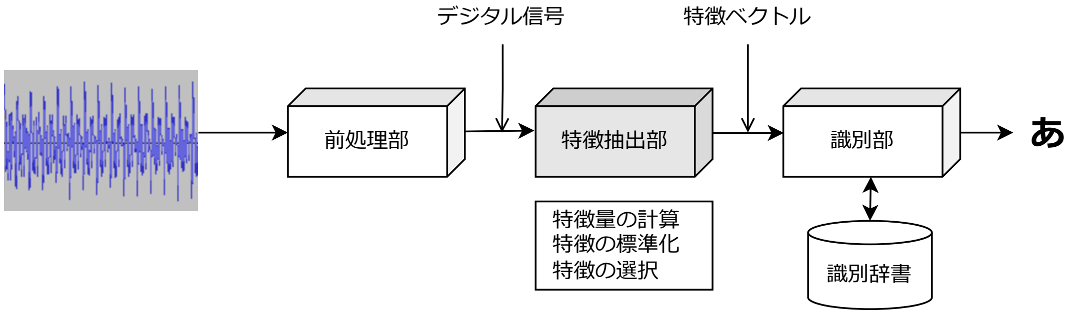
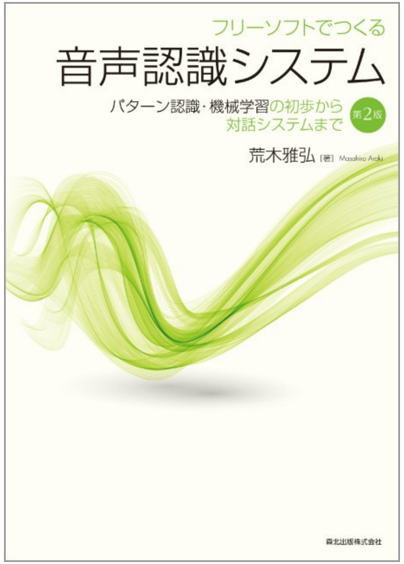
3. パターンの特徴を調べよう



- 3.1 変動に強い特徴とは
- 3.2 特徴のスケールを揃える
- 3.3 特徴は多いほどよいか

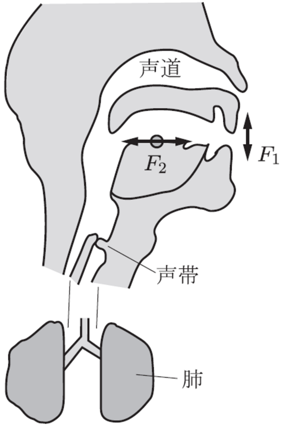


- 荒木雅弘: 『フリーソフトでつくる音声認識システム(第2版)』(森北出版, 2017年)
- スライドとJupyter notebook
- サポートページ

3.1 変動に強い特徴とは

3.1.1 音声の場合 (1/6)

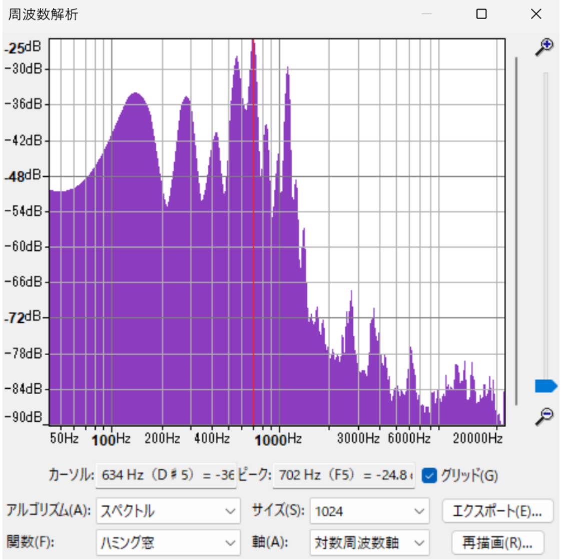
- 音素の違いとは
  - 声帯振動の有無(パルス波 / 雑音)
  - 母音:声道(口の開き具合・舌の位置など)の変形による共振周波数の違い
  - 子音:声道や口唇の変化パターンの違い



