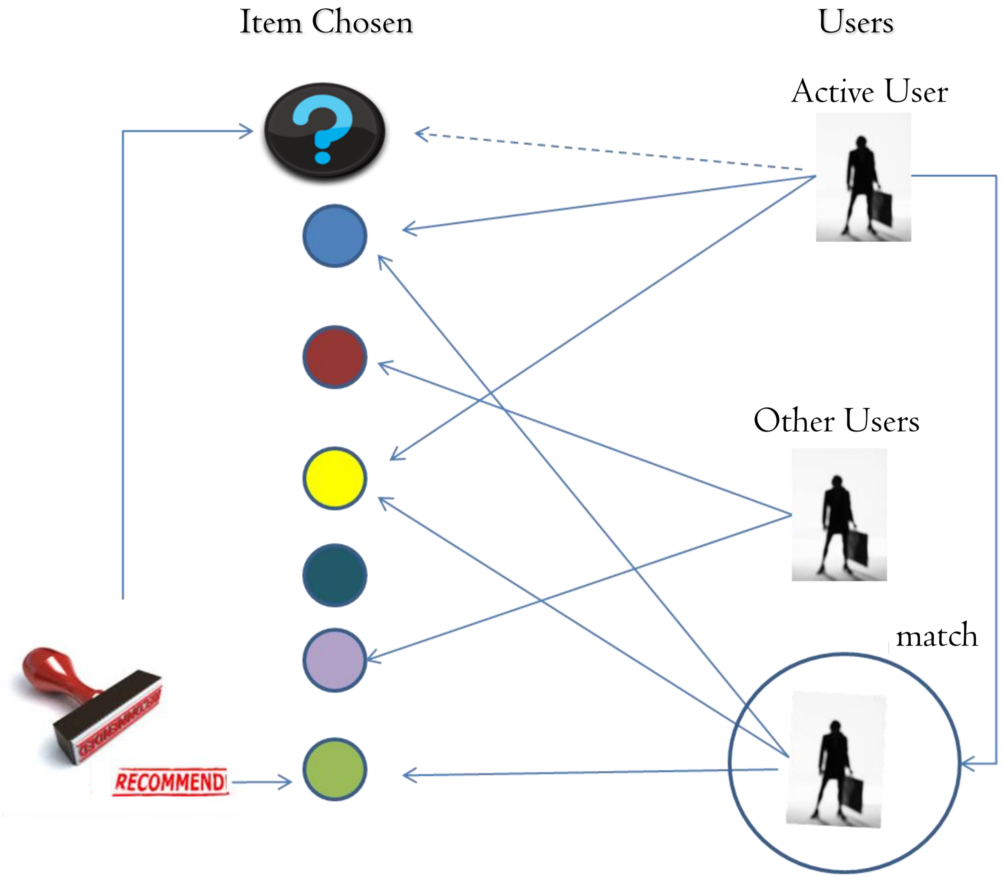
推薦システム

Recommender Systems

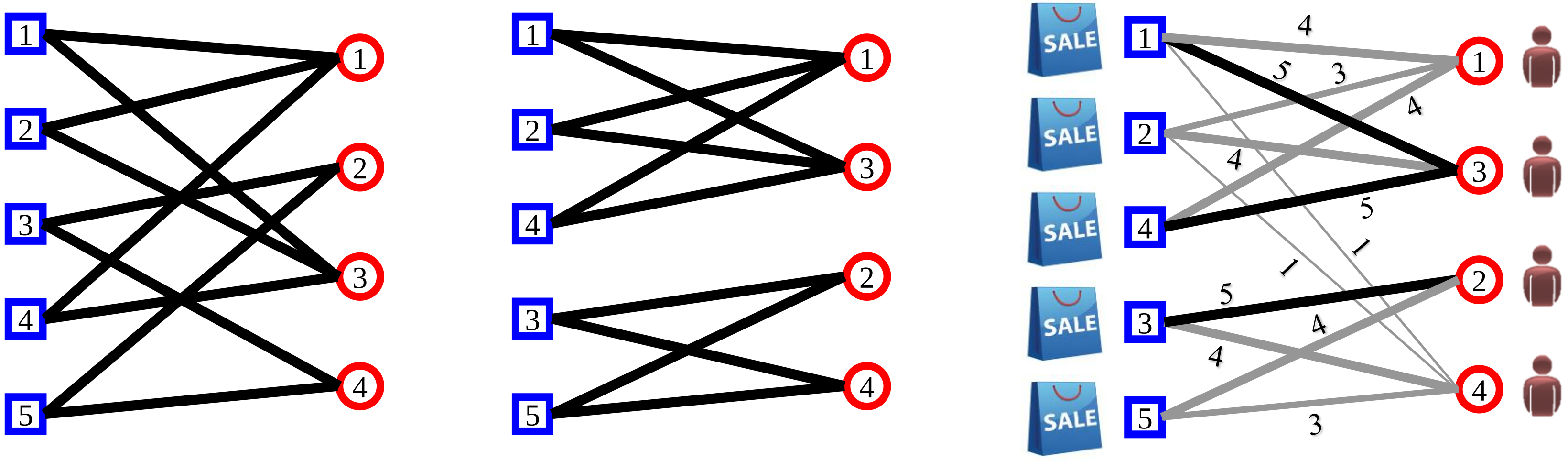
Collaborative filtering and content-based filtering



An Introduction to Recommendation Engines
http://dataconomy.com/an-introduction-to-recommendation-engines/



Bipartite graph representation and clustering



e.g.,

- products vs customers
- keywords vs web pages

n

m

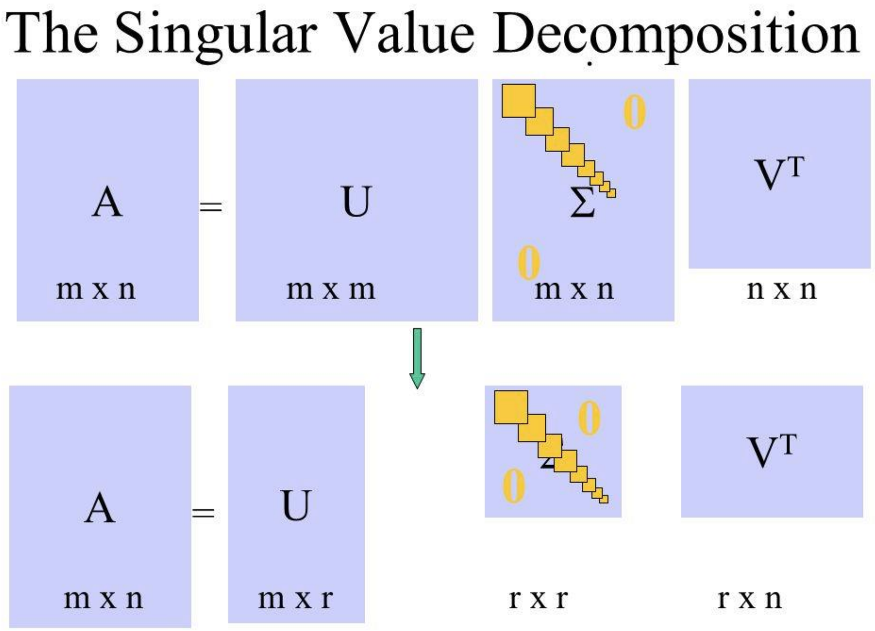
☆	C1	C2	C3	C4	⋯
P1	4	0	5	1	⋯
P2	3	0	4	1	⋯
P3	0	5	0	4	⋯
P4	4	0	5	0	⋯
P5	0	4	0	3	⋯
⋮	⋮	⋮	⋮	⋮	⋱

要点１：表の行(または列)には線形従属性がある.

【数学に翻訳すると】

表は階数が小さい行列（低ランク行列）

- 特異値分解で低ランク行列の内訳がわかる.
- 特異ベクトルで商品と顧客を同時に分類できる.


