

Frustratingly Easy Domain Adaptation

Hal Daumé III. Proceedings of the 45th Annual Meeting of the
Association of Computational Linguistics, pp.256-263, June, 2007

M1 飯塚卓哉

論文概要

- 機械学習の一種である**転移学習**の中の**ドメイン適応**への新たなアプローチを提案
- 非常に簡単（Perlで10行）な実装を行うだけで，従来のアプローチよりも良い性能を発揮
- マルチドメイン適応にも拡張可能

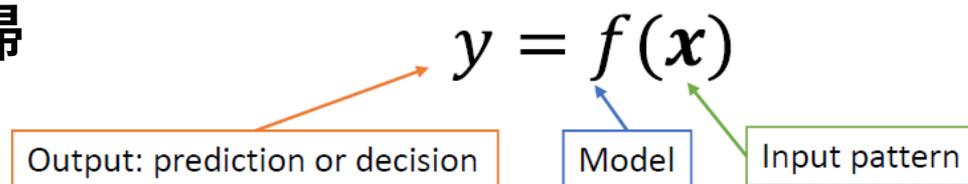
目標

- 転移学習とは何か理解する
- ドメイン適応とは何か理解する
- 本論文の提案する手法を理解する

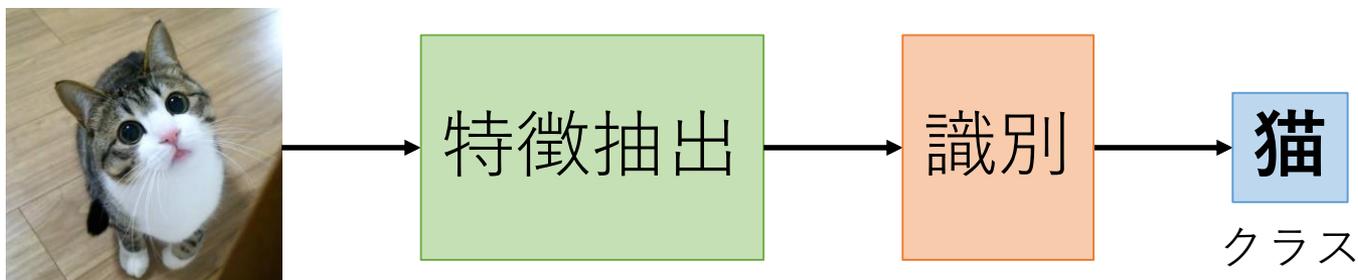
目次

0. 機械學習基礎
1. Introduction
2. Problem Formalization and Prior work
3. Adaptation by Feature Augmentation
4. Results
5. Discussion

分類と回帰



- 回帰 (regression)
 - 従属変数 y が連続変数である場合.
 - 独立変数 x が1次元の場合を単回帰, 2次元以上を重回帰という.
 - 例) 国民の所得から経済全体の消費を推定する.
- 分類, 識別 (classification)
 - 従属変数 y が有限個の離散カテゴリの場合.
 - 例) 画像を入力して, そこに写る動物の種類 (犬, 猫, 馬など) を推定する.



特徴抽出と特徴ベクトル

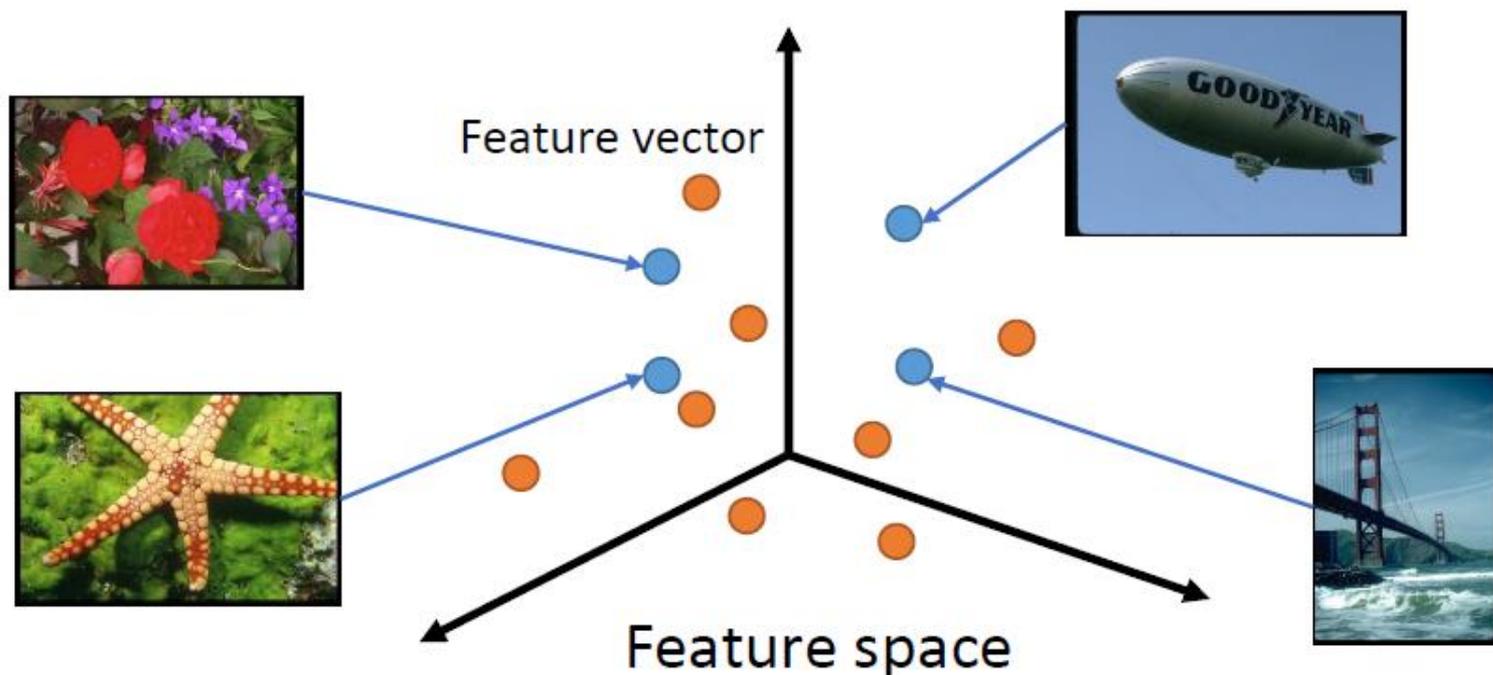
- 特徴抽出 (feature extraction)
 - 多くの情報を持ったパターンから、認識に本質的な情報を抽出すること.
 - パターンを圧縮した効率のよい空間の点として表現する.
 - ノイズ除去などの前処理も含む.
- 特徴ベクトル (feature vector)
 - 例えば文字認識の場合、特徴量として、線の傾き、長さ、曲率などが考えられる.
 - それぞれの特徴を数値で表現し、それらを組としたベクトルが利用される.



