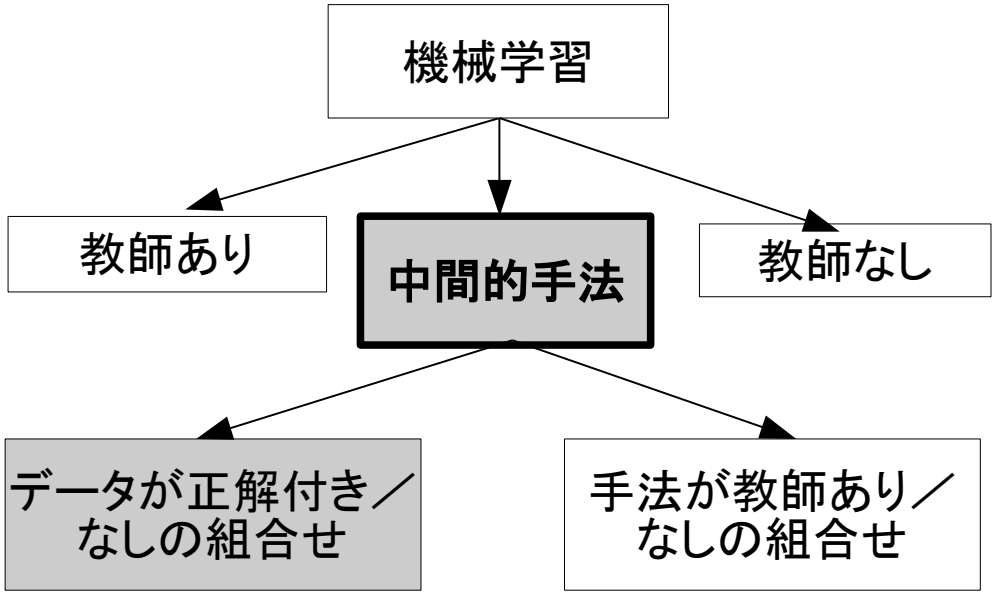
Section 4

- 中間的手法(13,14章)