(a) 発声と調音の仕組み

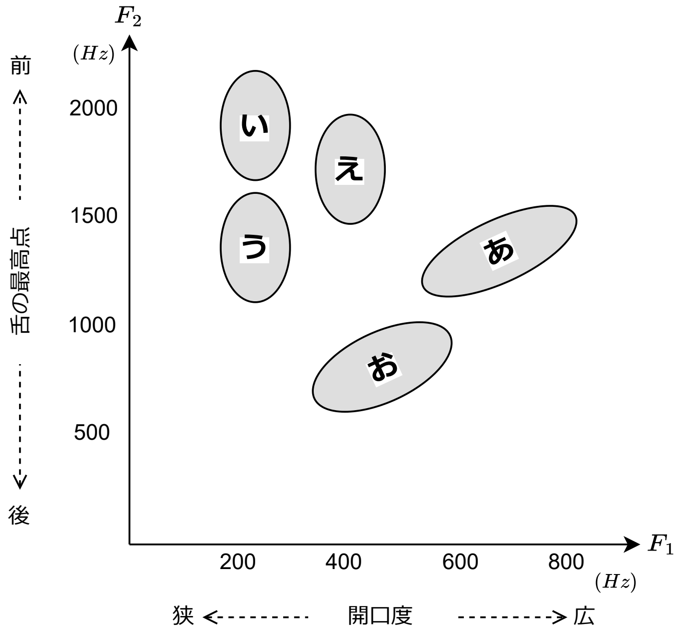
3.1.1 音声の場合（2/6）

- 音声のスペクトル
  - 音声波形中の一定時間を切り出し、フーリエ変換によって周波数成分に分解したもの
  - 音編集ソフト Audacity で表示したスペクトル



