修 士 学 位 論 文

ベイズ規則による確率密度比の推定を用いた語義曖昧性解消の領域適応

平成　２６　年度

茨 城 大 学 大 学 院 理 工 学 研 究 科

情 報 工 学 専 攻

菊池　裕紀

目 次

表 目 次

図 目 次

第1章　はじめに

1.1　研究背景

自然言語処理における語義曖昧性解消（Word Sense Disambiguation, WSD）とは，文書や会話の中に現れた多義語が，文脈中ではどの語義で使われているのかを判定するタスクのことである．多くの言語では，多義語と呼ばれる一つで複数の語義を有する単語が存在する．そのため文書中では文脈に応じて用いられる語義が異なる可能性があるためこのようなタスクが存在する．このタスクに対しては様々な研究が行われているが，確率的な言語モデルを用いて周辺単語の品詞や共起関係などを特徴としてコーパスや訓練データなどから分類器を学習させて語義を判定させる帰納学習のアプローチをとることが一般的である．

WSD をはじめとした自然言語処理のタスクでは，教師付き学習が利用されるが，そこでは訓練データとテストデータが属する領域が異なる領域適応の問題が生じている [14][17]．WSD をタスクとして考えてみると，例えば,「ゴルフ」という単語には少なくとも sport と car の 2 つの意味が存在する.「ゴルフ」の語義の判定を考えた場合，学習元となるコーパスがスポーツに関連するコーパスであれば主に sport と判定される規則が学習されやすいが，その規則を車関係の文書に適用した場合，語義を誤って判定してしまう確率が高くなってしまうという問題が領域適応の問題である．自然言語処理では，WSD 以外の様々なタスクでも領域適応の問題が生じるため，近年，活発に研究が行われている．このような問題に対処する手法は大きく分けて 2 つ存在し，1 つは事例ベースの手法，もう 1 つは素性ベースの手法である [15]．本論文では，事例ベースの手法を利用する．

事例ベースの手法を利用する場合には，領域適応の問題に共変量シフトを仮定することが多い．対象単語 w の用例を $\boldsymbol{x}$，w の語義の集合を C とする．用例 $\boldsymbol{x}$ 内の w の語義が $c \in C$ である確率を $P(c|\boldsymbol{x})$ とすると，WSD は $\arg\max_{c \in C} P(c|\boldsymbol{x})$ を求めることで解決できる．領域適応では，ソース領域 S から得られた訓練データを用いることで $P(c|\boldsymbol{x})$ を推定するので，求まるのはソース領域 S 上の条件付き分布 $P_S(c|\boldsymbol{x})$ となる．しかし，実際に求めたいのはターゲット領域 T 上の $P_T(c|\boldsymbol{x})$ である．このため，領域適応の問題は $P_S(c|\boldsymbol{x}) \neq P_T(c|\boldsymbol{x})$ から発生するように思われるが，実際はそうではない．用例 $\boldsymbol{x}$ がどのような領域に現れようと，その用例 $\boldsymbol{x}$ 内の対象単語 w の語義が変化するとは考えにくい．そこで，共変量シフトでは以下を仮定する．

$$P_S(c|\boldsymbol{x}) = P_T(c|\boldsymbol{x}) \tag{1.1}$$

$$P_S(\boldsymbol{x}) \neq P_T(\boldsymbol{x}) \tag{1.2}$$

式 1.1 は，同じ用例 $\boldsymbol{x}$ が異なる領域で現れた場合でもその用例が示す意味は同じであることを示している．これは自然言語処理では通常成立している．問題は式 1.2 にあり，ソース領域 S とターゲット領域 T 上の事前確率の違いが原因となり領域適応の問題が生じているとされる．この場合，用例 $\boldsymbol{x}$ の確率密度比 $w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ を重みとして，重み付き対数尤度を最大化するパラメータを求めることで，$P_T(c|\boldsymbol{x})$ を構築するアプローチがとられる [18].

