データサイエンス応用コース
（スパースモデリング）

高野　渉
大阪大学

スパースコーディング

あるデータ $\boldsymbol{x} \in R^l$ を規定ベクトル $\boldsymbol{d}_k \in R^l, k = 1,2,\cdots,m$ の線形結合

$$\boldsymbol{x} = \sum_{k=1}^{m} c_k \boldsymbol{d}_k$$

で表されると仮定する。

係数ベクトル $\boldsymbol{c} = (c_1, c_2, \cdots, c_m)^T \in R^n$ のうち少数の c_k のみが非ゼロの値をとり、それ以外はゼロとなるように規定ベクトル $\boldsymbol{d}_k$ を設計することができた場合、データ $\boldsymbol{x}$ の次元より少ない個数の c_k のみで $\boldsymbol{x}$ を表現できたことになる。

全体に対して、少数の状態をスパースと呼ぶ。

スパースコーディング

規定ベクトル $\boldsymbol{d}_k$ と係数ベクトル $\boldsymbol{c}$ を見つけるのは、以下の最適化問題を解くことに定式化することができる

$$\min_{\boldsymbol{c},\boldsymbol{d}_k} \left\| \boldsymbol{x} - \sum_{k=1}^{m} c_k \, \boldsymbol{d}_k \right\|_2^2 + \varphi(\boldsymbol{c})$$

第１項：元のデータ$\boldsymbol{x}$を規定ベクトルと係数ベクトルで再構成できる性能

第２項：係数ベクトルがスパースとなるように機能するための正則化項

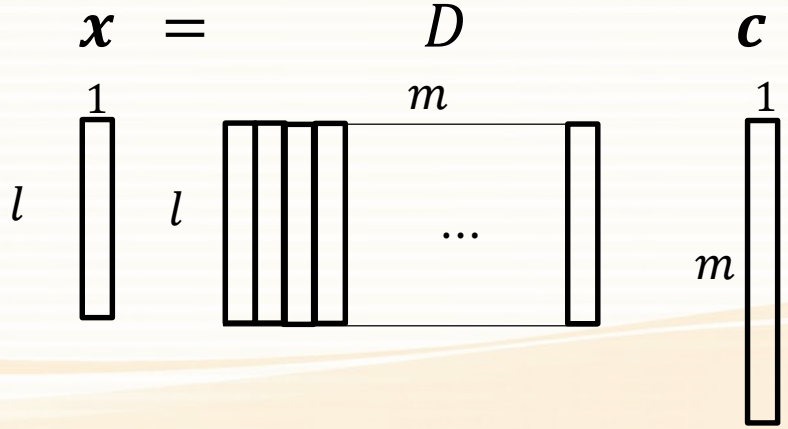
（例）　$\varphi(\boldsymbol{c}) = \lambda \sum_{k=1}^{m} e^{c_k^2} \qquad (\lambda > 0)$

スパースコーディング

あるデータ $\boldsymbol{x} \in R^l$ を規定ベクトル $\boldsymbol{d}_k \in R^l, k = 1,2,\cdots,m$ の線形結合として表現した式

$$\boldsymbol{x} = \sum_{k=1}^{m} c_k \boldsymbol{d}_k$$

これは辞書 $D = (\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_m) \in R^{l \times m}$ を用いて以下のベクトル・行列表現にて書くことができる。



$\boldsymbol{c} = (c_1, c_2, \cdots, c_m)^{\boldsymbol{T}}$ は係数ベクトルである。

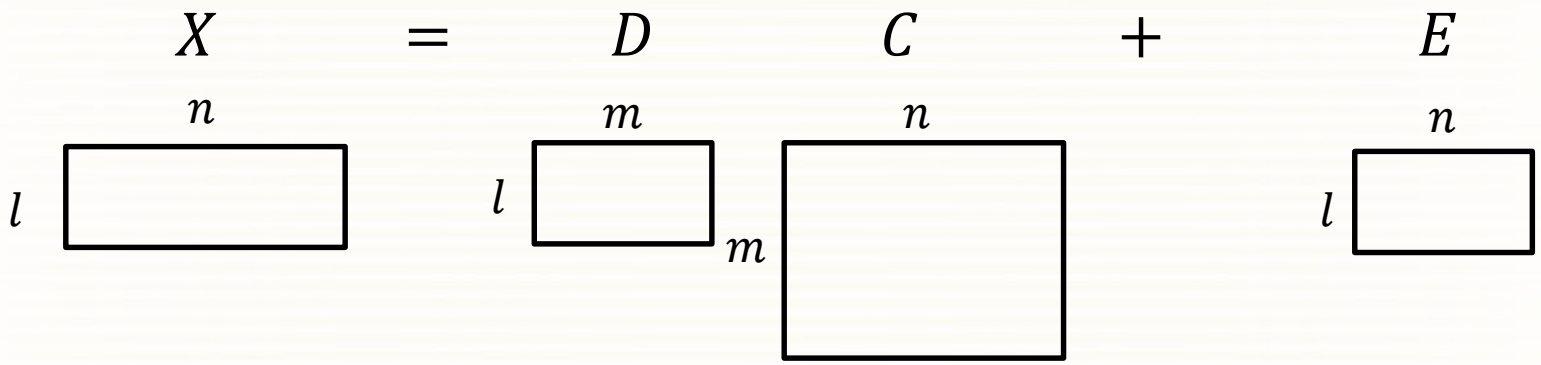
スパースコーディング

観察行列 $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) \in R^{l \times n}$

辞書行列 $D = (\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_m) \in R^{l \times m}$

係数行列 $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n) \in R^{m \times n}$

誤差行列 $E = (\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \cdots, \boldsymbol{\varepsilon}_n) \in R^{l \times n}$　　$\boldsymbol{\varepsilon}_k \in R^l \sim N(0, \sigma^2 I)$

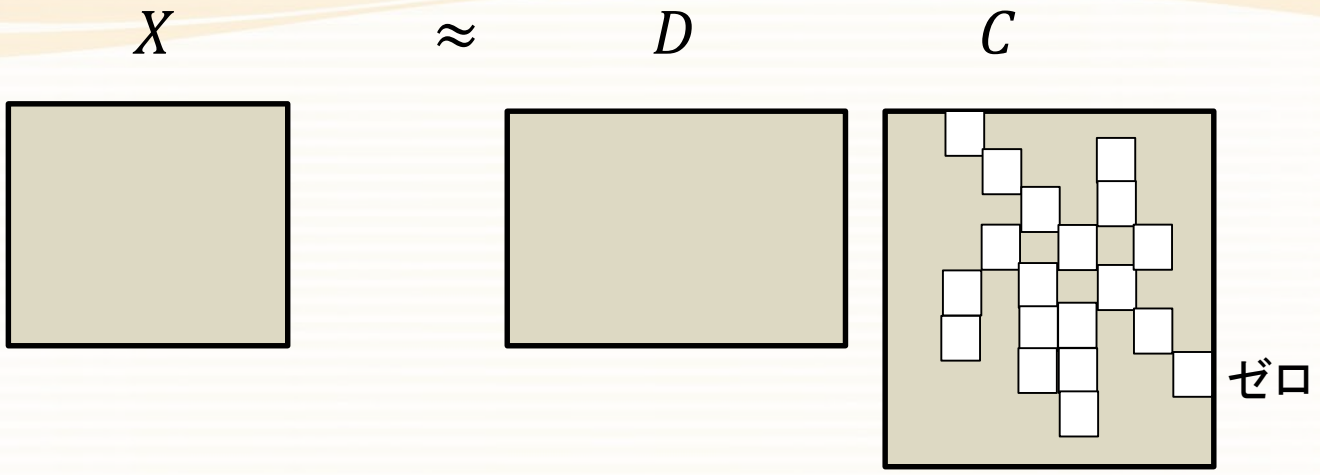


スパースコーディングは、観察行列 Xから辞書行列Dと
係数行列Cを最適化する

$$\min_{C,D} \quad \|X - D\,C\|_2^2 + \lambda \sum_{i=1}^{n} \|\boldsymbol{c}_i\|_p^p$$

損失関数フロベニウスノルム　　正則化項　l_pノルム

スパースコーディング



最適な辞書行列Dと係数行列Cは、

辞書行列Dを固定して、係数行列Cを最適化する

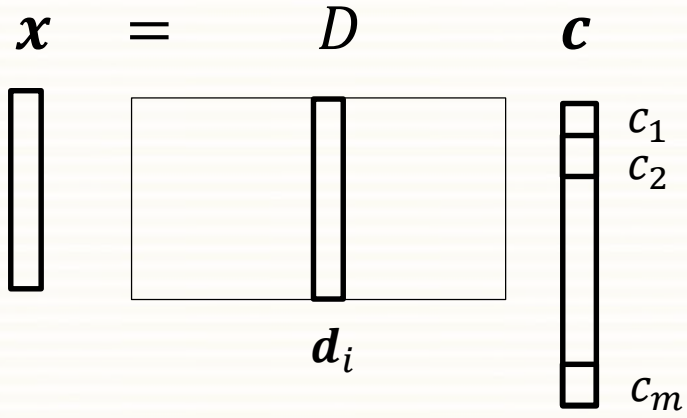
係数行列Cを固定して、辞書行列Dを最適化する

の手順を繰り返して求める

貪欲法

辞書行列Dを固定して、スパース表現を求めるアルゴリズムを考える。

観測データ $\boldsymbol{x} \in R^l$ を辞書行列Dの中の列ベクトル $\boldsymbol{d}_i$ の線形和として表す

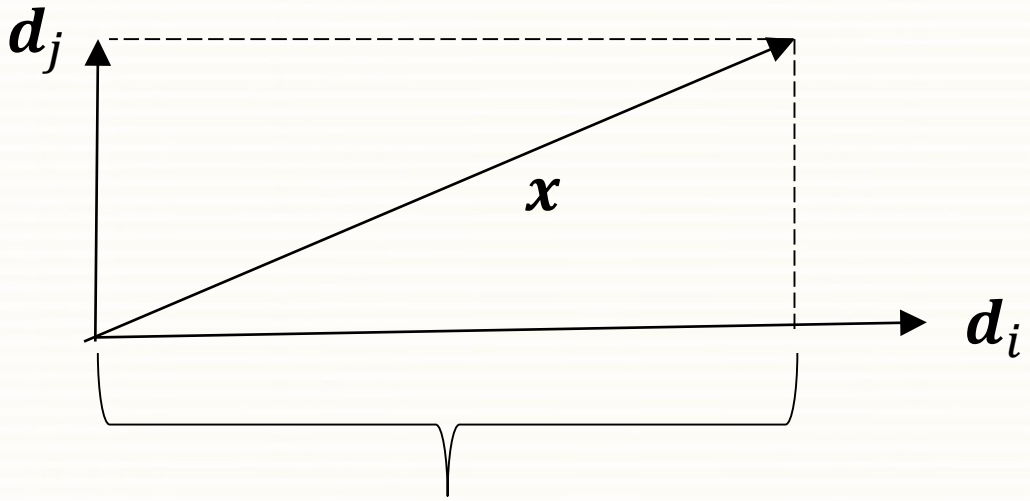


$$\boldsymbol{x} = c_1 \boldsymbol{d}_1 + c_2 \boldsymbol{d}_2 + \cdots + c_m \boldsymbol{d}_m$$

少しずつ採用する $\boldsymbol{d}_i$の数を増やしていく。
最もよく観測データを近似する $\boldsymbol{d}_i$を順々にサポートSに追加する

貪欲法

最もよく観測データを近似する $\boldsymbol{d}_i$ の選択方法



観測データ $\boldsymbol{x}$ を列ベクトル $\boldsymbol{d}_i$ にて最もよく近似する($c_i\boldsymbol{d}_i$)には、

誤差 $e = (\boldsymbol{x} - c_i\boldsymbol{d}_i)^T(\boldsymbol{x} - c_i\boldsymbol{d}_i)$

最適性の条件 $\dfrac{\partial e}{\partial c_i} = 2(\boldsymbol{x} - c_i\boldsymbol{d}_i)^T\boldsymbol{d}_i = 0$

貪欲法

この方程式を解くと、

$$c_i = \frac{\boldsymbol{x}^T \boldsymbol{d}_i}{\boldsymbol{d}_i{}^T \boldsymbol{d}_i}$$

これを誤差eに代入すると

$$e = (\boldsymbol{x} - \frac{\boldsymbol{x}^T \boldsymbol{d}_i}{\boldsymbol{d}_i{}^T \boldsymbol{d}_i} \boldsymbol{d}_i)^T (\boldsymbol{x} - \frac{\boldsymbol{x}^T \boldsymbol{d}_i}{\boldsymbol{d}_i{}^T \boldsymbol{d}_i} \boldsymbol{d}_i)$$

$$= \boldsymbol{x}^T \boldsymbol{x} - 2 \left(\frac{\boldsymbol{x}^T \boldsymbol{d}_i}{\boldsymbol{d}_i{}^T \boldsymbol{d}_i} \right) \boldsymbol{x}^T \boldsymbol{d}_i + \left(\frac{\boldsymbol{x}^T \boldsymbol{d}_i}{\boldsymbol{d}_i{}^T \boldsymbol{d}_i} \right)^2 \boldsymbol{d}_i{}^T \boldsymbol{d}_i$$

$$= \boldsymbol{x}^T \boldsymbol{x} - \frac{(\boldsymbol{x}^T \boldsymbol{d}_i)^2}{\boldsymbol{d}_i{}^T \boldsymbol{d}_i}$$

Orthogonal Matching Pursuit (OMP)

観測データ $\boldsymbol{x}$を最もよく近似する $c_i\boldsymbol{d}_i$ を求める。

この $c_i\boldsymbol{d}_i$で表せないベクトル$\boldsymbol{x} - c_i\boldsymbol{d}_i$を求める。

ベクトル$\boldsymbol{x} - c_i\boldsymbol{d}_i$を最もよく近似する $c_j\boldsymbol{d}_j$ を求める。

この $c_j\boldsymbol{d}_j$で表せないベクトル$\boldsymbol{x} - c_i\boldsymbol{d}_i - c_j\boldsymbol{d}_j$を求める。

ベクトル$\boldsymbol{x} - c_i\boldsymbol{d}_i - c_j\boldsymbol{d}_j$を最もよく近似する $c_k\boldsymbol{d}_k$ を求める。

これを繰り返していく。

Orthogonal Matching Pursuit (OMP)

初期化 : $k = 0$, サポートを空集合 $S^k = \emptyset$,

$$\boldsymbol{c}^{(k)} = 0, \quad \boldsymbol{r}^{(k)} = \boldsymbol{x} - D\boldsymbol{c}^{(k)},$$

Step1 : ベクトル$\boldsymbol{r}^{(k)}$と各列ベクトル $\boldsymbol{d}_i$による近似との誤差を計算

$$e_i = \min_{c_i} (\boldsymbol{r}^{(\boldsymbol{k})} - c_i \boldsymbol{d}_i)^T (\boldsymbol{r}^{(\boldsymbol{k})} - c_i \boldsymbol{d}_i)$$