3.1.1 音声の場合（3/6）

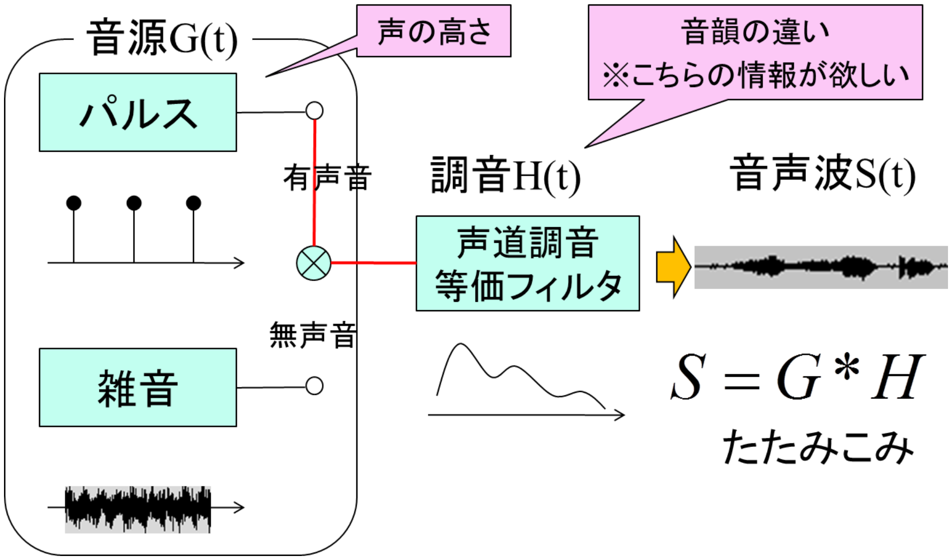
- 共振周波数はスペクトルのピークとして現れる



(b) 日本語母音識別のための特徴空間（男声）

3.1.1 音声の場合（4/6）

- 音声生成過程のモデル



3.1.1 音声の場合（5/6）

- 調音フィルタ特性 H（スペクトル包絡）の取り出し
  - 音声信号 $S = G * H$

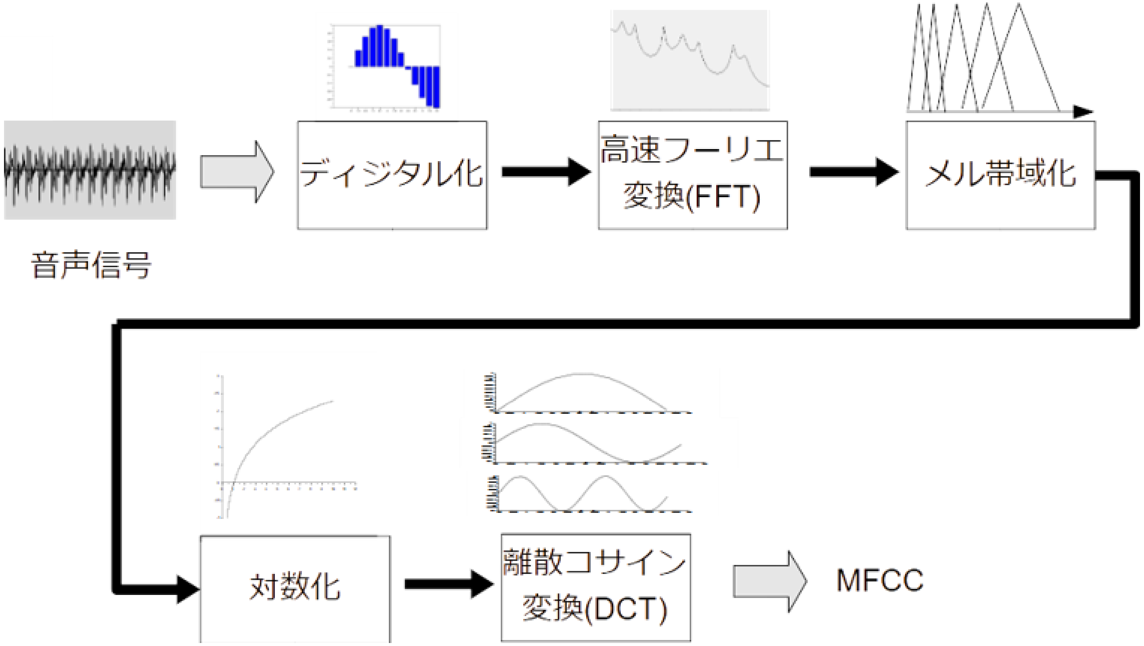
    $\downarrow$ フーリエ変換
  - スペクトル $S = G \cdot H$

    $\downarrow$ 絶対値・メルフィルタ・対数化（人間の聴覚特性を反映）
  - 対数スペクトル $\log |S|^2 = \log |G|^2 + \log |H|^2$

    $\downarrow$ 離散コサイン変換
  - ケプストラム $DCT \log |S|^2 = DCT \log |G|^2 + DCT \log |H|^2$
    - 音のスペクトルを信号とみなして周波数分析したもの
    - 音源 G（スペクトルの調波構造）の成分が高周波域、調音フィルタ H の成分が低周波域に現れる

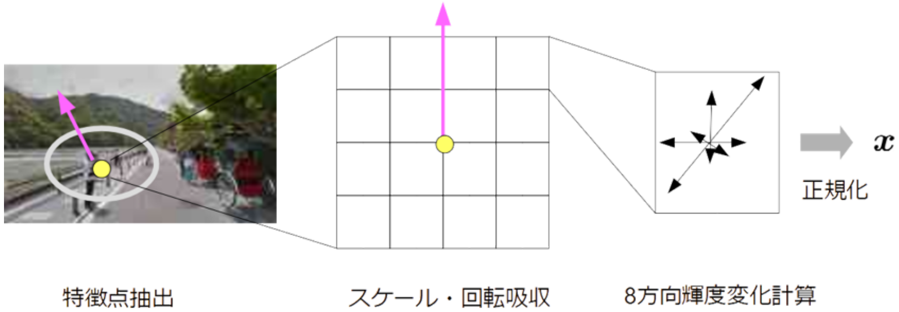
3.1.1 音声の場合 (6/6)

- MFCC (Mel Frequency Cepstrum Coefficient)
  - 離散コサイン変換結果の低次(12次まで)の係数をスペクトル包絡の情報とみなす