そこで確率密度比の適切な算出が必要となってくる．確率密度比の算出手法は大きく 2 つに分けられる．1 つは $P_S(\boldsymbol{x})$ と $P_T(\boldsymbol{x})$ をそれぞれ推定してその比 $w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ をとる手法，もう 1 つは $w(\boldsymbol{x})$ を直接モデル化する手法 [32] である．前者の研究としては，Jiang の研究 [8]，齋木の研究 [33] と新納の研究 [28] があり，後者としては金森が提案した uLSIF[10] がある．しかし，WSD の領域適応をタスクとして扱う場合，事例数よりも素性数のほうが圧倒的に数が多くなるため，実際の確率密度比よりも推定された値は小さい値をとる傾向がある．この問題に対処するために推定された確率密度比を p 乗 $(0 < p < 1)$ する手法 [31] や相対確率密度比をとる手法 [19]，またソース領域 S とターゲット領域 T のコーパスを合わせたものを新たなソース領域 S のコーパスとみなす確率密度比を求める手法 [29] など，確率密度比を上方に修正する手法がある．

1.2 研究目的

本研究では，WSD の領域適応の問題に共変量シフトを仮定し，その問題を確率密度比を重みとした重み付け学習によって解決するアプローチをとる．重み付け学習には，ロジスティック回帰や最大エントロピー法などを用いるのが一般的だが，損失関数ベースの手法であれば重み付け学習は可能であることが分かっている．このため本実験では，使用する識別器として SVM を採用する．また，重みとして用いられる確率密度比の推定手法が重要となってくるが，$P_S(\boldsymbol{x})$ と $P_T(\boldsymbol{x})$ を推定して比を取る手法である NB 法 [28][29] と $w(\boldsymbol{x})$ を直接モデル化する uLSIF[10] を取り上げて効果を確認することを目的とする．前者の手法に関しては，まだ重み付け SVM での効果は確認されていない．そのため，この NB 法によって算出された確率密度比の重み付き SVM での効果の確認と直接モデル化する手法である uLSIF との有意差の確認が主な話題となる．

1.3 論文構成

第 2 章では，WSD のタスクの説明と手法についての説明を記述する．第 3 章では，分類問題に関する説明と本研究で使用するサポートベクターマシンの説明を記述する．第 4 章では，領域適応に関する説明とその根本の概念となる転移学習に関する言及を記述する．第 5 章では，領域適応の原因と仮定されている共変量シフトに関して記述する．第 6 章では，確率密度比の算出手法である NB 法と uLSIF に関して，その理論について記述する．第 7 章では，本実験に関する説明と結果を記述する．第 8 章では，本実験の考察を記述する．第 9 章では，結論と今後の展望について記述する．

第2章　語義曖昧性解消

2.1　概要

自然言語処理における語義曖昧性解消（Word Sense Disambiguation, WSD）とは，ある文書や会話の中に現れた単語やフレーズなどを対象とし，その文中ではどの意味で用いられているのかを判定するタスクのことである．1つの単語を取り上げたとき，その単語が複数の意味を有する場合がある．このような単語は多義語と呼ばれ，実際に文中で使用される場合は複数の語義の中から1つに限定されて用いられる．例えば，英単語の *fire* には少なくとも以下の4つの語義が存在する[1]．

1. uncontrolled flames, lights, and heat that destroy and damege things
2. to force someone to leave their job
3. to shoot bullets or bombs
4. to make someone feel interested in something and excited about it

それぞれの意味に対して次のような例文が考えられる[2]。

1. The *fire* was burning brightly.
2. He was *fired* from the job.
3. The enemy *fire* a rifle.
4. Her story *fired* my imagination.

上記の4つの例文にはそれぞれ *fire* という英単語が現れており，それぞれが文中で箇条書きの番号に対応した特定の意味で用いられている．我々は，会話の流れや文脈から総合的に判断し，単語の意味を推測する．WSD の問題は，形態素解析や機械翻訳など多くの自然言語処理の分野で課題となっている．

語義の判定には，主に確率的な言語モデルを用いて，周辺単語の品詞や共起関係を特徴としてコーパスや訓練データを使用して学習した分類器を利用して語義を判定する機械学習のアプローチがとられる．機械学習の流れを図 2.1 に示す．分類器の学習に使用するデータを訓練データ，そのデータが属する領域をソース領域という．学習した分類器を適用するデータをテストデータといい，そのデータが属する領域をターゲット領域という．従来は，ソース領域とターゲット領域は同一の領域であることがほとんどでった．しかし，属する領域が異なる場合も考えられる．この場合は領域適応という，ソース領域とターゲット領域が異なる場合のタスクとなる．領域適応についての説明は第 3 章で記述することとする．

