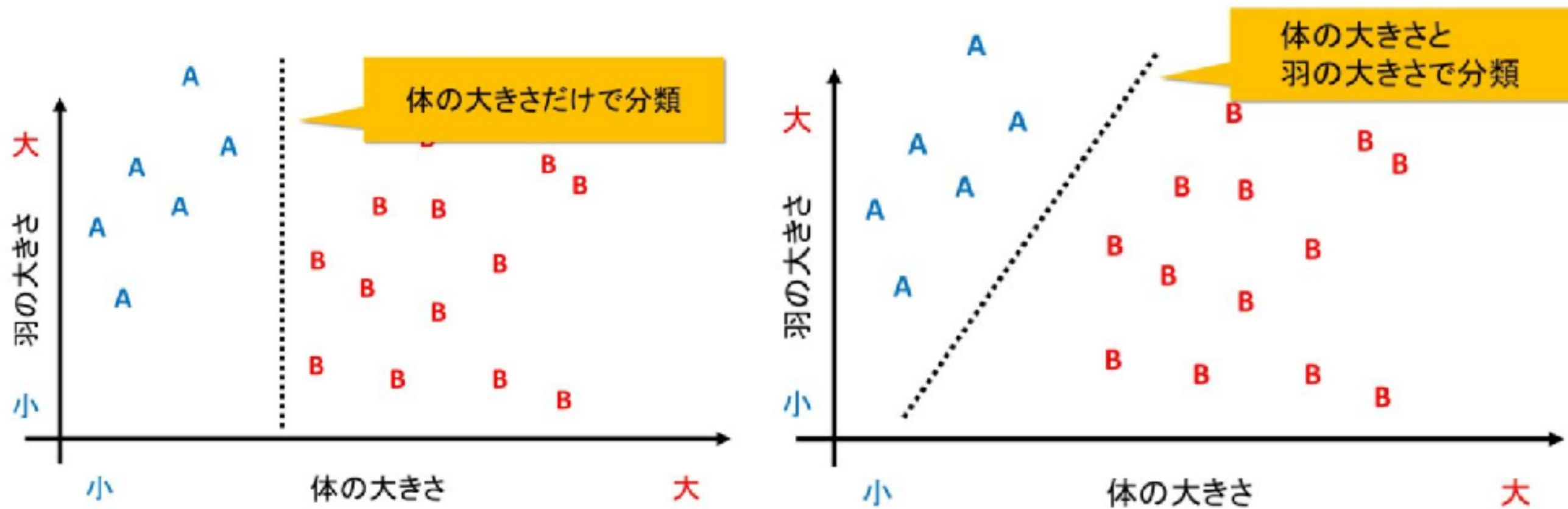


マルチインスタンスSVM

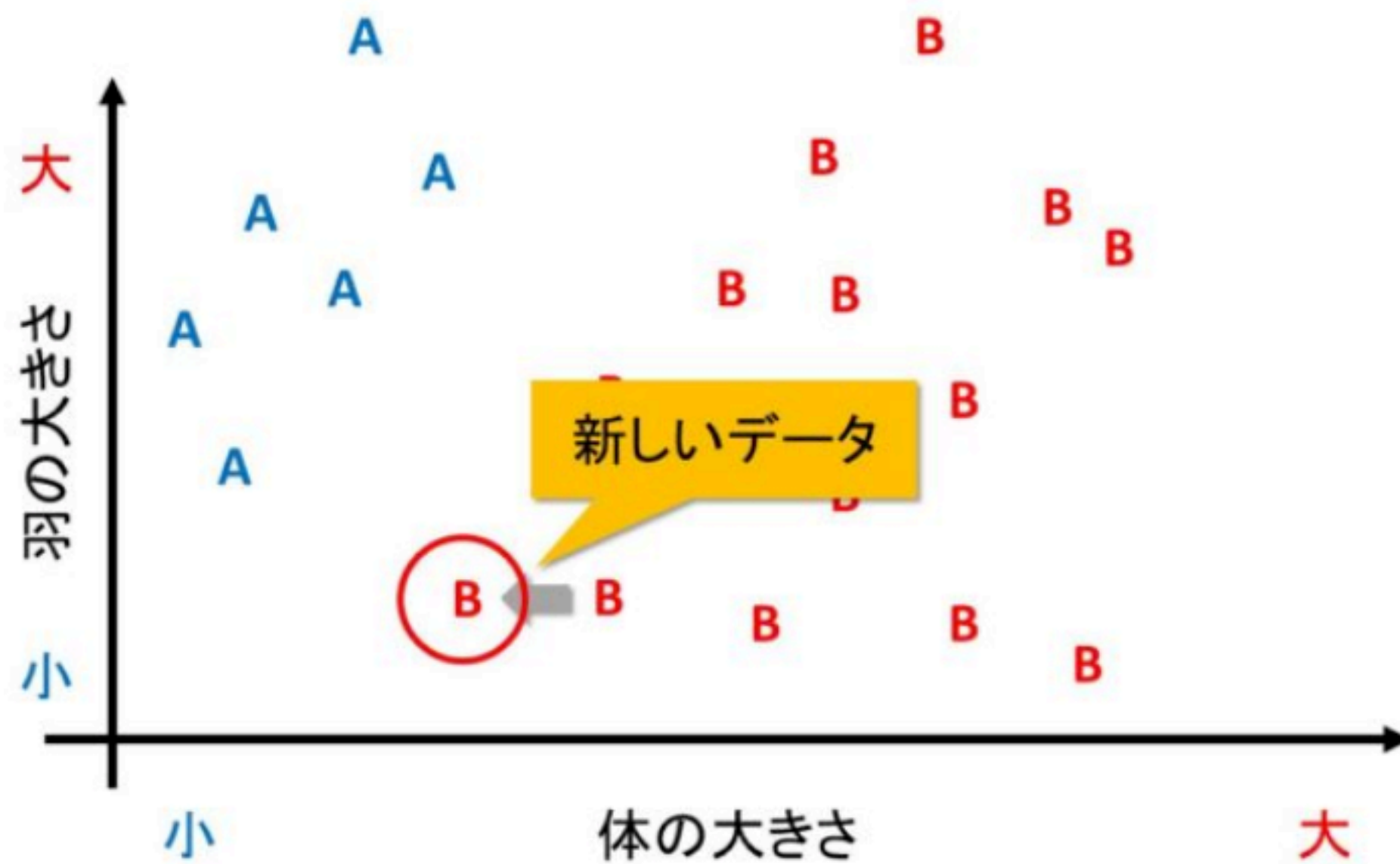
Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2002. Support vector machines for multiple-instance learning. In Proceedings of the 15th International Conference on Neural Information Processing Systems (NIPS'02), Suzanna Becker, Sebastian Thrun, and Klaus Obermayer (Eds.). MIT Press, Cambridge, MA, USA, 577-584.