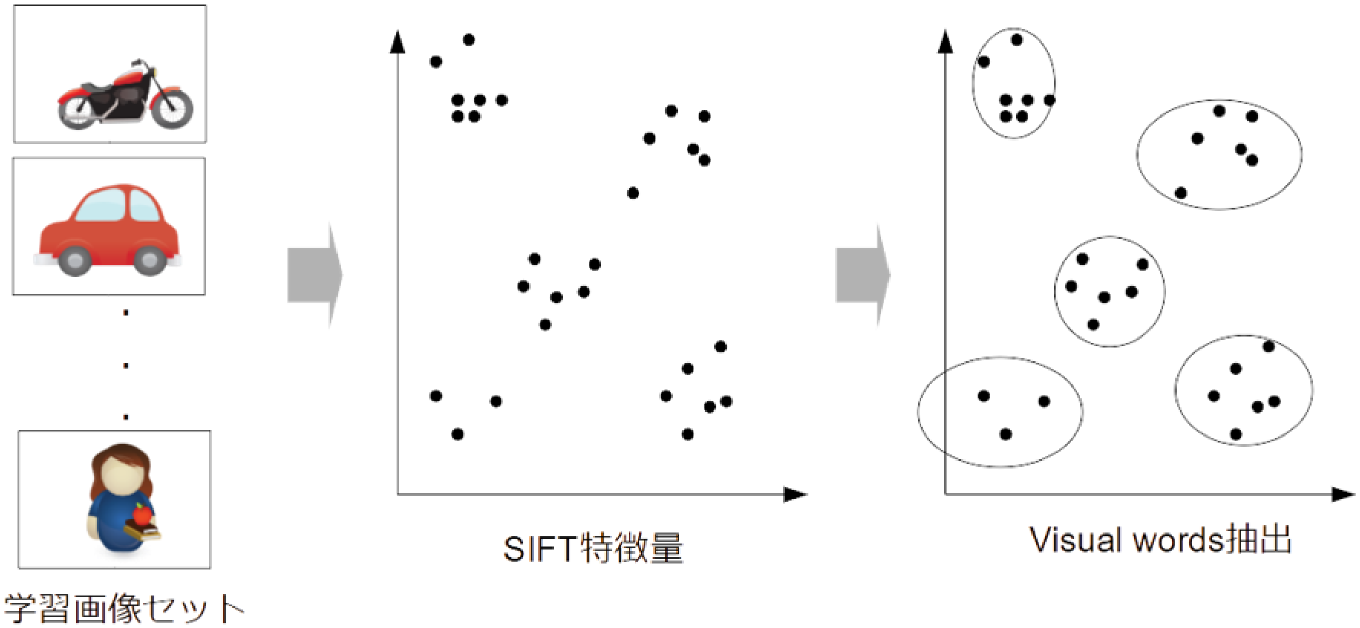
3.1.2 画像の場合 (1/2)

- 画像の変動
  - 明るさの変化, 拡大･縮小, 回転など
- SIFT特徴量
  - 2枚の画像の対応抽出などに有効



3.1.2 画像の場合 (2/2)

- Bag of Visual Words
  - SIFT特徴量の似ているベクトルを単語と見なし、出現頻度を特徴として識別問題に適用



(補足)自然言語処理の場合

- Bag of Words
  - 応用:文書分類
    - 例)商品レビューを 肯定的/否定的 に分類する
  - 文書を単語の多重集合(bag)とみなす
    - 文例)「顔認証はヤバいぐらい便利」
    - 形態素解析「顔認証　は　ヤバい　ぐらい　便利」

単語の種類数＝次元数

(0, ..., 0, 1, 0, ..., 1, 0, 1, ...)