$$\mathbf{x} = (x_1, x_2, \dots, x_d)^T$$

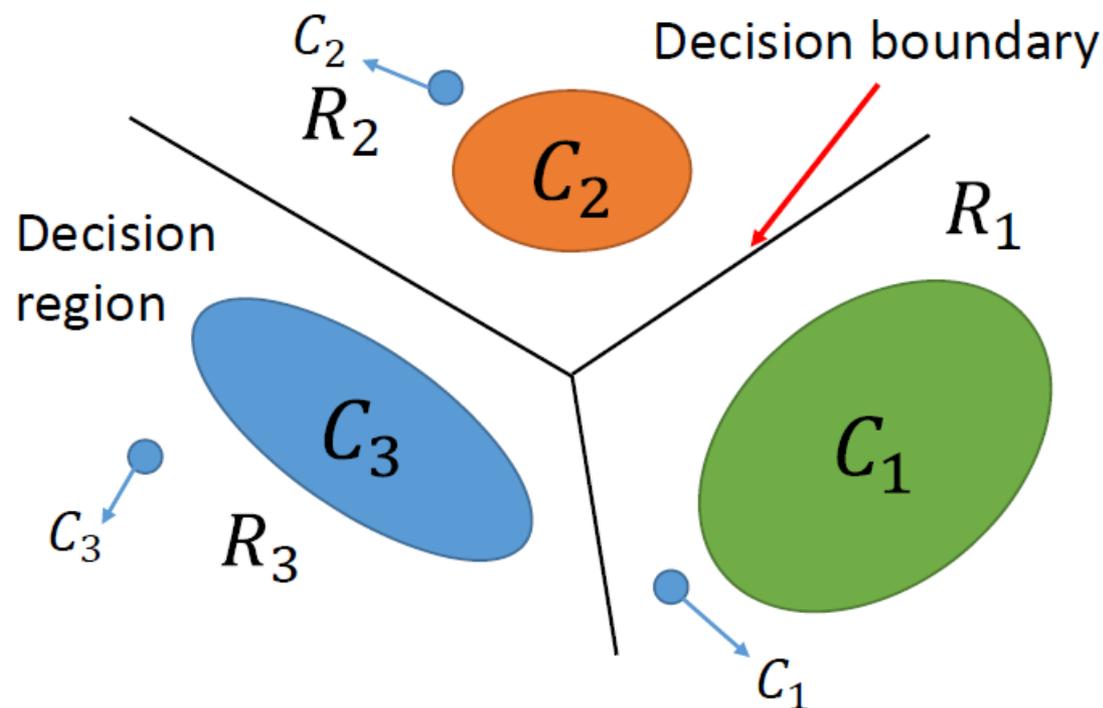
線の傾き 長さ 曲率

特徴空間



- 特徴ベクトルによって張られる空間を特徴空間 (feature space) という.
- 1つのパターンは d 次元の特徴空間の1点として表現される.

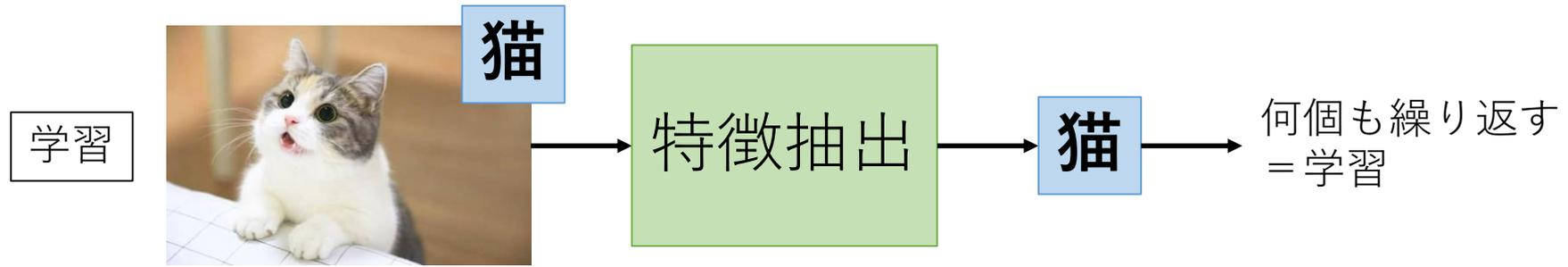
決定境界・決定領域



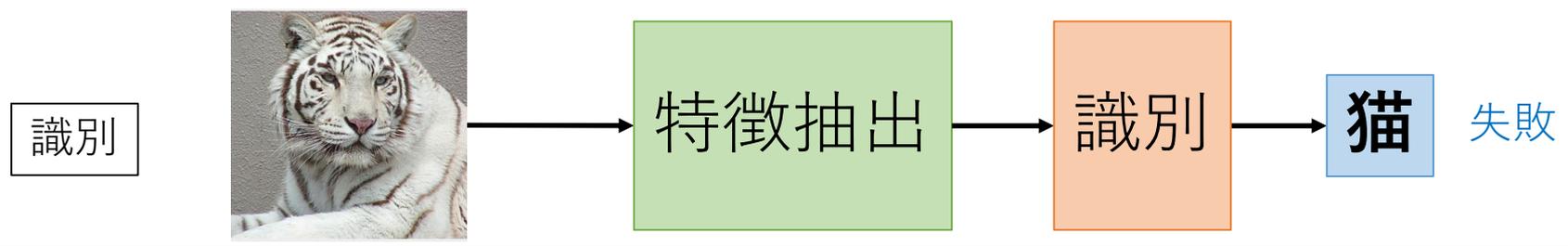
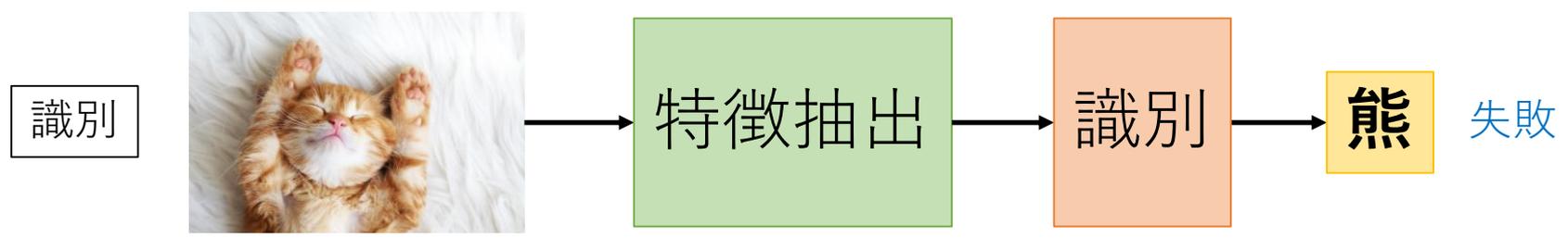
- 決定境界 (decision boundary)
 - クラス間を分離する境界
- 決定領域 (decision region)
 - クラスラベルの付与された分割された領域

→ 識別器 (関数) を設計する

教師あり学習 (Supervised learning)

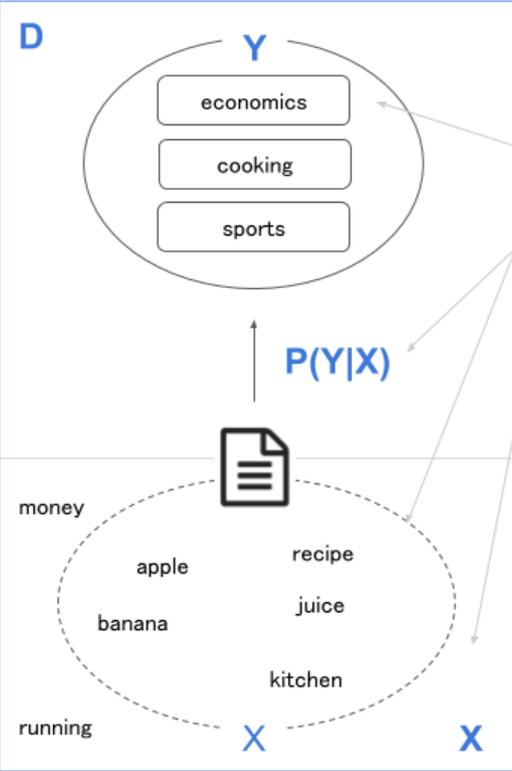


ラベル付きデータが不十分だと.....