The Singular Value Decomposition

Singular value decomposition: sum of rank-one matrices

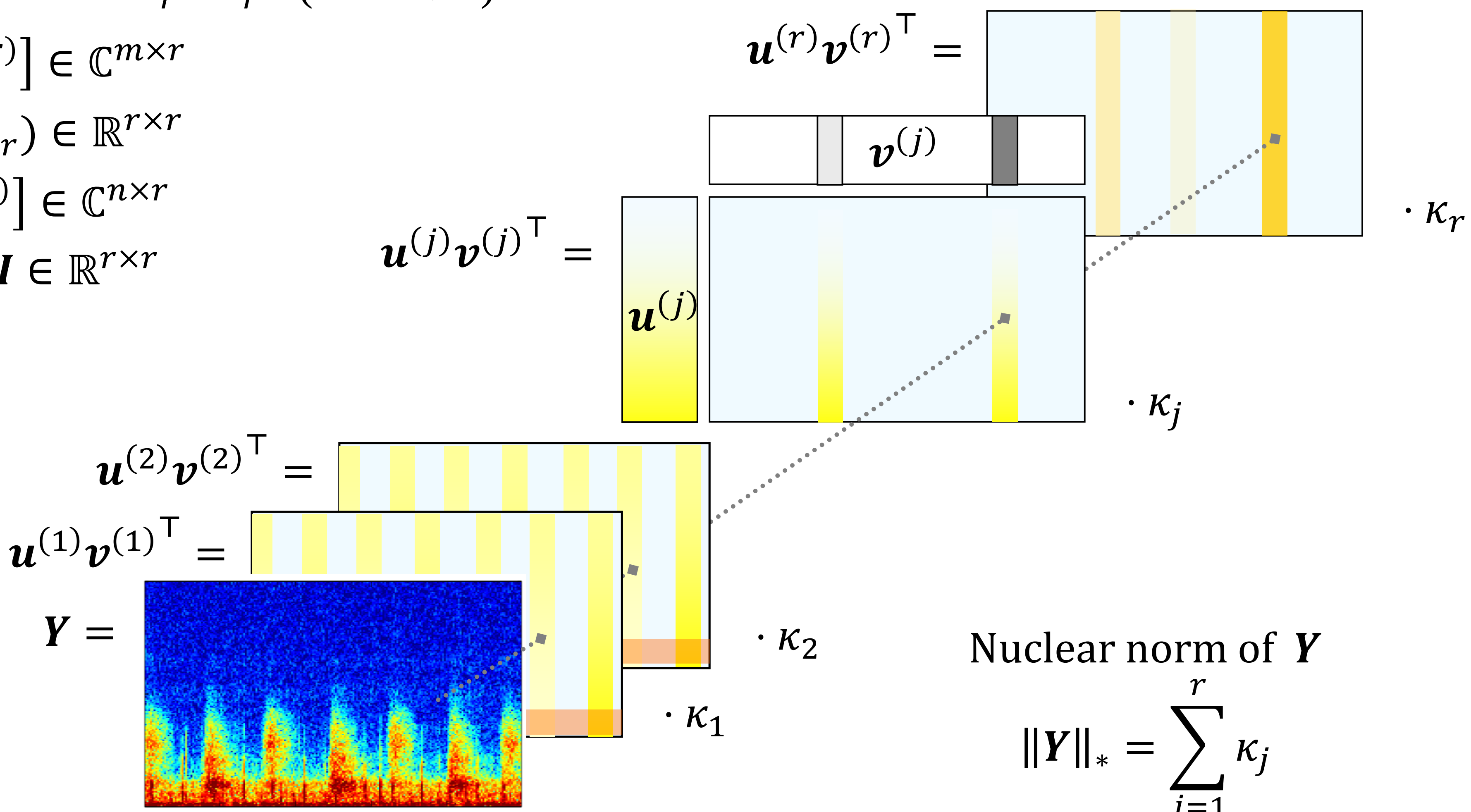
$$\boldsymbol{Y} = \sum_{j=1}^{r} \kappa_j \, \boldsymbol{u}^{(j)} \boldsymbol{v}^{(j)\top} = \boldsymbol{U}_r \boldsymbol{K} \boldsymbol{V}_r^\top \quad \text{(thin SVD)}$$

$$\boldsymbol{U}_r = \left[\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(r)}\right] \in \mathbb{C}^{m \times r}$$

$$\boldsymbol{K} = \mathrm{diag}(\kappa_1, \dots, \kappa_r) \in \mathbb{R}^{r \times r}$$

$$\boldsymbol{V}_r = \left[\boldsymbol{v}^{(1)}, \dots, \boldsymbol{v}^{(r)}\right] \in \mathbb{C}^{n \times r}$$

$$\boldsymbol{U}_r^\top \boldsymbol{U}_r = \boldsymbol{V}_r^\top \boldsymbol{V}_r = \boldsymbol{I} \in \mathbb{R}^{r \times r}$$



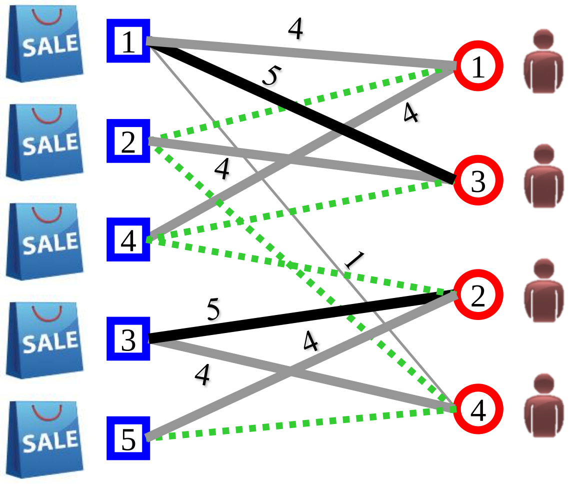
$\boldsymbol{u}^{(r)} \boldsymbol{v}^{(r)\top} =$
$\boldsymbol{v}^{(j)}$
$\cdot \kappa_r$
$\boldsymbol{u}^{(j)} \boldsymbol{v}^{(j)\top} =$
$\boldsymbol{u}^{(j)}$
$\cdot \kappa_j$
$\boldsymbol{u}^{(2)} \boldsymbol{v}^{(2)\top} =$
$\boldsymbol{u}^{(1)} \boldsymbol{v}^{(1)\top} =$
$\boldsymbol{Y} =$
$\cdot \kappa_2$
$\cdot \kappa_1$

Nuclear norm of $\boldsymbol{Y}$

$$\|\boldsymbol{Y}\|_* = \sum_{j=1}^{r} \kappa_j$$

In practice, we are given incomplete data

- ***Y*** has many missing entries
  they are indicated by 0 in ***R***.
- Prior knowledge: ***Y*** is redundant;
  Y consists of linearly dependent rows/columns,
  i.e., ***Y*** is low-rank.
- Estimate the missing entries, and
  recommend the products according to the estimates.



	C1	C2	C3	C4
P1	1	1	1	1
P2	0	1	1	0
P3	1	1	1	1
P4	1	0	0	1
P5	1	1	1	0

$= \boldsymbol{R}$

★	C1	C2	C3	C4
P1	4	0	5	1
P2	?	0	4	?
P3	0	5	0	4
P4	4	?	?	0
P5	0	4	0	?

$= \boldsymbol{Y}$

要点２：商品vs顧客の表の穴を埋める必要がある.

【数学に翻訳すると】

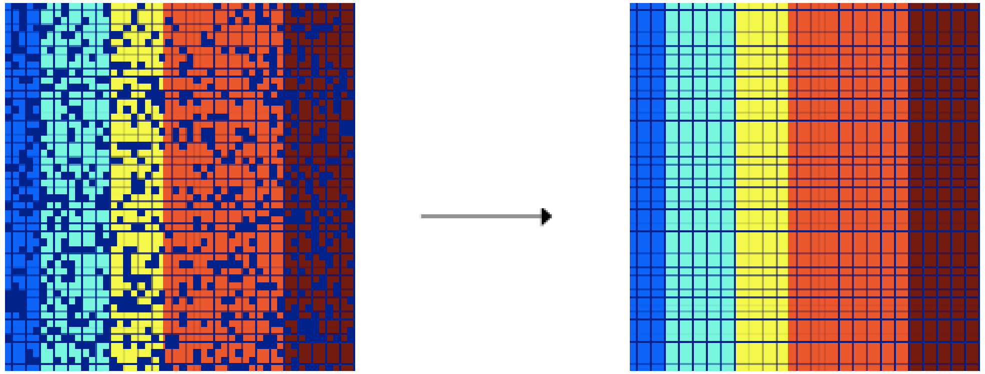
既知の値と辻褄が合うような

低ランク行列を求める問題を解く.

