

## 飼いならし — 飼育・野生混在データからの学習

## Taming — Learning from Wild and Tame Data

神島 敏弘      濱崎 雅弘      赤穂 昭太郎  
Toshihiro Kamishima      Masahiro Hamasaki      Shotaro Akaho

\*1 産業技術総合研究所

National Institute of Advanced Industrial Science and Technology (AIST)

We propose a learning framework that adopts two types of data sets: tame and wild. The tame examples are labeled based on a consistent criterion, while the wild examples are not. Examples of tame examples are the Web pages classified according to the carefully designed criteria. Social bookmarks tagged according to the mixed concepts can be considered as wild data.

## 1. はじめに

本研究では飼いならし (taming) とよぶ学習問題を提案し、その解法を示す。この学習の訓練事例には、飼育データ (tame data) と野生データ (wild data) の2種類の事例集合が混在している。飼育データでは、これから学習したい目標概念と無矛盾なラベルが注意深く選ばれて与えられている。もう一方の、野生データのラベルは、厳密には管理されておらず、目標概念に合致しているものも、そうでないものもあり、完全には信頼はできない。ここで、管理コストが大きい飼育データを大量に準備するのは困難だが、野生データは大量に獲得できると仮定する。この大量の野生データを用いて、飼育データの場合よりも、より高精度の予測を行うことが飼いならし学習問題の目標である。

野生データの一例として、協調タグ付け (collaborative tagging) によって得られるデータがある。協調タグ付けは、del.icio.us<sup>\*1</sup> などのようなソーシャルブックマークのようなサービスである。利用者は、好きな Web ページを登録し、その内容、特徴、カテゴリなどを表すタグを付与できる。これらのタグは、自身の登録ページの検索や分類に役立つ。さらに、登録ページやそのタグをに他の利用者と共有することで、他の利用者が登録したページを検索したり、嗜好が似ている利用者を見つけたりすることもできる。

この協調タグ付けでは、各利用者が個人的な規準に従って自由にタグを付加できる。そのため、多様な規準に基づくタグが使われることを Golder と Huberman は指摘している [Golder 06]。Golder らは、この多様性の原因を幾つか挙げているが、中でも、一般-特殊の階層からの語の選択の問題を指摘している。例えば、プログラム言語 python のホームページには、python という内容を限定した特殊なタグを付けてもよいし、より一般的で広い内容を表す programming というタグを付けてもよい。こうした特殊-一般の度合いは統制されず、多様なものが用いられる。他にも、関連はあるが、異なった概念を表す同形の語である polysemy の問題もある<sup>\*2</sup>。例えば、data mining という語を、データを用いたマーケティングでの統計手法の一つと考える利用者もいるし、より大規模データからの統計的な知識発見とする利用者もいる。これらの多様性のため、ある利用者がラベル付けしたタグは、他の利用者にとって適切とは限らず、

検索などに利用すると不都合を生じたりする。

この問題に対処するため、整合性のある概念に基づいてラベル付けされた事例集合があるとし、これを飼育データと呼ぶ。だが、こうした整合性のあるラベル付けには人的・時間的なコストが必要になり、準備できるデータの量は一般に少なくなる。そのため、統計的な規則性をこのデータだけから発見するのは難しい。そこで、飼育データに加えて、多くの利用者によって比較的に大量に収集できる協調タグ付けで得られたデータも併用する。このデータ集合は、厳密には管理されておらず、飼育データと同じ概念に基づいてラベル付けされているデータも、そうでないものも含んでいる。このような事例集合を野生データと呼ぶ。飼育データと野生データは、そのデータ量と信頼性において相互に補完しあう関係にある。よって、これらを併用し、互いのデータの長所を活用すれば、正確な予測ができるだろう。そこで、飼育と野生データの両方を訓練に用いて、より高精度の予測を目標とする学習問題を考え、これを飼いならしと呼ぶ。この問題の解法は、少数の正確な観測データと、多数の不正確な観測データがある他の状況にも応用できるだろう。本論文では、この問題に対して、バギング (bagging) [Breiman 96] を改造した解法を開発し、その有効性を協調タグ付けの問題で検証する。