転移学習 (Transfer learning)

ドメイン (Domain)



X: 観測データの集合($x_1 \sim x_n$ で構成)
X: 観測データの母集団
Y: ラベルの空間($y_1 \sim y_n$ で構成)
 $P(Y|X)$: Xが観測されたときにラベルがYである確率

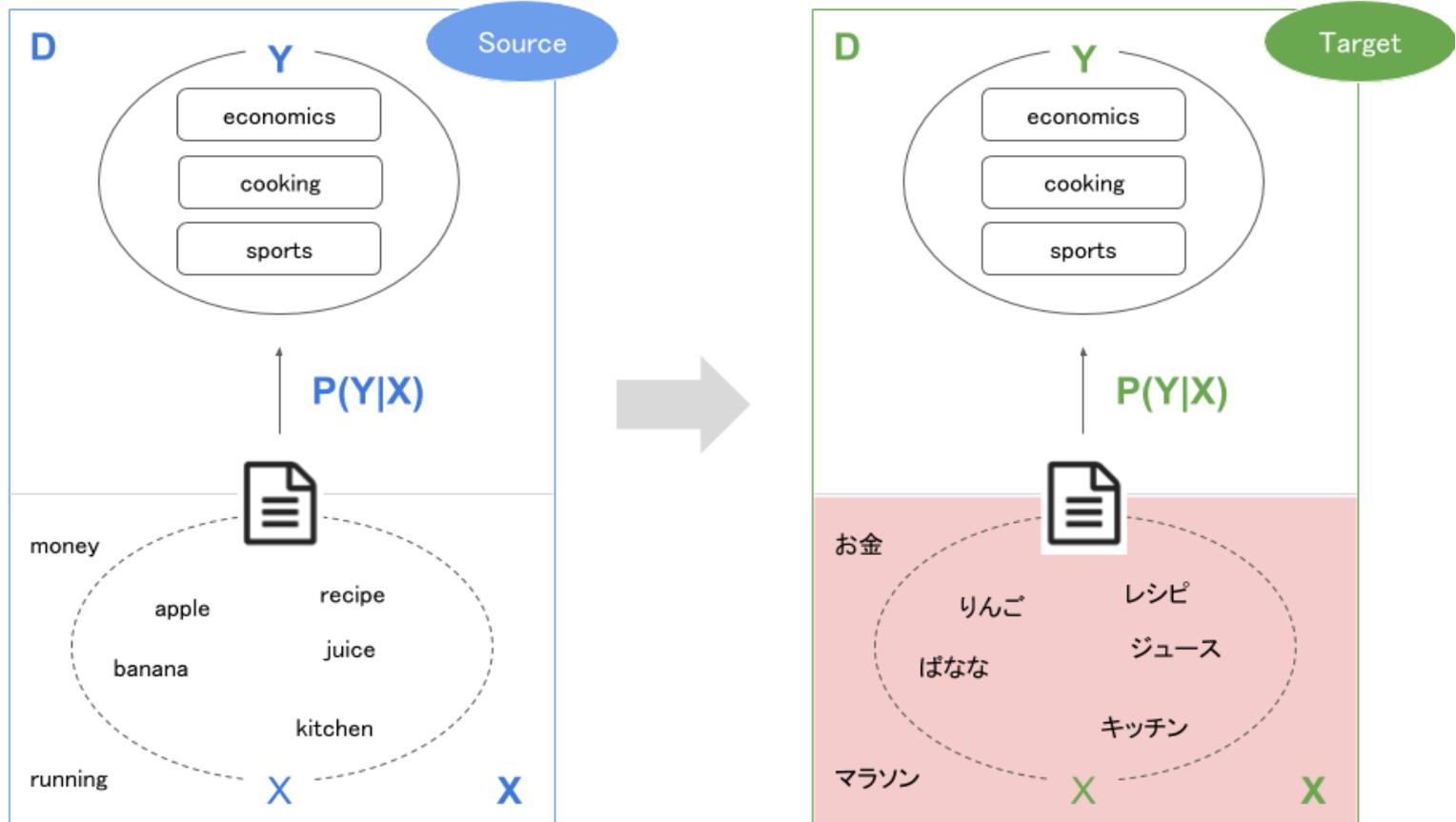
ドメイン (D)

転移学習
ソースドメイン (D_S) で構築したモデルを, ターゲットドメイン (D_T) に, より少ないデータで適応させる

ソースドメイン (D_S) : 十分なラベル付きデータが取得可能
ターゲットドメイン (D_T) : ラベル付きデータが不足

転移学習 (Transfer learning)

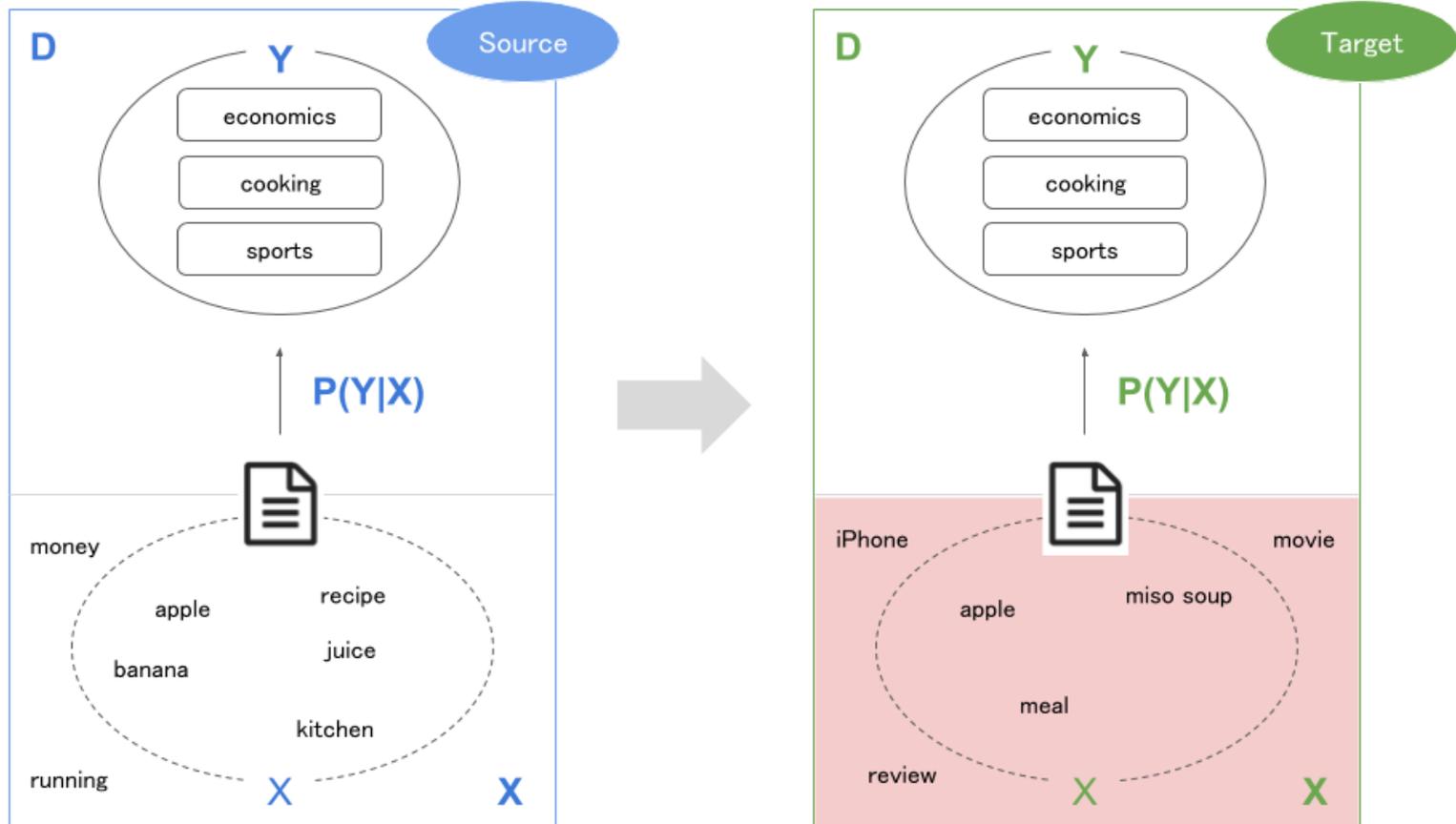
観測データが異なる ($X_S \neq X_T$)



- タスクは同じだが、観測データが違う
- Cross-xxx Adaptation

転移学習 (Transfer learning)