近似されなかったベクトル

$$= \boldsymbol{r}^{(\boldsymbol{k})^T} \boldsymbol{r}^{(\boldsymbol{k})} - \frac{\left(\boldsymbol{r}^{(\boldsymbol{k})^T} \boldsymbol{d}_i\right)^2}{\boldsymbol{d}_i^T \boldsymbol{d}_i}$$

Orthogonal Matching Pursuit (OMP)

Step2 : 誤差が最小となる列ベクトルをサポート集合Sに追加

$$\hat{\imath} = \arg\min_{i \notin S^{(k)}} e_i \,, \qquad S^{(k+1)} = S^{(k)} \cup \{\hat{\imath}\}$$

Step3 : サポート集合に登録されたベクトルを用いた場合の係数ベクトルを計算。ただし、係数ベクトルの非ゼロ要素数があらかじめ決められた数値以下であること。

$$\hat{\boldsymbol{c}}^{(k+1)} = \arg\min_{\boldsymbol{c}} \left(\boldsymbol{x} - D_{S^{(k+1)}}\,\boldsymbol{c}\right)^T \left(\boldsymbol{x} - D_{S^{(k+1)}}\,\boldsymbol{c}\right)$$

$$= \left({D_{S^{(k+1)}}}^T D_{S^{(k+1)}}\right)^{-1} {D_{S^{(k+1)}}}^T \boldsymbol{x}$$

Orthogonal Matching Pursuit (OMP)

Step4 : サポート集合の辞書行列$D_{S^{(k+1)}}$と係数ベクトル$\hat{\boldsymbol{c}}^{(k+1)}$を用いたベクトル$D_{S^{(k+1)}}\hat{\boldsymbol{c}}^{(k+1)}$と観測ベクトル$\boldsymbol{x}$の誤差を計算

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{x} - D_{S^{(k+1)}}\hat{\boldsymbol{c}}^{(k+1)}$$

$$= \left(I - \left(D_{S^{(k+1)}}{}^{T} D_{S^{(k+1)}}\right)^{-1} D_{S^{(k+1)}}{}^{T}\right)\boldsymbol{x}$$

Step5 : $k \to k+1$にしてstep2に戻って繰り返す

このようにして、各観測データ$\boldsymbol{x}_k$に対して、辞書行列を固定した場合のスパースな係数行列$\boldsymbol{c}_k$を求めることができる。

辞書行列の最適化

OMPを通じて求められた係数行列 C を用いて辞書行列 Dの最適化を行う。

$$\widehat{D} = \arg\min_{D} \ \|X - D\ C\|_2^2$$

$$= X\ C^T\ (C\ C^T)^{-1}$$

【フロベニウスノルムの微分公式】

$$\frac{\partial}{\partial D}\ \|X - D\ C\|_2^2 = 2(DC - X)C^T$$

$$\frac{\partial}{\partial D}\ \|X - \ C\ D\|_2^2 = 2\ C^T(CD - X)$$

フロベニウスノルムの微分

公式 $\frac{\partial}{\partial D}\|X - D\,C\|_2^2 = 2(DC - X)C^T$ の導出だけ確認する

$X \in R^{l\times n}, D \in R^{l\times m}, C \in R^{m\times n}$

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{l1} & \cdots & x_{ln} \end{pmatrix}, \qquad D = (\boldsymbol{d}_1 \quad \cdots \quad \boldsymbol{d}_m) = \begin{pmatrix} \hat{\boldsymbol{d}}_1{}^T \\ \vdots \\ \hat{\boldsymbol{d}}_l{}^T \end{pmatrix}$$

$$C = (\boldsymbol{c}_1 \quad \cdots \quad \boldsymbol{c}_n)$$

$$DC = \begin{pmatrix} \hat{\boldsymbol{d}}_1{}^T \\ \vdots \\ \hat{\boldsymbol{d}}_l{}^T \end{pmatrix} (\boldsymbol{c}_1 \quad \cdots \quad \boldsymbol{c}_n) = \begin{pmatrix} \hat{\boldsymbol{d}}_1{}^T\boldsymbol{c}_1 & \cdots & \hat{\boldsymbol{d}}_1{}^T\boldsymbol{c}_n \\ \vdots & \ddots & \vdots \\ \hat{\boldsymbol{d}}_l{}^T\boldsymbol{c}_1 & \cdots & \hat{\boldsymbol{d}}_l{}^T\boldsymbol{c}_n \end{pmatrix}$$

フロベニウスノルムの微分

行列 DC の(i,j)成分は、$\hat{\boldsymbol{d}}_i{}^T \boldsymbol{c}_j$

フロベニウスノルムの２乗は、

$$\|X - D\ C\|_2^2 = \sum_{i,j} \left(x_{ij} - \hat{\boldsymbol{d}}_i{}^T \boldsymbol{c}_j \right)^2$$

このスカラー関数のベクトル$\hat{\boldsymbol{d}}_i$による微分は、

$$\frac{\partial}{\partial \hat{\boldsymbol{d}}_i} \|X - D\ C\|_2^2 = 2 \sum_j \left(\hat{\boldsymbol{d}}_i{}^T \boldsymbol{c}_j - x_{ij} \right) \boldsymbol{c}_j{}^T$$

フロベニウスノルムの微分

したがって、行列Dによる微分は、

$$\frac{\partial}{\partial D}\ \|X - D\ C\|_2^2 = 2\begin{pmatrix} \hat{\boldsymbol{d}}_1{}^T \sum_j \boldsymbol{c}_j\ \boldsymbol{c}_j{}^T \\ \vdots \\ \hat{\boldsymbol{d}}_l{}^T \sum_j \boldsymbol{c}_j\ \boldsymbol{c}_j{}^T \end{pmatrix} - 2\begin{pmatrix} \sum_j x_{1j}\boldsymbol{c}_j{}^T \\ \vdots \\ \sum_j x_{lj}\boldsymbol{c}_j{}^T \end{pmatrix}$$

ここで、

$$\sum_j \boldsymbol{c}_j\ \boldsymbol{c}_j{}^T = (\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n)\begin{pmatrix} \boldsymbol{c}_1{}^T \\ \boldsymbol{c}_2{}^T \\ \vdots \\ \boldsymbol{c}_n{}^T \end{pmatrix} = C\ C^T$$

フロベニウスノルムの微分

$$\sum_j x_{ij}\boldsymbol{c}_j{}^T = (x_{i1}, x_{i2}, \cdots, x_{in}) \begin{pmatrix} \boldsymbol{c}_1{}^T \\ \boldsymbol{c}_2{}^T \\ \vdots \\ \boldsymbol{c}_n{}^T \end{pmatrix}$$

これはXの第 i 行ベクトル

$$\begin{pmatrix} \sum_j x_{1j}\boldsymbol{c}_j{}^T \\ \vdots \\ \sum_j x_{lj}\boldsymbol{c}_j{}^T \end{pmatrix} = \begin{pmatrix} x_{11}, x_{12}, \cdots, x_{1n} \\ \vdots \\ x_{l1}, x_{l2}, \cdots, x_{ln} \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_1{}^T \\ \boldsymbol{c}_2{}^T \\ \vdots \\ \boldsymbol{c}_n{}^T \end{pmatrix} = X\,C^T$$

フロベニウスノルムの微分

最適性の条件

$$\frac{\partial}{\partial D} \|X - D\ C\|_2^2 = 2DC\ C^T - 2X\ C^T = 0$$

より、行列Dの最適解は

$$D = X\ C^T\ (C\ C^T)^{-1}$$

として得られる。

Method of Optimal Direction (MOD)

観察行列 $X = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n) \in R^{l \times n}$

辞書行列 $D = (\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_m) \in R^{l \times m}$

係数行列 $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_n) \in R^{m \times n}$

$$\min_{C,D} \quad \|X - D\ C\|_2^2$$

を解くアルゴリズムは、

最適な辞書行列Dと係数行列Cは、

辞書行列Dを固定して、係数行列Cを最適化する (OMP)

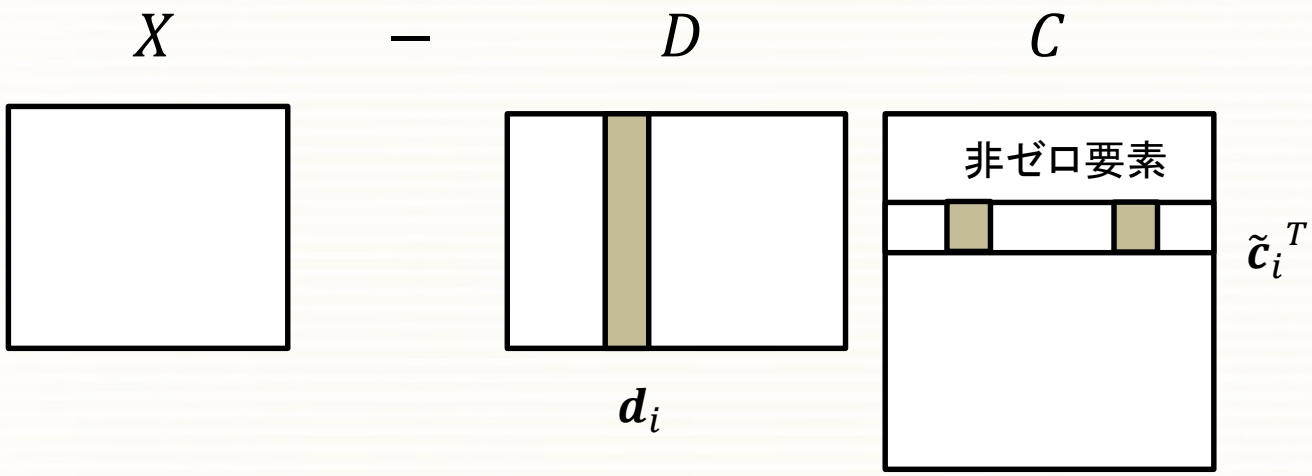
係数行列Cを固定して、辞書行列Dを最適化する (辞書最適化)

の繰り返しとなる。

K – Singular Value Decomposition (K-SVD)

係数行列Cを固定して、辞書行列Dを最適化する方法の一種であり、

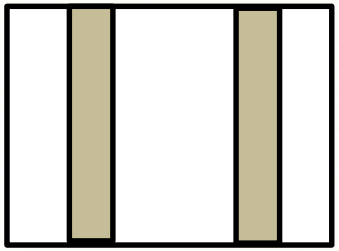
辞書行列Dの列ごとに更新していく計算方法



辞書行列D の第 i 列 $\boldsymbol{d}_i$,
係数行列C の第 i 行$\tilde{\boldsymbol{c}}_i^T$に注目する。

K – Singular Value Decomposition (K-SVD)

X $-$ D C

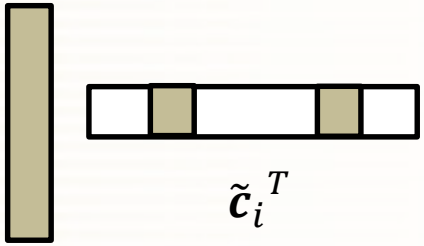


$\boldsymbol{d}_i$ が関係する部分

辞書行列から第 i 列 $\boldsymbol{d}_i$ を除いた行列

係数行列から第 i 行 $\tilde{\boldsymbol{c}}_i^{\,T}$ を除いた行列

$+$



$\boldsymbol{d}_i$ $\tilde{\boldsymbol{c}}_i^{\,T}$

$$X - \left(\sum_{j \neq i} \boldsymbol{d}_j \tilde{\boldsymbol{c}}_j^{\,T} + \boldsymbol{d}_i \tilde{\boldsymbol{c}}_i^{\,T} \right)$$

$$X - \sum_{j \neq i} \boldsymbol{d}_j \tilde{\boldsymbol{c}}_j^{\,T} = \boldsymbol{d}_i \tilde{\boldsymbol{c}}_i^{\,T}$$

K – Singular Value Decomposition (K-SVD)

この部分（$\boldsymbol{d}_i \tilde{\boldsymbol{c}}_i{}^T$）は、特異値分解と
みることができる。

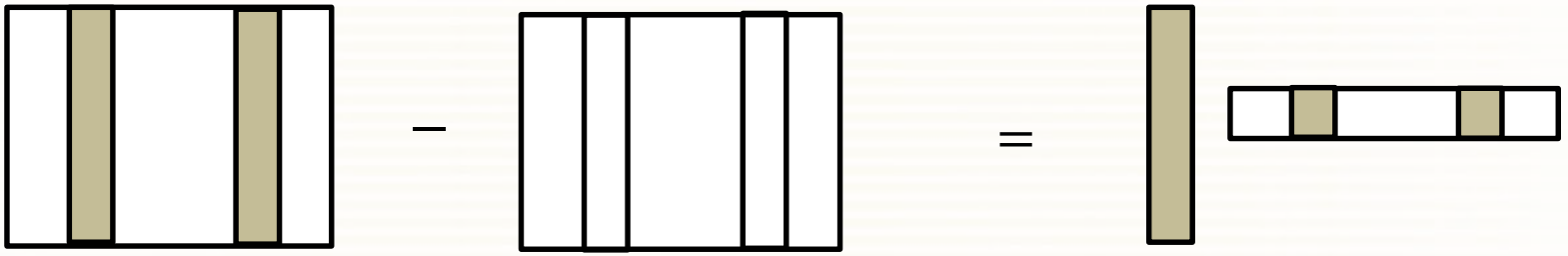
したがって、 $X - \sum_{j \neq i} \boldsymbol{d}_j \tilde{\boldsymbol{c}}_j{}^T$ を計算して、

特異値分解（特異値1つ）すると、$\boldsymbol{d}_i$ と $\tilde{\boldsymbol{c}}_i{}^T$ が求められることになる。

しかし、これだけでは、$\tilde{\boldsymbol{c}}_i{}^T$ がスパース（非ゼロ要素数が少数）に
なる保証はない。

K – Singular Value Decomposition (K-SVD)

$\tilde{\boldsymbol{c}}_i^T$の非ゼロ要素の位置情報を活用する。



$$X - \sum_{j \neq i} \boldsymbol{d}_j \tilde{\boldsymbol{c}}_j^T = \boldsymbol{d}_i \tilde{\boldsymbol{c}}_i^T$$

この行列の特異値分解から$\boldsymbol{d}_i$を求めることができる

K – Singular Value Decomposition (K-SVD)

$$\min_{C,D} \quad \|X - D\ C\|_2^2$$

K-SVDを利用したアルゴリズムをまとめる。

初期化 : 辞書行列Dの各列ベクトルを単位ベクトルに正規化する

Step1 : OMPにしたがって、係数行列を計算する

Step2 : $\boldsymbol{d}_i$ が関係する観測データの集合Qを求める

Step3 : 観測データと近似の残差行列を計算する

$$E = X - \sum_{j \neq i} \boldsymbol{d}_j \tilde{\boldsymbol{c}}_j{}^T$$

K – Singular Value Decomposition (K-SVD)