2. 節では飼いならしの学習問題とその解法を示し、3. 節では、協調タグ付けのタグ予測問題に適用し、その有効性を示す。4. 節で関連研究とまとめを述べる。

## 2. 飼いならし学習問題とその解法

本節では、飼いならし学習問題と、この問題のためにバギングを改造して開発した BaggTaming について述べる。

## 2.1 飼いならし学習問題

まず、飼いならし学習問題について述べる。飼いならしは、回帰など他の教師あり学習問題についても定義できるが、ここではクラス分類問題を扱う。対象を特徴ベクトル  $\mathbf{x}$  で記述し、この対象が分類されるクラスを  $c$  と記す。クラスと特徴ベクトルの具体例の対  $(c_i, \mathbf{x}_i)$  の集合である訓練集合から、任意の特徴ベクトル  $\mathbf{x}$  に対して、これが分類されるべき適切なクラスを予測する分類器を獲得するのがクラス分類問題である。

通常のクラス分類問題では、この訓練集合は1種類だけである。この訓練事例は、獲得すべき目標概念を表すクラスと特徴ベクトルの同時分布  $P[c, \mathbf{x}]$  から独立にサンプリングされたと仮定する。一方、飼いならしの場合は、飼育データと野生デー

連絡先: 神島 敏弘, <http://www.kamishima.net/>

\*1 <http://del.icio.us/>

\*2 無関係な同形の語である homonymy とは区別する

タの2種類の訓練事例集合を用いる。飼育データに含まれる事例は、通常のクラス分類と同様に、獲得すべき目標概念を表す同時分布  $P[c, \mathbf{x}]$  から独立にサンプリングされたと仮定する。この飼育データを  $\mathcal{D}_T = \{(c_i, \mathbf{x}_i)\}_{i=1}^{N_T}$  と記す。ただし、 $N_T = |\mathcal{D}_T|$  である。野生データには、目標概念の分布  $P[c, \mathbf{x}]$  から独立にサンプリングされた事例以外にも、別の概念を表す分布から生成された事例も含まれる。目標概念の事例は、少なくとも  $N_T$  の2~3倍は含まれているとし、どの事例が目標概念の分布から生成されたかは未知である。この野生データを  $\mathcal{D}_W = \{(c_i, \mathbf{x}_i)\}_{i=1}^{N_W}$  と記す。ただし、 $N_W = |\mathcal{D}_W|$  であり、野生データは飼育データより十分に多い、すなわち、 $N_W \gg N_T$  と仮定する。

前節の協調タグ付けの問題での例を示す。一人の利用者がタグ付けする、もしくは、図書分類のように厳密に定めた規準に基づいてタグ付けするとしよう。このとき、特徴ベクトルとクラスの間には高い一貫性が保たれ、獲得すべき目標概念の分布に十分に高い精度で従っているだろう。よって、こうしてタグ付けされたデータは、飼育データとして利用できる。厳密にタグ付けの規準を共有するため、こうしたタグ付けを複数で行うのは困難であり、多数のWebページにタグをつけるのは難しい。よって、相対的に飼育データ数  $N_T$  は小さくならざるおえない。一方、協調タグ付けによって、多数の利用者が、個々の規準に基づいて自由に与えたタグを考えよう。こうしたタグの中には、獲得すべき目標概念と同じ規準で付けられたものが存在するだろう。すなわち、いろいろな特殊性の度合いのなかで目標概念と同じものを選ぶことや、polysemyのある語でも、目標概念と同じ意味でタグを選ぶことは十分に高い頻度でなされるだろう。それと同時に、違う特殊性の語を選んだり、別の意味でタグ付けをすることもあり得るだろう。よって、協調タグ付けによるタグは、一部は目標概念と一致するが、残りは一致しないようなデータとなっており、野生データとして扱うのが適切である。また、協調タグ付けでは、非常に多数の利用者がタグを付けるため、多くの野生データを収集でき、一般に  $N_W \gg N_T$  となる。

