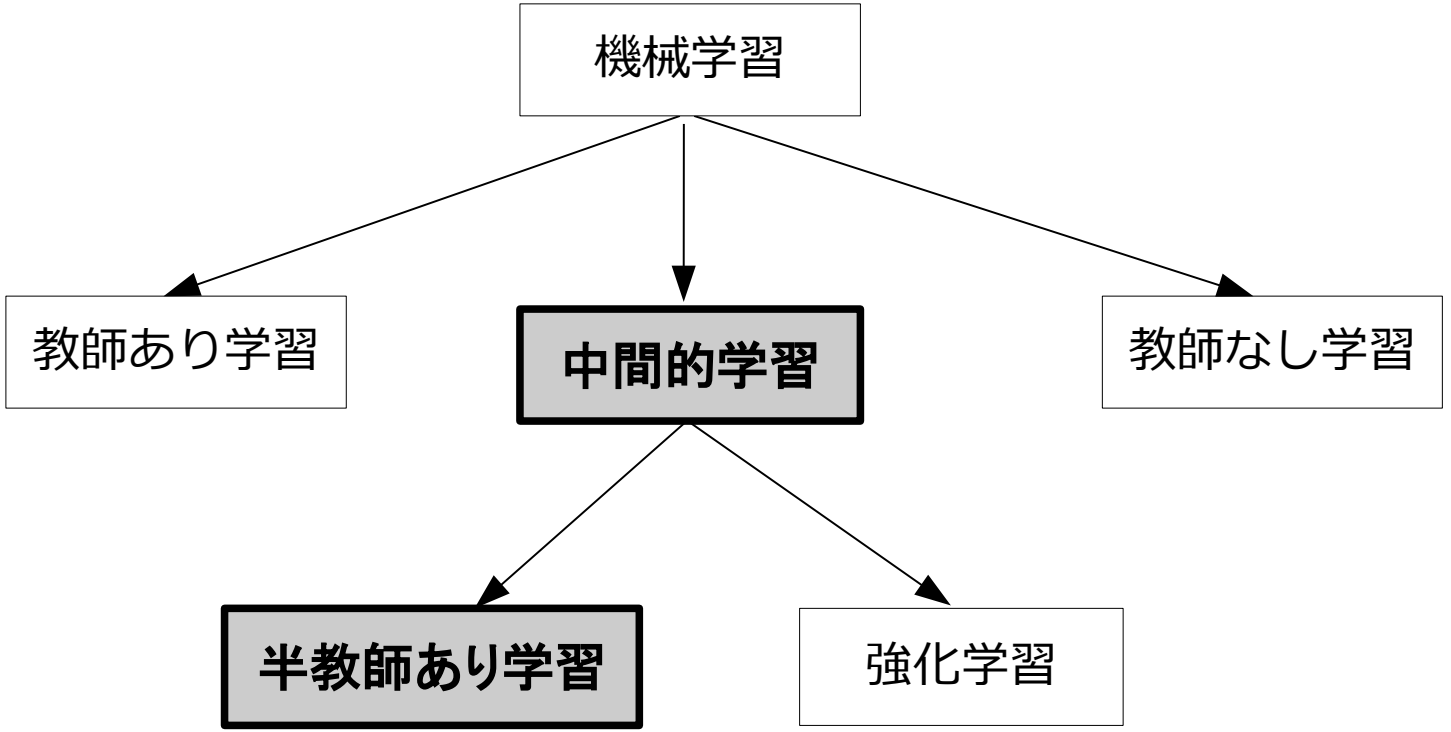
14. 半教師あり学習



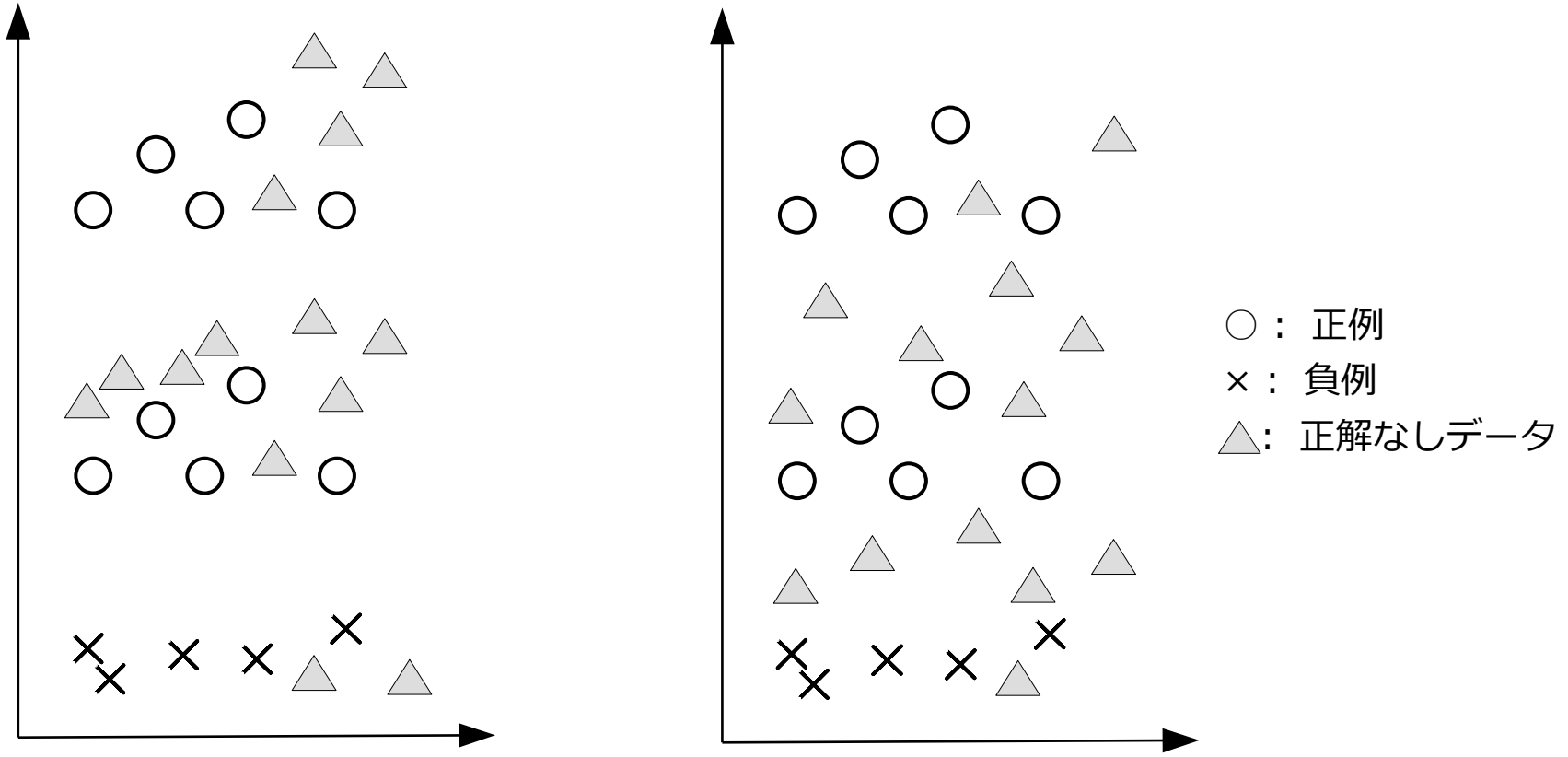
学習に用いるデータが中間的

14.1 半教師あり学習とは

- 半教師あり学習の設定
  - 正解情報が一部の学習データにのみ与えられている
- 例： web 文書の P/N 判定
  - データ自体はクローラで容易に収集できる
  - タグ付け作業はコストが掛かり、大量の正解付きデータが得られることはあまり期待できない

14.1.1 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条件
  - 正解なしデータから得られる $p(\boldsymbol{x})$ に関する情報が，$p(y \mid \boldsymbol{x})$ の推定に役立つこと