Step4 : 残差行列Eから集合Qの列だけを残したE^Qを求める

Step5 : E^Qに特異値分解を適用して、第１左固有値ベクトルを辞書行列 D の第 i 列ベクトル $\boldsymbol{d}_i$として採用する

Step6 : Step1に戻って、同じ手続きを繰り返す

K-SVDは、OMPの逆行列計算がないため、計算不安定の問題が解消される。一方、列ベクトル毎に特異値分解を実行する必要があり計算コストが高い。

Iterative Reweighted Least Square (IRLS)

観測データ $\boldsymbol{x}$ から係数ベクトル $\boldsymbol{c}$ を求めるときに、

l_p ノルムを重み付き l_2 ノルムとして表現することで

l_p ノルム正則化問題を近似的に解く収束計算方法

k 回目の収束計算にて求められた係数ベクトルを $\boldsymbol{c}^{(k)}$ と置く。

重み行列 W_k を以下に定義する

$$W_k = \begin{bmatrix} \left|c_1^{(k)}\right|^{1-\frac{p}{2}} & & 0 \\ & \ddots & \\ 0 & & \left|c_m^{(k)}\right|^{1-\frac{p}{2}} \end{bmatrix}$$

Iterative Reweighted Least Square (IRLS)

重み行列 W_k の逆行列は

$$W_k^{-1} = \begin{bmatrix} \left|c_1^{(k)}\right|^{-1+\frac{p}{2}} & & 0 \\ & \ddots & \\ 0 & & \left|c_m^{(k)}\right|^{-1+\frac{p}{2}} \end{bmatrix}$$

となる。ただし、W_kの対角成分がゼロの場合、

その位置に対応する W_k^{-1}の成分をゼロとする。

Iterative Reweighted Least Square (IRLS)

係数ベクトル$\boldsymbol{c}^{(k)}$の l_p ノルムをW_k^{-1}を用いて表す。

$$\left\|W_k^{-1}\boldsymbol{c}^{(k)}\right\|_2^2 = \left|c_1^{(k)}\right|^p + \left|c_2^{(k)}\right|^p + \cdots + \left|c_m^{(k)}\right|^p = \left\|\boldsymbol{c}^{(k)}\right\|_p^p$$

辞書行列を用いた $D\boldsymbol{c}$ によって観測データ$\boldsymbol{x}$を表しながら、
係数ベクトル$\boldsymbol{c}$ がスパースとなるような $\boldsymbol{c}$ を求める問題を
以下の最適化問題とする。

$$\min_{\boldsymbol{c}} \left\|W_k^{-1}\boldsymbol{c}\right\|_2^2$$

$$\text{subject to } \|\boldsymbol{x} - D\boldsymbol{c}\|_2 = 0$$

Iterative Reweighted Least Square (IRLS)

拘束条件付き最適化問題であり、

これをラグランジュ未定乗数法で解く。

ラグランジュ関数：

$$L = \frac{1}{2}\,\boldsymbol{c}^T W_k^{-1} W_k^{-1} \boldsymbol{c} + \boldsymbol{\lambda}^T (\boldsymbol{x} - D\boldsymbol{c})$$

最適性の条件より

$$\frac{\partial L}{\partial \boldsymbol{c}} = \boldsymbol{c}^T W_k^{-1} W_k^{-1} - \boldsymbol{\lambda}^T D = 0 \qquad \cdots ①$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = (\boldsymbol{x} - D\boldsymbol{c})^T = 0 \qquad \cdots ②$$

Iterative Reweighted Least Square (IRLS)

式①より　　$W_k^{-1} W_k^{-1} \boldsymbol{c} = D^T \boldsymbol{\lambda}$

$$\boldsymbol{c} = W_k W_k D^T \boldsymbol{\lambda} \quad \cdots ③$$

式②より　　$\boldsymbol{x} = D\boldsymbol{c} \quad \cdots ④$

式③を式④に代入して

$$\boldsymbol{x} = D W_k W_k D^T \boldsymbol{\lambda}$$

$$\boldsymbol{\lambda} = (D W_k W_k D^T)^{-1} \boldsymbol{x}$$

これを式③に代入すると

$$\boldsymbol{c} = W_k W_k D^T (D W_k W_k D^T)^{-1} \boldsymbol{x}$$

係数ベクトルを求めることができる。

データサイエンス応用コース
（テキスト処理）

高野　渉
大阪大学

文書の特徴量表現

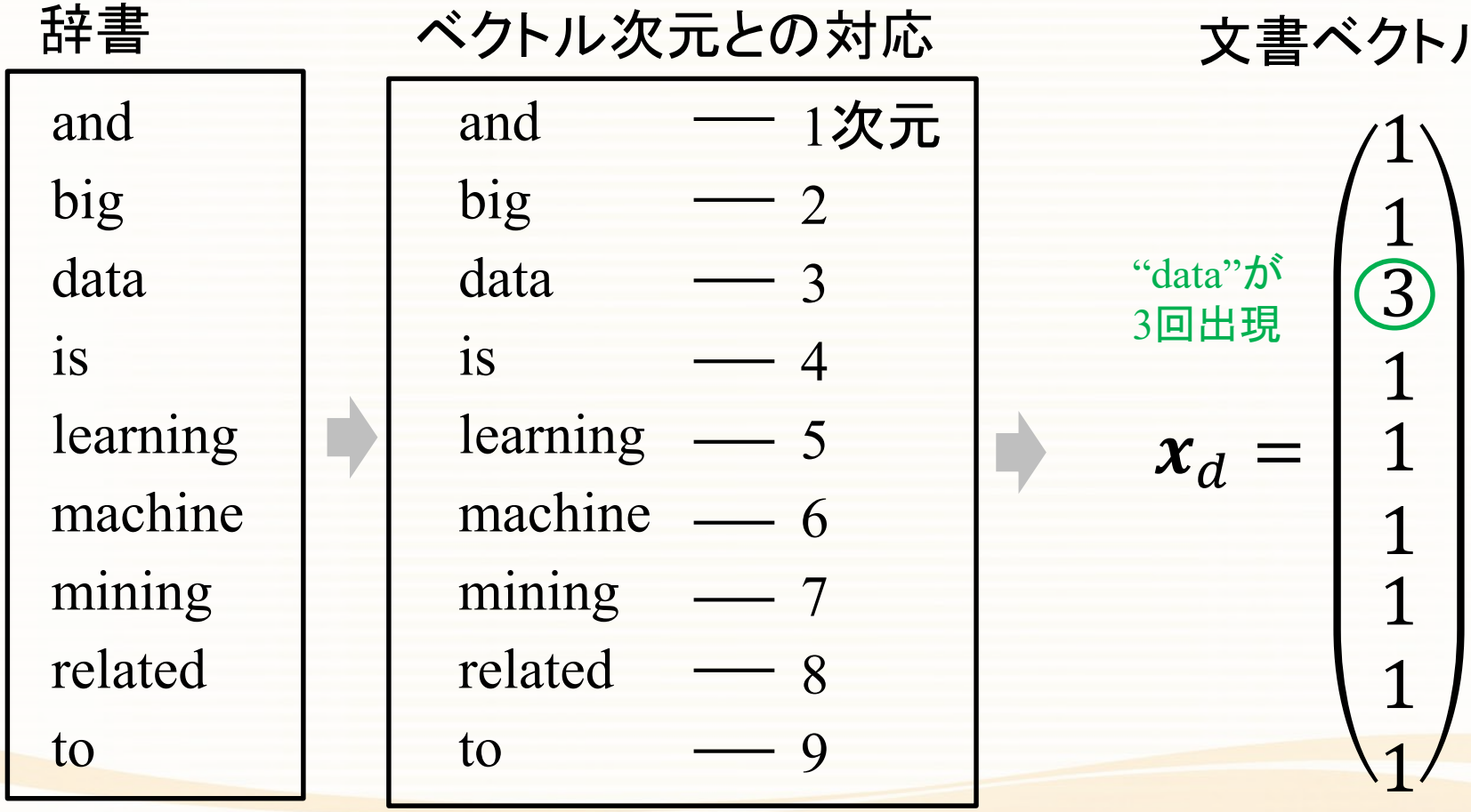
文書をベクトルにて表現することを考える
ベクトルの各要素が特徴を表しており、これを素性と呼ぶ。
（自然言語処理では素性、機械学習一般では特徴量）

文書 d をベクトル $\boldsymbol{x}_d$ として数値化する。
ベクトル $\boldsymbol{x}_d$ の各要素の位置と単語の種類を対応付け、
各要素の値を文書 d の中におけるその単語ωの頻度$n_d(\omega)$とする。

文書の特徴量表現

（例1）文書

“data science is related to data mining, machine learning and big data”



各単語が文章中で何回現れているかを表現しているが、
文の語順や構造の情報が失われている。

文書の特徴量表現

（例2）

文書 d_1 : “ a cat bites a mouse” (猫がねずみを噛む)

文書 d_2 : “ a mouse bites a cat” (ねずみが猫を噛む)

辞書と次元

a	1
bites	2
cat	3
mouse	4

文書ベクトル

$$\boldsymbol{x}_{d_1} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 1 \end{pmatrix} \qquad \boldsymbol{x}_{d_2} = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

文書 d_1と d_2 を数値化した文書ベクトル$\boldsymbol{x}_{d_1}$と$\boldsymbol{x}_{d_2}$は同じになってしまう

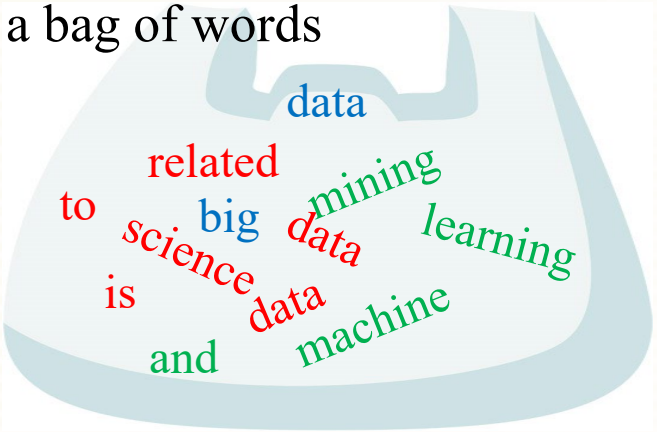
$d_1 \neq d_2$ $\qquad$ $\boldsymbol{x}_{d_1} = \boldsymbol{x}_{d_2}$

Bag of Words

各単語が文書中に出現する回数で、文書を数値化する方法を“bag of words”と呼ぶ。
（文書を単語に分割して、バラバラになった単語が袋の中にある状態から連想される名称）

document

data science is related to data mining, machine learning and big data



例2のように異なる文書が同じベクトルとして数値化されるのをどう解消すればいいだろうか？

Word n-gram

連続した n 個の単語のまとまりを1つの語彙として辞書に登録する。
この単語のまとまりを単語n-gramと呼ぶ。
特に $n = 1$の場合は、unigram
　　$n = 2$の場合は、bigram
　　$n = 3$の場合は、trigram
と呼ぶ。

（例2）の文書
　文書 d_1 : “ a cat bites a mouse” (猫がねずみを噛む)
　文書 d_2 : “ a mouse bites a cat” (ねずみが猫を噛む)
に対して、単語bigramを用いた場合の文書ベクトルを求めてみる。

Word n-gram

	a	bites	cat	mouse
a	g_{11}	g_{12}	g_{13}	g_{14}
bites	g_{21}	g_{22}	g_{23}	g_{24}
cat	g_{31}	g_{32}	g_{33}	g_{34}
mouse	g_{41}	g_{42}	g_{43}	g_{44}

“a” の後に “cat”が続く単語bigramをg_{13}として表している。

単語bigramと次元の関係

g_{11}	g_{12}	g_{13}	g_{14}	g_{21}	g_{22}	g_{23}	g_{24}	g_{31}	g_{32}	g_{33}	g_{34}	g_{41}	g_{42}	g_{43}	g_{44}
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16次元

文書 d_1とd_2の単語bigram 表現

d_1 : “a cat bites a mouse”

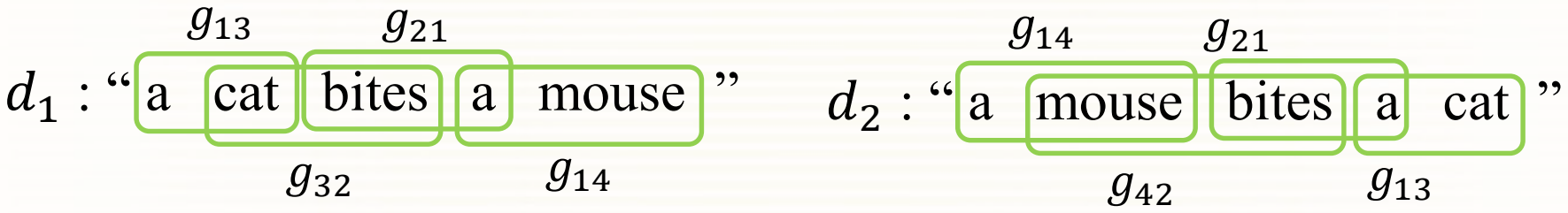
g_{13} g_{21} g_{32} g_{14}

d_2 : “a mouse bites a cat”

g_{14} g_{21} g_{42} g_{13}

Word n-gram

文書 d_1とd_2の単語bigram 表現



文書 d_1とd_2のベクトル表現

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
g_{11}	g_{12}	g_{13}	g_{14}	g_{21}	g_{22}	g_{23}	g_{24}	g_{31}	g_{32}	g_{33}	g_{34}	g_{41}	g_{42}	g_{43}	g_{44}
$\boldsymbol{x}_{d_1}$ = (0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0)
$\boldsymbol{x}_{d_2}$ = (0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0)

$\boldsymbol{x}_{d_1} \neq \boldsymbol{x}_{d_2}$ となり、文書d_1とd_2を異なる
ベクトルとして表すことができる。

類似度計算

文書 d_1，d_2 を bag of words を通じてベクトル$\boldsymbol{x}_{d_1}$，$\boldsymbol{x}_{d_2}$に数値化した。