中間的手法



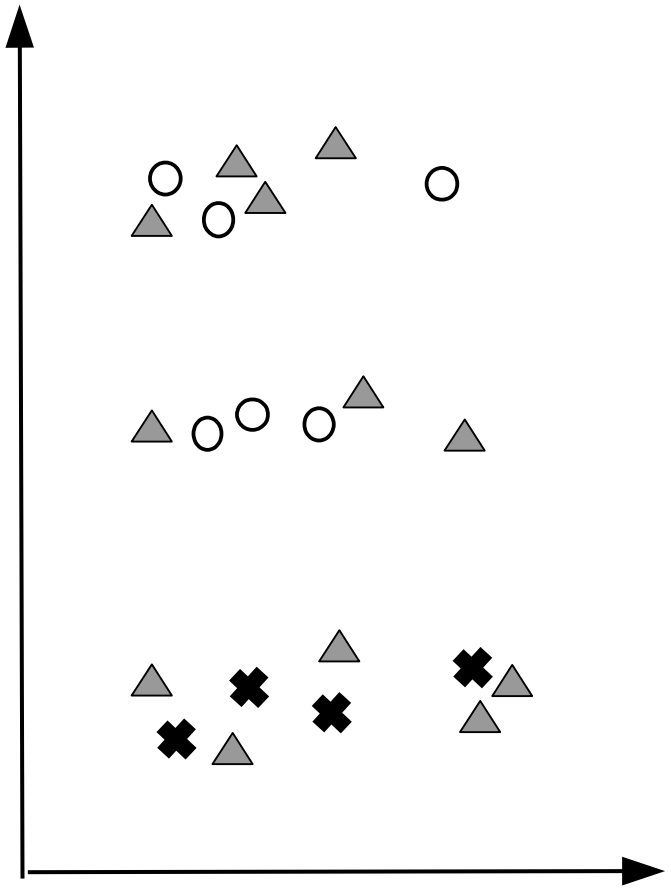
半教師あり学習

強化学習

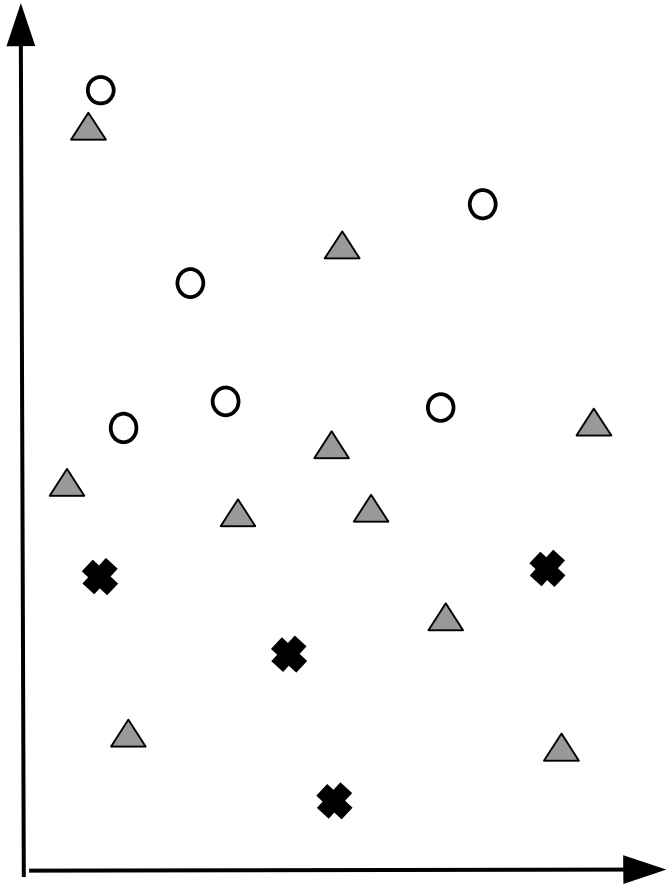
13.1 半教師あり学習とは

13.1.1 数値特徴の場合

- 半教師あり学習に適した数値特徴データの性質



半教師あり学習に適するデータ



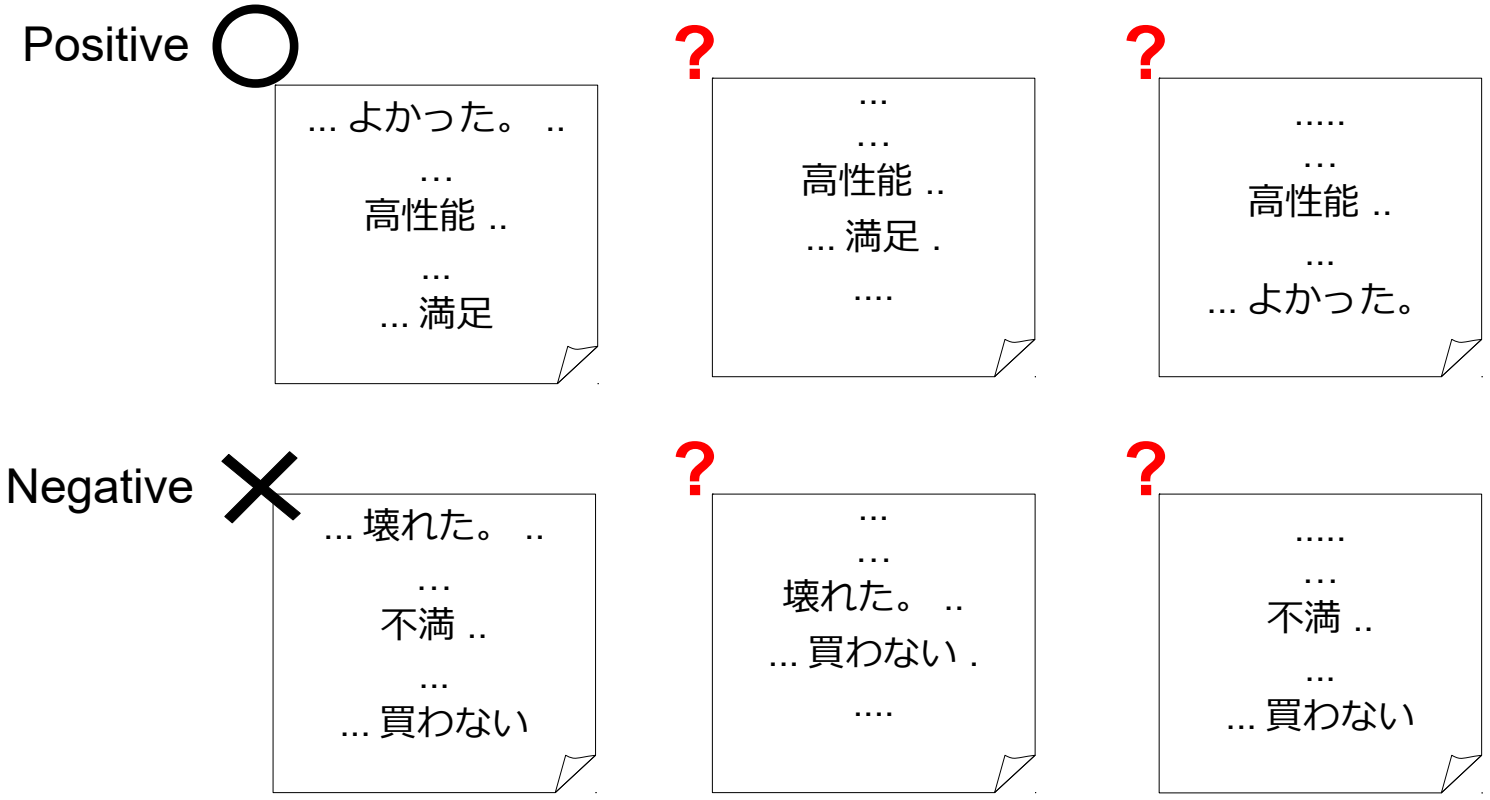
半教師あり学習に適さないデータ

13.1.1　数値特徴の場合

- 半教師あり学習が可能なデータ
  - 半教師あり平滑性仮定
    - 二つの入力が高密度領域で近ければ、出力も関連している
  - クラスタ仮定
    - もし入力が同じクラスタに属するなら、それらは同じクラスになりやすい
  - 低密度分離
    - 識別境界は低密度領域にある
  - 多様体仮定
    - 高次元のデータは、低次元の多様体上に写像できる
      - 多様体：局所的に線形空間と見なせる空間