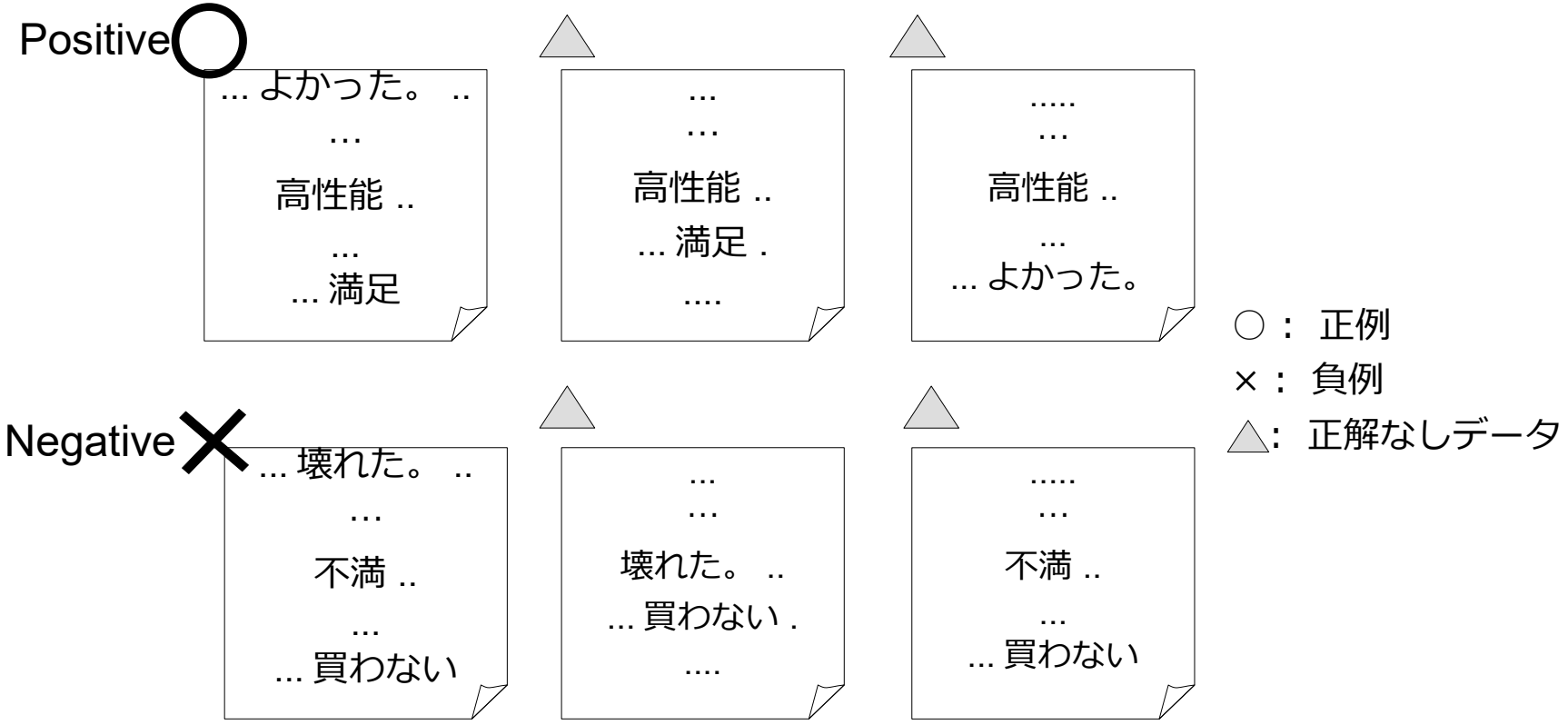
(a) 半教師あり学習に適するデータ　　(b) 半教師あり学習に適さないデータ

14.1.1　数値特徴の場合

- 半教師あり学習に適したデータ
  - 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

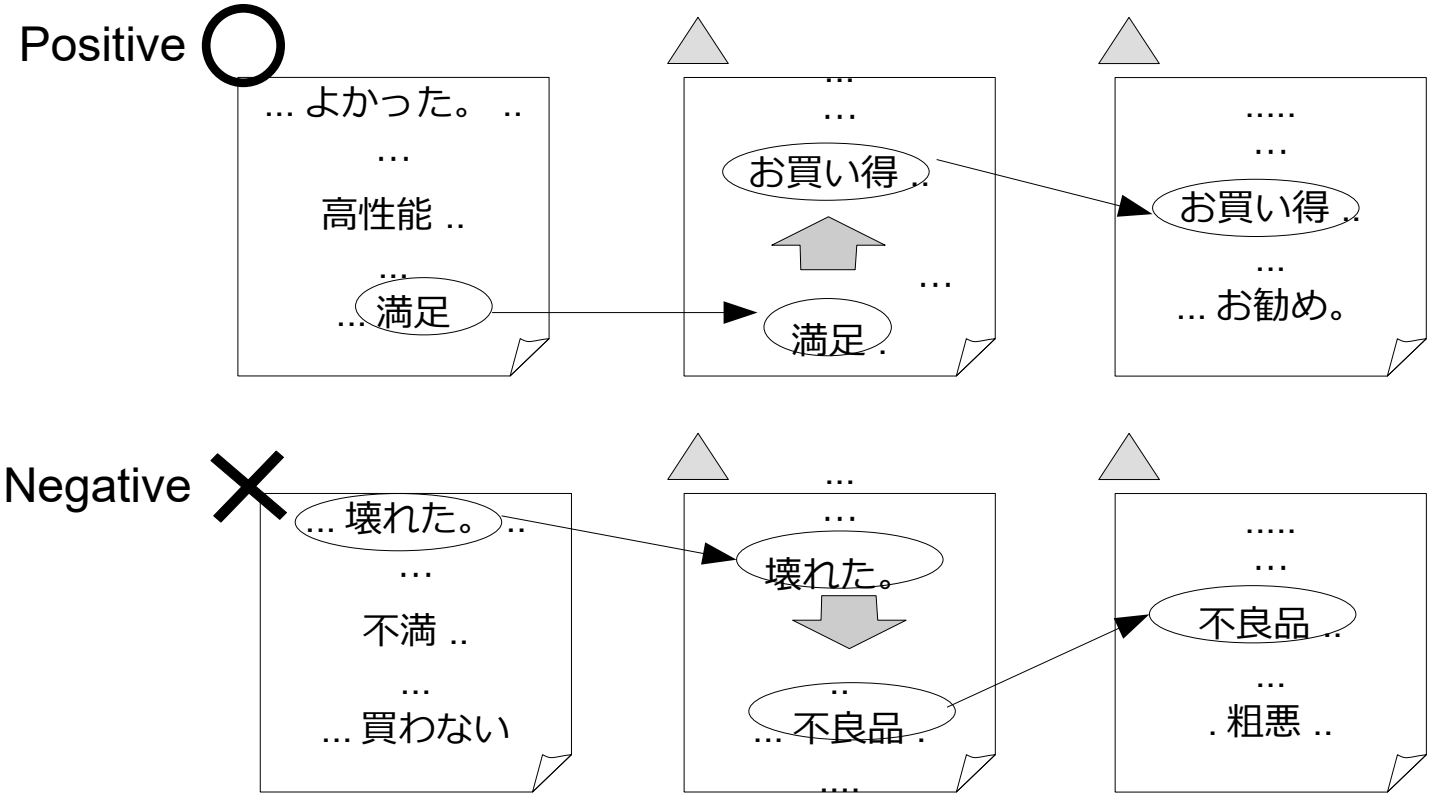
14.1.2 カテゴリ特徴の場合