これら文書ベクトルを用いて、文書間が似ているかどうかの数値指標（類似度）を計算することができる。

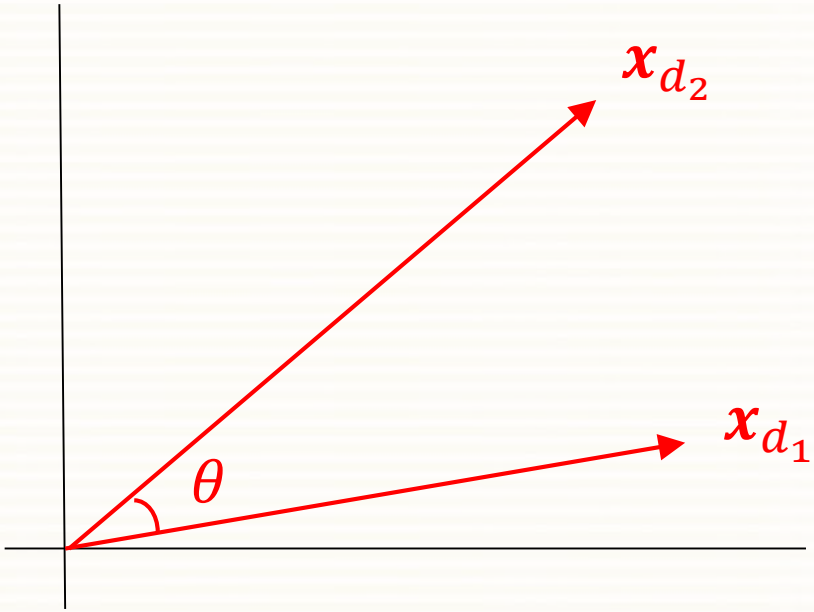
類似度の代表として、

- 余弦類似度
- l_2 ノルム

などを挙げることができる。

余弦類似度

ベクトル$\boldsymbol{x}_{d_1}$, $\boldsymbol{x}_{d_2}$のなす角度 θ を用いて類似度を表現する。



ベクトル$\boldsymbol{x}_{d_1}$, $\boldsymbol{x}_{d_2}$の内積より余弦を計算

$$\cos\theta = \frac{\boldsymbol{x}_{d_1}{}^{T}}{|\boldsymbol{x}_{d_1}|}\frac{\boldsymbol{x}_{d_2}}{|\boldsymbol{x}_{d_2}|} \qquad (-1 \leq \cos\theta \leq 1)$$

余弦類似度

ベクトル$\boldsymbol{x}_{d_1}$, $\boldsymbol{x}_{d_2}$の方向が一致するとき、
$\theta = 0$であり、余弦 $\cos\theta$ は最大値 1 をとる。

ベクトル$\boldsymbol{x}_{d_1}$, $\boldsymbol{x}_{d_2}$の方向が反対のとき、
$\theta = 180^\circ$ であり、余弦 $\cos\theta$ は最小値−1をとる。

すなわち、余弦$\cos\theta$が2つのベクトルの方向の一致度を数値化できることになり、余弦$\cos\theta$をベクトル間の類似度として使用できる

余弦類似度

例2の単語bigramを用いた場合の
文書 d_1とd_2のベクトル表現は以下であった

$$\boldsymbol{x}_{d_1} = (0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

$$\boldsymbol{x}_{d_2} = (0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0)$$

文書 d_1とd_2の余弦類似度は

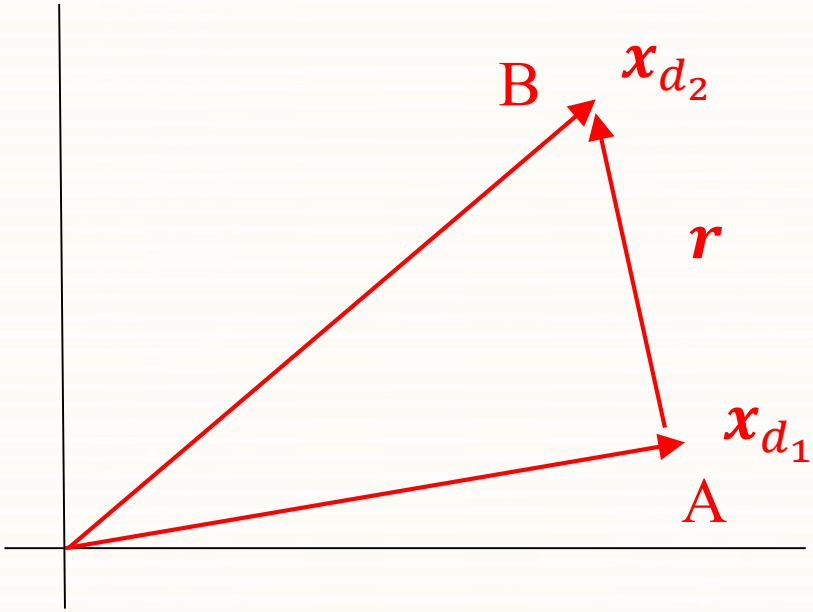
$$|\boldsymbol{x}_{d_1}| = |\boldsymbol{x}_{d_2}| = 2$$

$$\cos\theta = \frac{\boldsymbol{x}_{d_1}{}^T}{|\boldsymbol{x}_{d_1}|} \frac{\boldsymbol{x}_{d_2}}{|\boldsymbol{x}_{d_2}|} = \frac{1+1+1}{2\times 2} = \frac{3}{4}$$

と求められる。

l_2ノルム非類似度

ベクトル$\boldsymbol{x}_{d_1}$, $\boldsymbol{x}_{d_2}$のユークリッド距離 $|\boldsymbol{r}|$ を用いて
非類似度を表現する。



ベクトル$\boldsymbol{x}_{d_1}$, $\boldsymbol{x}_{d_2}$ の差 $\boldsymbol{r} = \boldsymbol{x}_{d_2} - \boldsymbol{x}_{d_1}$

そのl_2ノルム$|\boldsymbol{r}| = \sqrt{\boldsymbol{r}^T\boldsymbol{r}}$ ($\boldsymbol{r}$の各成分の2乗和の平方根)が

AからBまでの距離であり、それをベクトル間の非類似度とする

l_2ノルム非類似度

例2の単語bigramを用いた場合

$$\boldsymbol{x}_{d_1} = (0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

$$\boldsymbol{x}_{d_2} = (0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0)$$

この2つのベクトルの差分 $\boldsymbol{r} = \boldsymbol{x}_{d_2} - \boldsymbol{x}_{d_1}$

$$\boldsymbol{r} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0)$$

$$|\boldsymbol{r}| = \sqrt{0+0+0+0+0+0+0+0+0+(-1)^2+0+0+0+1^2+0+0}$$

$$= \sqrt{2}$$

一般に文書ベクトル $\boldsymbol{x}$ は高次元である。

これを低次元化して解析を視覚化するなどの工夫を行うことがある。

潜在意味解析（Latent Semantic Analysis）

文書：$d_1, d_2, \cdots, d_m$　　単語：$\omega_1, \omega_2, \cdots, \omega_n$

文書の特徴ベクトル：$\boldsymbol{x}_{d_1}, \boldsymbol{x}_{d_2}, \cdots, \boldsymbol{x}_{d_m}$

（特徴ベクトル$\boldsymbol{x}_{d_k} \in R^n$ の i 次元目は、文書d_kにおける単語ω_iの頻度

つまり、k 番目の文書にi 番目の単語が出る回数）

単語の特徴ベクトル：$\boldsymbol{x}_{\omega_1}, \boldsymbol{x}_{\omega_2}, \cdots, \boldsymbol{x}_{\omega_n}$

（特徴ベクトル$\boldsymbol{x}_{\omega_k} \in R^m$ の i 次元目は、文書d_iにおける単語ω_kの頻度

つまり、i 番目の文書にk 番目の単語が出る回数）

文書集合の行列：$D = \left[\boldsymbol{x}_{d_1}\ \boldsymbol{x}_{d_2}\ \cdots\ \boldsymbol{x}_{d_m}\right] = \begin{bmatrix} \boldsymbol{x}_{\omega_1}{}^T \\ \boldsymbol{x}_{\omega_2}{}^T \\ \vdots \\ \boldsymbol{x}_{\omega_n}{}^T \end{bmatrix} \in R^{n \times m}$

潜在意味解析（Latent Semantic Analysis）

$$D^T D = \begin{bmatrix} \boldsymbol{x}_{d_1}{}^T \\ \boldsymbol{x}_{d_2}{}^T \\ \vdots \\ \boldsymbol{x}_{d_m}{}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{d_1} \boldsymbol{x}_{d_2} \cdots \boldsymbol{x}_{d_m} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{d_1}{}^T \boldsymbol{x}_{d_1} & \cdots & \boldsymbol{x}_{d_1}{}^T \boldsymbol{x}_{d_m} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}_{d_m}{}^T \boldsymbol{x}_{d_1} & \cdots & \boldsymbol{x}_{d_m}{}^T \boldsymbol{x}_{d_m} \end{bmatrix}$$

行列 $D^T D$の(i, j)成分 $\boldsymbol{x}_{d_i}{}^T \boldsymbol{x}_{d_j}$ は、文書 d_i と d_j に同じ単語頻度で記述されていれば大きな値をとることになる。

同じ単語頻度ということは、文書 d_i と d_jの意味が似ていると解釈することができるので、$\boldsymbol{x}_{d_i}{}^T \boldsymbol{x}_{d_j}$ を文書間の類似度と定義する。

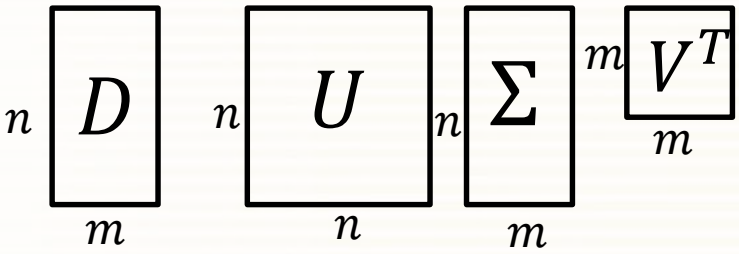
$\boldsymbol{x}_{d_i}{}^T \boldsymbol{x}_{d_j}$ は文書ベクトル間の内積であり、余弦類似度に比例することからも文書間の類似度とみるのは妥当である。

潜在意味解析（Latent Semantic Analysis）

行列 $D^T D$ を文書の類似度行列と呼ぶ。

行列 D の特異値分解を考える

$$D = U \Sigma V^T$$



U, V : 直交行列

$(UU^T = U^T U = I_n)$

$(VV^T = V^T V = I_m)$

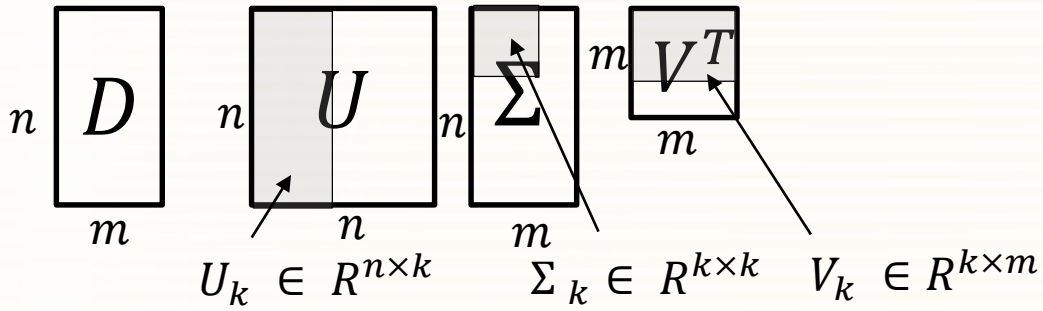
$n \geq m$ の場合

$$\Sigma = \begin{bmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_m \\ \hline & \mathbf{0} & \end{bmatrix}$$

特異値 $(|\sigma_1| \geq \cdots \geq |\sigma_m|)$

潜在意味解析（Latent Semantic Analysis）

k 個の大きい特異値を用いて、文書行列を近似することができる



$$D_k = U_k \Sigma_k V_k^T$$

この文書行列を用いた時の文書の類似度行列は

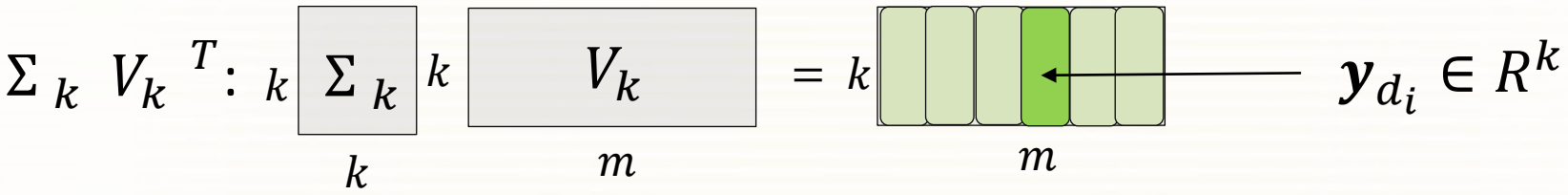
$$D_k^T D_k = \left(U_k \Sigma_k V_k^T\right)^T U_k \Sigma_k V_k^T$$

$$= V_k \Sigma_k U_k^T U_k \Sigma_k V_k^T$$

$$= V_k \Sigma_k \Sigma_k V_k^T \quad （U_k は直交行列なので）$$

潜在意味解析（Latent Semantic Analysis）

$$D_k{}^T D_k = V_k \, \Sigma_k \, \Sigma_k \, V_k{}^T = \left(\Sigma_k \, V_k{}^T \right)^T \Sigma_k \, V_k{}^T$$



文書の類似度行列$D_k{}^T D_k$は以下のように表すことができる。

$$D_k{}^T D_k = \begin{bmatrix} \boldsymbol{y}_{d_1}{}^T \\ \vdots \\ \boldsymbol{y}_{d_m}{}^T \end{bmatrix} [\boldsymbol{y}_{d_1} \quad \cdots \quad \boldsymbol{y}_{d_m}]$$

潜在意味解析（Latent Semantic Analysis）

もとの文書の類似度行列 $D^T D$ は以下であった。

$$D^T D = \begin{bmatrix} \boldsymbol{x}_{d_1}{}^T \\ \boldsymbol{x}_{d_2}{}^T \\ \vdots \\ \boldsymbol{x}_{d_m}{}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{d_1} \boldsymbol{x}_{d_2} \cdots \boldsymbol{x}_{d_m} \end{bmatrix}$$