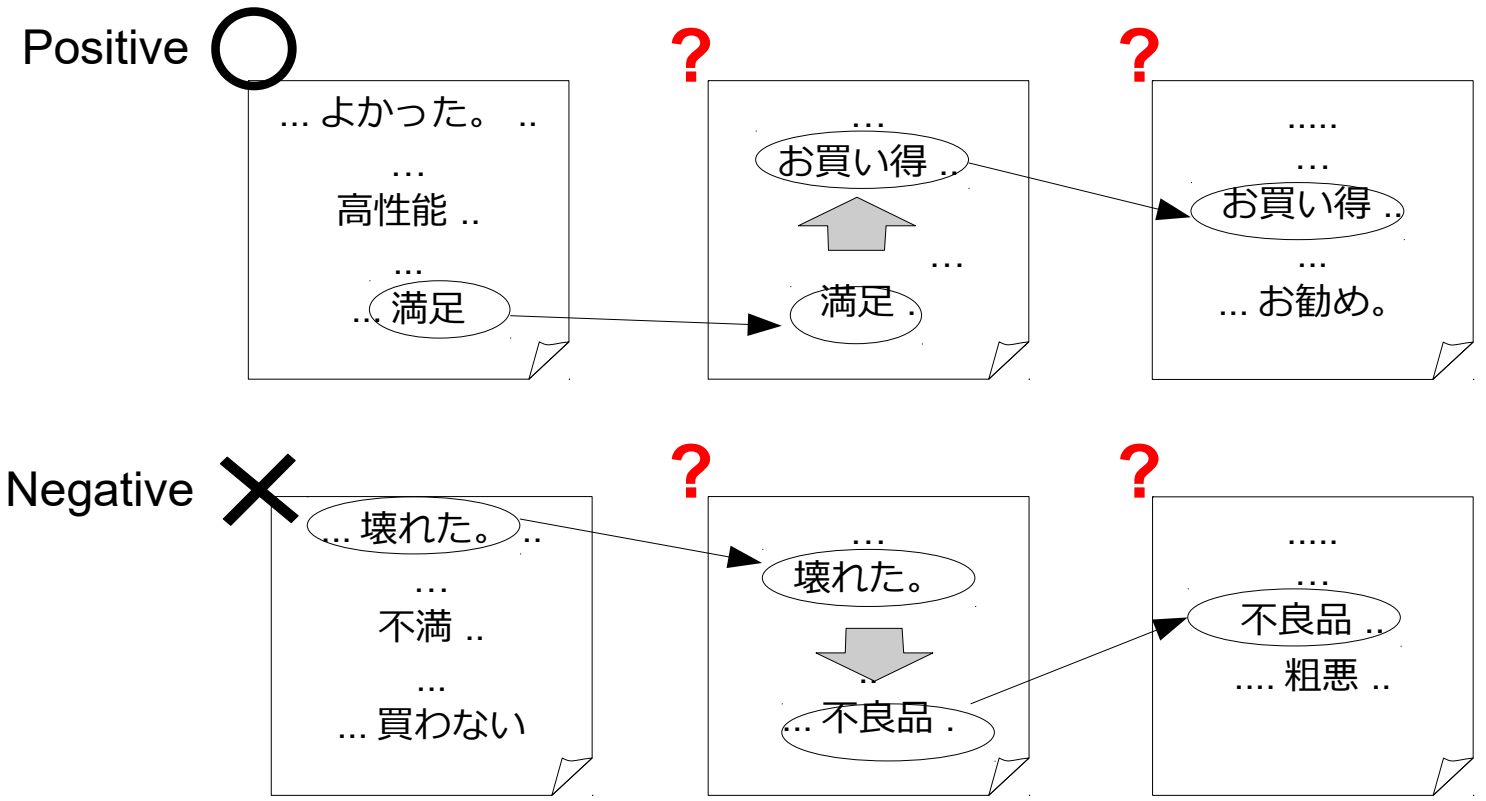
13.1.2 カテゴリ特徴の場合

- オーバーラップ
  - 文書からの評判分析の例



13.1.2 カテゴリ特徴の場合

- 特徴の伝播



13.1.3 半教師あり学習のアルゴリズム

- 半教師あり学習の基本的な考え方
  - 正解付きデータで識別器を作成
  - 正解なしデータで識別器のパラメータを調整
- 識別器に対する要求
  - 確信度の出力：正解なしデータに対する出力を信用するかどうかの判定に必要