他参考サイト多数

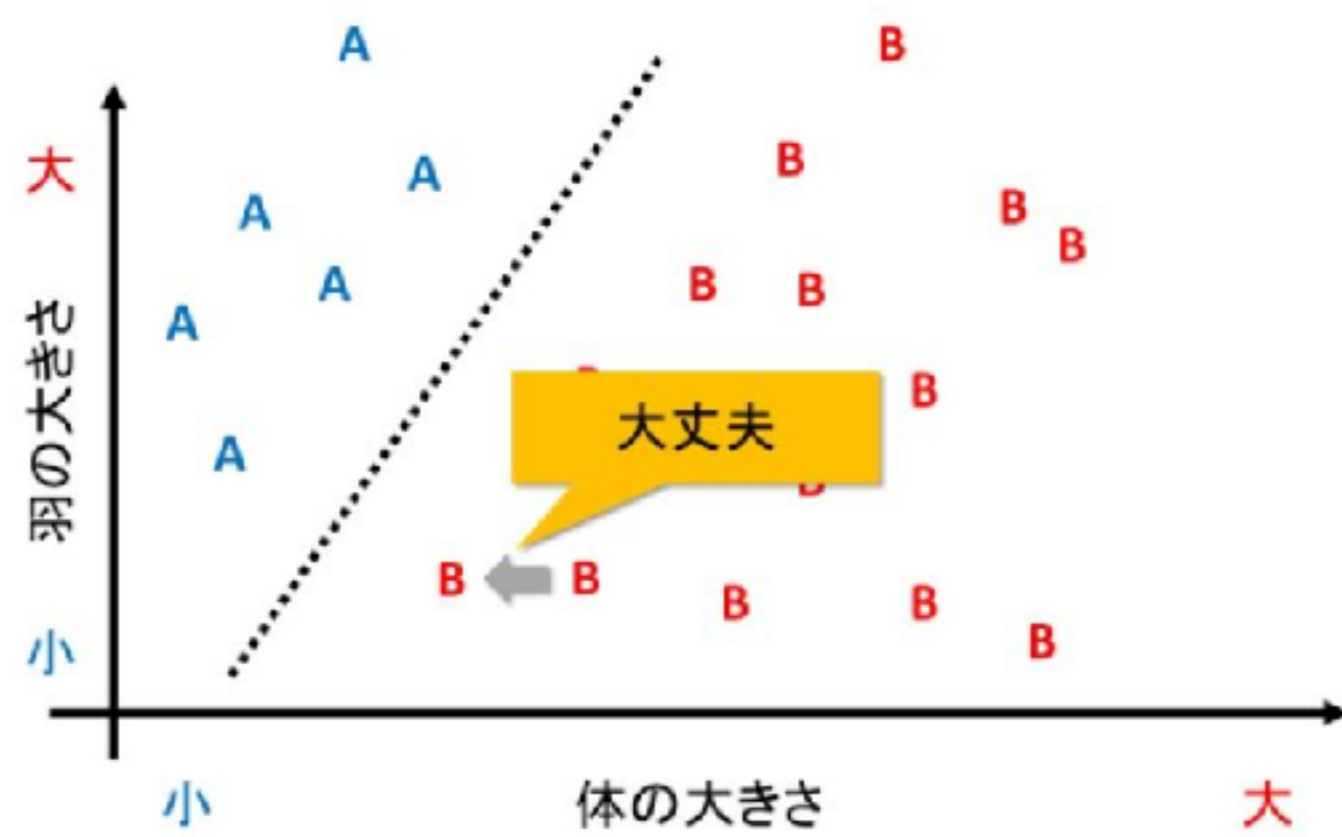