- 文書の P/N 判定の例
  - 特徴語が抽出できていると仮定



14.1.2 カテゴリ特徴の場合

- 特徴の伝播
  - オーバーラップした特徴語によって、判定に寄与する新たな特徴語が見つかる



14.1.3 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

14.2 自己学習

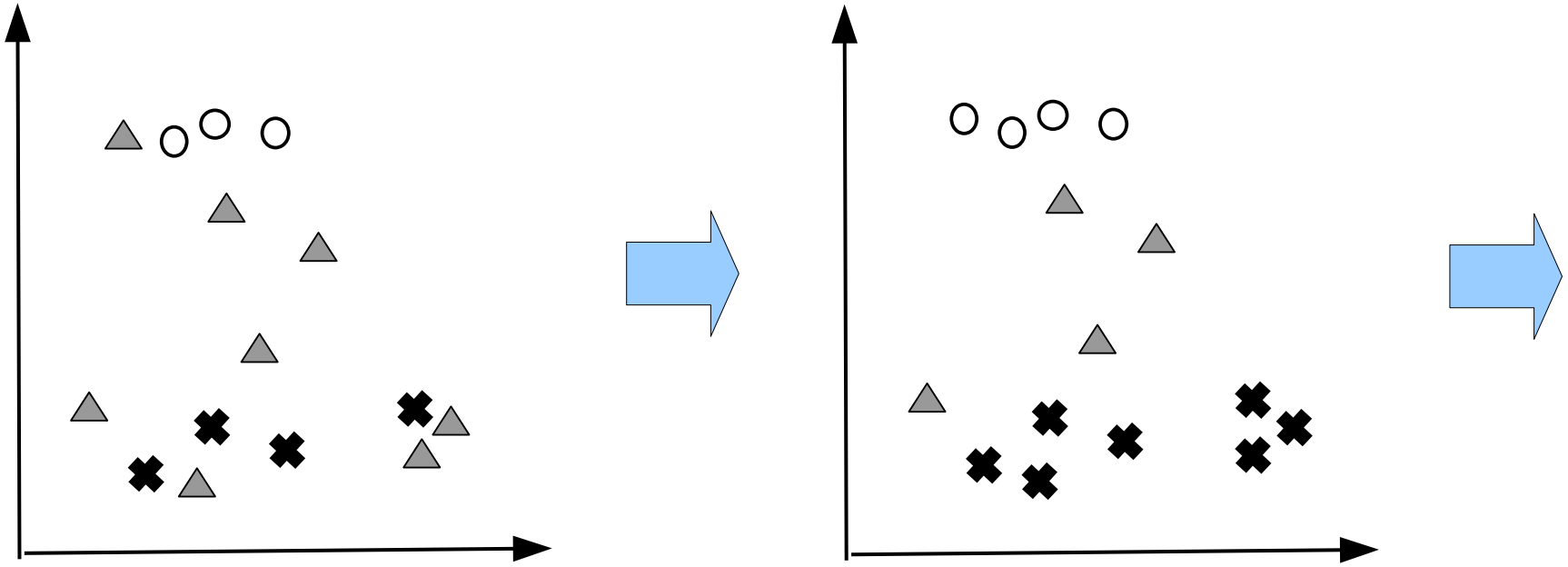
- 自己学習のアルゴリズム