13.2 自己学習

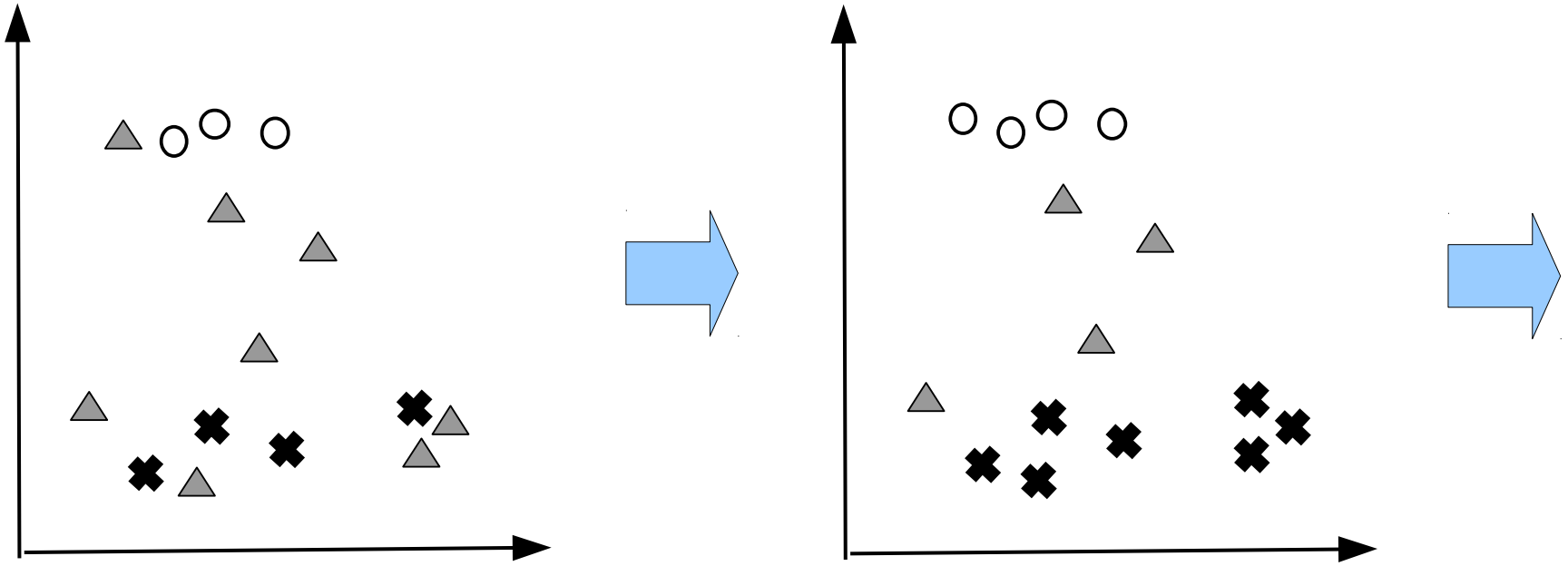
- 自己学習のアルゴリズム

1.正解付きデータで初期識別器を作成

2.正解なしデータの識別結果のうち、確信度の高いものを、正解付きデータとみなす

3.新しい正解付きデータで、識別器を学習

4. 2, 3 を繰り返す

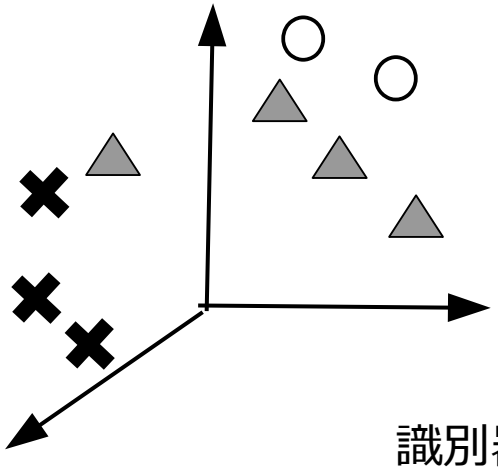


13.2 自己学習

- 自己学習の性質
  - クラスタ仮定や低密度分離が満たされるデータに対しては、高い性能が期待できる
  - 低密度分離が満たされていない場合、初期識別器の誤りが拡大してゆく可能性がある