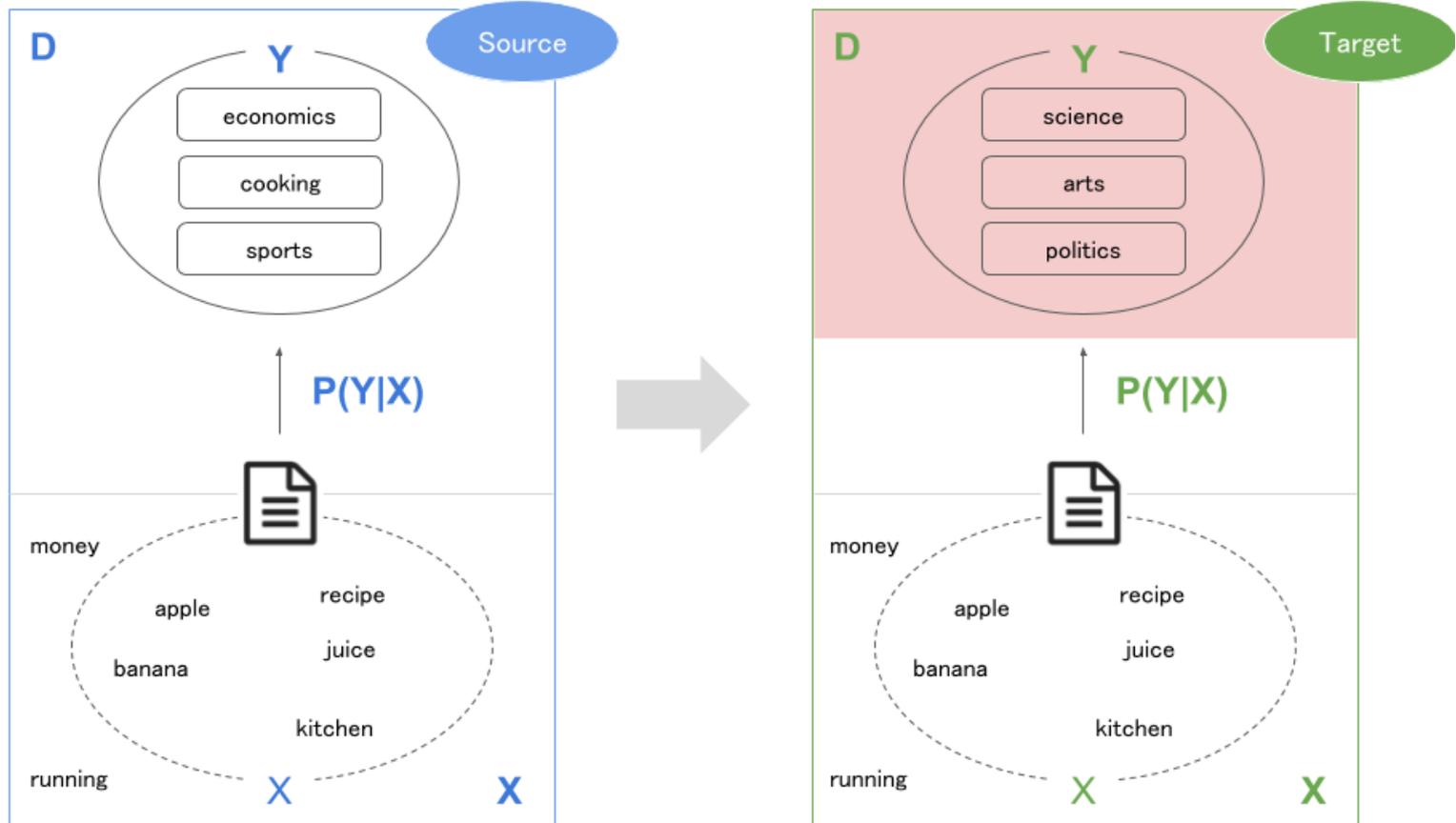
観測データの分布が異なる ($P(X_S) \neq P(X_T)$)



- 単語の出現頻度がドメインによって違う
- **Domain Adaptation**

転移学習 (Transfer learning)

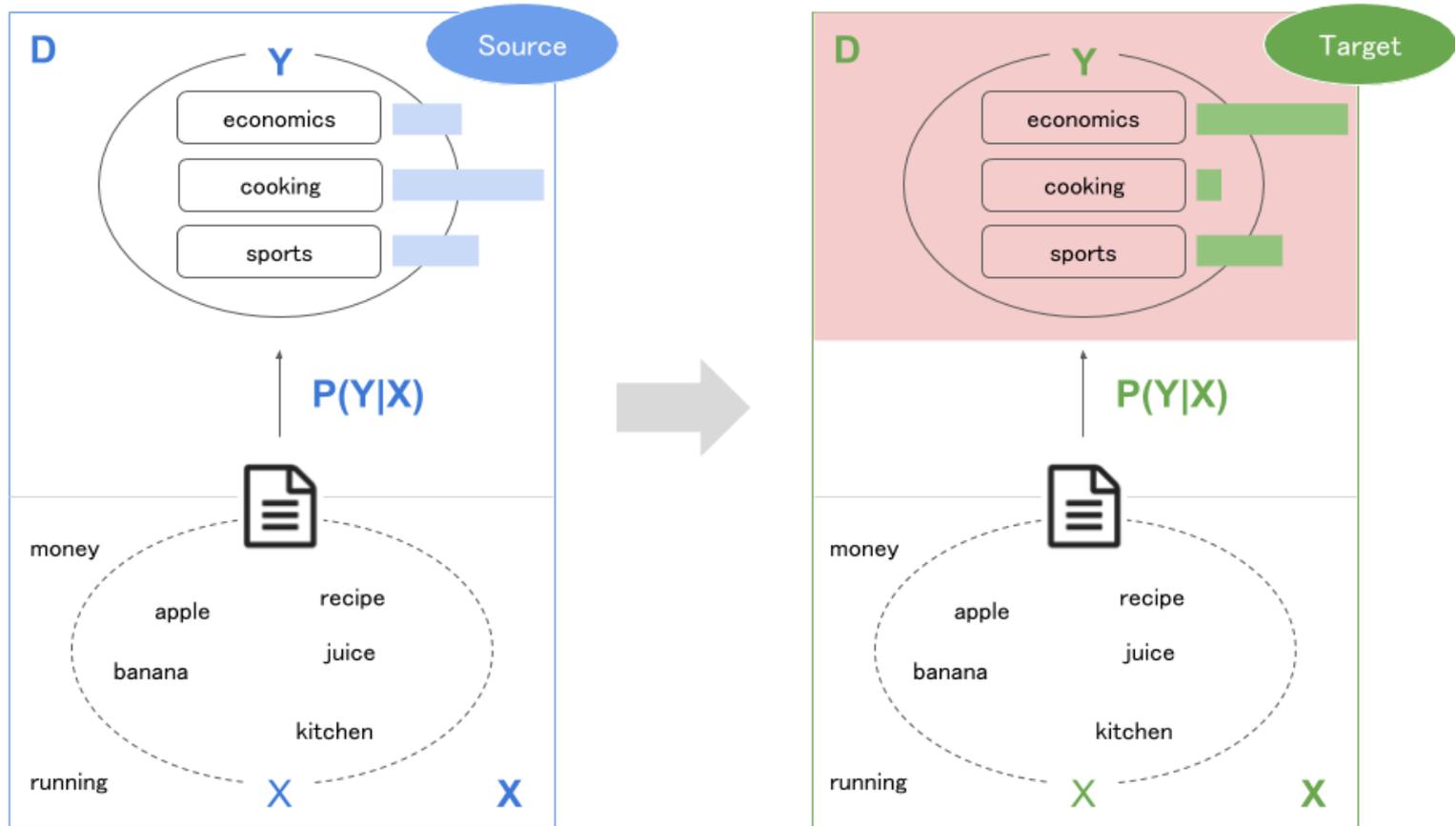
ラベルが異なる ($Y_S \neq Y_T$)



- 予測ラベルが異なる
- Fine Tuningと呼ばれたりする

転移学習 (Transfer learning)