飼育データと野生データは補完し合う関係にある。すなわち、飼育データは信頼できるがその数は少なく、野生データは信頼できないが多数ある。もし、野生データから目標概念に従うデータを取り出して、飼育データと併用することができれば、飼育データ単独から学習する場合よりも、より多くの情報を利用できるだろう。こうしてより正確にラベルを予測できる分類器を獲得するのが飼いならし学習問題の目標である。

## 2.2 BaggTaming

飼いならし学習問題の解法として、**BaggTaming (Bootstrap AGGgregated TAMING)** と呼ぶ手法を開発した。この方法はバギング (bagging; Bootstrap AGGgregated TAMING) [Breiman 96] の考えに基づいているので、まずこのバギングについて述べる。

バギングは、次の手続きを  $t = 1, \dots, T$  について繰り返す。

1. 訓練事例集合  $\mathcal{D}$  からブートストラップサンプリング、すなわち、事例を復元抽出して、訓練事例集合  $\mathcal{D}_t$  を得る。
2.  $\mathcal{D}_t$  を訓練事例集合として、弱分類器  $\hat{f}_t(\mathbf{x})$  を学習する。

この手続きにより、 $T$  個の弱分類器  $\hat{f}_1(\mathbf{x}), \dots, \hat{f}_T(\mathbf{x})$  が得られる。そして、これらの分類器の凝集 (aggregation) によって最終結果を決める。凝集は、クラス分類問題では多数決により行う。

このバギングにより、予測精度が向上する理由については [Breiman 96, Breiman 98] など、バイアス-バリエーションの考え

```

1:  $t = 1$ 
2: while  $t \leq T$  do
3:    $s = 1$ 
4:    $\mathcal{D}_W$  から事例を復元抽出して訓練事例集合  $\mathcal{D}_t$  を生成
5:    $\mathcal{D}_t$  から弱学習器  $\hat{f}_t(\mathbf{x})$  を学習
6:    $\hat{f}_t(\mathbf{x})$  の、飼育データ  $\mathcal{D}_T$  に対する経験正解率  $p_t$  を計算
7:   if  $p \geq \text{AccLimit}$  then  $t = t + 1$ , ステップ 2 へ
8:   if  $s \geq \text{FailureLimit}$  then
       今回のループで  $p_t$  の最も高い分類器を  $\hat{f}_t(\mathbf{x})$  に設定
        $t = t + 1$ , ステップ 2 へ
9:    $s = s + 1$ , ステップ 4 へ
10: 弱学習器  $\hat{f}_1(\mathbf{x}), \dots, \hat{f}_T(\mathbf{x})$  を、正解率  $p_1, \dots, p_T$  と共に出力

```

図 1: BaggTaming アルゴリズム

に基づいて簡潔な説明がなされている。バイアス-バリエーションの理論では、誤差を、予測に用いたモデルに由来する誤差であるバイアス、学習に用いた訓練集合のサンプリングの揺らぎに由来するバリエーション、本質的に減らせない誤差の三つの部分に分解する。多様な関数を近似できる低バイアスなモデルでは、バイアスに由来する誤差は小さくなるが、サンプリングの揺らぎに由来するバリエーションは大きくなる。一方、高バイアスモデルではその逆の状況が生じるとされている。バギングは、いろいろな訓練事例を生成し、それらから学習した弱分類器を凝集することで、サンプルに由来する誤差、すなわち、バリエーションを小さくする一方で、バイアスに由来する誤差はそのままに保つことができる。そのため、低バイアスモデルを使えば、バギングによって汎化誤差を小さくできる。一方、Fisher 判別分析のような高バイアスのモデルでは、もともと誤差の中でバリエーションの占める要素が少なく、バギングではあまり誤差を減らせない。