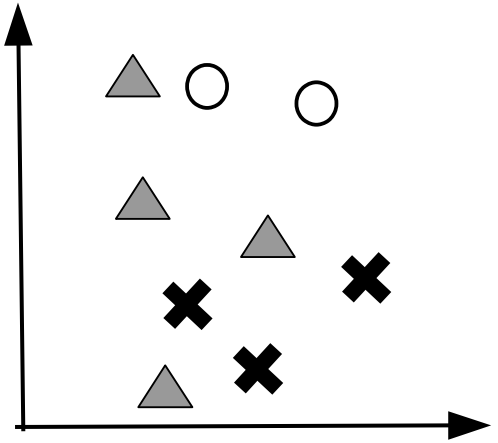
13.3 共訓練

- 共訓練とは
  - 判断基準が異なる識別器を交互に用いる
  - 片方の確信度が高いデータを、相手が正解付きデータとみなして学習



$\{x_1, x_2, x_3, x_4, x_5\}$

識別器 1　　識別器 2



識別器 1 で確信度の高い
正解なしデータ

識別器 2 の学習

識別器 1 の学習

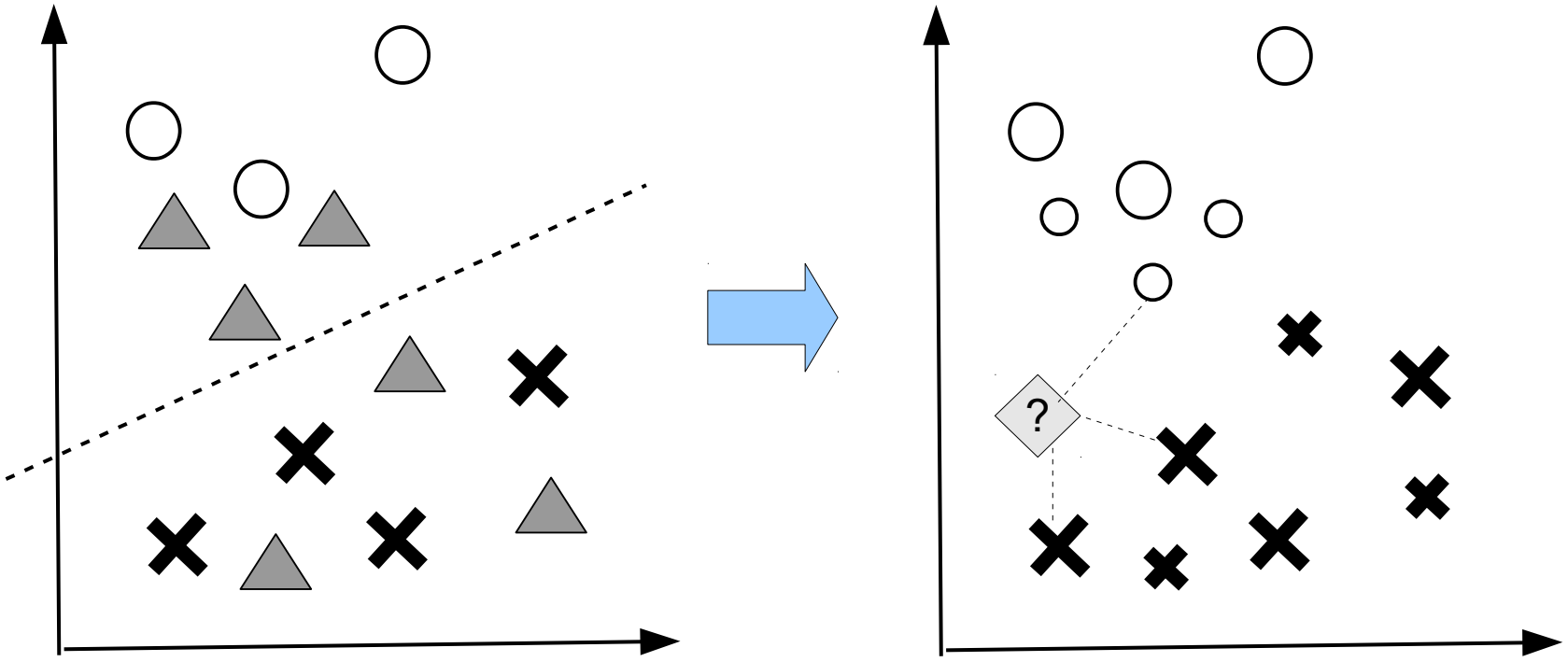
識別器 2 で確信度の高い
正解なしデータ

13.3 共訓練

- 共訓練の特徴
  - 学習初期の誤りに対して頑健
- 共訓練の問題点
  - それぞれが識別空間として機能する特徴集合を、どのようにして作成するか
  - 全ての特徴を用いる識別器よりも高性能な識別器が作成できるか

13.4 YATSI アルゴリズム

- YATSI(Yet Another Two-Stage Idea) アルゴリズムの考え方
  - 繰り返し学習による誤りの増幅を避ける



正解付きデータで作った識別器で全データを識別

正解付きデータ :1
識別後の正解なしデータ :0.1
の重みで k-NN

調整可能

ラベル伝搬法

- ラベル伝搬法の考え方
  - 特徴空間上のデータをノードとみなし、類似度に基づいたグラフ構造を構築する
  - 近くのノードは同じクラスになりやすいという仮定で、正解なしデータの予測を行う
  - 評価関数（最小化）

$$J(\boldsymbol{f}) = \sum_{i=1}^{l} (y_i - f_i)^2 + \lambda \sum_{i<j} w_{ij} (f_i - f_j)^2$$