ラベルの分布が異なる ($P(Y_S) \neq P(Y_T)$)



- 予測ラベルの出現頻度が異なる

1. Introduction

- ドメイン適応は自然言語処理（Natural Language Processing, NLP）の分野で研究が盛ん
 - 大量のラベルデータがある特定のドメインで入手可能だから（newswireなど）
- 本研究のアプローチは、ドメイン適応問題を、通常の教師付き学習問題に変換すること
 - ソース・ターゲットデータの状態空間を拡張する
- 教師付き学習の場合のみを対象にする
 - 半教師付き学習の場合は対象外
 - ソースデータから大量の注釈付きコーパスを取得可能で、ターゲットデータの内の少量のコーパスに注釈をつけることに少しだけお金と労力をかけられる場合。

※コーパスとは

- 例文集（文字・音声問わず）
- 注釈付け（annotation）をされているものが作成されている
 - ex) 品詞タグ付きコーパス，構文構造付きコーパス

```
# S-ID:950101004-003 KNP:96/10/27 MOD: MEMO:?
0 3D
ロシア ろしあ * 名詞 地名 **
側 がわ * 接尾辞 名詞性名詞接尾辞 **
は は * 助詞 副助詞 **
1 2D
首都 しゆと * 名詞 普通名詞 **
制圧 せいあつ * 名詞 サ変名詞 **
の の * 助詞 接続助詞 **
2 3D
最終 さいしゅう * 名詞 普通名詞 **
段階 だんかい * 名詞 普通名詞 **
に に * 助詞 格助詞 *** 3 4D
入った はいった 入る 動詞 * 子音動詞ラ行 タ形
と と * 助詞 格助詞 **
4 -1D
み みみる 動詞 * 母音動詞 未然形
られる られる られる 接尾辞 動詞性接尾辞 母音動詞 基本形
。 。 * 特殊 句点 **
EOS
```

京大コーパス→

2. Problem Formalization and Prior work

表記法

\mathcal{X} : 入力空間

\mathcal{Y} : 出力空間

\mathcal{D}^s : ソースデータの分布

\mathcal{D}^t : ターゲットデータの分布

$D^s \sim \mathcal{D}^s$: ソースデータからのサンプル (N 個)

$D^t \sim \mathcal{D}^t$: ターゲットデータからのサンプル (M 個)

目標

$N \gg M$ のときに、ターゲットドメインで期待損失の少ない関数 $h : \mathcal{X} \rightarrow \mathcal{Y}$ を学習する。

基本のドメイン適応

- SRCONLY : ソースデータだけでモデルを学習
- TGTONLY : ターゲットデータだけでモデルを学習
- ALL : 二つのデータセットの和集合でモデルを学習

問題点

- D^s が D^t の効果をかき消す

→ WEIGHTED : ソースドメインに重みをかけて影響を等しくする。重みは交差検証で決定

2. Problem Formalization and Prior work

基本のドメイン適応

- PRED：まずSRCONLYを学習。そのモデルをターゲットデータに適用した分類結果を、ターゲットデータでモデルを構築する際の特徴量のひとつとして利用。
- LININT：SRCONLYとTGTONLYの線形補間モデル
→ これらはかなり性能がいい

2009年時点でより性能の良いモデルは2つ。

PRIOR Model

- SRCONLYの学習結果をターゲットデータでの学習の重みとして扱う
- 多くの学習アルゴリズム（エントロピー最大化，サポートベクターマシン，平均化パーセプトロン，単純ベイズ）は重み正則化項を持つ ($\lambda \|\mathbf{w}\|_2^2$)
- これを，SRCONLYで学習した重み \mathbf{w}^s を用いて， $\lambda \|\mathbf{w} - \mathbf{w}^s\|_2^2$ に置き換えてやる
- ターゲットデータで学習したモデルはSRCONLYで学習したモデルに似た重みを持つようになる

2. Problem Formalization and Prior work

Daumé III and Marcu(2006)のモデル

- 3つのモデルを学習
- ソース固有の情報を捉えるモデルと、ターゲット固有の情報を捉えるモデルと、generalなモデル
- ソースデータはソース固有の情報とgeneralな情報を、ターゲットデータはターゲット固有の情報とgeneralな情報を持っていると考える

→ 性能はよかったが、PRIOR Modelより10~15倍遅かった

3. Adaptation by Feature Augmentation

手法：特徴空間の拡張

- 拡張ソースデータ : ソース固有, generalなもののみ含む
- 拡張ターゲットデータ : ターゲット固有, generalなもののみ含む

$\mathcal{X} = \mathbb{R}^F, \mathcal{Y} = \{-1, 1\}$ として考える

- 拡張入力空間は $\tilde{\mathcal{X}} = \mathbb{R}^{3F}$ と定義する
- 写像 $\Phi^s, \Phi^t : \mathcal{X} \rightarrow \tilde{\mathcal{X}}$ をソースデータ, ターゲットデータの写像として以下のように定義

$$\Phi^s(\mathbf{x}) = \langle \mathbf{x}, \mathbf{x}, \mathbf{0} \rangle, \quad \Phi^t(\mathbf{x}) = \langle \mathbf{x}, \mathbf{0}, \mathbf{x} \rangle \quad \text{※} \mathbf{0} = \langle 0, 0, \dots, 0 \rangle \in \mathbb{R}^F$$

つまり, ソースデータの特徴ベクトルには Φ^s を, ターゲットデータの特徴ベクトルには Φ^t を適用して, 特徴空間を $\tilde{\mathcal{X}} = \mathbb{R}^{3F}$ として通常の教師あり学習をするだけ

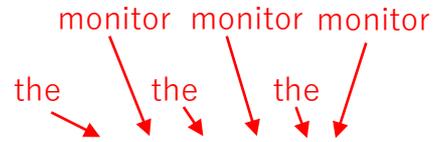
3. Adaptation by Feature Augmentation

$$\Phi^s(x) = \langle x, x, \mathbf{0} \rangle, \quad \Phi^t(x) = \langle x, \mathbf{0}, x \rangle$$

なぜこれでいいのか？

スピーチへのタグ付け問題を考えてみる

- ソースドメイン : Wall Street Journal
 - ターゲットドメイン: コンピュータハードウェアのレビュー
- “the” : どちらのドメインでも限定詞
 “monitor” : WSJでは動詞, レビューでは名詞であることが多そう



簡単に $\mathcal{X} = \mathbb{R}^2$ として考えてみる.

→ “the” : x_1 “monitor” : x_2 → 拡張状態空間 $\tilde{\mathcal{X}}$

$$\left\{ \begin{array}{l} \text{ソースドメイン} : \langle \bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, 0, 0 \rangle \\ \text{ターゲットドメイン} : \langle \bar{x}_1, \bar{x}_2, 0, 0, \bar{x}_5, \bar{x}_6 \rangle \end{array} \right.$$

\bar{x}_1, \bar{x}_2 は二つの指標のgeneralな特徴量, \bar{x}_3, \bar{x}_4 はソース固有の特徴量, \bar{x}_5, \bar{x}_6 はターゲット固有の特徴量

もし, モデルがうまく学習できたら,

- “限定詞”の重みベクトルは $\langle 1, 0, 0, 0, 0, 0 \rangle$ のようになる
- “名詞”の重みベクトルは $\langle 0, 0, 0, 0, 0, 1 \rangle$ のようになる
- “動詞”の重みベクトルは $\langle 0, 0, 0, 1, 0, 0 \rangle$ のようになる

3. Adaptation by Feature Augmentation

カーネル化 ver.