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

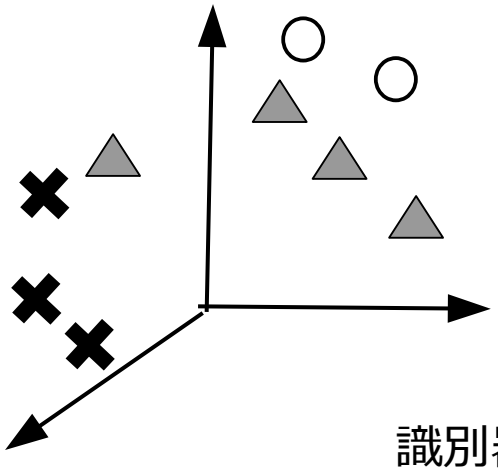


14.2 自己学習

- 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

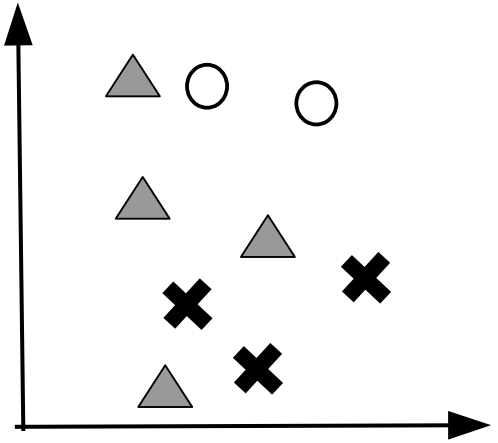
14.3 共訓練