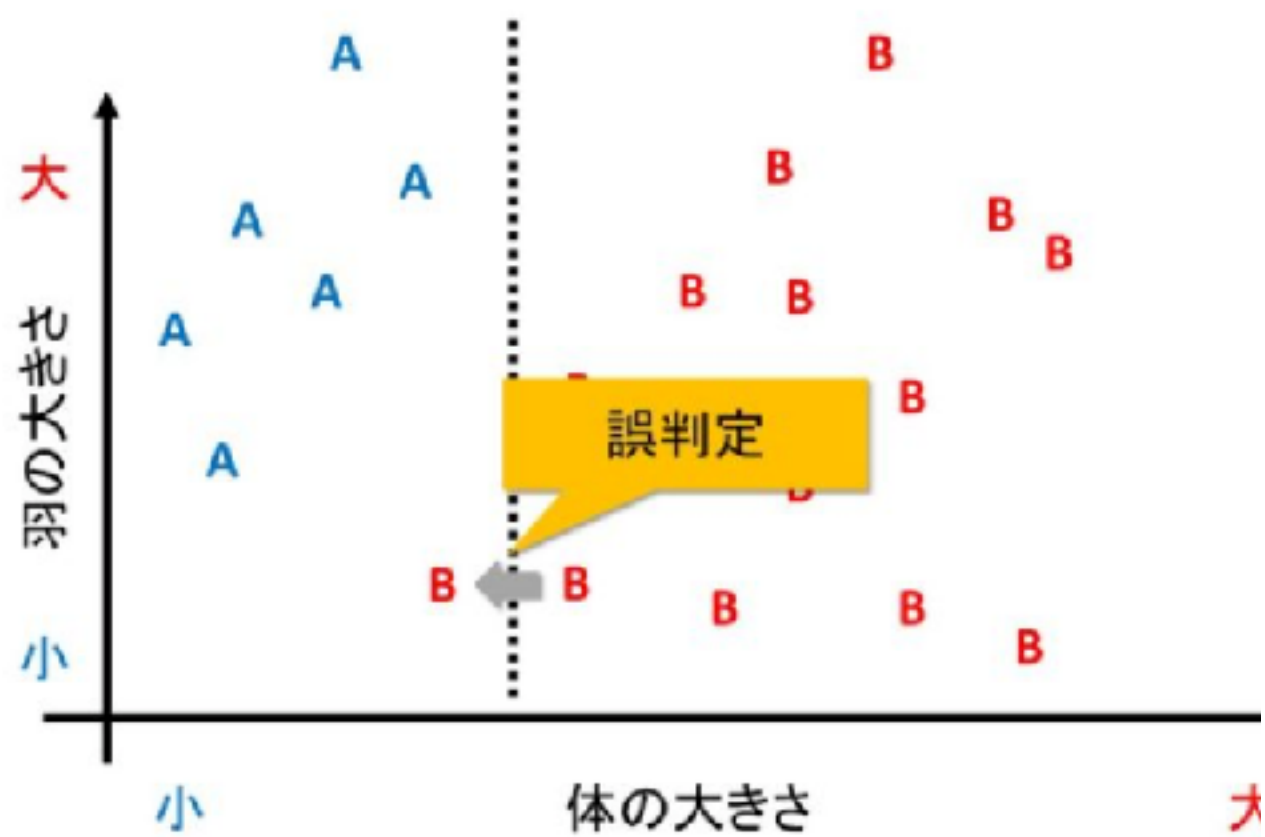
SVMとは