- データ点 x は半正定値カーネル $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ により定まる再生核ヒルベルト空間 \mathcal{X} の要素とする
→ K は \mathcal{X} 内の二つの（一般には無限次元の）ベクトルの内積で書くことができる：

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{X}}$$

Φ に関して、 Φ^s, Φ^t を以下のように定義する：

$$\Phi^s(x) = \langle \Phi(x), \Phi(x), \mathbf{0} \rangle, \quad \Phi^t(x) = \langle \Phi(x), \mathbf{0}, \Phi(x) \rangle$$

- x, x' を同じドメインから取ってきたとき、拡張カーネル $\check{K}(x, x')$ は：

$$\check{K}(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{X}} + \langle \Phi(x), \Phi(x') \rangle_{\mathcal{X}} = 2K(x, x')$$

- x, x' を違うドメインから取ってきたとき、拡張カーネル $\check{K}(x, x')$ は：

$$\check{K}(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{X}} = K(x, x')$$

つまり、
$$\check{K}(x, x') = \begin{cases} 2K(x, x') & \text{same domain} \\ K(x, x') & \text{diff. domain} \end{cases}$$

同じドメインの特徴は、異なるドメインの特徴より2重視される

= 識別のときにターゲットドメインのデータはソースドメインのデータより2倍の影響力を持つ

??????

3. Adaptation by Feature Augmentation

カーネル化 ver.

簡単に言うと...

- 同じドメインの内積は, $\langle x, x, 0 \rangle \cdot \langle x, x, 0 \rangle = x \cdot x + x \cdot x + 0 = 2x \cdot x$
- 違うドメインの内積は, $\langle x, 0, x \rangle \cdot \langle x, x, 0 \rangle = x \cdot x + 0 + 0 = x \cdot x$

同じドメインの方が2倍になるので同じドメインの特徴量は2重視される

3. Adaptation by Feature Augmentation

Analysis

- L2ノルム正則化 ($\lambda \|\mathbf{w}\|_2^2$) を行う識別器で、拡張特徴空間での重みを学習したとする
- w_s を重みベクトルのソースとgeneralの成分の合計, w_t をターゲットとgeneralの成分の合計とする → 重みベクトルの正則化条件は $\|w_g\|^2 + \|w_s - w_g\|^2 + \|w_t - w_g\|^2$ (w_g はこれを最小化するように選ばれる)
= PRIORモデルに似た, $\|w_s - w_t\|^2$ に比例する正則化器になる

解釈

- 提案したモデルは w_s と w_t を順にではなく、同時に最適化する
- さらに、一つの教師付き学習アルゴリズムがソース/ターゲットとgeneralの重みの間のトレードオフを調整してくれる

→ PRIORモデルよりも性能がいい

??????

3. Adaptation by Feature Augmentation

Multi-domain adaptation

- 提案手法はソースデータが複数あるときでも使える
- そういうことはよくある（ソースデータがより具体的な複数のドメインに分割してある時など）

2ドメイン（ソースとターゲット）のときは特徴空間を $\mathbb{R}^F \rightarrow \mathbb{R}^{3F}$ に拡張したので、
 K ドメインのときは特徴空間を $\mathbb{R} \rightarrow \mathbb{R}^{(K+1)F}$ に拡張すればよい

4. Results

Tasks : 文章ラベリング

- 固有表現抽出
- 構文解析
- 品詞タグ付け

Datasets:

- ACE-NER : 固有表現抽出に使用. Broadcast News(br), Broadcast Conversations(bc), Newswire(nw), Weblog(wl), Usenet(un), Conversational Telephone Speech(cts).
- CoNLL-NE : 固有表現抽出に使用. ソースドメインが2006 ACE, ターゲットドメインがCoNLL 2003 NER.
- PubMed-POS : 品詞タグ付けに使用. ソースドメインがPenn TreebankのWSJ部分, ターゲットドメインがPubMed.
- CNN-Recap : 再大文字化に使用. ソースドメインがnewswire, ターゲットドメインが自動音声認識のアウトプット文.
- Treebank-Chunk : 構文解析に使用. WSJドメイン, ATIS switchboard ドメイン, Brown corpus (8つのサブドメインからなる) で構成.
- Treebank Brown : Treebank-ChunkのBrown corpusを一つのドメインとみなしたバージョン.

Task	Dom	# Tr	# De	# Te	# Ft
ACE-NER	bn	52,998	6,625	6,626	80k
	bc	38,073	4,759	4,761	109k
	nw	44,364	5,546	5,547	113k
	wl	35,883	4,485	4,487	109k
	un	35,083	4,385	4,387	96k
CoNLL-NE	cts	39,677	4,960	4,961	54k
	src	256,145	-	-	368k
tgt	29,791	5,258	8,806	88k	
PubMed-POS	src	950,028	-	-	571k
	tgt	11,264	1,987	14,554	39k
CNN-Recap	src	2,000,000	-	-	368k
	tgt	39,684	7,003	8,075	88k
Treebank-Chunk	wsj	191,209	29,455	38,440	94k
	swbd3	45,282	5,596	41,840	55k
	br-cf	58,201	8,307	7,607	144k
	br-cg	67,429	9,444	6,897	149k
	br-ck	51,379	6,061	9,451	121k
	br-cl	47,382	5,101	5,880	95k
	br-cm	11,696	1,324	1,594	51k
	br-cn	56,057	6,751	7,847	115k
	br-cp	55,318	7,477	5,977	112k
br-cr	16,742	2,522	2,712	65k	

- 平均化パーセプトロンをベースにした学習アルゴリズムを使用 (Daumé III et al. (2007)で提案)

4. Results

Experimental Results

各ドメイン適応手法による誤差率の比較

Task	Dom	SRCONLY	TGTONLY	ALL	WEIGHT	PRED	LININT	PRIOR	AUGMENT	T<S	Win
ACE- NER	bn	4.98	2.37	2.29	2.23	2.11	2.21	2.06	1.98	+	+
	bc	4.54	4.07	3.55	3.53	3.89	4.01	3.47	3.47	+	+
	nw	4.78	3.71	3.86	3.65	3.56	3.79	3.68	3.39	+	+
	wl	2.45	2.45	2.12	2.12	2.45	2.33	2.41	2.12	=	+
	un	3.67	2.46	2.48	2.40	2.18	2.10	2.03	1.91	+	+
	cts	2.08	0.46	0.40	0.40	0.46	0.44	0.34	0.32	+	+
CoNLL	tgt	2.49	2.95	1.80	1.75	2.13	1.77	1.89	1.76		+
PubMed	tgt	12.02	4.15	5.43	4.15	4.14	3.95	3.99	3.61	+	+
CNN	tgt	10.29	3.82	3.67	3.45	3.46	3.44	3.35	3.37	+	+
	wsj	6.63	4.35	4.33	4.30	4.32	4.32	4.27	4.11	+	+
	swbd3	15.90	4.15	4.50	4.10	4.13	4.09	3.60	3.51	+	+
	br-cf	5.16	6.27	4.85	4.80	4.78	4.72	5.22	5.15		
Tree	br-cg	4.32	5.36	4.16	4.15	4.27	4.30	4.25	4.90		
bank-	br-ck	5.05	6.32	5.05	4.98	5.01	5.05	5.27	5.41		
Chunk	br-cl	5.66	6.60	5.42	5.39	5.39	5.53	5.99	5.73		
	br-cm	3.57	6.59	3.14	3.11	3.15	3.31	4.08	4.89		
	br-cn	4.60	5.56	4.27	4.22	4.20	4.19	4.48	4.42		
	br-cp	4.82	5.62	4.63	4.57	4.55	4.55	4.87	4.78		
	br-cr	5.78	9.13	5.71	5.19	5.20	5.15	6.71	6.30		
Treebank-brown		6.35	5.75	4.80	4.75	4.81	4.72	4.72	4.65	+	+