[1] ロングマン現代英英辞典 [４訂新版] より
[2] ジーニアス英和辞典 [第４版] より

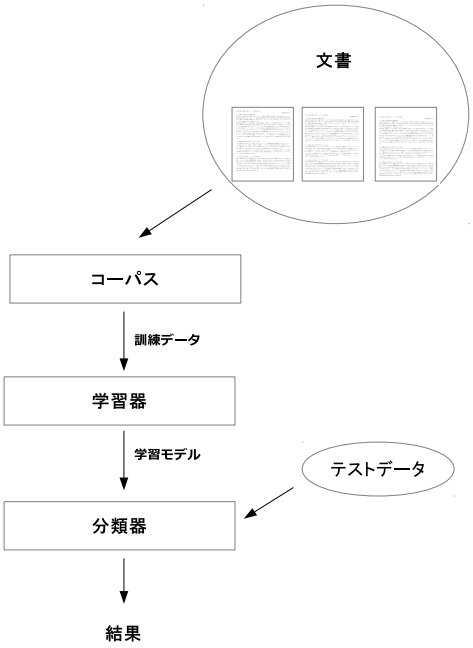

図 2.1: 機械学習の流れ

分類器の学習には文書から大量に集めたコーパスから得られるデータを用いる．このコーパスには，単に文書を集めたコーパスの他に文書中の単語に語義の情報や構文構造などの付加情報を含む自然言語処理に特化したコーパスが存在する．注釈つきコーパスを使用することで高い語義識別の精度を得ることができるが，大量の文書に付加情報を人手で付けていく作業は手間がかかり相当な時間と費用を必要とする。コスト面での負荷が膨大にかかってくる．学習データとなるものには，文書中の単語に品詞を付けた POS タグ，対象単語の文脈に含まれる単語を集めたもの，その周辺単語の分類語彙表の番号などがある．

学習する際の手法は，教師付き学習（supervised learning），半教師付き学習（semi-supervised learning），教師無し学習（unsupervised learning）が存在し，学習に利用するデータの性質によって準ずる手法が変化する．教師あり学習は，教示データを含むデータを用いて学習を行う場合の手法である．教示データが付与されたデータはラベル付きデータ，教示データが付与されていないデータをラベルなしデータと呼ばれている．ラベル付きデータの作成は人手で作成されるため，莫大な人的資源や時間，資金が必要となってくる．学習コストが高い分，精度は良い．半教師付き学習は，ラベル付きデータに加えてラベルなしデータも用いる場合の手法である．ラベルあり・ラベルなしが混在しているデータを用いることで教師付き学習よりも精度を向上させることが目標となる．教師なし学習は，ラベル付きデータを一切用いない場合の学習手法である．ラベル付きデータを用いる上記の 2 つの手法より識別精度には若干の不安が残る．しかし，教示データが必要ない分，学習コストはあまりかからない．教師なし学習の精度を向上させることができれば，莫大な資源の節約となり理想的な手法と言える．本研究では，教師なし学習を採用している．

2.2 関連研究

WSD に関する研究は古くから行われている．先行研究の手法としては辞書の定義文を用いる手法，用例をベースとした手法，語義のタグ付きコーパスを用いた機械学習による手法などが存在する．この 3 つのうちでは，機械学習による手法が有能であるとされている．しかし，機械学習に基づいた手法ではソース領域に出現する単語に対してしか語義の曖昧性を解消できない．そこで玉垣ら [23] は，さまざまな単語に対して WSD を行うためには機械学習による分類器とそれ以外の分類器を組み合わせて用いることが有効であるとし，辞書の用例を用いた分類器，語義の出現条件を用いた分類器，語義のラベル付きコーパスを用いた分類器の 3 種類を組み合わせることによって読解支援システムのための WSD に関する研究を行っている．杉山ら [30] は，用例をクラスタリングした結果から得られた素性を用いて語義曖昧性解消の教師あり領域適応の実験を行っている．類似した用例の素性を用いて分類器を作成することで精度の改善に成功している．WSD のための辞書作成としては，胡ら [25] の研究が挙げられる．胡らは wikipedia の「曖昧さ回避」のリンクを辿ることで語義に関する情報を集め，新規語・造語に対応した辞書の作成を試みている．

第3章　分類問題

3.1　概要

自然言語処理の分野において，あらかじめ分かっているグループに観測事例を分けたいというタスクが存在する．前章で説明した語義曖昧性解消もそうだが，日常生活では「可燃ごみ」「不燃ごみ」のどちらかに分けるゴミの分別や電子メールを「会社」「友人」「スパム」などに分けたい場合などが考えられる．このようにあらかじめ決められたグループにデータを分けることを分類（classification）といい，分ける単位となるグループをクラス（class）と呼ぶ．