$D_k{}^T D_k$ と $D^T D$ を比較すると、

文書ベクトル $\boldsymbol{x}_d \in R^n$ が $\boldsymbol{y}_d \in R^k$ に置き換わっている

見ることができる。

すなわち、高次元の文書ベクトル $\boldsymbol{x}_d$ が低次元ベクトル $\boldsymbol{y}_d$ で表現できることを示唆している。

潜在意味解析（Latent Semantic Analysis）

低次元ベクトル $\boldsymbol{y}_d$ の計算について

$$U = \begin{matrix} n \\ \end{matrix} \overset{\begin{matrix} k & n-k \end{matrix}}{\begin{bmatrix} U_k & U_{n-k} \end{bmatrix}} \qquad \Sigma = \begin{matrix} n \\ m-k \end{matrix} \overset{\begin{matrix} k & m-k \end{matrix}}{\begin{bmatrix} \Sigma_k & \\ & \Sigma_{m-k} \end{bmatrix}} \qquad V = \begin{matrix} m \\ \end{matrix} \overset{\begin{matrix} k & m-k \end{matrix}}{\begin{bmatrix} V_k & V_{m-k} \end{bmatrix}} \begin{matrix} \\ m \end{matrix}$$

$$D = \begin{bmatrix} U_k & U_{n-k} \end{bmatrix} \begin{bmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{m-k} \end{bmatrix} \begin{bmatrix} {V_k}^T \\ {V_{m-k}}^T \end{bmatrix}$$

$$= \begin{bmatrix} U_k\Sigma_k & U_{n-k}\Sigma_{m-k} \end{bmatrix} \begin{bmatrix} {V_k}^T \\ {V_{m-k}}^T \end{bmatrix}$$

$$= U_k\Sigma_k{V_k}^T + U_{n-k}\Sigma_{m-k}{V_{m-k}}^T$$

潜在意味解析（Latent Semantic Analysis）

両辺に左から ${U_k}^T$ をかけると

$$ {U_k}^T D = {U_k}^T U_k \Sigma_k {V_k}^T + {U_k}^T U_{n-k} \Sigma_{m-k} {V_{m-k}}^T $$

Uは直交行列なので、 ${U_k}^T U_k$ は単位行列

Uは直交行列なので、 ${U_k}^T U_{n-k}$はゼロ行列

$$ = \Sigma_k {V_k}^T $$

$$ = [\boldsymbol{y}_{d_1} \quad \cdots \quad \boldsymbol{y}_{d_m}] $$

${U_k}^T D$ によってすべての低次元ベクトル$\boldsymbol{y_d}$ が求められることになる

潜在意味解析（Latent Semantic Analysis）

同様に

$$D\,D^T = \begin{bmatrix} \boldsymbol{x}_{\omega_1}{}^T \\ \boldsymbol{x}_{\omega_2}{}^T \\ \vdots \\ \boldsymbol{x}_{\omega_n}{}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\omega_1} \boldsymbol{x}_{\omega_2} \cdots \boldsymbol{x}_{\omega_n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{\omega_1}{}^T \boldsymbol{x}_{\omega_1} & \cdots & \boldsymbol{x}_{\omega_1}{}^T \boldsymbol{x}_{\omega_n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}_{\omega_n}{}^T \boldsymbol{x}_{\omega_1} & \cdots & \boldsymbol{x}_{\omega_n}{}^T \boldsymbol{x}_{\omega_n} \end{bmatrix}$$

行列 $D\,D^T$ の(i,j)成分　$\boldsymbol{x}_{\omega_i}{}^T \boldsymbol{x}_{\omega_j}$ は、単語 ω_i と ω_j が同じ文書に出現していれば大きな値をとることになる。同じ文書に出現しているということは、単語 ω_i と ω_jの意味が似ていると解釈することができるので、$\boldsymbol{x}_{\omega_i}{}^T \boldsymbol{x}_{\omega_j}$ を単語間の類似度と定義する。

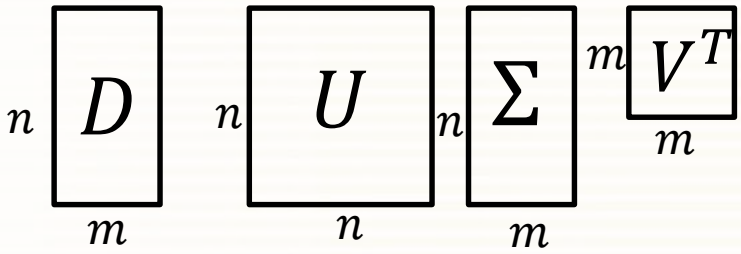
$\boldsymbol{x}_{\omega_i}{}^T \boldsymbol{x}_{\omega_j}$ は単語ベクトル間の内積であり、余弦類似度に比例することからも単語間の類似度とみるのは妥当である。

潜在意味解析（Latent Semantic Analysis）

行列 D D^T を単語の類似度行列と呼ぶ。

行列 D の特異値分解を考える

$$D = U \Sigma V^T$$



U, V : 直交行列

$(UU^T = U^T U = I_n)$

$(VV^T = V^T V = I_m)$

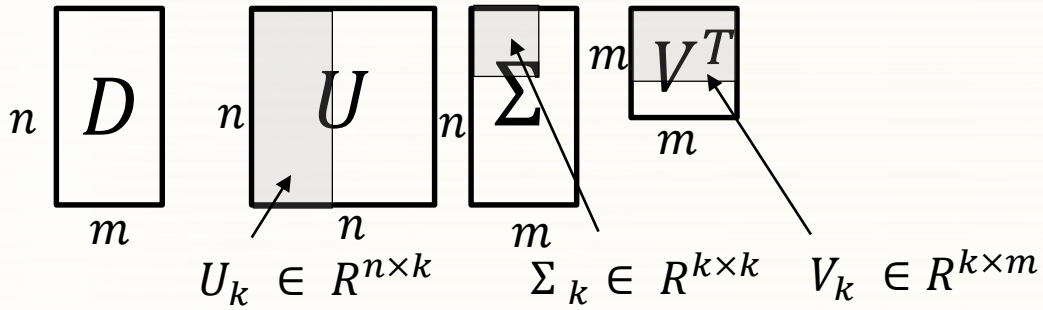
$n \geq m$ の場合

$$\Sigma = \begin{bmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_m \\ \hline & \mathbf{0} & \end{bmatrix}$$

特異値 $(|\sigma_1| \geq \cdots \geq |\sigma_m|)$

潜在意味解析（Latent Semantic Analysis）

k 個の大きい特異値を用いて、文書行列を近似することができる



$$D_k = U_k \Sigma_k {V_k}^T$$

この文書行列を用いた時の単語の類似度行列は

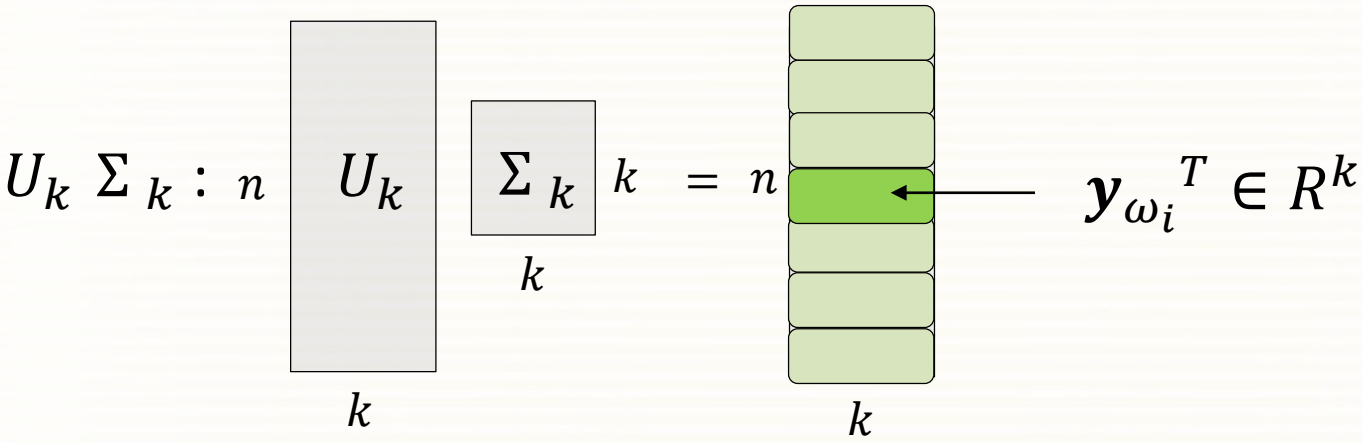
$$D_k {D_k}^T = U_k \Sigma_k {V_k}^T \left(U_k \Sigma_k {V_k}^T\right)^T$$

$$= U_k \Sigma_k {V_k}^T V_k \Sigma_k {U_k}^T$$

$$= U_k \Sigma_k \Sigma_k {U_k}^T \quad （V_k は直交行列なので）$$

潜在意味解析（Latent Semantic Analysis）

$$D_k\ {D_k}^T = U_k\ \Sigma_k\ (U_k\ \Sigma_k)^T$$



単語の類似度行列 ${D_k\ {D_k}^T}$ は以下のように表すことができる。

$$D_k\ {D_k}^T = \begin{bmatrix} {\boldsymbol{y}_{\omega_1}}^T \\ \vdots \\ {\boldsymbol{y}_{\omega_n}}^T \end{bmatrix} [\boldsymbol{y}_{\omega_1} \quad \cdots \quad \boldsymbol{y}_{\omega_n}]$$

潜在意味解析（Latent Semantic Analysis）

もとの単語の類似度行列 $D\ D^T$ は以下であった。

$$D\ D^T = \begin{bmatrix} \boldsymbol{x}_{\omega_1}{}^T \\ \boldsymbol{x}_{\omega_2}{}^T \\ \vdots \\ \boldsymbol{x}_{\omega_n}{}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\omega_1} & \boldsymbol{x}_{\omega_2} & \cdots & \boldsymbol{x}_{\omega_n} \end{bmatrix}$$

$D_k\ D_k{}^T$ と $D\ D^T$ を比較すると、

単語ベクトル $\boldsymbol{x}_\omega \in R^m$ が $\boldsymbol{y}_\omega \in R^k$ に置き換わっている

見ることができる。

すなわち、高次元の単語ベクトル $\boldsymbol{x}_\omega$ が低次元ベクトル $\boldsymbol{y}_\omega$

で表現できることを示唆している。

音声処理

スマートスピーカー

インターネットの接続と音声認識・音声操作が可能な、AIアシスタントを搭載したスピーカー。声だけであらゆる操作ができる。

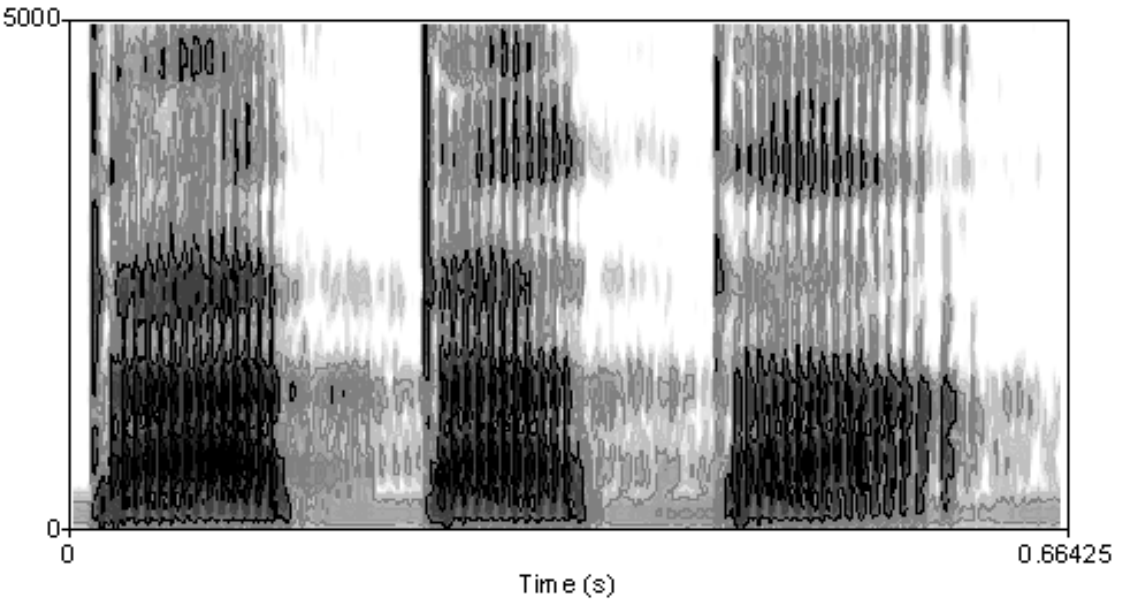
Google Home, Amazon Alexa, Apple Siri ...

Hey Google,

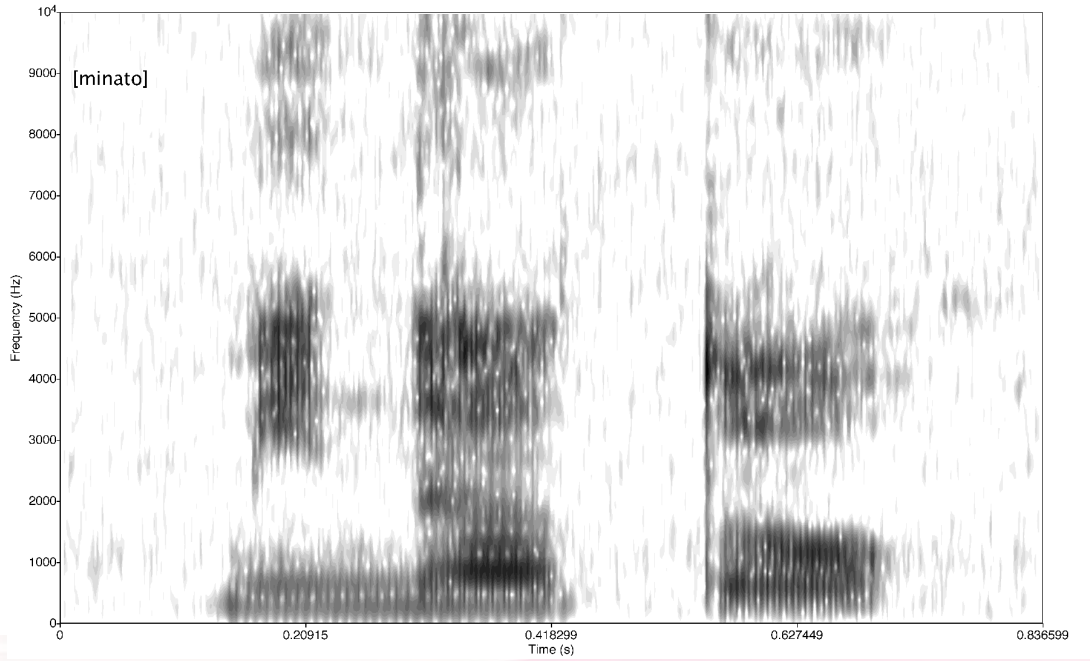
What’s the weather today?
Latest news, please.
Play some songs by Utada Hikaru.
Timer for 10 minutes!
Bonjour!