- 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

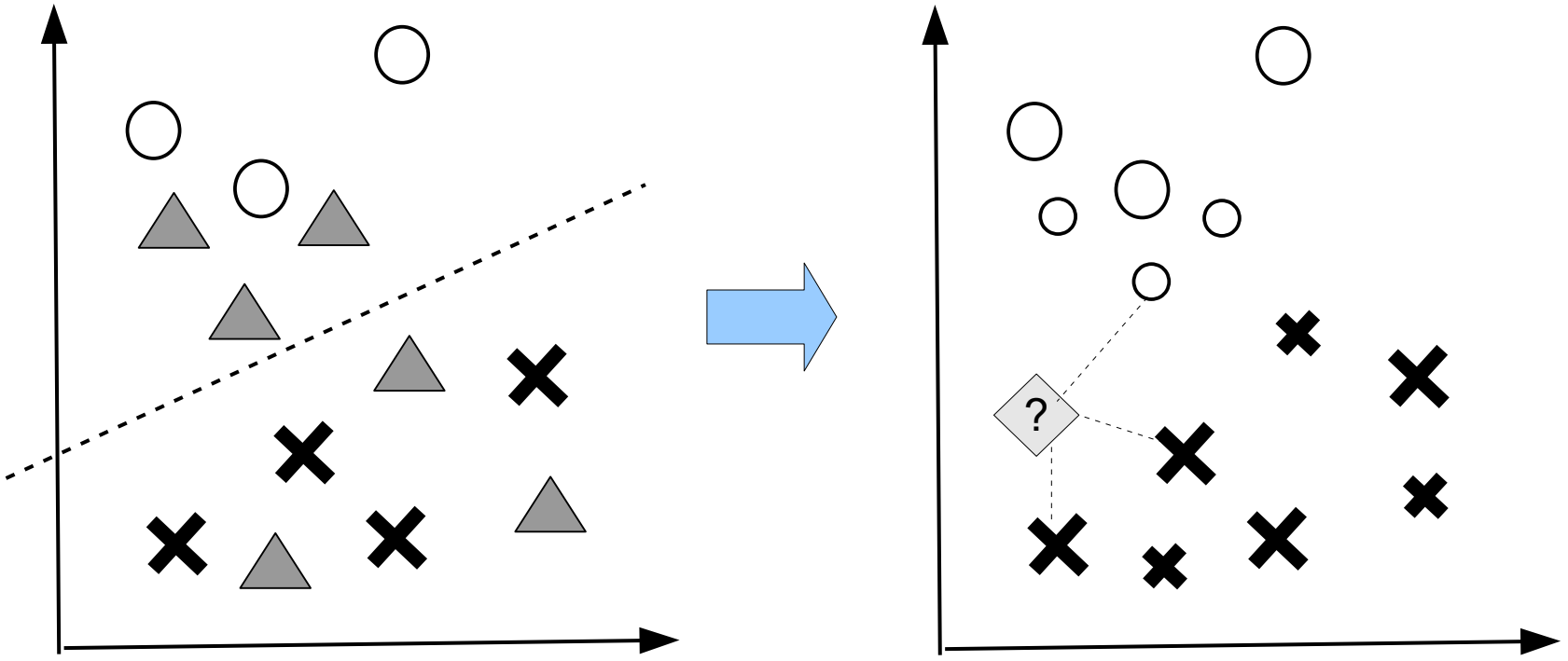
識別器 2 で確信度の高い
正解なしデータ

14.3 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
  - すべ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