このバギングの考えを利用するのが BaggTaming である。通常のバギングでは、与えられた訓練集合から復元抽出をする。その代わりに、元の訓練集合から得た事例の特徴ベクトルに乱数を加え、より多様な訓練集合から弱学習器を作ることで、バギングの性能を改善できる [Breiman 98]。このアイデアに基づいて、飼育データから復元抽出するよりも、より多様な事例が多数含まれている野生データから復元抽出することで、バリエーションを積極的に減らす。しかし、ここで問題がある。野生データには、目標概念とは異なる分布から生成された事例が含まれていて、しかも、それがどれかも未知である。そこで、目標概念から得られていることが確実な飼育データを利用する。具体的には、野生データから学習した分類器で、飼育データを分類し、その正解率が十分に高ければ、学習に用いた野生データの訓練事例には、目標概念に従って生成された事例が多数含まれているとみなす。もし、飼育データに対する正解率が十分でなければ、その弱分類器は廃棄し、野生データから別の訓練事例集合をサンプリングして弱分類器を再び生成する。この手続きを繰り返すことで、目標概念を表す、多様な弱分類器を獲得できる。こうして、 $T$  個の弱分類器を生成し、これらの多数決で分類を行うのが BaggTaming である。

以上の手続きを図 1 に示す。ステップ 2 からのループでは、 $T$  個の弱分類器を生成する。ステップ 3 では、本ループで弱学習器の生成に失敗した回数  $s$  を初期化する。ステップ 4~6 では、野生データから訓練事例をサンプリングし、弱学習器  $\hat{f}_t(\mathbf{x})$  を獲得、飼育データに対する正解率  $p_t$  を計算する。この正解率が AccLimit より大きければその弱学習器を受理して、次の弱学習器を生成するのがステップ 7 である。なお、次節

の実験では、AccLimit には、飼育データ全て、野生データ全て、および両方のデータ全て、それぞれで訓練した弱学習器の、飼育データに対する正解率を用いた。一般に、飼育、両方、野生の順に受理規準は緩くなる。ステップ 8 は、実用的な時間でアルゴリズムを終わらせるため、弱学習器の棄却回数が FailureLimit 回を超えたら、本ループで一番正解率の高かった弱学習器を  $\hat{f}_t(\mathbf{x})$  として採用し、次の弱分類器の生成に移る。ステップ 9 は失敗回数を増やして、弱学習器の生成をやり直す。以上の手続きで、 $T$  個の弱分類器が得られるが、それと共に、各弱分類器の正解率  $p_t$  も出力しておく。

BaggTaming で、新たな入力  $\mathbf{x}$  に対する分類は、通常のバギングと同様に多数決で行うが、若干異なる点もある。前述のように  $p_t$  は、訓練に用いた野生データに、目標概念から得られたものが多く含まれていればより高くなるだろう。そこで、 $p_t$  の単調関数で重み付けして投票すれば、より精度の高い予測ができると考える。適切な非減少関数については今後の課題とし、今回は  $p_t$  そのものを重みとして用いた。形式的には、特徴ベクトル  $\mathbf{x}$  の所属するクラスを、次式で決める。

$$\hat{c} = \arg \max_{c \in C} \sum_{t=1}^T p_t I[c = \hat{f}_t(\mathbf{x})] \quad (1)$$

ただし、 $I[\cdot]$  は指示関数で、 $C$  はクラスの定義域である。

### 3. 協調タグ付けでの実験

協調タグ付けでのタグ予測問題に BaggTaming を適用する。

#### 3.1 協調タグ付けのデータと実験設定

実験に用いた協調タグ付けデータは、del.icio.us を 2007 年 7 月にクロールして収集したものである。登録 URL 数、タグの種類数、利用者数はそれぞれ、762,454, 172,817, および 6,488。違う利用者が同じ URL にタグを付けるので、URL とタグの対の総数は 3,198,185 であった。