推定した行列について,

- 表の既知の値との誤差
- 行列の階数

を共に最小化する問題

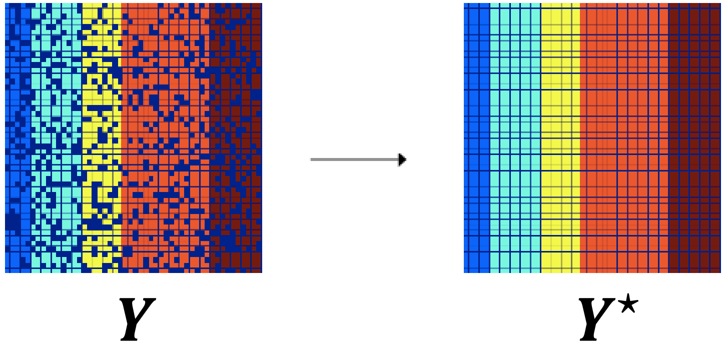


Low-rank matrix completion

$$\boldsymbol{Y}^{\star} = \arg\min_{\boldsymbol{A}} \frac{1}{2}\|\boldsymbol{R} * (\boldsymbol{A} - \boldsymbol{Y})\|_F^2 + w\|\boldsymbol{A}\|_*$$

Errors on known entries

Convex relaxation of matrix rank



- $\boldsymbol{Y}$ products vs customers table
- $\boldsymbol{R}$ $R_{ij} = 1$ if Y_{ij} is known, $R_{ij} = 0$ otherwise.
- $\boldsymbol{R} * (\boldsymbol{A} - \boldsymbol{Y})$ matrix of errors masked by $\boldsymbol{R}$. "*" indicates element-by-element multiplication.
- $\|\boldsymbol{R} * (\boldsymbol{A} - \boldsymbol{Y})\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(R_{ij}\left(A_{ij} - Y_{ij}\right)\right)^2$ amount of errors.

  $\|\cdot\|_F$ denotes the Frobenius norm.
- $\|\boldsymbol{A}\|_*$ nuclear norm of $\boldsymbol{A}$. Its minimization induces low-rankness of $\boldsymbol{A}$.
- w balancing parameter between the error and low-rank regularization.

ADMM

Alternating direction method of multipliers [Gabay&Mercier, 76]

$$\underset{(\boldsymbol{x},\boldsymbol{z})}{\text{Minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{z}) \quad \text{subject to} \quad \boldsymbol{z} = \boldsymbol{G}\boldsymbol{x}$$

$$\boldsymbol{x}^{(k+1)} \coloneqq \underset{\boldsymbol{x}}{\arg\min}\, f(\boldsymbol{x}) + \frac{\rho}{2}\left\|\boldsymbol{G}\boldsymbol{x} - (\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)})\right\|_2^2$$

$$\boldsymbol{z}^{(k+1)} \coloneqq \underset{\boldsymbol{z}}{\arg\min}\, g(\boldsymbol{z}) + \frac{\rho}{2}\left\|\boldsymbol{z} - (\boldsymbol{G}\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)})\right\|_2^2$$

$$\boldsymbol{u}^{(k+1)} \coloneqq \boldsymbol{u}^{(k)} + \boldsymbol{G}\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k+1)}$$

Low-rank matrix completion by ADMM

$$\boldsymbol{Y}^{\star} = \arg\min_{\boldsymbol{A}} \frac{1}{2} \|\boldsymbol{R} * (\boldsymbol{A} - \boldsymbol{Y})\|_F^2 + w\|\boldsymbol{A}\|_*$$

- Let $\boldsymbol{x} = \operatorname{vec} \boldsymbol{A}$ and let $\boldsymbol{M}$ be a linear mapping that extracts the known values from $\boldsymbol{Y}$ as $\boldsymbol{y} = \boldsymbol{M} \operatorname{vec} \boldsymbol{Y}$. The above equation becomes

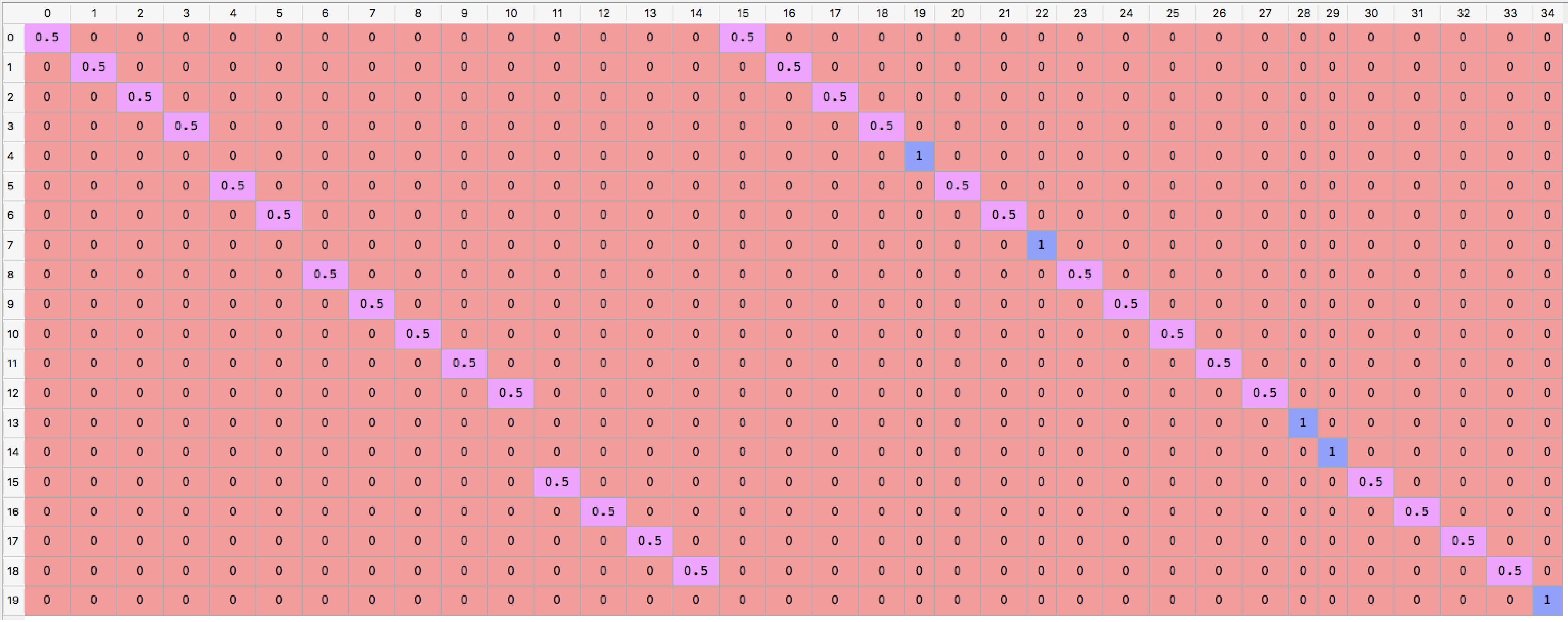
$$(\boldsymbol{y}^{\star}, \boldsymbol{z}^{\star}) = \operatorname*{argmin}_{(\boldsymbol{x},\boldsymbol{z})} \ \frac{1}{2}\|\boldsymbol{z}_1 - \boldsymbol{y}\|_2^2 + w\|\operatorname{mat} \boldsymbol{z}_2\|_* \quad \text{subject to} \quad \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{x}.$$