14.4 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

14.5 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

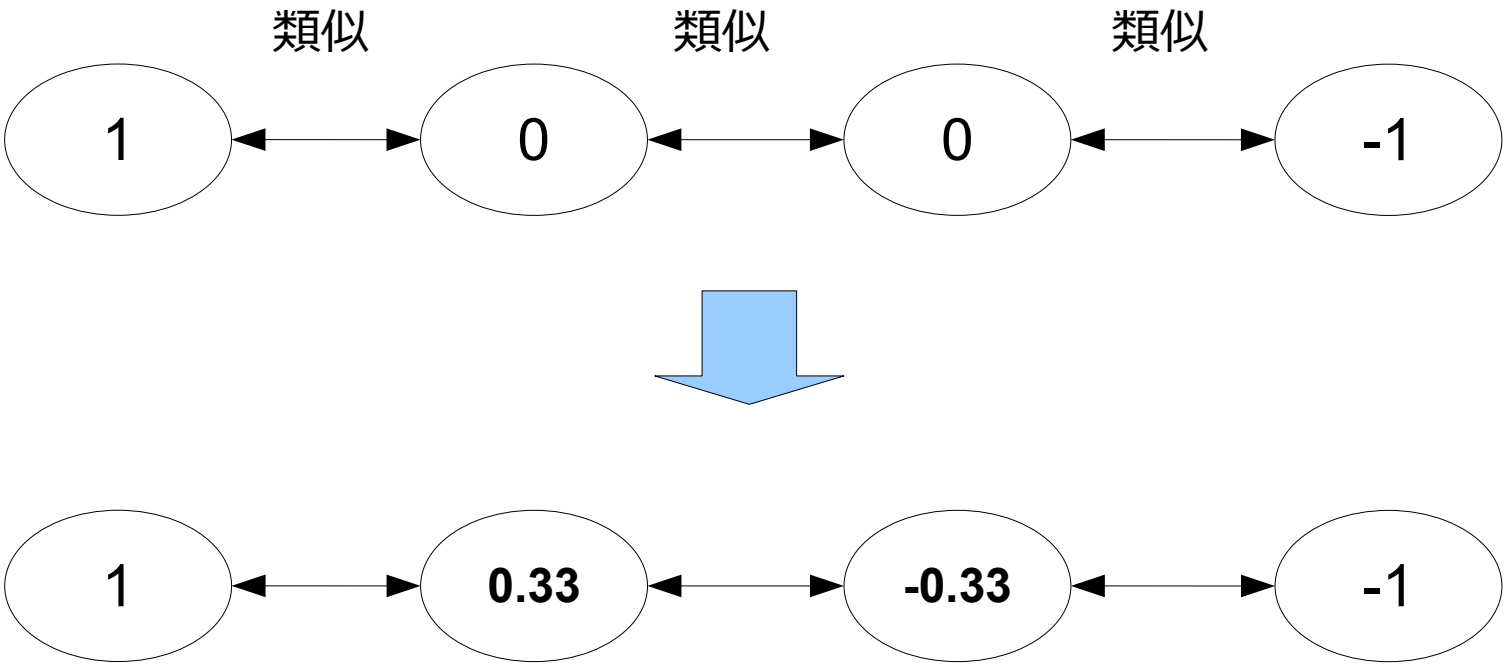
14.5 ラベル伝搬法

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

14.5 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



補足

データ拡張

- 学習の前提
  - モデル $p_\theta(y \mid x)$ のパラメータ θ を最適化する
- 正解付きデータ L の拡張
  - データ x に対して、正解 y^* やデータの性質を変えないような変換 $q(\hat{x} \mid x)$ を施して、本物らしく見えるデータ $\hat{x}$ を作り出す。
  - 正解付きデータ拡張に対する最適化（対数尤度最大化）

$$\min_\theta \mathcal{J}_{\mathrm{da}}(\theta) = \mathop{\mathbb{E}}_{x, y^* \in L} \mathop{\mathbb{E}}_{\hat{x} \sim q(\hat{x}|x)} \left[-\log p_\theta(y^* \mid \hat{x}) \right]$$

データ拡張

- データ拡張の方法
  - 画像
    - 基本：移動、回転、拡大・縮小
    - AutoAugment
      - 対象画像に応じて様々な変換の組み合わせを強化学習で学習し、検証用データで評価
  - 自然言語
    - Back translation
      - 翻訳後、元に戻すことで水増し
    - TF-IDF word replacement
      - TF-IDF 値に基づく乱数（高いほど高い確率）で単語を置換

データ拡張

- 正解なしデータの拡張
  - 基本的なアイディア
    - 正解なしデータ U に対してノイズ ε を加えた出力 $p_\theta(y \mid x,\varepsilon)$ と元の出力 $p_\theta(y \mid x)$ が近くなるように学習する
  - 正解なしデータに対する最適化（確率分布間の距離を最小化）

$$\min_\theta \ \mathcal{J}_{\mathrm{UDA}}(\theta) = \mathop{\mathbb{E}}_{x \in U} \mathop{\mathbb{E}}_{\hat{x} \sim q(\hat{x}|x)} \left[\mathcal{D}_{\mathrm{KL}} \left(p_{\tilde{\theta}}(y \mid x) \,\middle\|\, p_\theta(y \mid \hat{x}) \right) \right]$$