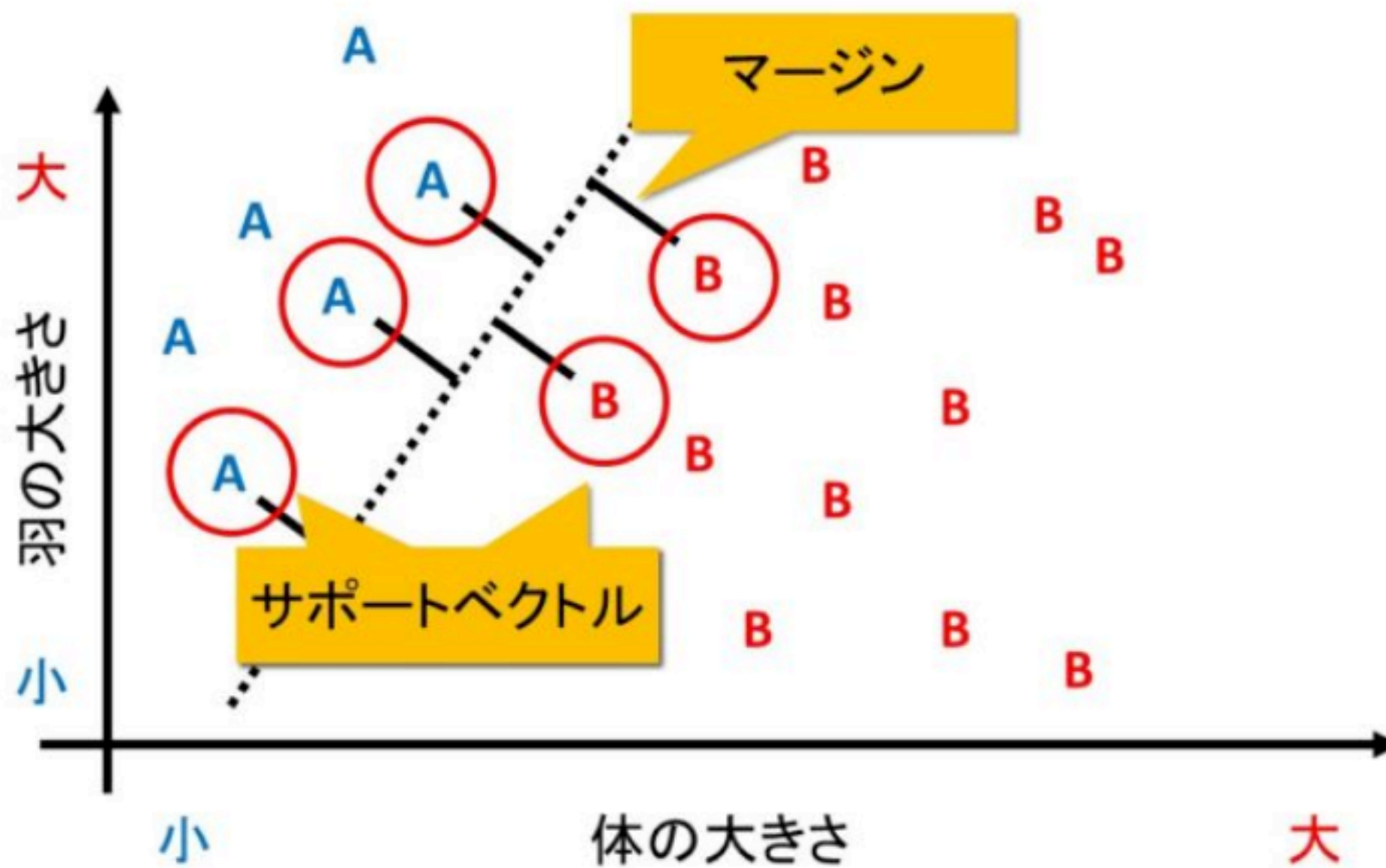
SVMとは



SVMとは

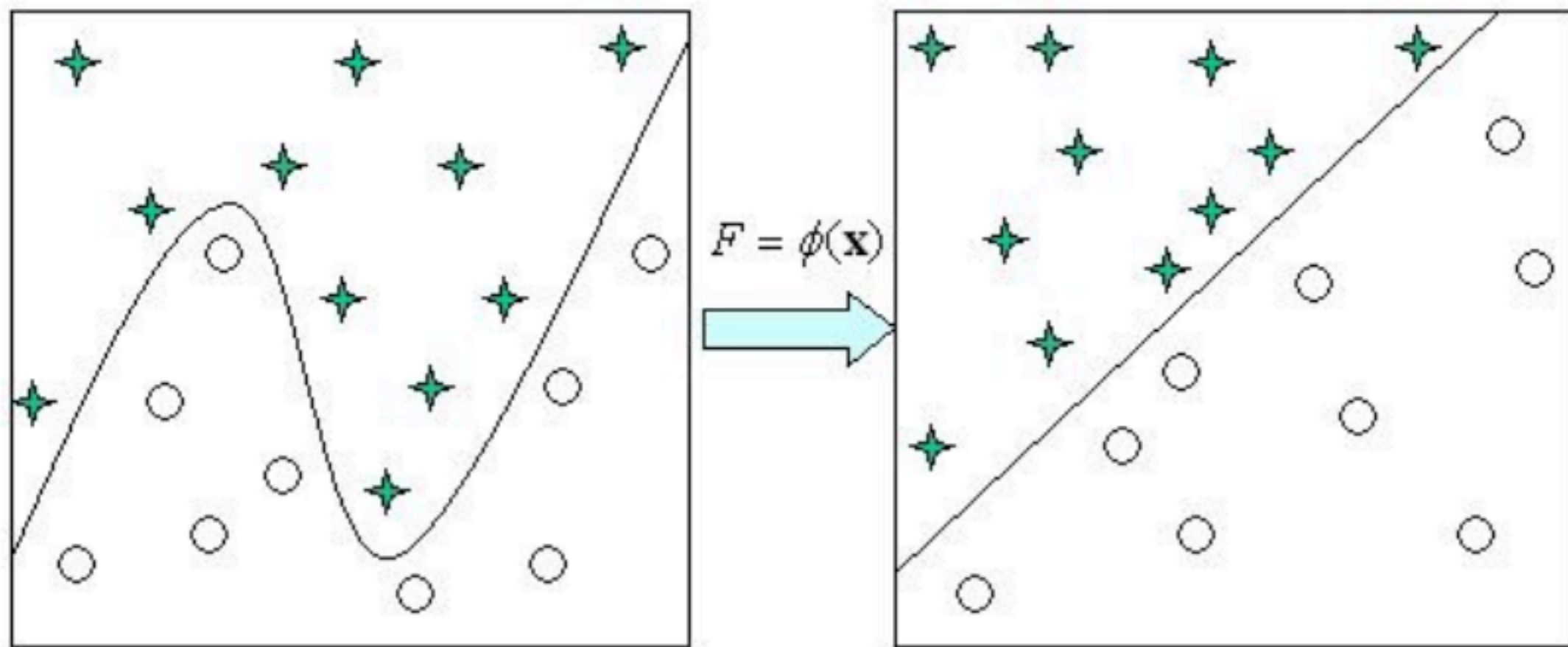


SVMとは



カーネル法

- 非線型データ構造を変換し，線形構造にする



マルチインスタンス学習とは

- この訓練事例にラベルが与えられるのではなく
バグと呼ばれる訓練事例の集合にラベルが与えられる

バグについて

- 各バグは複数の事例から構成
- 各事例は正事例か負事例に属する
- バグが正事例を一つでも含んでいれば正バグ
- バグが負事例"のみ"を含んでいれば負バグ

ラベルが与えられた時

- 負バッグ
 - 含まれる事例が全て負事例
- 正バッグ
 - 正事例と負事例の判断ができない
- 正バッグに含まれる事例のラベルを推定
 - 分類境界を求める