- ADMM: $(\boldsymbol{x}^{\star}, \boldsymbol{z}^{\star}) = \arg\min_{(\boldsymbol{x},\boldsymbol{z})} \ f(\boldsymbol{x}) + g(\boldsymbol{z})$ subject to $\boldsymbol{z} \ = \ \boldsymbol{G} \ \boldsymbol{x}$
  - $\boldsymbol{x}^{(k+1)} \coloneqq \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{\rho}{2}\left\|\boldsymbol{G}\boldsymbol{x} - \left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right)\right\|_2^2 = (\boldsymbol{G}^{\top}\boldsymbol{G})^{-1}\, \boldsymbol{G}^{\top}\left(\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}\right)$
  - $\boldsymbol{z}^{(k+1)} \coloneqq \arg\min_{\boldsymbol{z}} g(\boldsymbol{z}) + \frac{\rho}{2}\left\|\boldsymbol{z} - (\boldsymbol{G}\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)})\right\|_2^2$
    - $\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \end{bmatrix} = \boldsymbol{G}\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)}$, $\quad \begin{cases} \boldsymbol{z}_1^{(k+1)} = \operatorname{prox}_{\frac{1}{2\rho}\|\cdot - \boldsymbol{y}\|_2^2}(\boldsymbol{q}_1) = (\rho\boldsymbol{q}_1 + \boldsymbol{y})/(\rho + 1) \\ \boldsymbol{z}_2^{(k+1)} = \operatorname{prox}_{\frac{w}{\rho}\|\operatorname{mat}\cdot\|_*}(\boldsymbol{q}_2) = \operatorname{svt}(\operatorname{mat} \boldsymbol{q}_2, w/\rho) \end{cases}$
  - $\boldsymbol{u}^{(k+1)} \coloneqq \boldsymbol{u}^{(k)} + \boldsymbol{G}\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k+1)}$

$$\boxed{(\boldsymbol{G}^\top \boldsymbol{G})^{-1}\ \boldsymbol{G}^\top}$$

```
R = R == 1
onesR = np.ones(m*n)
onesR[R.ravel()] = 0.5
pinvG =sp.hstack((0.5*sp.eye(m*n,format='csr')[R.ravel()].T,
                  sp.diags(onesR,format='csr')))
```

Python numpy array
(row-major order, $\boldsymbol{y} = \mathrm{vec}(\boldsymbol{Y}^\top)$)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

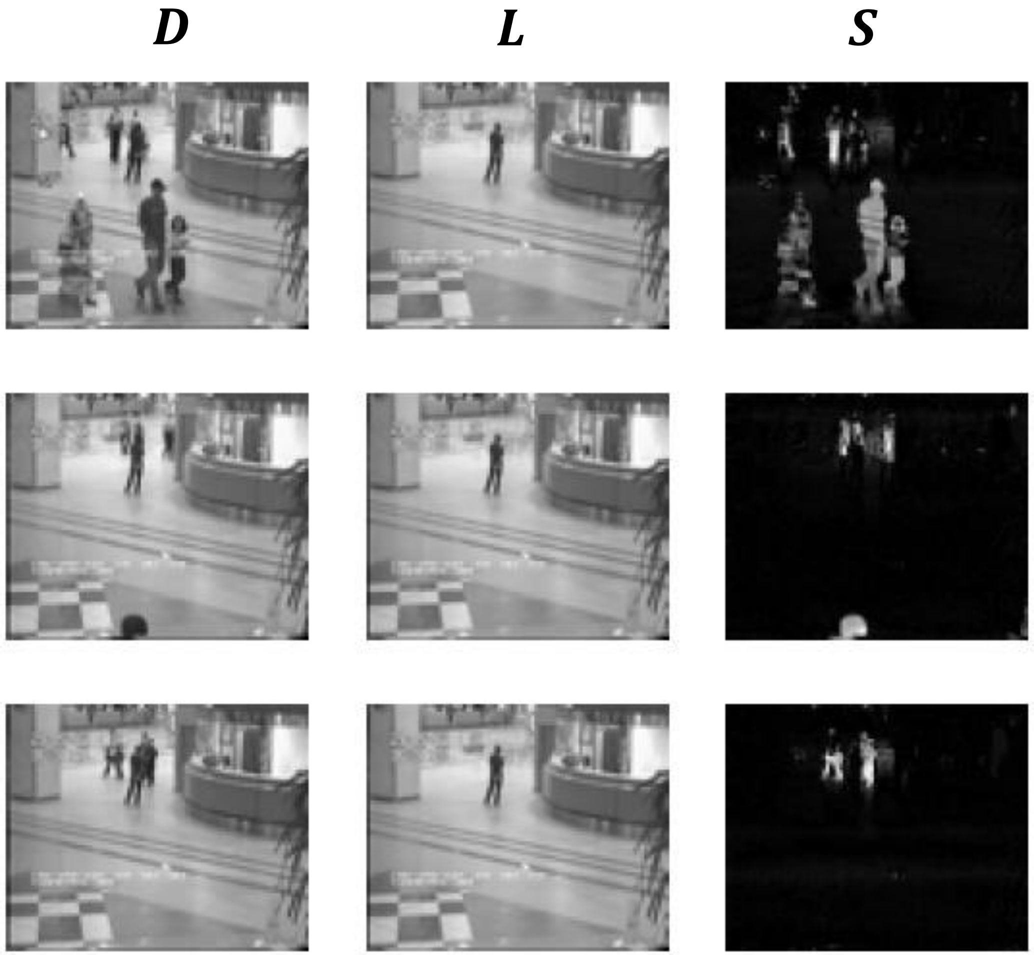
凸最適化の応用
Practical Convex Optimization

ロバスト主成分分析

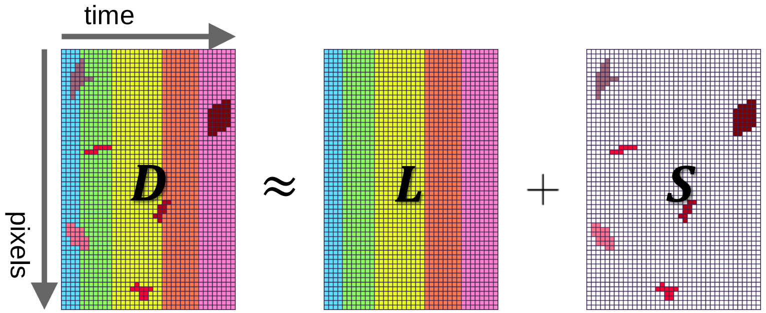
Robust Principal Component Analysis (RPCA)

Background subtraction

[Waters+, 11]

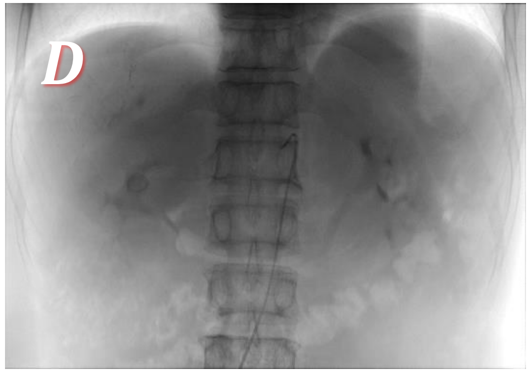


"SpaRCS: Recovering low-rank and sparse matrices from compressive measurements."



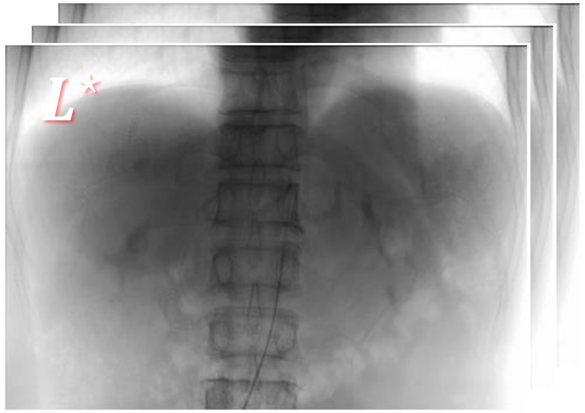
Digital angiography w/o breath holding

[Kokura+17]