- 提案手法が最も性能がいい
- 二番手はPRIORモデルで、これは既往研究と一致

4. Results

Experimental Results

各ドメイン適応手法による誤差率の比較

Task	Dom	SRCONLY	TGTONLY	ALL	WEIGHT	PRED	LININT	PRIOR	AUGMENT	T<S	Win
ACE- NER	bn	4.98	2.37	2.29	2.23	2.11	2.21	2.06	1.98	+	+
	bc	4.54	4.07	3.55	3.53	3.89	4.01	3.47	3.47	+	+
	nw	4.78	3.71	3.86	3.65	3.56	3.79	3.68	3.39	+	+
	wl	2.45	2.45	2.12	2.12	2.45	2.33	2.41	2.12	=	+
	un	3.67	2.46	2.48	2.40	2.18	2.10	2.03	1.91	+	+
	cts	2.08	0.46	0.40	0.40	0.46	0.44	0.34	0.32	+	+
CoNLL	tgt	2.49	2.95	1.80	1.75	2.13	1.77	1.89	1.76		+
PubMed	tgt	12.02	4.15	5.43	4.15	4.14	3.95	3.99	3.61	+	+
CNN	tgt	10.29	3.82	3.67	3.45	3.46	3.44	3.35	3.37	+	+
Tree bank- Chunk	wsj	6.63	4.35	4.33	4.30	4.32	4.32	4.27	4.11	+	+
	swbd3	15.90	4.15	4.50	4.10	4.13	4.09	3.60	3.51	+	+
	br-cf	5.16	6.27	4.85	4.80	4.78	4.72	5.22	5.15		
	br-cg	4.32	5.36	4.16	4.15	4.27	4.30	4.25	4.90		
	br-ck	5.05	6.32	5.05	4.98	5.01	5.05	5.27	5.41		
	br-cl	5.66	6.60	5.42	5.39	5.39	5.53	5.99	5.73		
	br-cm	3.57	6.59	3.14	3.11	3.15	3.31	4.08	4.89		
	br-cn	4.60	5.56	4.27	4.22	4.20	4.19	4.48	4.42		
	br-cp	4.82	5.62	4.63	4.57	4.55	4.55	4.87	4.78		
	br-cr	5.78	9.13	5.71	5.19	5.20	5.15	6.71	6.30		
Treebank-brown		6.35	5.75	4.80	4.75	4.81	4.72	4.72	4.65	+	+

- 提案手法の性能が悪く、特に決まった性能の良い手法もない
- これらは全てSRCONLYがTGTONLYより性能がいい
- ソースドメインとターゲットドメインが似通っているために、ドメイン適応の効果が弱い

4. Results

Experimental Results

Task	Dom	SRCONLY	TGTONLY	ALL	WEIGHT	PRED	LININT	PRIOR	AUGMENT	T<S	Win
ACE- NER	bn	4.98	2.37	2.29	2.23	2.11	2.21	2.06	1.98	+	+
	bc	4.54	4.07	3.55	3.53	3.89	4.01	3.47	3.47	+	+
	nw	4.78	3.71	3.86	3.65	3.56	3.79	3.68	3.39	+	+
	wl	2.45	2.45	2.12	2.12	2.45	2.33	2.41	2.12	=	+
	un	3.67	2.46	2.48	2.40	2.18	2.10	2.03	1.91	+	+
	cts	2.08	0.46	0.40	0.40	0.46	0.44	0.34	0.32	+	+
CoNLL	tgt	2.49	2.95	1.80	1.75	2.13	1.77	1.89	1.76		+
PubMed	tgt	12.02	4.15	5.43	4.15	4.14	3.95	3.99	3.61	+	+
CNN	tgt	10.29	3.82	3.67	3.45	3.46	3.44	3.35	3.37	+	+
Tree bank- Chunk	wsj	6.63	4.35	4.33	4.30	4.32	4.32	4.27	4.11	+	+
	swbd3	15.90	4.15	4.50	4.10	4.13	4.09	3.60	3.51	+	+
	br-cf	5.16	6.27	4.85	4.80	4.78	4.72	5.22	5.15		
	br-cg	4.32	5.36	4.16	4.15	4.27	4.30	4.25	4.90		
	br-ck	5.05	6.32	5.05	4.98	5.01	5.05	5.27	5.41		
	br-cl	5.66	6.60	5.42	5.39	5.39	5.53	5.99	5.73		
	br-cm	3.57	6.59	3.14	3.11	3.15	3.31	4.08	4.89		
	br-cn	4.60	5.56	4.27	4.22	4.20	4.19	4.48	4.42		
	br-cp	4.82	5.62	4.63	4.57	4.55	4.55	4.87	4.78		
	br-cr	5.78	9.13	5.71	5.19	5.20	5.15	6.71	6.30		
Treebank-brown		6.35	5.75	4.80	4.75	4.81	4.72	4.72	4.65	+	+

- ちなみに、MEGAMモデル (Daumé III and Marcu (2006)) でもやってみたところ、ほぼすべてのドメインで表の最良の結果と同等の結果になったが、学習にかかる時間が約10倍で、交差検証も5~10回ほど必要だった。

4. Results

Model Introspection — 状態空間の拡張は効果があったのか？

- ACE-NER（固有表現抽出）で学習された重みに着目
- 全ての特徴量 f は “cross-domain” な特徴量と、各ドメイン固有の特徴量の7パターンある

「現在の単語は大文字で始まり、一つ以上の小文字が続く」という特徴量の重みベクトルの値に着目

- `/Aa+/` という特徴量は固有表現の良い指標
- broadcast newsには大文字はないので、重みは0
- usenetの重みが負になっているのは、e-mailアドレスやURLがこのパターンに合致しているせい

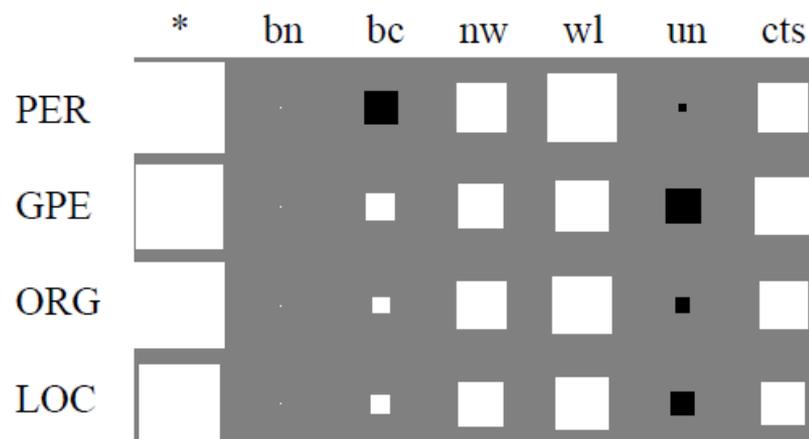


Figure 1: Hinton diagram for feature `/Aa+/` at current position.

白：positive weights
黒：negative weights

4. Results

Model Introspection

「現在の単語が 'bush'」という特徴量の重みベクトルの値に着目

- /bush/は基本的に「人」
- broadcast conversationのようなBushを大文字表記するドメインでは重みが0に近い
- conversational telephone speechでは/bush/は人ではなく植物
- usenetではbushは小文字表記することが多いので重みが正で大きい

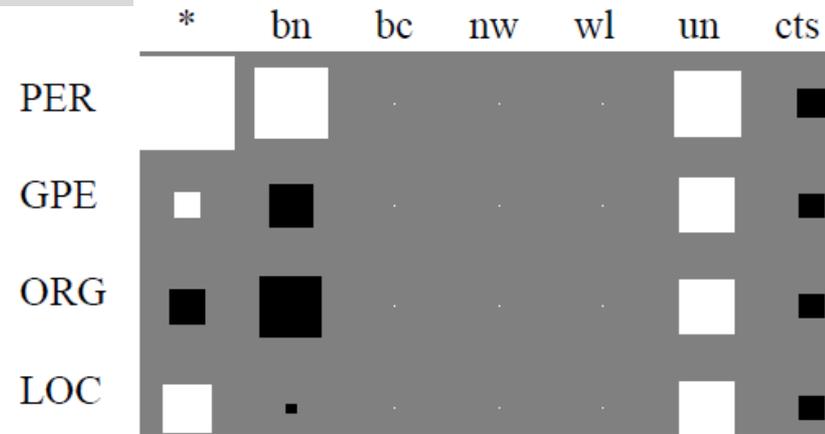


Figure 2: Hinton diagram for feature /bush/ at current position.