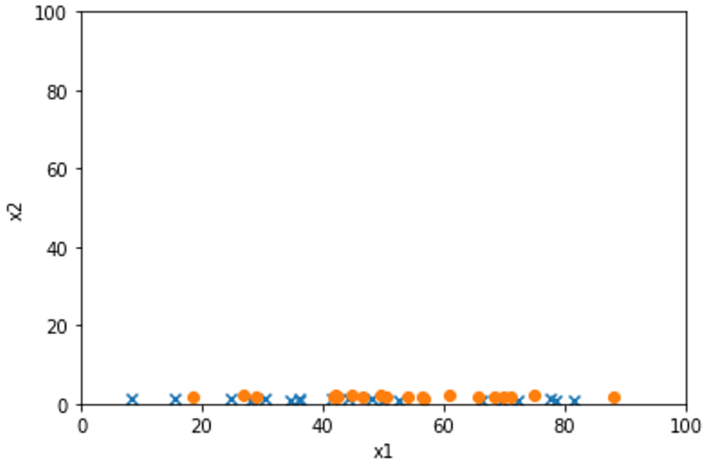
顔認証　ヤバい　便利

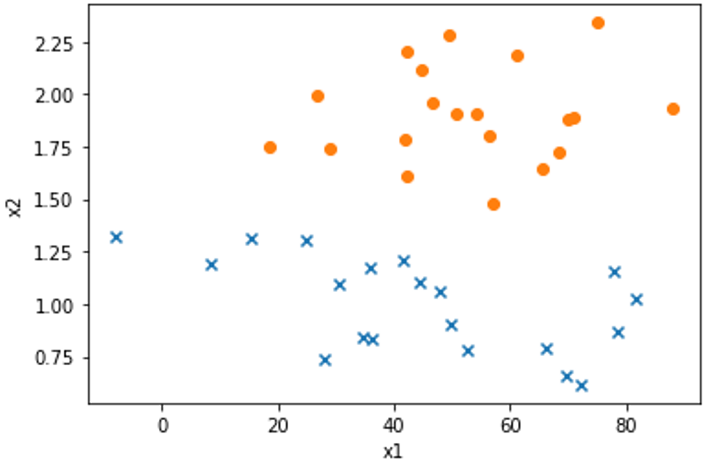
Positive

分類ラベル

3.2 特徴のスケールを揃える (1/2)

- 各軸で値のスケールが異なる場合
  - 値の標準化が必要





3.2 特徴のスケールを揃える (2/2)

- スケールの揃え方
  - 特徴空間の単位超立方体の体積を軸伸縮の前後で一定に保ち、かつパターン相互の距離を最小化 $\rightarrow$ 各軸の分散を等しくする
- 平均値を0にしておく
  - 学習における初期値の調整が不要
- 標準化の式(平均0、分散1への変換)

$$x'_i = \frac{x_i - m}{\sigma} \qquad m: \text{平均、} \sigma: \text{標準偏差}$$

3.3 特徴は多いほどよいか

3.3.1 偶然に見つかってしまってはまずい（1/4）

（1） 偶然の傾向とは

- 特徴は多いほどよいか
    - 特徴が多く、データ数が少ないと、偶然の傾向が現れるかもしれない
    - 特徴の次元数が高いほど、偶然の傾向が発見される可能性が高い

3.3.1 偶然に見つかってしまってはまずい（2/4）

（2）学習に必要なパターン数

- 超平面の容量 $2(d+1)$
  - $p(n,d)$：d 次元空間上で、適当に配置された n 個のパターンを任意に2クラスに分けたとき、超平面により線形分離できる確率

$$
\begin{aligned}
n < 2(d+1) &\Rightarrow p(n,d) \sim 1 \\
n = 2(d+1) &\Rightarrow p(n,d) = 1/2 \\
n > 2(d+1) &\Rightarrow p(n,d) \sim 0
\end{aligned}
$$

3.3.1 偶然に見つかってしまってはまずい (3/4)

- 例題3.3
  - データ数: 4、次元数: 1 $\Rightarrow p(4,1) = 1/2$

○	○	○	○
○	○	○	×
○	○	×	○
○	○	×	×
○	×	○	○
○	×	○	×
○	×	×	○
○	×	×	×
×	○	○	○
×	○	○	×
×	○	×	○
×	○	×	×
×	×	○	○
×	×	○	×
×	×	×	○
×	×	×	×

(a) データの配置

○	○	○	○
○	○	○	×
○	○	×	○
○	○	×	×
○	×	○	○
○	×	○	×
○	×	×	○
○	×	×	×
×	○	○	○
×	○	○	×
×	○	×	○
×	○	×	×
×	×	○	○
×	×	○	×
×	×	×	○
×	×	×	×

(b) 二つに分離可能

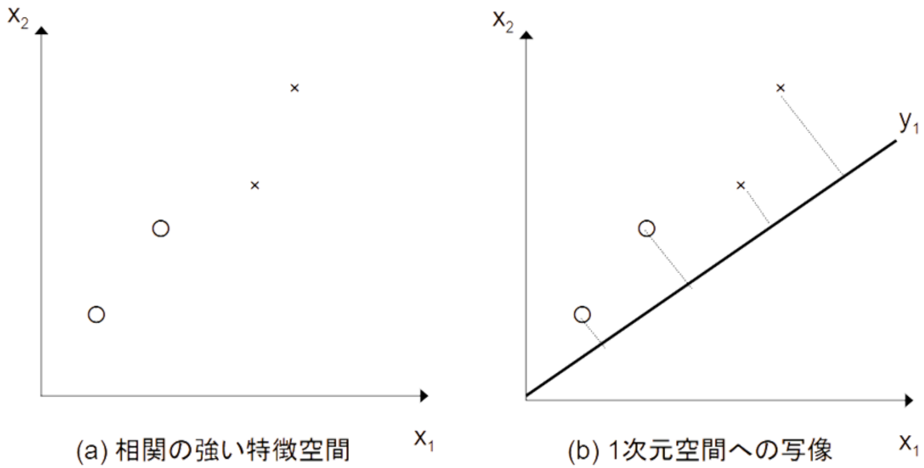
3.3.1 偶然に見つかってしまってはまずい（4/4）

（3）見つかるはずのないものが見つかった？

- $n >> 2(d+1)$ のとき
  - もし、この条件で識別面が見つかったとしたら
    → 偶然には存在しえないものが見つかった
    → その識別面は必然的に存在していた