予測値と正解ラベルを近づける

隣接ノードの予測値を近づける

f_i ： i 番目のノードの予測値

y_i ： i 番目のノードの正解ラベル { -1, 0, 1}

w_{ij} ：i 番目のノードとj 番目のノードの結合の有無

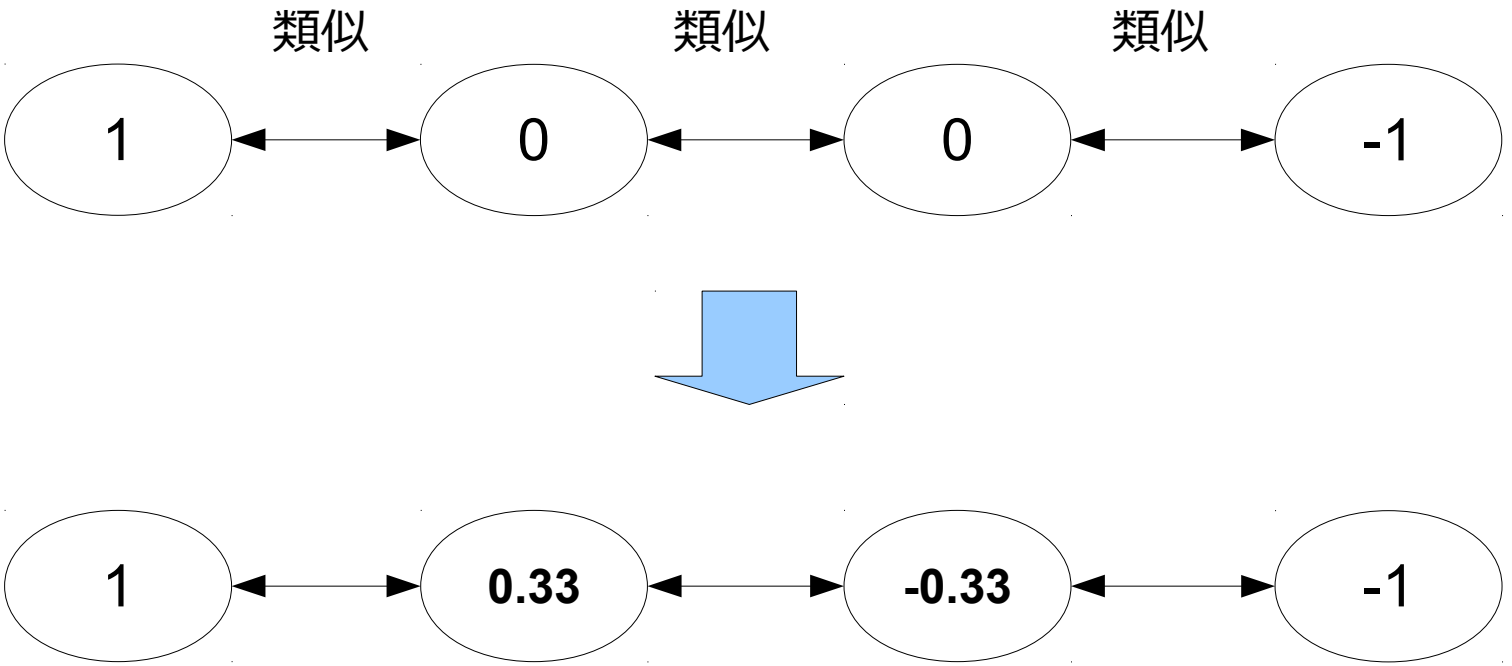
ラベル伝搬法

1.データ間の類似度に基づいて、データをノードとしたグラフを構築

- 類似度の基準
  - RBF　$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2)$
    - 全ノードが結合
    - 連続値の類似度が与えられる
  - K-NN
    - 近傍の k 個のノードが結合
    - 結合の有無は 0 または 1 で表現
    - 省メモリ