そのタグが付けられている URL の種類数が多い上位 20 件のタグを選んだ。これらのタグそれぞれを目標タグとよび、任意の URL に目標タグが付加されるかどうかを予測する 2 値分類問題を扱う。各目標タグごとに、そのタグを最も多く付けた利用者に注目する。この利用者が一人で付けたタグは、この利用者の考えるタグ付け規準に高い精度で従っているだろう。そこで、この利用者を飼育利用者として、飼育利用者がタグ付けした全 URL 中で目標タグを付けたものを正例、付けていないものを負例として、飼育データとした。同様に、目標タグの上位 2~20 位までの利用者を野生利用者とし、彼らのタグ付けから野生データを生成した。この野生データには、飼育利用者と同じ規準でタグ付された URL も、そうでないものも含まれるだろう。

各 URL は、目標タグ以外の上位 100 の人気タグを選び、全利用者のうち、第  $i$  位の人気タグを、その URL に付加した利用者数を  $i$  番目の要素とする特徴ベクトルで表した。全ての特徴量が 0 となるような URL を、飼育・野生データから除いた結果、20 種類の各目標タグに対してする訓練事例数は表 1 のようになった。

弱学習器には、[McCallum 98] の多項分布モデルを用いた単純ベイズ分類器を用いた。やや高バイアスな分類器だが、高速に学習できるので、採用した。予測精度は 5 分割の交差確認で評価した。すなわち、各目標タグごとに、飼育と野生の両方のデータを 5 個のブロックに分割した。これらのうち、一つを順次テスト用に選び、残りを訓練用に用いる手続きを 5 回繰り返して、正解率のマイクロ平均を求めた。

表 1: 協調タグ付けデータの事例数

タグ	飼育	野生	タグ	飼育	野生
blog	2908	28214	web2.0	1784	13829
design	2511	26791	politics	1234	13709
reference	2355	22847	news	2473	13429
software	2658	22529	howto	1685	13407
music	2898	19725	imported	405	12862
programming	1697	18668	linux	1535	12231
web	2296	18503	blogs	1465	12217
tools	2365	18488	tutorial	1883	12001
video	2538	16734	games	2097	11291
art	2054	16521	free	1960	11258

表 2: 訓練サンプル数と受理規準を変えたときの失敗率

サンプル数	飼育	両方	野生
100%	0.477	0.067	0.001
50%	0.438	0.095	0.013
20%	0.405	0.143	0.038

### 3.2 実験結果

比較実験の前に、BaggTaming のパラメータ設定するための調査を行った。弱学習器の棄却回数の限界値 FailureLimit は、計算時間が大きくなりすぎない範囲でできるだけ大きな値として 100 に設定した。生成する弱学習器の数  $T$  が 10 の場合の失敗率、すなわち、弱学習器が FailureLimit 回棄却されて、代替として最良の弱学習器を採用した割合を、表 2 に示す。各行には、弱学習器の訓練事例としてサンプリングした事例数を、飼育データの大きさに対して 100%、50%、20% と変えた結果を示した。2.2 節で述べたように、弱学習器を受理する規準 AccLimit には、飼育データ全て、野生データ全て、および両方のデータ全て、それぞれで訓練した弱学習器の、飼育データに対する正解率を用いた。これらの結果を表の各列に示す。一般に、飼育、両方、野生の順に、受理規準が厳しくなる。受理規準が厳しいと計算時間はかかるが、正解率が高くなるのが期待される。表 2 では、全飼育データの正解率を用いると受理規準が厳密過ぎ、40%以上の弱学習器が受理されず、効率的ではない。そこで、10%前後の失敗率になる両方のデータの正解率を受理規準に用いた。弱学習器を学習するための訓練サンプルの大きさは、野生データ中に含まれる目標概念の事例数によって適切な値が決まるが、これは未知なので、ここでは中間的な飼育データ数の 50%とした。