3.3.2 特徴を減らそう（1/2）

- （1）力業で次元を減らす → 全ての組み合わせを評価する
- （2）スマートに主成分分析

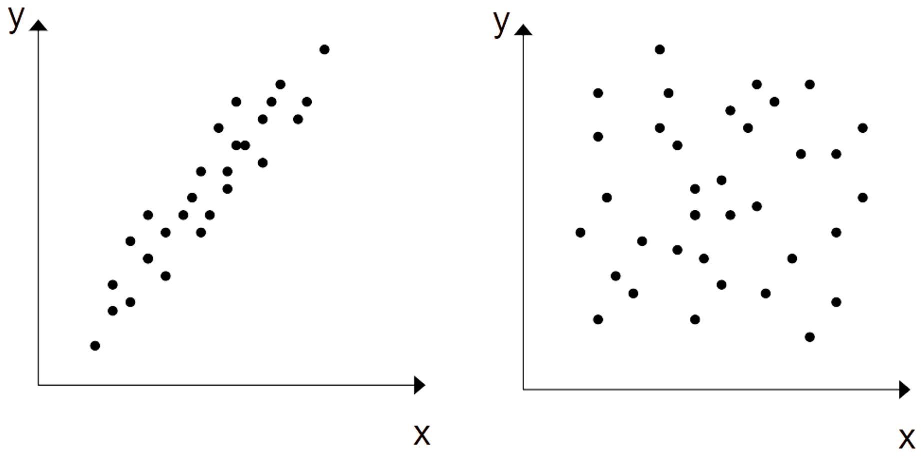


(補足)共分散行列とは (1/2)

- データの広がりを調べる $\rightarrow$ 共分散行列
- 1次元の場合
  - 平均 $m = \frac{1}{N}\sum_{x \in \chi} x$
  - 分散 $\sigma^2 = \frac{1}{N}\sum_{x \in \chi}(x - m)^2$
- 多次元の場合
  - 平均ベクトル $\boldsymbol{m} = \frac{1}{N}\sum_{\boldsymbol{x} \in \chi} \boldsymbol{x}$
  - 共分散行列 $\Sigma = \frac{1}{N}\sum_{\boldsymbol{x} \in \chi}(\boldsymbol{x} - \boldsymbol{m})(\boldsymbol{x} - \boldsymbol{m})^T$

(補足)共分散行列とは (2/2)

- 各軸の平均･分散が等しいデータを区別できる



$$\Sigma = \begin{pmatrix} x\text{の分散} & x\text{と}y\text{の相関} \\ x\text{と}y\text{の相関} & y\text{の分散} \end{pmatrix}$$

3.3.2 特徴を減らそう（2/2）

- 主成分分析（principal component analysis; PCA）
  - 主成分とは
    - データの分散が最大となる方向
  - PCA のアイディア
    - 主成分を分散の大きい順に少数抽出することで、データを少ない次元に投影（変換）する
  - PCA の手順
    1. データの標準化　$X_{std} = (X - m)/\sigma$
    2. 共分散行列の計算　$C = \frac{1}{N} X_{std}^T X_{std}$
    3. 共分散行列の固有値と固有ベクトルの計算　$Cv = \lambda v$　　λ：固有値, v：対応する固有ベクトル
    4. 固有値の大きい順に固有ベクトル（主成分）を並べ替える　$v_{sorted} = sort(v, by = \lambda)$
    5. データを主成分の空間に変換（投影）する　$X_{pca} = X_{std} \cdot v_{sorted}[: \tilde{d}]$

まとめ

- 特徴抽出部の役割
    - 特徴量の計算
        - 入力の種類および認識対象によって処理が異なる
    - 特徴の標準化
        - スケールの異なる特徴の識別に対する効果を公平にする
    - 特徴の選択
        - 実験的に有効な特徴を調べる
        - 低次元に変換する
- Jupyter notebook