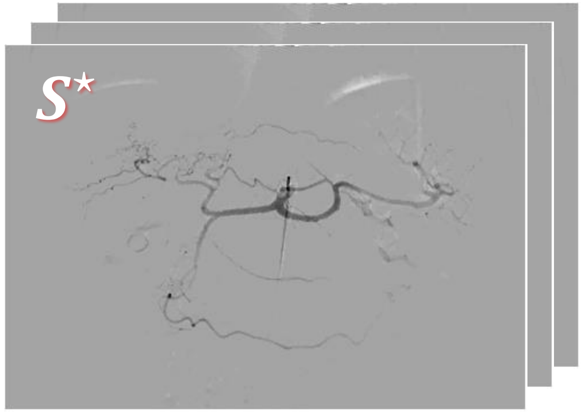
CT image sequence
under free breathing

≈



Fluoroscopic images

+

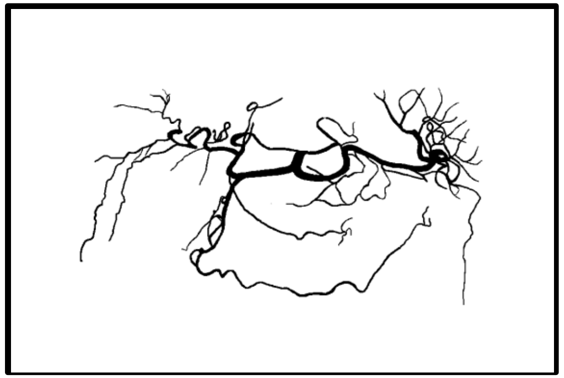


Angiographic images

$$(\boldsymbol{L}^\star,\boldsymbol{S}^\star) \coloneqq \operatorname*{argmin}_{(\boldsymbol{L},\boldsymbol{S})} \|\boldsymbol{L}\|_* + \lambda_S\|\boldsymbol{S}\|_G + \iota_C(\boldsymbol{D}-(\boldsymbol{L}+\boldsymbol{S}))$$

Our contributions:

- Total variation / group sparse regularization
- GPU-accelerated computing



Blood vessel registration

* Collaborating with Prof. Dr. Haneishi' lab., Center for Frontier Medical Engineering, Chiba University, Japan

ADMM

Alternating direction method of multipliers [Gabay&Mercier, 76]

$$\underset{(\boldsymbol{x},\boldsymbol{z})}{\text{Minimize}} \quad f(\boldsymbol{x}) + g(\boldsymbol{z}) \quad \text{subject to} \quad \boldsymbol{z} = \boldsymbol{G}\boldsymbol{x}$$

$$\boldsymbol{x}^{(k+1)} := \underset{\boldsymbol{x}}{\arg\min}\, f(\boldsymbol{x}) + \frac{\rho}{2}\left\|\boldsymbol{G}\boldsymbol{x} - (\boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)})\right\|_2^2$$

$$\boldsymbol{z}^{(k+1)} := \underset{\boldsymbol{z}}{\arg\min}\, g(\boldsymbol{z}) + \frac{\rho}{2}\left\|\boldsymbol{z} - (\boldsymbol{G}\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)})\right\|_2^2$$

$$\boldsymbol{u}^{(k+1)} := \boldsymbol{u}^{(k)} + \boldsymbol{G}\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k+1)}$$

L+S approximation by ADMM

- Letting $\boldsymbol{x}_L = \operatorname{vec} \boldsymbol{L}$, $\boldsymbol{x}_S = \operatorname{vec} \boldsymbol{S}$ and $\boldsymbol{d} = \operatorname{vec} \boldsymbol{D}$, SPCP becomes

$$(\boldsymbol{x}^\star, \boldsymbol{z}^\star) = \operatorname*{argmin}_{(\boldsymbol{x},\boldsymbol{z})} \frac{w_1}{2} \|\boldsymbol{z}_1 - \boldsymbol{d}\|_2^2 + w_2 \|\operatorname{mat} \boldsymbol{z}_2\|_* + w_3 \|\boldsymbol{z}_3\|_1 \quad \text{subject to} \quad \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \boldsymbol{z}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_L \\ \boldsymbol{x}_S \end{bmatrix}.$$

- ADMM:
  - $\boldsymbol{x}^{(k+1)} := \frac{1}{3} \begin{bmatrix} \boldsymbol{q}_1 + 2\boldsymbol{q}_2 - \boldsymbol{q}_3 \\ \boldsymbol{q}_1 - \boldsymbol{q}_2 + 2\boldsymbol{q}_3 \end{bmatrix}$ where $\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_3 \end{bmatrix} := \boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}$
  - $\begin{bmatrix} \boldsymbol{q}_1 \\ \boldsymbol{q}_2 \\ \boldsymbol{q}_3 \end{bmatrix} := \boldsymbol{G}\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)}$
  - $\boldsymbol{z}_1^{(k+1)} := \arg\min_{\zeta} \frac{w_1}{2} \|\zeta - \boldsymbol{d}\|_2^2 + \frac{\rho}{2} \|\zeta - \boldsymbol{q}_1\|_2^2 = \frac{1}{\rho + w_1} (\rho \boldsymbol{q}_1 + w_1 \boldsymbol{d})$
  - $\boldsymbol{z}_2^{(k+1)} := \arg\min_{\zeta} w_2 \|\operatorname{mat} \zeta\|_* + \frac{\rho}{2} \|\zeta - \boldsymbol{q}_2\|_2^2 = \operatorname{vec}\left(\boldsymbol{U} \operatorname{soft}\left(\boldsymbol{K}, \frac{w_2}{\rho}\right) \boldsymbol{V}^\top\right)$ where $\operatorname{mat} \boldsymbol{q}_2 = \boldsymbol{U}\boldsymbol{K}\boldsymbol{V}^\top$
  - $\boldsymbol{z}_3^{(k+1)} := \arg\min_{\zeta} w_3 \|\zeta\|_1 + \frac{\rho}{2} \|\zeta - \boldsymbol{q}_3\|_2^2 = \operatorname{soft}\left(\boldsymbol{q}_3, \frac{w_3}{\rho}\right)$
  - $\boldsymbol{u}^{(k+1)} := \boldsymbol{u}^{(k)} + \boldsymbol{G}\boldsymbol{x}^{(k+1)} - \boldsymbol{z}^{(k+1)}$

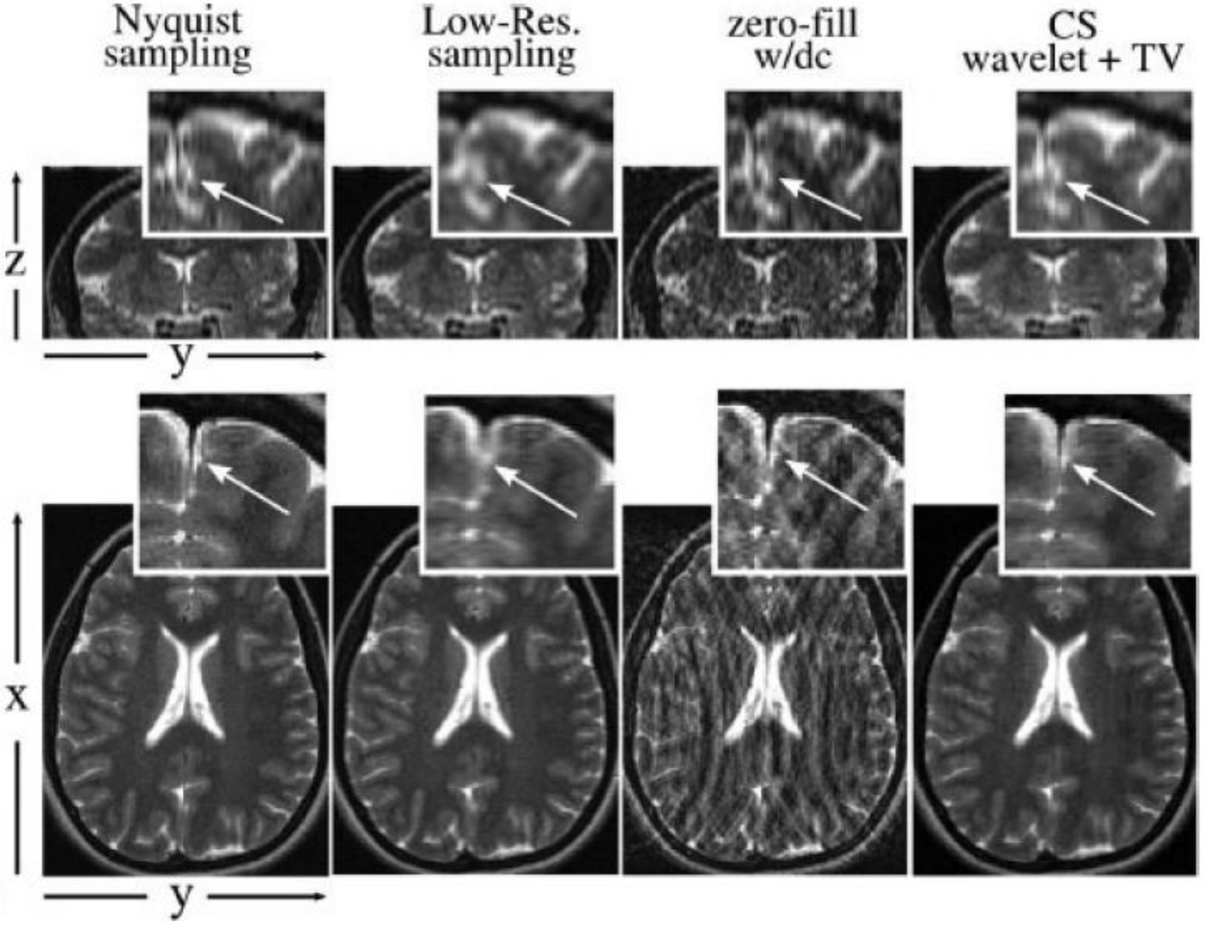
凸最適化の応用
Practical Convex Optimization

MRIの圧縮センシング
Sparse MRI

Sparse MRI

[Lustig+, 07]