音のスペクトログラム

横軸に時間、縦軸に周波数を取り、成分ごとの
音声パワー(dB)を擬似カラーもしくは白黒で示したもの。



男性の声「タタタ」



女性の声「みなと」

出典：Wikipedia

音声認識の流れ(1)

音　声
　↓ (マイクロフォン)
アナログ電気信号
　↓ (A/D変換器)
デジタル波形データ
　↓ (窓かけ)
一定区間のフレームに区切る
　↓ (離散フーリエ変換)
周波数成分(スペクトルデータ)
　↓ (絶対値の2乗)
周波数ごとのパワースペクトル　例：256次元のベクトル
　↓ (メルフィルターバンク)
微細構造を落としたスペクトル外形　例：24次元のベクトル

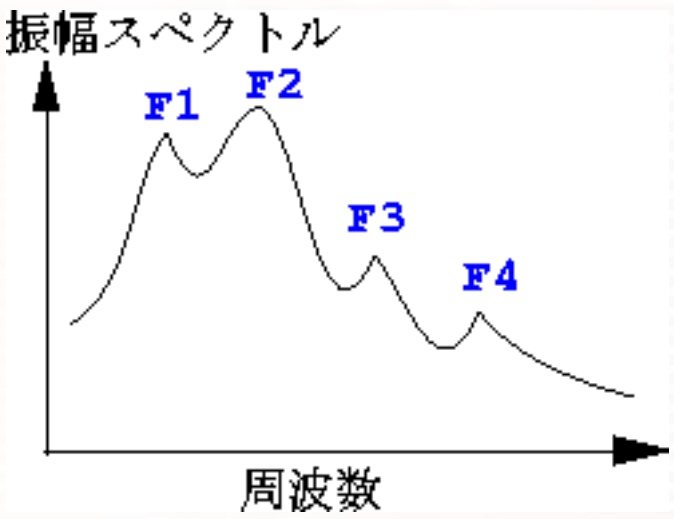
音声認識の流れ(2)

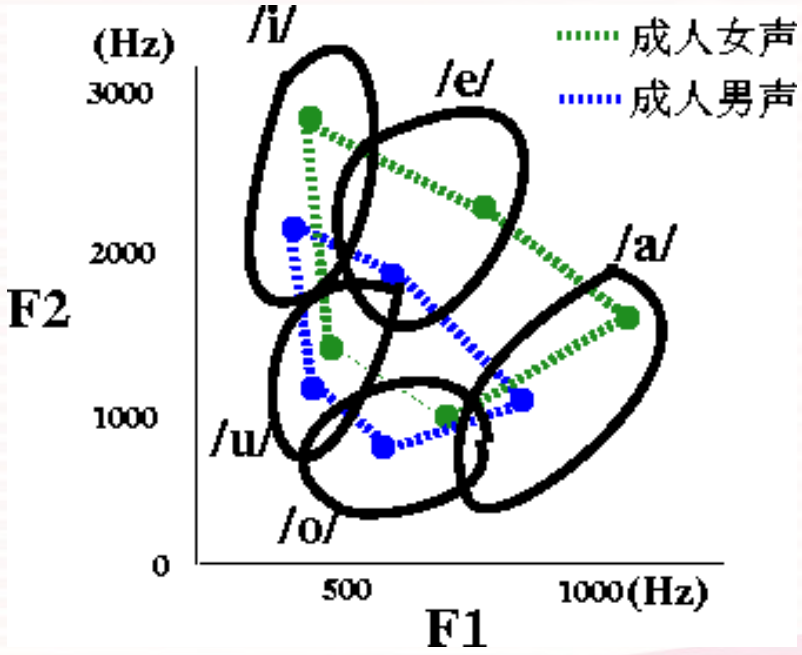
周波数ごとのパワースペクトル　例：256次元のベクトル
　↓ (人間の聴覚特性に合わせたメルフィルターバンク)
微細構造を落としたスペクトル外形　例：24次元のベクトル
　↓ (対数)
対数メルスペクトル
　↓ (離散コサイン変換)
メル周波数スペクトラム係数　　例：13次元のベクトル

最近は（ニューラルネットを用いた音声認識では）、
最後まで次元削減を行わずに、途中のスペクトルを
特徴量としてそのまま用いることも多い。

フォルマントと音素

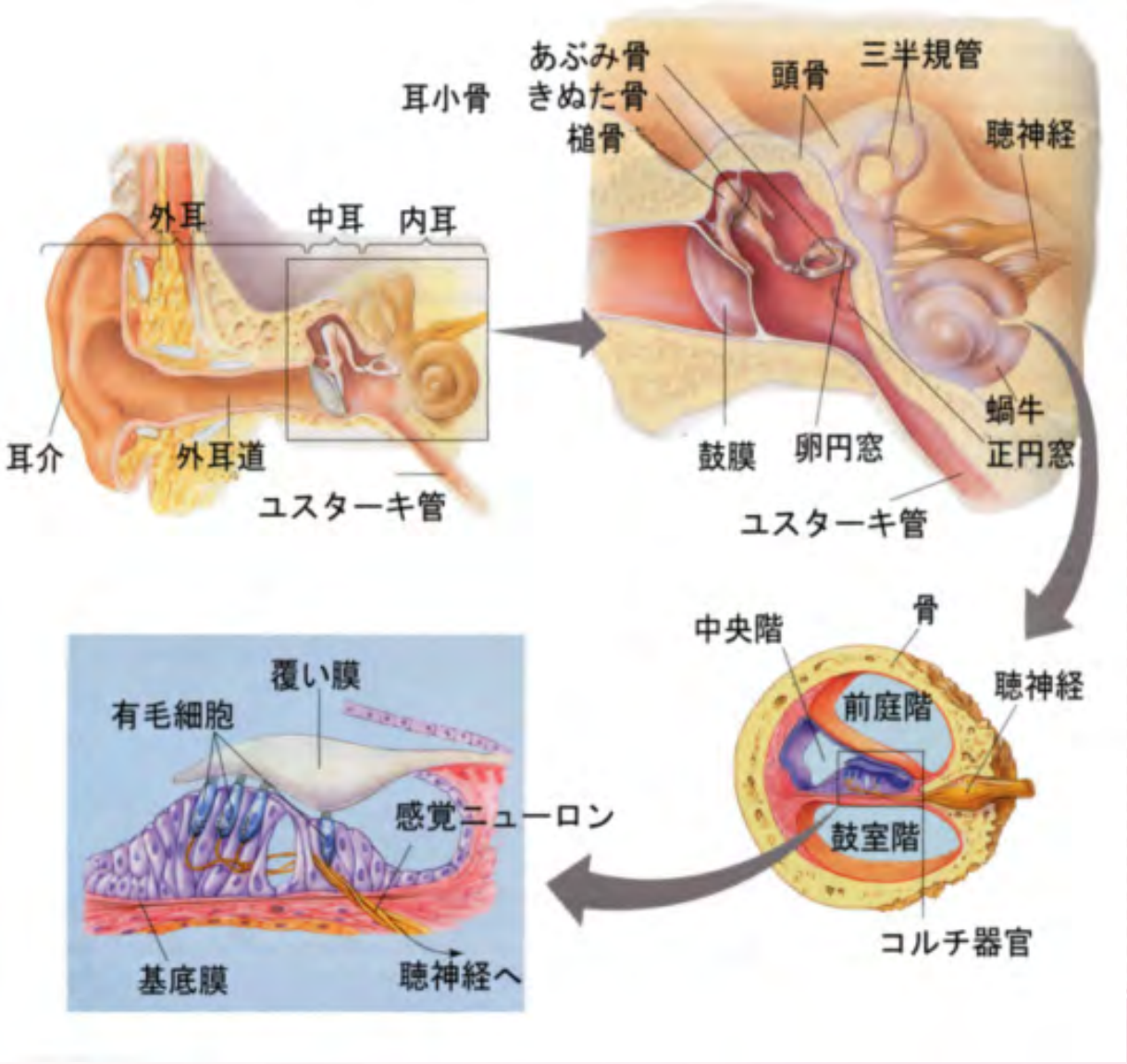
フォルマント：話者の音声のスペクトルのピーク。
周波数の低い順に、第1、第2、…と数字をあて、
F1, F2, … と表記する。





図の出典：https://media.sys.wakayama-u.ac.jp/kawahara-lab/LOCAL/diss/diss7/S3_6.htm

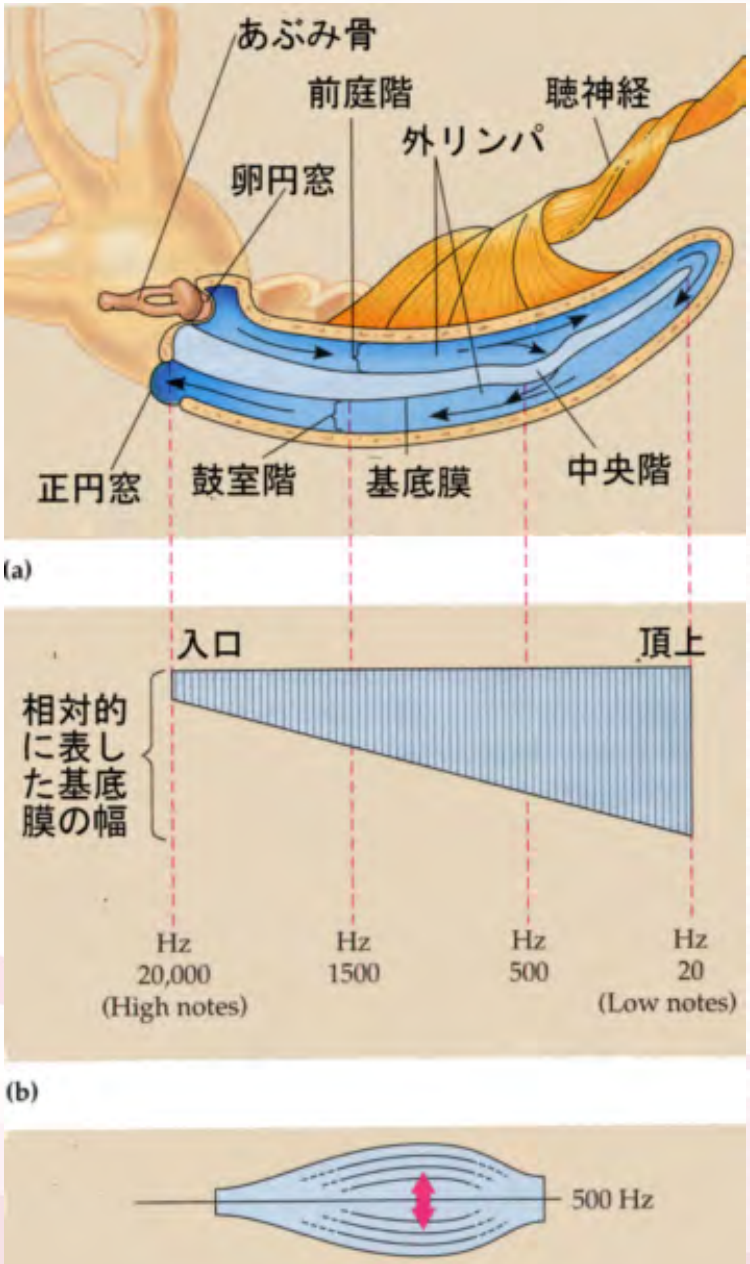
参考：ヒトの聴覚



音
↓
外耳道
↓
中耳の鼓膜
↓
耳小骨
↓
蝸牛の卵円窓
↓
前庭階の外リンパ
↓
中央階の内リンパ
↓
基底膜の振動
↓
有毛細胞の興奮
↓
聴神経を経て中枢へ

図の出典：http://www.tmd.ac.jp/artsci/biol/textlife/sense.htm

参考：ヒトの聴覚



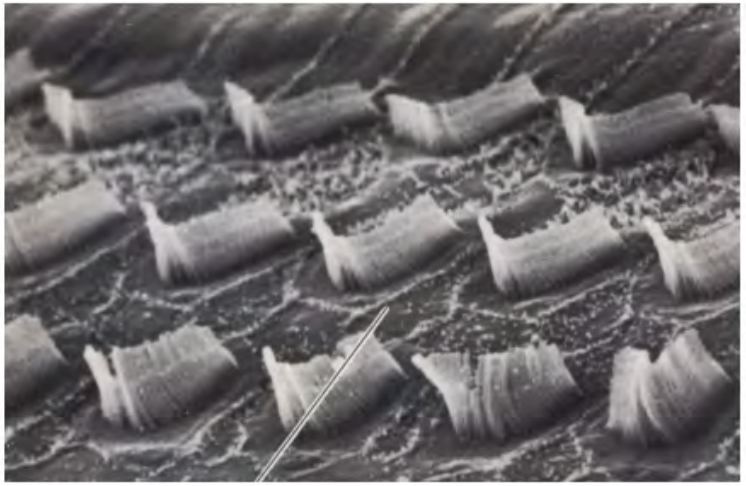
基底膜の幅は入り口が狭く（0.04 mm）、
一番奥（蝸牛の頂上）が広い（0.5 mm）。

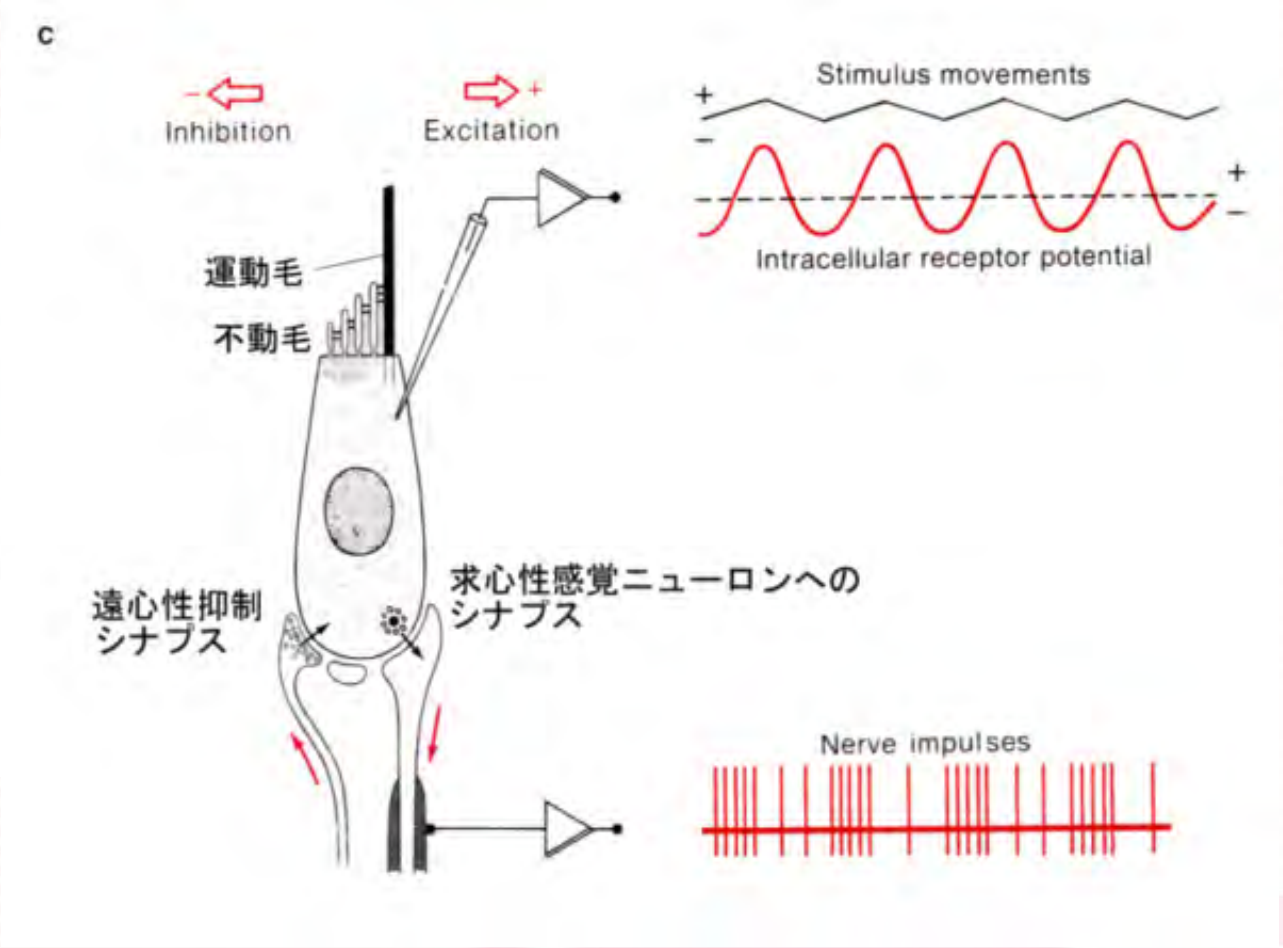
異なる高さ（振動数）をもつ音は、
異なる部分の基底膜を振動させ、
異なる有毛細胞を興奮させる。
有毛細胞の場所に応じて、異なる神経線維
がその信号を中枢に伝える。