提案手法 BaggTaming を、飼育データのみを訓練事例に用いたバギングと比較した実験結果を表 3 に示す。弱学習器数  $T$  はどちらも 10 個、その他の条件は上に記したとおりである。ALL~1/8 の列のラベルは、野生データはそのまま全て用いたままで、飼育データ数を、全て使った場合から、その 1/8 だけ用いた場合まで変化させた結果を示す。各列の左側には BaggTaming の、右側にはバギングの結果を示した。提案手法がバギングより正解率が高い場合は太字にしてある。飼育データ数を ALL~1/8 と変えたとき、提案手法がバギングより優れていたタグはそれぞれ 12, 15, 15, 17 個といずれも半分以上である。しかも、飼育データが少なく、飼いならしによる学習がより有用になる状況で、より結果が改善されている。この他に、飼育と野生データの両方を訓練事例とした通常のパギングとも比較したが、飼育データの場合と大差はなかった。この表では、提案手法の方が、バギングより悪くなっている場合があるが、サンプル数や受理規準の調節によって、これらも多くの場合回避できる。この調節を適切に行う方法については、

表 3: BaggTaming とバギングの予測正解率

タグ	ALL		1/2		1/4		1/8	
	BT	bagg	BT	bagg	BT	bagg	BT	bagg
blog	0.693	0.740	0.705	0.728	0.715	0.735	0.803	0.816
design	<b>0.744</b>	0.719	<b>0.761</b>	0.734	<b>0.801</b>	0.773	<b>0.727</b>	0.705
reference	<b>0.824</b>	0.819	<b>0.775</b>	0.758	<b>0.778</b>	0.750	<b>0.789</b>	0.747
software	0.760	0.773	<b>0.744</b>	0.720	<b>0.767</b>	0.718	<b>0.813</b>	0.745
music	<b>0.958</b>	0.955	<b>0.956</b>	0.938	<b>0.954</b>	0.927	<b>0.962</b>	0.929
programming	0.897	0.898	0.888	0.894	0.862	0.866	<b>0.829</b>	0.812
web	0.665	0.776	0.748	0.816	0.752	0.820	0.759	0.840
tools	<b>0.761</b>	0.748	<b>0.712</b>	0.681	<b>0.689</b>	0.635	<b>0.680</b>	0.584
video	0.886	0.892	<b>0.883</b>	0.878	<b>0.857</b>	0.814	<b>0.906</b>	0.868
art	<b>0.919</b>	0.902	<b>0.918</b>	0.882	<b>0.921</b>	0.871	<b>0.943</b>	0.862
web2.0	0.630	0.698	0.634	0.773	0.657	0.765	0.678	0.774
politics	0.636	0.667	0.713	0.720	0.800	0.803	<b>0.823</b>	0.820
news	<b>0.955</b>	0.806	<b>0.853</b>	0.735	<b>0.824</b>	0.294	<b>0.625</b>	0.375
howto	<b>0.901</b>	0.900	<b>0.872</b>	0.869	<b>0.871</b>	0.854	<b>0.883</b>	0.847
imported	<b>1.000</b>	0.971	<b>1.000</b>	0.965	<b>1.000</b>	0.930	<b>0.955</b>	0.864
linux	<b>0.791</b>	0.745	<b>0.800</b>	0.769	<b>0.865</b>	0.819	<b>0.868</b>	0.799
blogs	<b>0.936</b>	0.921	<b>0.934</b>	0.902	<b>0.934</b>	0.904	<b>0.929</b>	0.906
tutorial	<b>0.909</b>	0.891	<b>0.873</b>	0.835	<b>0.859</b>	0.809	<b>0.875</b>	0.802
games	0.959	0.964	<b>0.960</b>	0.958	<b>0.952</b>	0.943	<b>0.938</b>	0.931
free	<b>0.813</b>	0.805	<b>0.756</b>	0.690	<b>0.748</b>	0.678	<b>0.777</b>	0.608