$$\underset{\boldsymbol{m}}{\text{Minimize}}\ \|\boldsymbol{\Psi m}\|_1 + \alpha\, TV(\boldsymbol{m})\ \ \text{subject to}\ \ \|\boldsymbol{Fm} - \boldsymbol{y}\|_2 \le \varepsilon$$



$\boldsymbol{\Psi}$: ウェーブレット変換行列

$\boldsymbol{F}$: フーリエ変換行列

$$TV(\boldsymbol{m}) = \textstyle\sum_i \sqrt{(\boldsymbol{D}_x\boldsymbol{m})_i^2 + \left(\boldsymbol{D}_y\boldsymbol{m}\right)_i^2}\ :$$

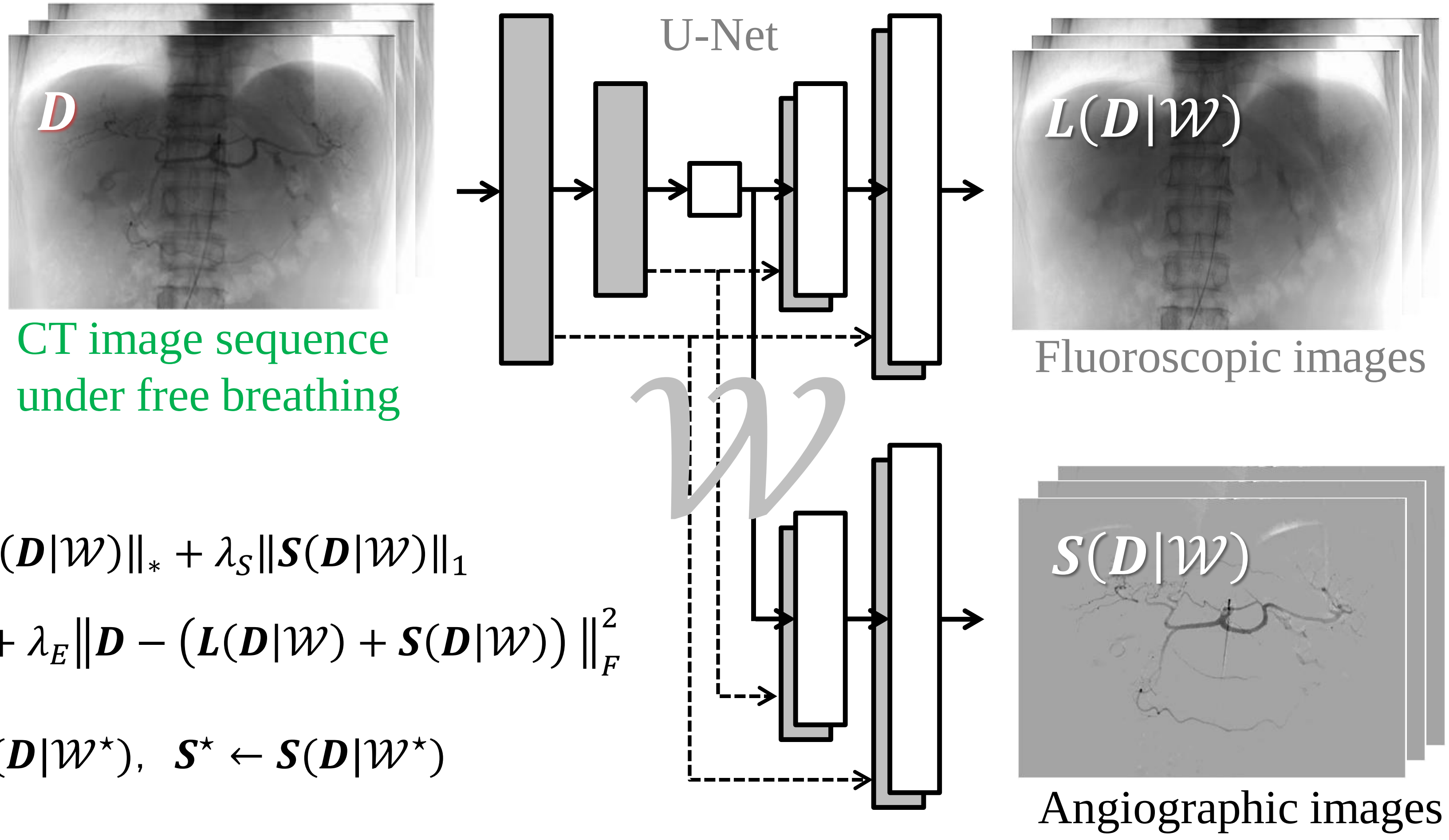
全変動（total variation; TV）
勾配強度の総和

https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnx0c2FrYWlsYWJ0aXBzfGd4OjU2YjEzM2RjODM5NzgxZmE

Sparse Modeling × Deep Learning

案1：Encoder-Decoderを応用する＋SpM

[Sakai, unpublished]



$$\mathcal{W}^\star \leftarrow \underset{\mathcal{W}}{\operatorname{argmin}} \|\boldsymbol{L}(\boldsymbol{D}|\mathcal{W})\|_* + \lambda_S \|\boldsymbol{S}(\boldsymbol{D}|\mathcal{W})\|_1 + \lambda_E \left\|\boldsymbol{D} - \left(\boldsymbol{L}(\boldsymbol{D}|\mathcal{W}) + \boldsymbol{S}(\boldsymbol{D}|\mathcal{W})\right)\right\|_F^2$$

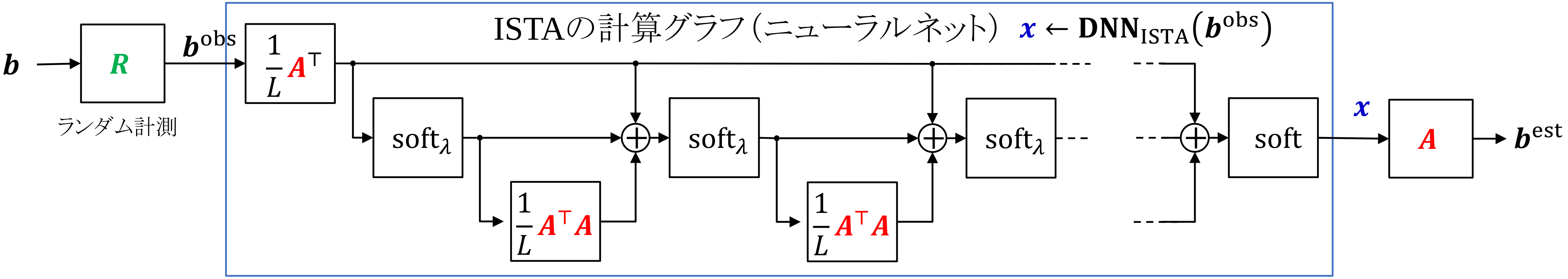
$$\boldsymbol{L}^\star \leftarrow \boldsymbol{L}(\boldsymbol{D}|\mathcal{W}^\star),\quad \boldsymbol{S}^\star \leftarrow \boldsymbol{S}(\boldsymbol{D}|\mathcal{W}^\star)$$

* This work is supported by JSPS KAKENHI (Grant-in-Aid for Scientific Research (B) 30345003) since 2019.