4. Results

Model Introspection

「現在の単語が 'the' 」という特徴量の重みベクトルの値に着目

- /the/という単語はbroadcast newsと conversational telephone speech以外で固有表現になっている
- bnとctsで重みが負になっているのは大文字化のせい

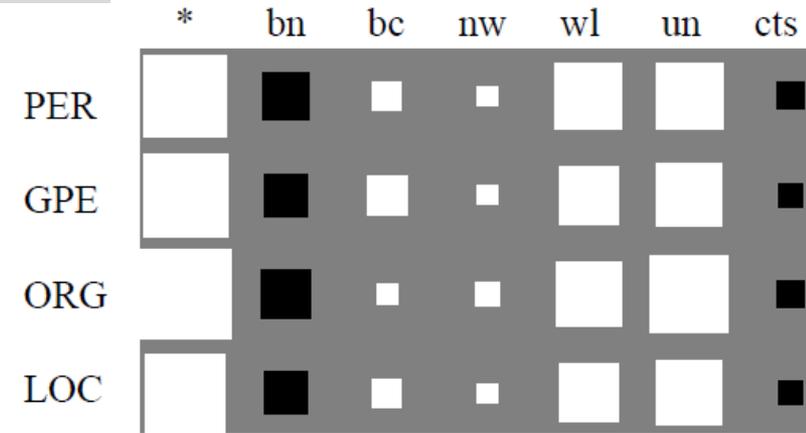


Figure 3: Hinton diagram for feature /the/ at current position.

4. Results

Model Introspection

「一つ前の単語が 'the'」という特徴量の重みベクトルの値に着目

- broadcast conversationでのみ、一つ前の単語が/the/というのが固有表現の良い指標となっている
- これはbc特有の,
“the Iraqi people”(IraqiはGPE)
“the Pentagon”(ORG)
“the Bush”(内閣など)(PER)
“the South”(LOC)
といった表現のせい

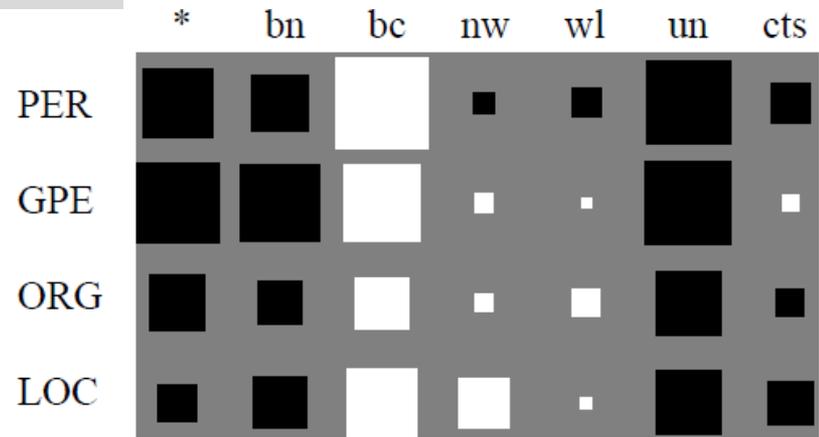


Figure 4: Hinton diagram for feature /the/ at previous position.

4. Results

Model Introspection

「現在の単語が俗称のリストの中にある」という特徴量の重みベクトルの値に着目

- これは「人名」の固有表現を抽出するには良い指標となる（他は良くない）
- bnでは全体的に良い指標であり，unでは全体的に良くない特徴量
- bnでは国や組織をそのリーダーの名前で言及することがよくある
- usenetでは俗称が多様すぎる

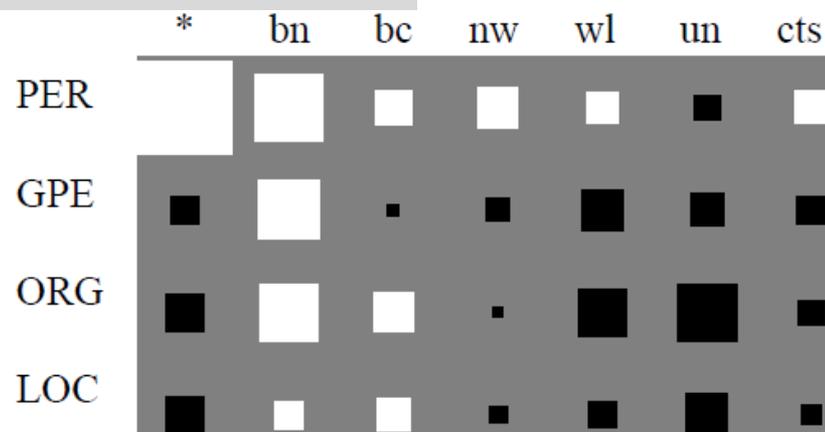


Figure 5: Hinton diagram for membership on a list of names at current position.

4. Results

Model Introspection

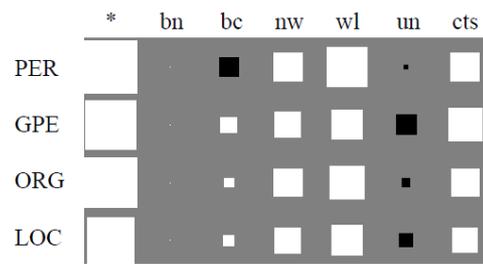


Figure 1: Hinton diagram for feature /Aa+/ at current position.

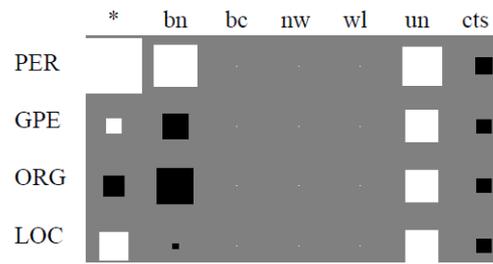


Figure 2: Hinton diagram for feature /bush/ at current position.

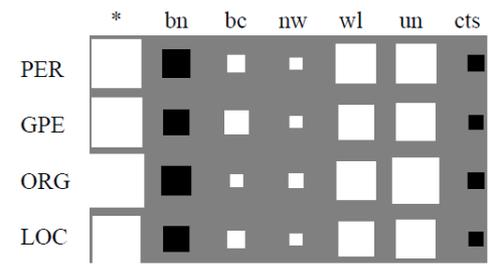


Figure 3: Hinton diagram for feature /the/ at current position.

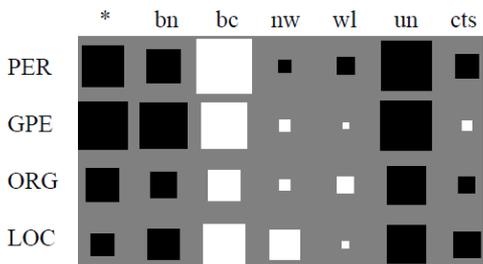


Figure 4: Hinton diagram for feature /the/ at previous position.

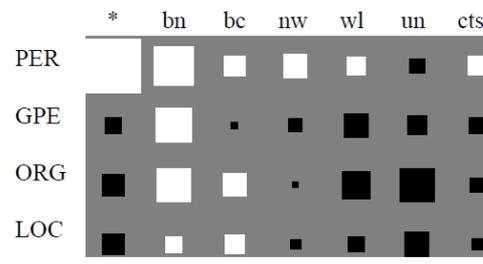


Figure 5: Hinton diagram for membership on a list of names at current position.

全ての特微量にわたって、同じように機能するドメインは無い
= ドメイン固有の特微量が存在する → 提案手法の妥当性

5. Discussion

- 状態空間の拡張はドメイン適応（転移学習）に有効な手法である

今後の展望

- 提案手法は学習を困難にすることなくより良い性能を発揮するが、学習をより簡単にするということを示したい
- カーネル化への拡張のところで、ドメインの類似性として“2”を用いることが導かれたが、実際にはドメインの類似性を示すハイパーパラメータ“a”を導入してチューニングするのがいい

感想

- 他分野の論文は慣れない表現が多くて苦戦した