図の出典：
http://www.tmd.ac.jp/artsci/biol/textlife/sense.htm

参考：ヒトの聴覚

有毛細胞





図の出典：
http://www.tmd.ac.jp/artsci/biol/textlife/sense.htm

音の知覚と対数

音圧レベル　$Lp = 20 \times \log (p/p_0)$　　単位：デシベル(dB)

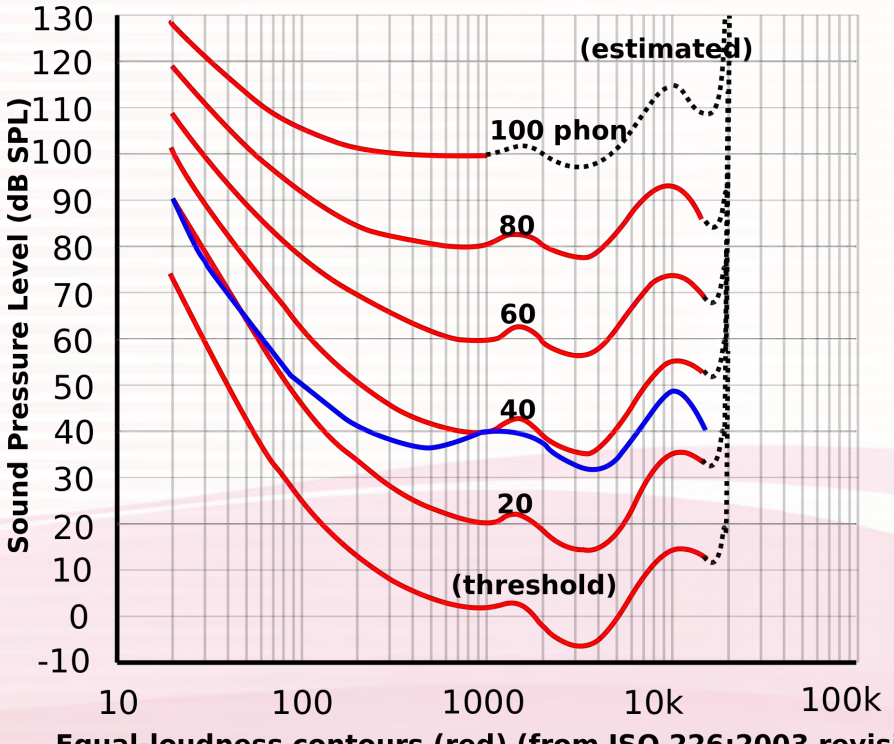
p: 音圧、p0: 基準となる音圧 (20 μPa)

Weber-Fechnerの法則

刺激強度 R に対して、それと
刺激 R + ΔR の刺激を弁別できる
最初の ΔR に対し、
ΔR / R = 定数 が成り立つ。

→ ΔE = ΔR / R を積分すると、
刺激 R に対する感覚量 E は
E = C × log R　(C: 定数)

等ラウドネス曲線（wikipedia）



特徴量の作成(1)

加法性歪の除去：環境雑音のパワーを減算

話者が話していない区間の平均値を周波数帯ごとに求め、
パワースペクトルから除去する（0以下は0に丸める）。

乗法性歪の除去：

対数メルスペクトルから、話者が話している区間の
周波数帯ごとの平均値を除去する。
マイクロフォンまでの距離やアンプゲインの影響
（伝達特性）を差し引き、ピークを鮮明化できる

特徴量の作成(2)

動的特徴量

前後のフレームの静的特徴量（対数メルスペクトルなど）の差分（Δ特徴量）、2階差分（ΔΔ特徴量）

標準化

特徴量の平均を0、分散を1にする
音響モデルとしてGMMを用いる場合は不要（平均・分散のばらつきも含めて学習するから）。
ニューラルネットを用いる場合に必要。

音響モデル

入力：フレームごとの特徴量（静的、Δ、ΔΔ）
前後数フレームを結合する
例：特徴量72個に、前後5フレームずつを結合した場合、72 × 11 = 792次元ベクトル。

出力：音声要素（セノン）

ニューラルネットワークで学習する。
・学習用のデータを使って、入力した特徴量に対する正解セノンの事後確率が高くなるようにパラメータを更新する（確率的勾配降下法）

音声要素（セノン）

音素は前後に来る音素によってスペクトログラムが変化する。(コンテキスト依存音素)

コンテキスト依存音素の種類は数万個であるが、
決定木を使ってクラスタリングを行うと、
その数を数千個に減らすことができ、これを
音素グループとして扱う。

音響モデルの学習アルゴリズム(1)

畳み込みニューラルネットワーク (CNN)
　重なって配置された部分領域ごとに複数のフィルターを適用

リカレントニューラルネットワーク (RNN)
長短期記憶ネットワーク (LSTM)
　長い時間(数百ミリ秒)の情報を記憶したい。
　RNNでは、各層への入力について、1つ前の時点でのその層の出力を下層からの入力と束ねる。→ 過去の入力の影響が残るが、急速に消え去ってしまう。
　LSTMでは、内部メモリを忘却ゲート・入力ゲート・出力ゲートで管理し、必要に応じて長く保持できる。

音響モデルの学習アルゴリズム(2)

混合正規分布モデル (GMM)
　音声スペクトルの形状を多次元正規分布の重ね合わせで表現し、音声要素(セノン)との対応を学習

隠れマルコフモデル (HMM)
　状態を遷移する確率と、状態が出現する確率を同時に学習

End-to-End 音声認識モデル
　音響モデルと言語モデルを1つのRNNにまとめる。
　入力が音響特徴量、出力が単語やサブワードの列になる。

言語モデル（前回の講義も参照）

N-gram

先行するN-1単語の条件付き確率としての当該単語の出現確率

Word2Vec

辞書にある単語を数百次元のベクトルに変換したもので、意味の似た単語は似たベクトルとなる。

言語モデル

パープレキシティ

単語列に対して、1単語あたりの平均の確率（相乗平均）の逆数（つまり単語の選択肢の数）。小さいほど言語モデルの性能は高く、音声認識の誤り率も小さくなる。

単語あたりのエントロピーをHとすると、

$H = - \Sigma \, 1/n \cdot P(w1, ..., wn) \log_2 P(w1, ..., wn)$

$Perplexity = 2^H$

音響モデルと言語モデルの統合

音響特徴量Xに対して、確率 $P(W \mid X)$ が最大になるような単語列 W を見つける。

ベイズの定理により

$P(W \mid X) \propto P(X \mid W)\, P(W)$

であるから、右辺の対数をとって

$\log P(X \mid W) + \log P(W)$

を最大にすればよい。第1項が音響モデルによる音響スコア、第2項が言語モデルによる言語スコアになる。

言語スコアに重みをつけ、単語数による罰則を課すと

$\log P(X \mid W) + \alpha \cdot \log P(W) + \beta \cdot Kw$

を最大にすることになる。

画像処理

コンピュータビジョン

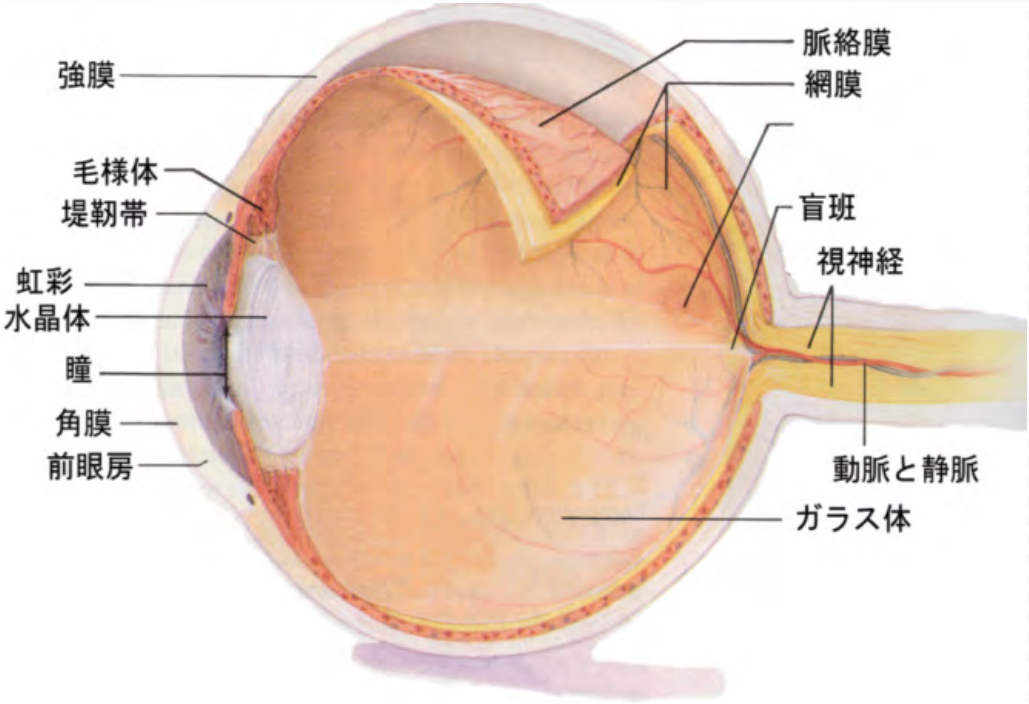
入力：静止画像もしくは動画像（映像）

出力：数値データ （画像解析の場合）
　　画像もしくは動画像（画像合成の場合）

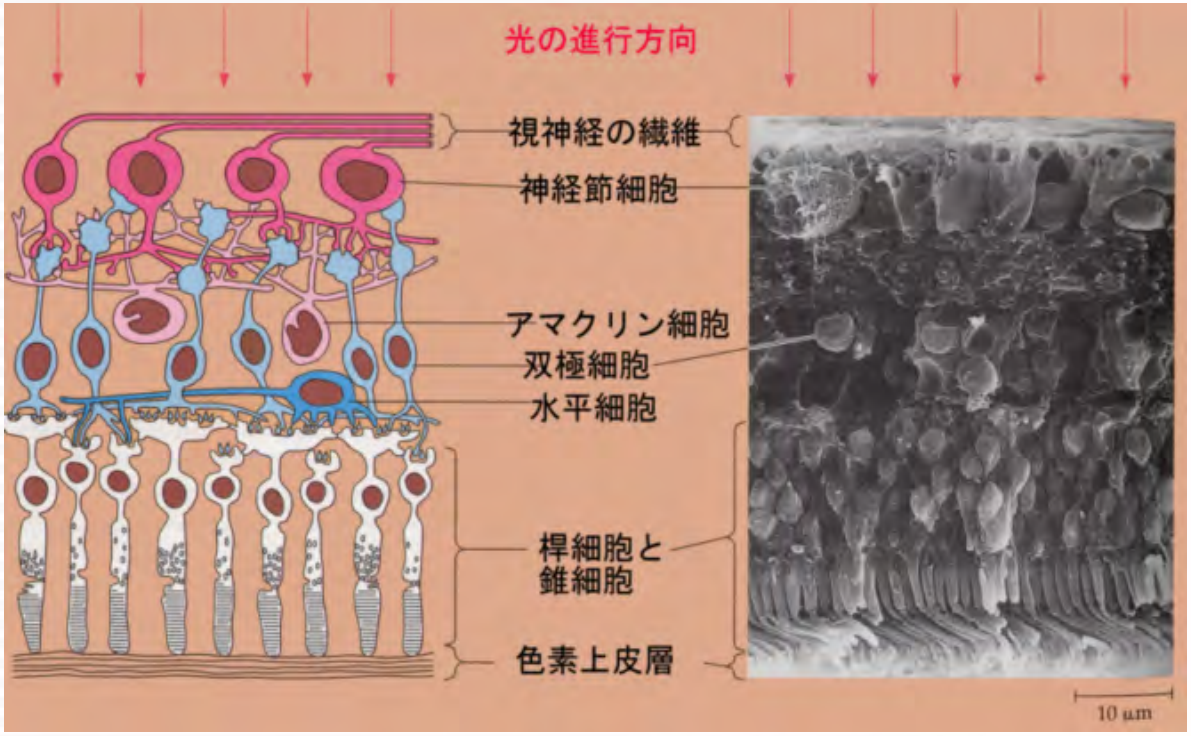
画像解析の応用：
(1) 画像計測
　地図作成、自動運転、環境測定
(2) 画像認識
　文字認識、自動翻訳、指紋認識、
　異常検知・見守り、医用画像診断

画像合成の応用：自由視点画像、欠損修復、拡張現実 (AR)