今後の課題としたい。

以上の実験より、信頼できないが大量に準備できる野生データも訓練事例に用いて、BaggTaming アルゴリズムを適用することで、飼育データのみで学習する場合より予測精度を向上できることが示された。

#### 4. 関連研究とまとめ

飼いならし学習と同様に、性質の異なる複数の訓練データ集合を用いる学習問題は幾つかある。ラベルあり・なしの両方のデータを用いる半教師あり学習 [Chapelle 06] とは、野生・飼育どちらのデータにもラベル情報が与えられている点が明確に異なる。別のドメインのデータを併用して、目標ドメインでの性能を改善するドメイン適応 [Daumé III 07] もある。両方のデータにラベルがある点では似ているが、ドメイン適応では、各ドメインごとに信頼できるラベルが与えられるのに対し、飼いならしでは同じドメインで、信頼性の異なるデータが与えられる。共変量シフト [杉山 06] は、訓練とテストデータで、入力の分布が異なる状況を扱う。ラベルが与えられる訓練集合一つであるので、やはり、飼いならしとは異なる問題である。帰納転移 (マルチタスク学習) [Thrun 96] では、以前に学習した知識を、新たな学習で活用する。この問題は、以前の学習で用いた訓練データラベルは全て信頼できる点がやはり異なる。協調タグ付けでタグを予測する研究も幾つかある。[Mishne 06, Chirita 07]。いずれの研究も、整合性のある自身のタグのみを利用するか、他の利用者とのタグ付けの整合性の問題は考慮していない。

本稿では、信頼できる少数の飼育データと、不整合のある大量の野生データから知識を獲得する飼いならし学習を提案し、バギングの考えに基づく手法を開発し、その有効性を実験により示した。今後は、理論的背景の強化、適応的なサンプリングによる効率化、確率モデルの変更による予測精度の向上、他の問題への適用を行う予定である。

謝辞: 松尾豊先生と藤井敦先生には数々のご教示をいただいた。株式会社ホットリンクには Web ページのクロールでご助力をいただいた。以上の方々に感謝する。

#### 参考文献

- [Breiman 96] Breiman, L.: Bagging Predictors, *Machine Learning*, Vol. 24, pp. 123–140 (1996)
- [Breiman 98] Breiman, L.: Arcing Classifiers, *The Annals of Statistics*, Vol. 26, No. 3, pp. 801–849 (1998)
- [Chapelle 06] Chapelle, O., Schölkopf, B., and Zien, A. eds.: *Semi-supervised Learning*, MIT Press (2006)
- [Chirita 07] Chirita, P.-A., Costache, S., Handschuh, S., and Nejd, W.: P-TAG: Large Scale Automatic Generation of Personalized Annotation TAGs for the Web, in *Proc. of The 16th Int'l Conf. on World Wide Web*, pp. 845–854 (2007)
- [Daumé III 07] Daumé III, H.: Frustratingly Easy Domain Adaptation, in *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 256–263 (2007)
- [Golder 06] Golder, S. A. and Huberman, B. A.: Usage Patterns of Collaborative Tagging Systems, *J. of Information Science*, Vol. 32, No. 2, pp. 198–208 (2006)
- [McCallum 98] McCallum, A. and Nigam, K.: A Comparison of Event Model for Naive Bayes Text Classification, in *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48 (1998)
- [Mishne 06] Mishne, G.: AutoTag: A Collaborative Approach to Automated Tag Assignment for Weblog Posts, in *Proc. of The 15th Int'l Conf. on World Wide Web*, pp. 953–954 (2006)
- [杉山 06] 杉山 将: 共変量シフト下での教師付き学習, 日本神経回路学会誌, Vol. 13, No. 3, pp. 111–118 (2006)
- [Thrun 96] Thrun, S.: Is Learning The  $n$ -th Thing Any Easier Than Learning The First?, in *Advances in Neural Information Processing Systems 8*, pp. 640–646 (1996)