ラベル伝搬法

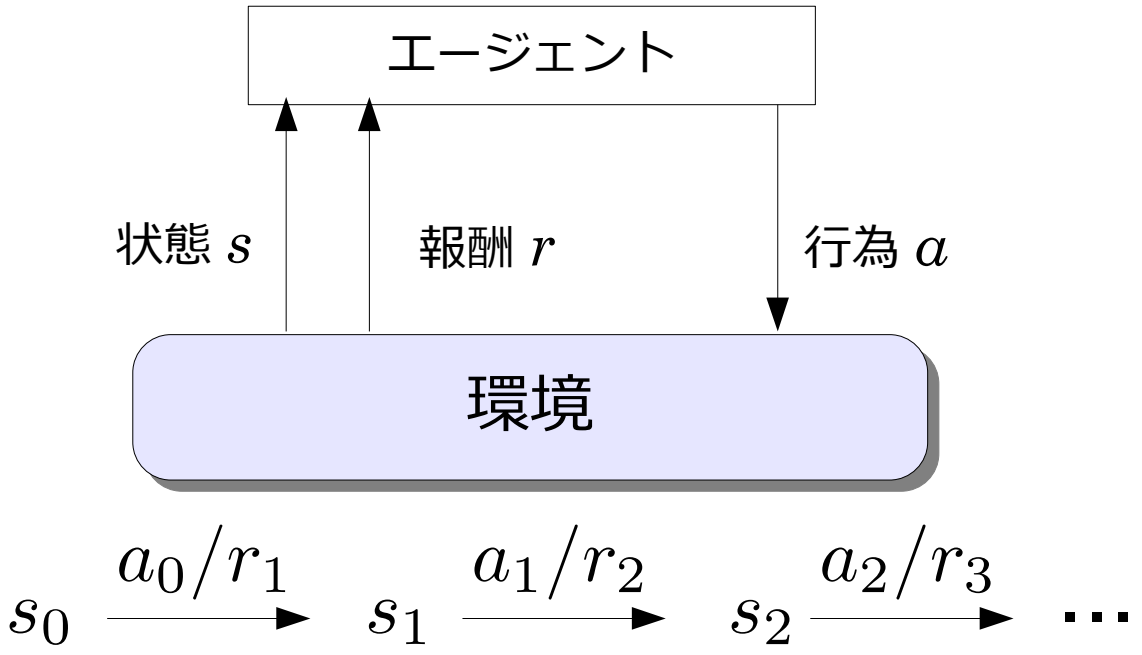
2.ラベル付きノードからラベルなしノードにラベルを伝播させる操作を繰り返し、隣接するノードがなるべく同じラベルを持つように最適化



14. 強化学習

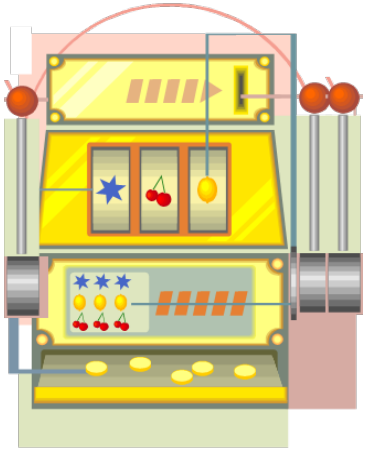
14.1 強化学習とは

- 強化学習の設定
  - 教師信号が間接的
  - 報酬が遅れて与えられる
  - 探索が可能
  - 状態が非確定的な場合がある



14.2 1 状態問題の定式化 -K-armed bandit 問題 -

- K-armed bandit の定義
  - K 本の腕を持つスロットマシン
  - i 番目の腕を引く行為：a_i
  - その行為の価値：$Q(a_i)$
    - 報酬 r が確定的な場合
      - 全ての可能な a_i を試み、$Q(a_i) = r(a_i)$ が最大となるa_iを探す
    - 報酬 r_t が確率的な場合



$$Q_{t+1}(a_i) = Q_t(a_i) + \eta(r_{t+1}(a_i) - Q_t(a_i))$$

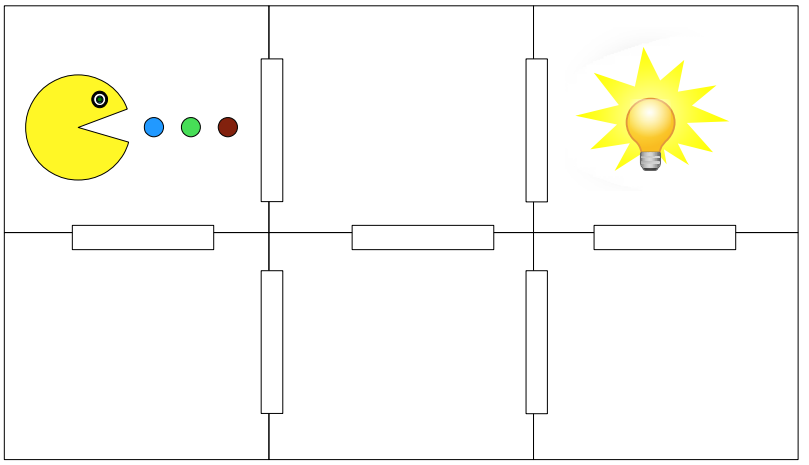
η は t の増加に伴って、減少させる

14.2 1 状態問題の定式化 -K-armed bandit 問題 -

- どのように a_i を選ぶか
  - 常に $Q_t(a_i)$ が最大のものを選ぶ
    - もっと良い行為があるのに見逃してしまうかもしれない
  - いろいろな a_i を何度も試みる
    - 無駄な行為を何度も行ってしまうかもしれない
- ε-greedy 法
  - 確率 1-ε で最良の行為を選び、確率 ε でランダムに行為を選ぶ
- Boltzmann 分布を利用した方法
  - 温度 k を導入し、 k が下がるにつれて確率的振る舞いが少なくなるようにする

14.3 マルコフ決定過程による定式化

- マルコフ決定過程
  - 状態遷移を伴う問題の定式化
  - 時刻 t における状態　$s_t \in S$
  - 時刻 t における行為　$a_t \in A(s_t)$
  - 報酬　$r_{t+1} \in \mathbb{R}$
    確率分布　$p(r_{t+1}|s_t, a_t)$
  - 次状態　$s_{t+1} \in S$
    確率分布　$P(s_{t+1}|s_t, a_t)$