案2：スパース解法のDNN化 ＝ unfold/unroll

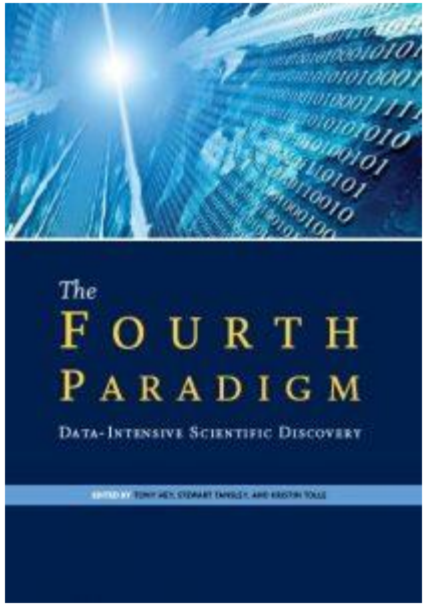
[Gregor&LeCun, ICML10]
[Sprechmann+, ICML12]
[Wang+, ICCV15]
[Papyan+, 16][Yang+, NIPS16]
IEEE SPM 2017Nov&2018Jan

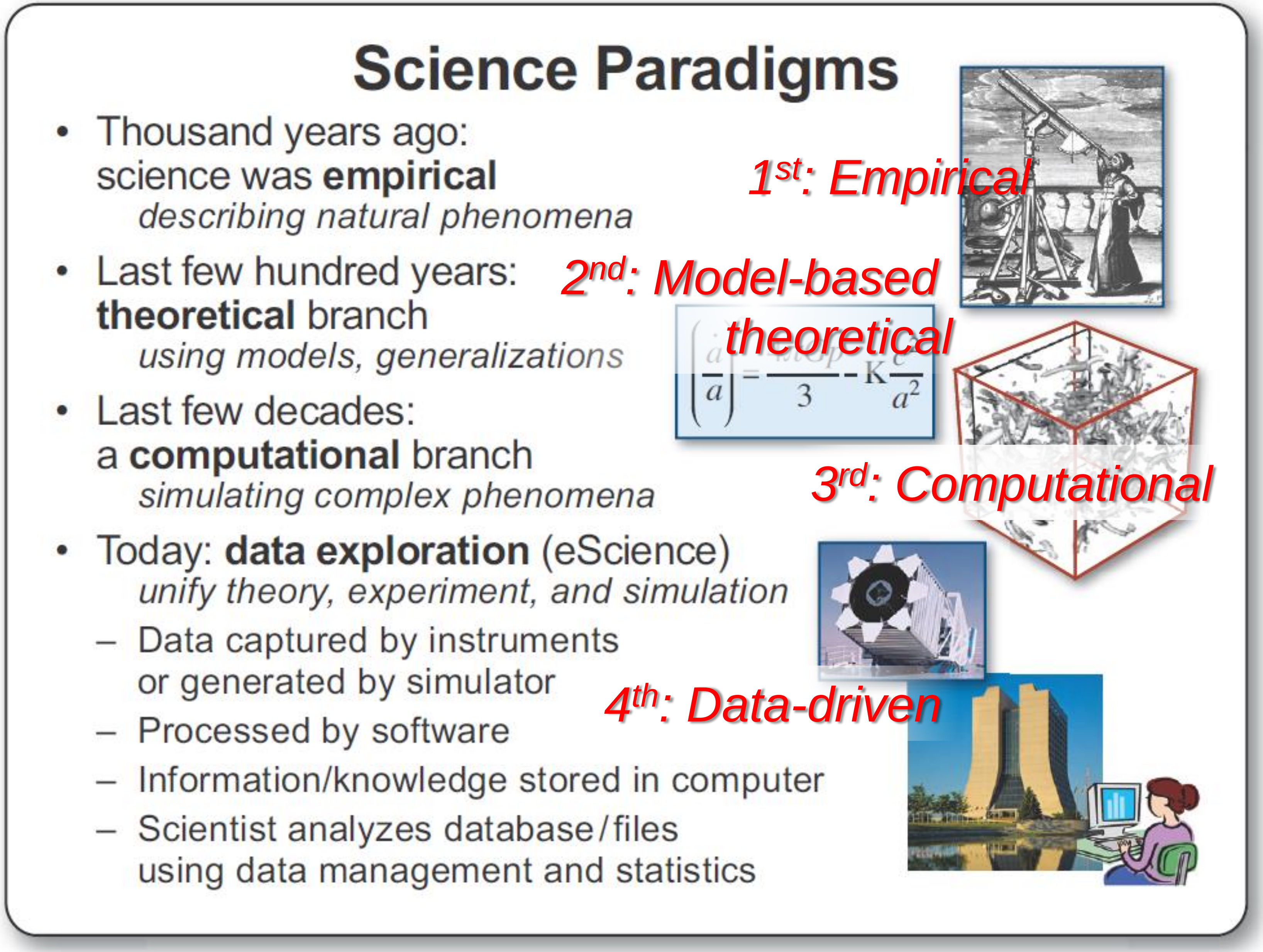
$$\boldsymbol{x}^{(k+1)} := \mathrm{soft}\left(\boldsymbol{x}^{(k)} + \frac{1}{L}\boldsymbol{A}^{\top}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)}),\ \frac{\lambda}{L} \right)$$



$\boldsymbol{b}$
$\boldsymbol{R}$
ランダム計測
$\boldsymbol{b}^{\mathrm{obs}}$
$\frac{1}{L}\boldsymbol{A}^{\top}$
ISTAの計算グラフ（ニューラルネット）
$\boldsymbol{x} \leftarrow \mathbf{DNN}_{\mathrm{ISTA}}(\boldsymbol{b}^{\mathrm{obs}})$
soft_{λ}
$\frac{1}{L}\boldsymbol{A}^{\top}\boldsymbol{A}$
$\oplus$
soft_{λ}
$\frac{1}{L}\boldsymbol{A}^{\top}\boldsymbol{A}$
$\oplus$
soft_{λ}
$\oplus$
soft
$\boldsymbol{x}$
$\boldsymbol{A}$
$\boldsymbol{b}^{\mathrm{est}}$

- 😄 学習前でも近似解を出力できる（数理モデルからの転移学習）
- 😄 少数のデータで学習（fine tuning）が可能　　$\mathrm{loss}(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$
- 😄 ハイパーパラメタ λ の学習も可能　　$\mathrm{loss}(\lambda) = \|\boldsymbol{x}(\lambda) - \boldsymbol{x}^{\mathrm{true}}\|$
- 😐 全結合層が多い





Science Paradigms

- Thousand years ago:
  science was **empirical**
  describing natural phenomena
- Last few hundred years:
  theoretical branch
  using models, generalizations
- Last few decades:
  a **computational** branch
  simulating complex phenomena
- Today: **data exploration** (eScience)
  unify theory, experiment, and simulation
  - Data captured by instruments
    or generated by simulator
  - Processed by software
  - Information/knowledge stored in computer
  - Scientist analyzes database/files
    using data management and statistics

1st: Empirical

2nd: Model-based theoretical

3rd: Computational

4th: Data-driven

FIGURE 1

The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, 2009.