参考：ヒトの視覚



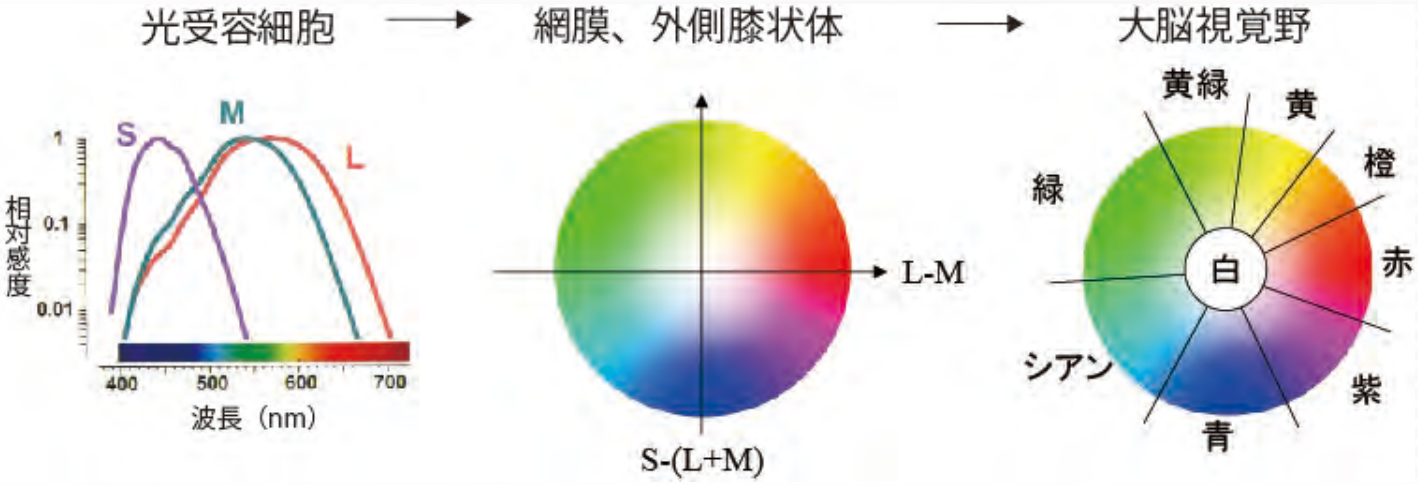
網膜

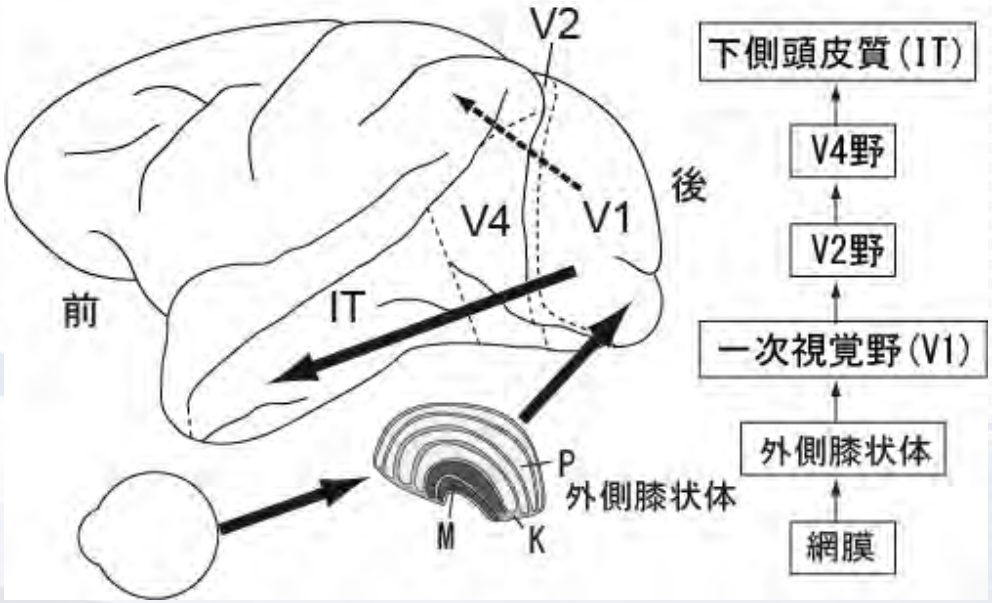


図の出典：http://www.tmd.ac.jp/artsci/biol/textlife/sense.htm

参考：ヒトの視覚

錐体細胞に含まれるオプシンには3種類あり、
赤色、緑色、青色の光を吸収しやすい（ヒトの場合）。





左図：マカクザルの視覚経路

図の出典：脳科学辞典
bsd.neuroinf.jp/wiki/色選択性細胞

画像のデータ表現

画素(pixel)の標本化（サンプリング）：空間方向のデジタル化

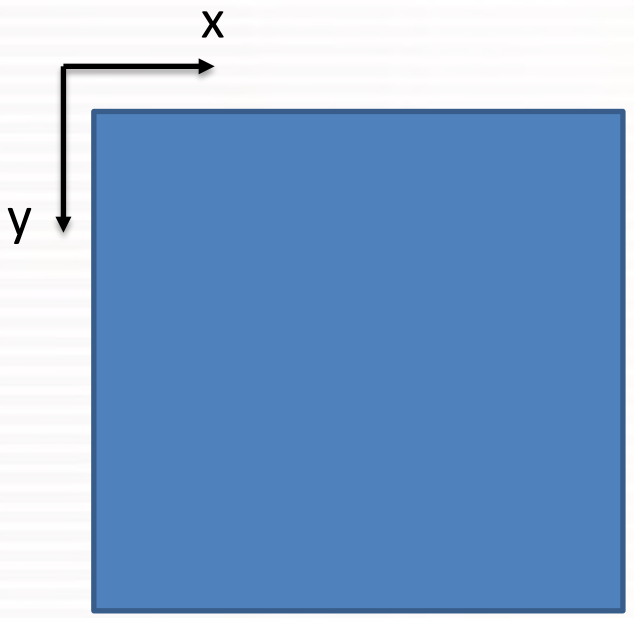
画素値の量子化：画素の明るさの情報を2^n階調
(n bitの場合)で表現

グレースケール画像：明暗のみで表現される。各画素に1つの値(明度)のみが記録

カラー画像：各画素がR, G, Bの3つの値をもっている。
RGBを混ぜて表現することで、人間が知覚可能な大半の色が表現できる。

画像のデータ表現: OpenCVの場合

```
import cv2 as cv
dat = cv.imread(“example.jpg”)
b = dat[60][120][0]
g = dat[60][120][1]
r = dat[60][120][2]
print(“R G B=“, r, g, b)
cv.imwrite(“output.jpg”, dat)
```



注意点（画像処理ライブラリに依存する）
60がy, 120がx
b, g, rの順番

画像の特徴量

画像のヒストグラム
　R, G, Bそれぞれの成分、もしくはグレースケール化した画像の明度値について、その分布（各値が画像上に出現する頻度）を可視化したもの。

OpenCVで使える特徴量の例：
Haar-Like 特徴
　画像の一部を矩形状に切り出し、明暗差を検出する

LBP (local binary pattern) 特徴
　局所的な輝度の分布の組み合わせを検出

HOG (histogram of oriented gradients) 特徴
　局所的な輝度の勾配方向の分布の組み合わせを検出

画像のフィルタ処理（畳み込み処理）

（例）平均化フィルタ

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

*

a	b	c
d	**e**	f
g	h	i

=

	(a+b+…+i)/9	

画像のフィルタ処理（畳み込み処理）

ガウシアンフィルタ

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

1次微分フィルタ

0	0	0
-1	0	1
0	0	0

or

0	-1	0
0	0	0
0	1	0

画像のフィルタ処理（畳み込み処理）

Prewittフィルタ
= 1次微分 * 平均化

-1	0	1
-1	0	1
-1	0	1

Sobelフィルタ
= 1次微分 * ガウシアン

-1	0	1
-2	0	2
-1	0	1

画像のフィルタ処理（畳み込み処理）

ラプラシアンフィルタ
(2次微分)

0	1	0
1	-4	1
0	1	0

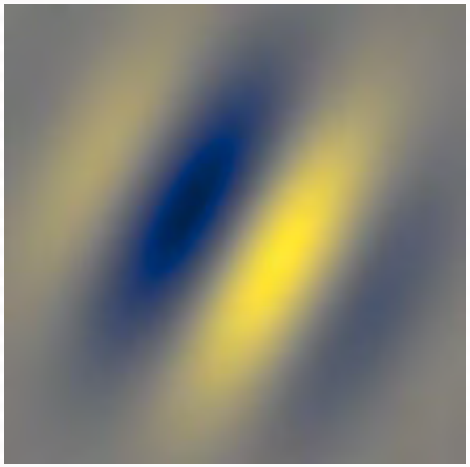
先鋭化フィルタ
(2次微分のk倍を引く)

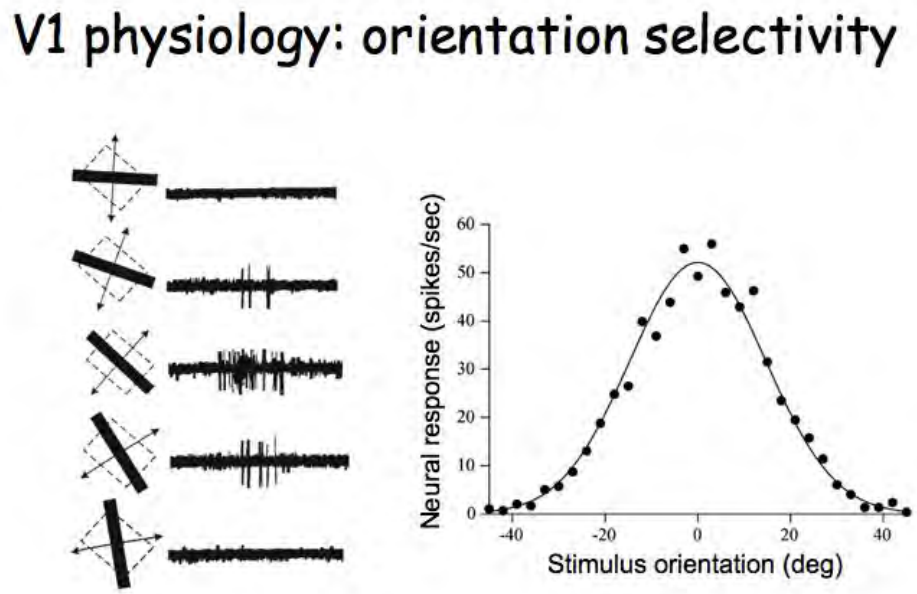
0	-k	0
-k	1-4k	-k
0	-k	0

Gaborフィルタと猫の一次視覚野

方向ごとに特定の周波数を検出することができるフィルタ。

ガウス関数(2次元正規分布の密度関数)と正弦波の積の形。
図の出典: Wikipedia





Hubel と Wiesel の研究 (1968)
www.cns.nyu.edu/~david/courses/perception/lecturenotes/V1/lgn-V1.html

特徴点検出の例：エッジの角

1. 微分フィルタを用いて、各座標 (u,v) の輝度勾配ベクトル (Ix, Iy) を求める

2. (Ix, Iy)の分布を、主成分分析(PCA)を用いて調べる。
   分散共分散行列の固有値をλ_1, λ_2とするとき、

   固有値が2つとも大きい：エッジの角である
   固有値が1つだけ大きい：直線的なエッジである
   固有値が2つとも小さい：平坦なパターンである

   ⇒ 固有値の閾値を設定して判定する (Shi-Tomasi オペレータ)
   ⇒ 判別式 $\lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$ を用いる (Harris オペレータ)

特徴点検出の例：2枚の画像の対応点探索

1. 点周辺の小領域の画像の明度値を比較する。

2. スケールの変化や、回転がある場合は、SIFT特徴量を用いる：

   輝度勾配ベクトルのヒストグラムから、画像の回転方向を決め、補正した後、4×4の小領域に分割する。

   8方位を用いて輝度勾配のヒストグラムを作成し、16の領域でヒストグラムを連結すると、128次元ベクトルができる。

3. 座標を変えながら、2枚の特徴ベクトルの間の距離を計算。

畳み込みニューラルネットワーク
Convolutional neural network (CNN) (復習)

画像認識で最もよく用いられる。

入力層
↓
畳み込み層：フィルタ処理で特徴をつかむ（何度も繰り返す）
↓
プーリング層：特徴を保ちつつデータを圧縮する（何度も繰り返す）
↓
全結合層：特徴的な所見を抽出・判定する
↓
出力層

畳み込みニューラルネットワーク
Convolutional neural network (CNN) (復習)

パディング：フィルタ処理によって画像の端が
欠けないように、画像の端をあらかじめ0で埋めて
からフィルタ処理を行う。

Max Pooling：各領域の最大値で代表させる。
多少の物体の位置ずれや傾きもカバーできる。

参考：実際の画像解析に取り組む

メディカルAIコース：オンライン講義資料
https://japan-medical-ai.github.io/medical-ai-course-materials/

扱われている題材：
MRI画像のセグメンテーション
血液の顕微鏡画像からの細胞検出
配列解析
モニタリングデータの時系列解析

画像生成

深層生成モデル(GANなど)による顔写真の自動生成

言語モデル GPT-3

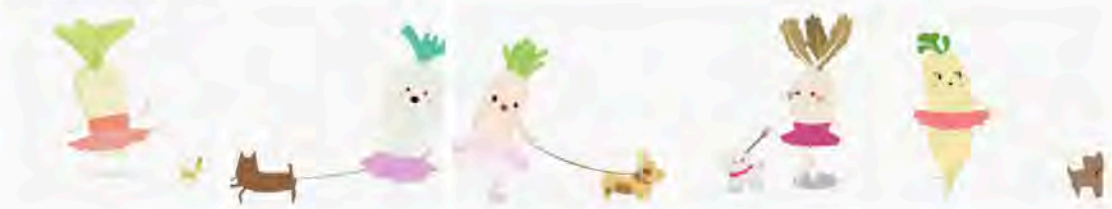
右の出典：
openai.com/blog/dall-e/

DALL·E[1] is a 12-billion parameter version of GPT-3 trained to generate images from text descriptions, using a dataset of text–image pairs. We've found that it has a diverse set of capabilities, including creating anthropomorphized versions of animals and objects, combining unrelated concepts in plausible ways, rendering text, and applying transformations to existing images.

TEXT PROMPT

an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES



View more or edit prompt↓

TEXT PROMPT

an armchair in the shape of an avocado [...]

AI-GENERATED IMAGES

View more or edit prompt↓

3次元画像解析

世界座標系の3次元座標 **Vw**
↓ (合同変換)
カメラ座標系の3次元座標 **Vc**
↓ (画面上へ投影)
画面上の2次元座標 **vu**

Vc = R**Vw** + t　　R: 直交行列 (回転行列)、 t: 並進ベクトル

回転行列 R のオイラー角表現： R = Rx(α) Ry(β) Rz(γ)

空間の想像力を養うためのクイズ
（世界座標系とカメラ座標系の変換）

無限に広い xy 平面の世界に自分が移り住んだと想像してほしい。
その xy 平面上には放物線 $y = x^2$ が描かれており、負の y 軸上に設置された高台からその放物線全体を写生したとする。
放物線の全体はどのように描かれることになるか？
特に地平線の近くで放物線はどのような様子になっているか？
実際に放物線の絵を描いてみよ。

解答：http://www.math.tohoku.ac.jp/~kuroki/LaTeX/OpenCampus2005.pdf