イメージ

正事例



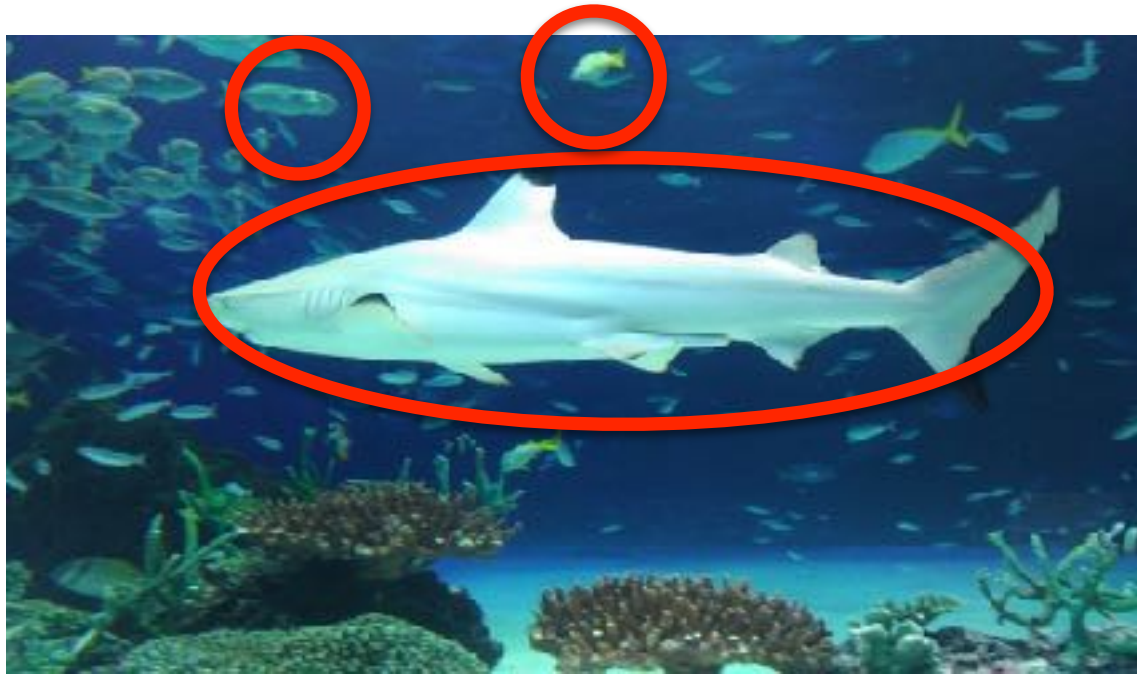
負事例



- バッグ：写真
- 事例：写真内の生き物
- ラベル：サメが写っているか

イメージ

正事例



負事例



どれがサメか？

サメではない

- どの生き物がサメかの判断も含め学習

マルチインスタンスSVMのアプローチ

- mi-SVM
 - 全ての事例のラベルを推定
 - 各事例のラベルに基づいて学習
- MI-SVM
 - 各バッグの代表事例を決定
 - 代表事例とバッグのラベルを用いて学習

- 分からん

```
initialize  $y_i = Y_I$  for  $i \in I$ 
REPEAT
  compute SVM solution  $w, b$  for data set with imputed labels
  compute outputs  $f_i = \langle w, x_i \rangle + b$  for all  $x_i$  in positive bags
  set  $y_i = \text{sgn}(f_i)$  for every  $i \in I$ ,  $Y_I = 1$ 
  FOR (every positive bag  $B_I$ )
    IF  $(\sum_{i \in I} (1 + y_i) / 2 == 0)$ 
      compute  $i^* = \arg \max_{i \in I} f_i$ 
      set  $y_{i^*} = 1$ 
    END
  END
  END
WHILE (imputed labels have changed)
  OUTPUT  $(w, b)$ 
```

Figure 2: Pseudo-code for mi-SVM optimization heuristics (synchronous update).

MI-SVM

- 分からん

```
initialize  $\mathbf{x}_I = \sum_{i \in I} \mathbf{x}_i / |I|$  for every positive bag  $B_I$ 
REPEAT
  compute QP solution  $\mathbf{w}, b$  for data set with
    positive examples  $\{\mathbf{x}_I : Y_I = 1\}$ 
  compute outputs  $f_i = \langle \mathbf{w}, \mathbf{x}_i \rangle + b$  for all  $\mathbf{x}_i$  in positive bags
  set  $\mathbf{x}_I = \mathbf{x}_{s(I)}$ ,  $s(I) = \arg \max_{i \in I} f_i$  for every  $I$ ,  $Y_I = 1$ 
WHILE (selector variables  $s(I)$  have changed)
OUTPUT  $(\mathbf{w}, b)$ 
```

Figure 3: Pseudo-code for MI-SVM optimization heuristics (synchronous update).

まとめ

- mi(MI)-SVMの式をざっくりと理解して実装したい
 - ライブラリがあるならそっちで・・・
- マルチインスタンス学習が無理でもSVMは実装する
(カーネルトリックするしないは考える必要がありそう)