I) Mathematical modeling

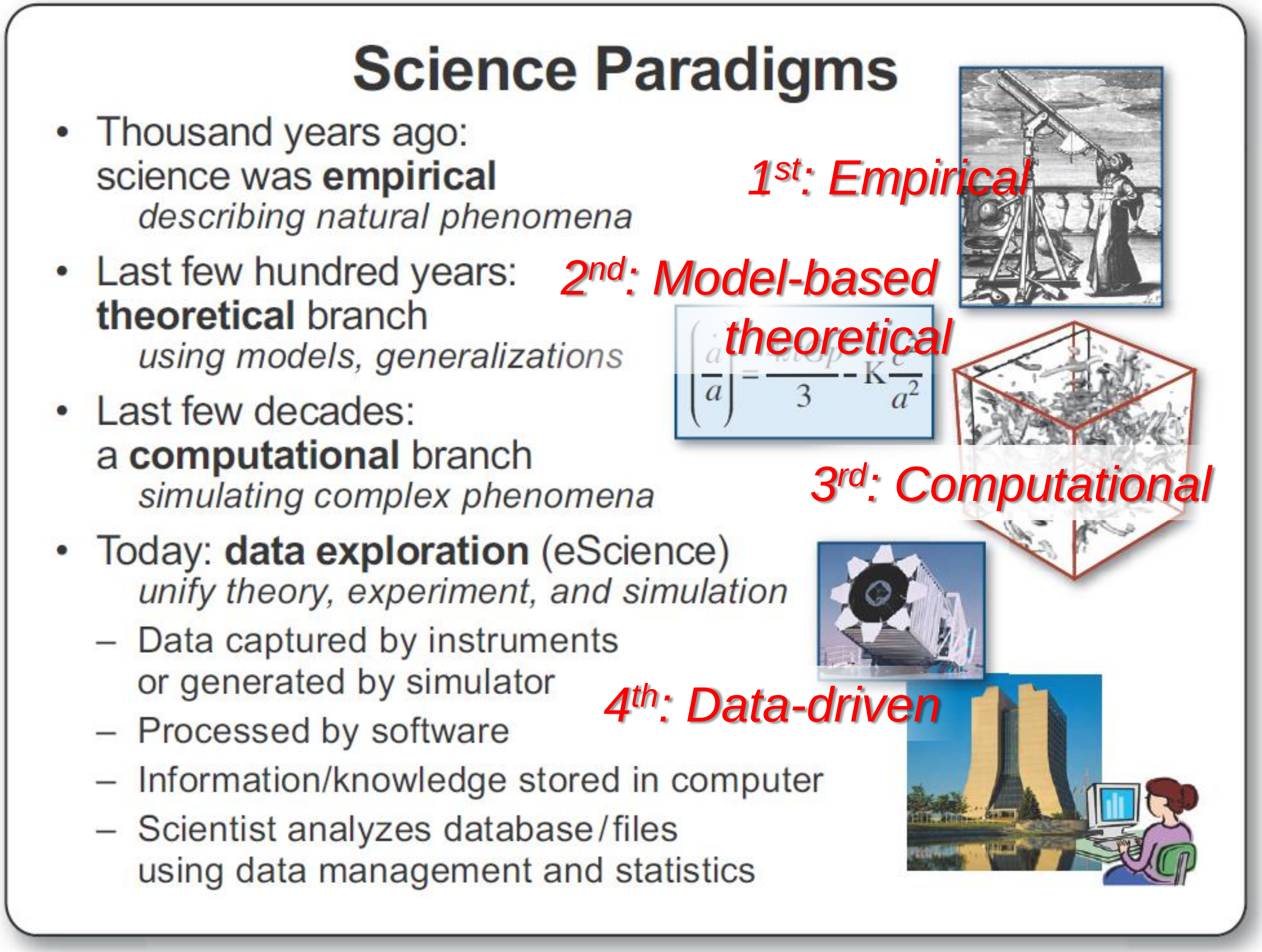
$$d = f(m)$$

II) Data assimilation with priors

$$m \approx \phi(d)$$

How can we exploit the mathematical model-based knowledges in the 4th paradigm?

-- *T. Sakai, 2018*



Science Paradigms

- Thousand years ago: science was **empirical**
  describing natural phenomena
- Last few hundred years: **theoretical** branch
  using models, generalizations
- Last few decades: a **computational** branch
  simulating complex phenomena
- Today: **data exploration** (eScience)
  unify theory, experiment, and simulation
  - Data captured by instruments or generated by simulator
  - Processed by software
  - Information/knowledge stored in computer
  - Scientist analyzes database/files using data management and statistics

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{4\pi G\rho}{3} - K\frac{c^2}{a^2}$$

1st: Empirical

2nd: Model-based theoretical

3rd: Computational

4th: Data-driven

FIGURE 1

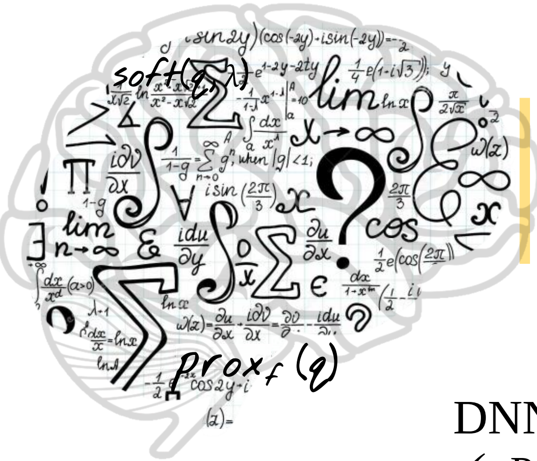
The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, 2009.

Science Paradigm 3.5

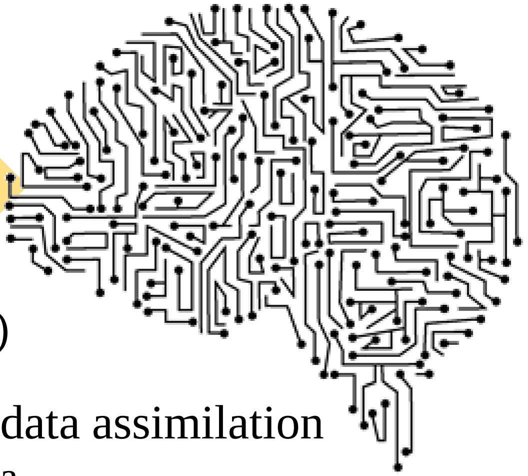
Transfer learning from mathematical models

(I) math. modeling, e.g., $\boldsymbol{d} \leftarrow \boldsymbol{f}(\boldsymbol{m})$

- Representation of data $\boldsymbol{d}$
- Formulation of the low of nature
- Forward problem (simulation)



[A] Transfer



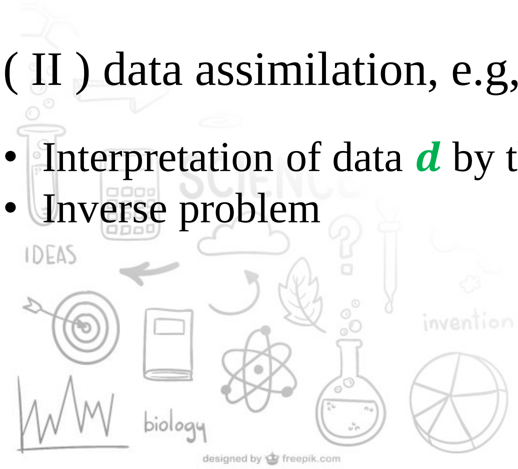
$\boldsymbol{\phi}(\boldsymbol{d}) \approx \mathbf{DNN}_{\boldsymbol{\phi}}(\boldsymbol{d})$

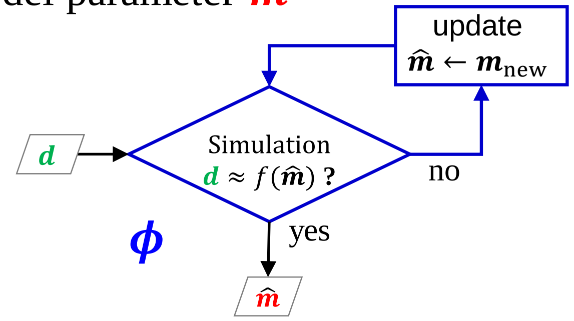
DNN representation of data assimilation

✓ Pretraining without data
✓ Theoretically guaranteed performance

(II) data assimilation, e.g, $\boldsymbol{m} \leftarrow \boldsymbol{\phi}(\boldsymbol{d})$

- Interpretation of data $\boldsymbol{d}$ by the model parameter $\boldsymbol{m}$
- Inverse problem




$\boldsymbol{d}$
Simulation
$\boldsymbol{d} \approx f(\widehat{\boldsymbol{m}})$?
no
update
$\widehat{\boldsymbol{m}} \leftarrow \boldsymbol{m}_{\text{new}}$
yes
$\widehat{\boldsymbol{m}}$
$\boldsymbol{\phi}$

[B] Fine Tuning

✓ performance improvement by training
✓ acquisition of differential knowledges