$\tilde{\theta}$: 現在のパラメータを定数としてコピー

確率分布間の距離

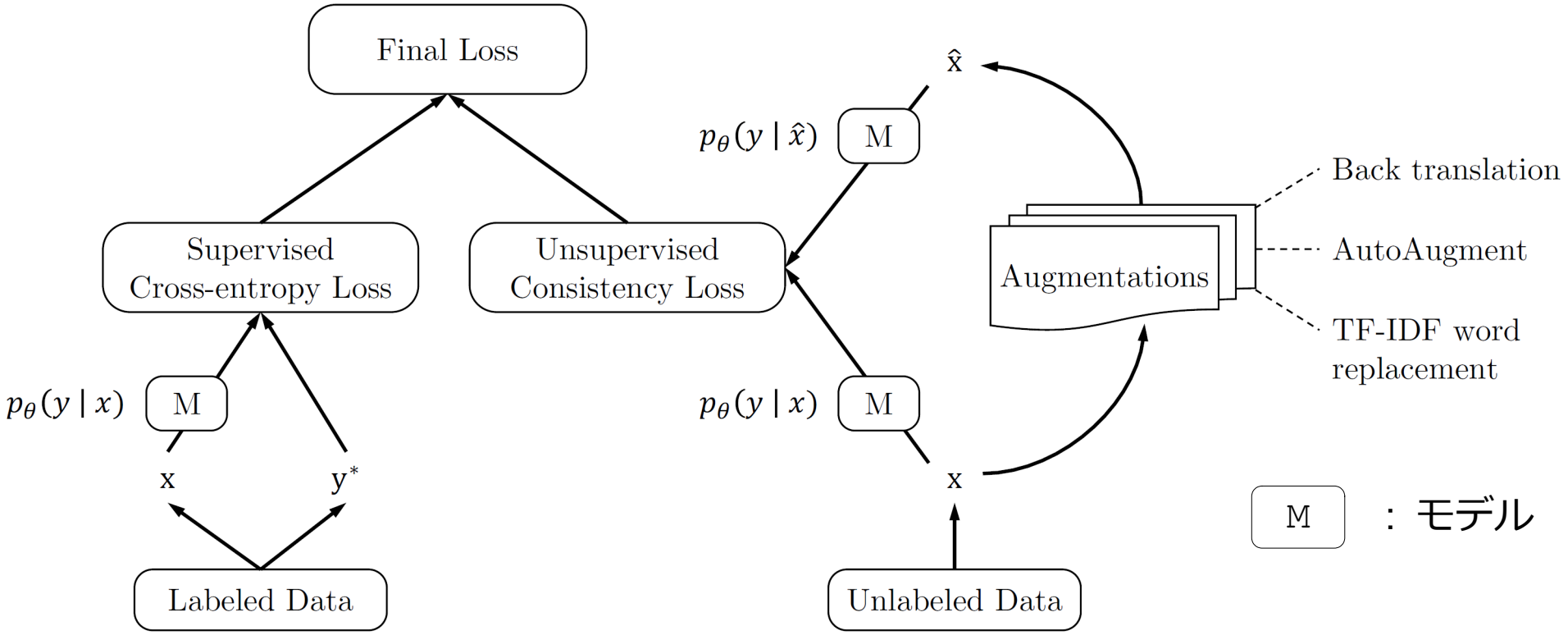
- KL (Kullback-Leibler) ダイバージェンス
  - 2 つの確率分布がどの程度似ているかを表す尺度
  - 順序を入れ替えれば値が変わるので、正確には距離ではない

$$D_{KL}(p(x)\|q(x)) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} dx$$

    - 同じ確率分布では値が 0 となる
    - 確率分布が似ていないほど大きな値となる

データ拡張

- データ拡張に基づく半教師あり学習での最適化



Xie et.al.: Unsupervised Data Augmentation, https://arxiv.org/abs/1904.12848