14.3 マルコフ決定過程による定式化

- 強化学習の学習目標
  - 最適政策　π^*
    - 状態から行為へのマッピング
    - 累積報酬の期待値が最大となる政策
  - 累積報酬の期待値

$$V^{\pi}(s_t) = \mathbb{E}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots)$$

$$= \mathbb{E}(\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i})$$

γ : 割引率　$0 \leqq \gamma < 1$

14.3 マルコフ決定過程による定式化

- 最適政策に対する期待報酬

$$
\begin{aligned}
V^*(s_t) &= \max_{a_t} Q^*(s_t, a_t) \\
&= \max_{a_t} \mathbb{E}\left(\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i}\right) \\
&= \max_{a_t} \mathbb{E}\left(r_{t+1} + \gamma \sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i+1}\right) \\
&= \max_{a_t} \mathbb{E}\left(r_{t+1} + \gamma V^*(s_{t+1})\right)
\end{aligned}
$$

14.3 マルコフ決定過程による定式化

- 状態遷移確率を明示

$$V^*(s_t) = \max_{a_t}(\mathbb{E}(r_{t+1}) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)V^*(s_{t+1}))$$

- Q 値による書き換え

$$Q^*(s_t, a_t) = \mathbb{E}(r_{t+1}) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$$

ベルマン方程式

14.4 モデルベースの手法

- 環境のモデル（状態遷移確率、報酬の確率分布）が与えられた場合の Q 値の求め方

Algorithm 14.1 Value iteration アルゴリズム

$V(s)$ を任意の値で初期化

repeat

  for all $s \in S$ **do**

    for all $a \in A$ **do**

      $Q(s,a) \leftarrow \mathbb{E}(r|s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V(s')$

    end for

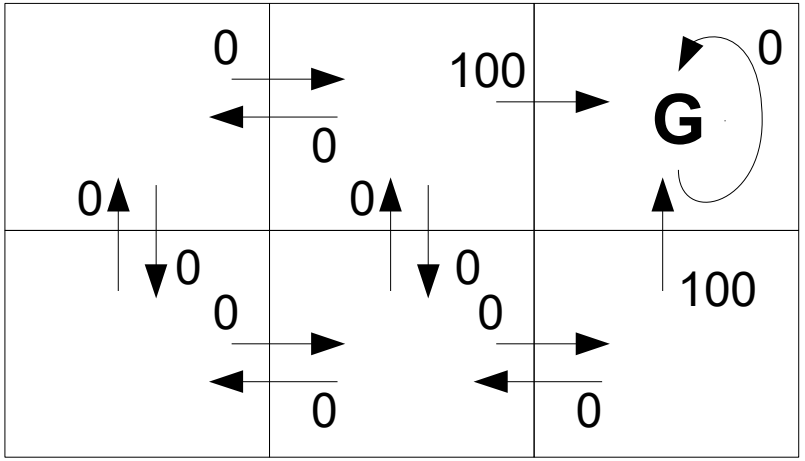
    $V(s) \leftarrow \max_a Q(s,a)$

  end for

until $V(s)$ が収束

14.5 モデルフリーの手法

- 報酬と遷移が決定的な TD 学習



- ベルマン方程式

$$Q(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$$

14.5 モデルフリーの手法

Algorithm 14.2 TD 学習 (報酬と遷移が決定的な場合)

$Q(s,a)$ を 0 に初期化

for all エピソード **do**

  repeat

    探索基準に基づき行為 a を選択

    行為 a を実行し、報酬 r と次状態 s' を観測

    以下の式で Q を更新

$$Q(s,a) \leftarrow r + \gamma \max_{a'} Q(s',a')$$

    $s \leftarrow s'$

  until s が終了状態

end for

14.5 モデルフリーの手法

- 報酬と遷移が確率的な TD (Temporal Difference) 学習
  - ベルマン方程式

$$Q(s,a) \leftarrow Q(s,a) + \eta(\underbrace{r + \gamma \max_{a'} Q(s',a')}_{\text{TD 誤差}} - Q(s,a))$$

    - 理論的には、各状態に無限回訪問可能な場合に収束
    - 実用的には無限回の訪問は不可能なので、状態推定関数等を用いて、複数の状態を同一とみなす等の工夫が必要

Deep Q-learning

- $Q(s, a)$ の推定に DNN を用いる
  - ネットワークの誤差に TD 誤差を用いる
  - 一部の問題においては、状態を推定しなくとも、局面そのものをネットワークの入力にできる
    - 例）ゲーム

Section4 のまとめ

- 半教師あり学習
  - 数値特徴の場合：一定の性質を満たす場合に有効
  - カテゴリ特徴の場合：言語データで有効な場合がある
  - 手法：自己学習、共訓練、ラベル伝搬法
- 強化学習
  - 変化する状態に対する最適な行為を求める学習
  - マルコフ決定過程による定式化を行い、 Q 値を最適化する