この分類問題を解決するためには，分類器（classifier）を作成し，観測されたデータがどのクラスに属するのかを判定するアプローチがとられる．分類器の作成には大きく分けて以下の2つの手法が存在する．

- 人間が分類規則を書く方法 (規則ベース手法)
- データから自動的に分類器を作成する方法

規則ベースの手法では，人が持っている知識・直観などを規則とする．先のごみの分別を例にとると,「素材が紙である物は燃える」という知識を利用することで「可燃ごみ」に分類する規則が習得できる．しかし，この手法は膨大なコストを必要とする．人的資源，時間，費用など多くのコストがかかりあまり現実的な手法とはいえない．データから自動的に分類器を作成する方法が採用されることが一般的である．与えられるデータの集合を以下のように仮定する．

$$D = (x^{(1)}, c^{(1)}), (x^{(2)}, c^{(2)}), \cdots, (x^{(|D|)}, c^{(|D|)}) \tag{3.1}$$

ここで $x^{(1)}, x^{(2)}, \cdots, x^{(|D|)}$ はそれぞれ事例を表している．$c^{(1)}, c^{(2)}, \cdots, c^{(|D|)}$ は事例が属するクラスを表している．このようなデータを用いて分類規則を抽出することを学習といい，このようにしてできた分類器にはいくつかの種類が存在する．以下では分類器としてナイーブベイズ分類器（Naive Bayes Classifier）とサポートベクターマシン（Support Vector Machine, SVM）を説明する．なお，本論文ではSVMを使用している．

3.2　ナイーブベイズ分類器

ナイーブベイズ分類器は，古くからある分類器で現在も使われており，適切な使い方をすれば高い性能を発揮することも多い．この分類器は確率に基づいた分類器であり，文章 x に対して事後確率 $P(c|x)$ が最大となるクラス $c \in$C を求めることを目的としている．この事後確率の式は，ベイズの公式を利用して以下のように表すことができる．

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)} \tag{3.2}$$

この右辺が最大となるクラス c に文章 x を分類することになる．ここで分母の $P(x)$ はクラス c には依存しないため，省略することができる．つまり，分子の最大化問題を解けばいい．

$$\begin{aligned} c_{\max} &= arg\max_{c} \frac{P(c)P(x|c)}{P(x)} \\ &= arg\max_{c} P(c)P(x|c) \end{aligned} \tag{3.3}$$

式 3.3 の右辺を求めることが出来れば良い．ここで $P(c)$ は，以下に示す式で値を求めることが可能である．

$$P(c) = \frac{\text{クラス } c \text{ に属する文書数}}{\text{総文書数}} \tag{3.4}$$

式 3.4 からも分かるように，$P(c)$ の値を計算することは非常に簡単である．

しかし，ここで問題となるのが $P(x|c)$ の計算である．この $P(x|c)$ は、文章 x がクラス c のもとで起こる条件付確率である．文章 x において単語の種類とその組み合わせを考えると，起こりうる文章 x は膨大な数にのぼる．全ての文章 x についてそれぞれがデータの中で何回起こるかを調べ，$P(x|c)$ を最尤推定で求めるのは非現実的である．

そこで問題を単純化してみる．ナイーブベイズ分類器では，この文章 x に単純化したモデルを仮定して $P(x|c)$ の値を求めている．モデルには以下の 2 種類が存在する．

- 多変数ベルヌーイモデル
- 多項モデル

それぞれについて解説する．

3.2.1 多変数ベルヌーイモデル（Multivariate Bernoulli model）

導入

語彙 V(単語の集合) を考える．多変数ベルヌーイモデルでは，語彙 V に含まれる各単語 w とクラス c について，ベルヌーイ分布に従う確率変数 $X_{w,c}$ を考える．

ベルヌーイ分布とは，ある事象が起こる確率を p、起こらない確率を $q = 1 - p$ として，それをモデル化したものである．つまり，2 つの事象しか現われないものをモデル化した分布と言える．各確率変数は，w が事例内で出現するときに ”1 ”となり，そうでないときに ”0 ”になるとする．各ベルヌーイ分布を規定するパラメータである「$X_{w,c}$ が 1 となる確立」は，$p_{w,c}$ で表すことにする．また，これらのベルヌーイ分布は互いに独立であると仮定する．つまり，c が与えられたときの独立なので，条件付独立である．ここで，わかりやすさのために $p_c = P(c)$ としておく．多変数ベルヌーイモデルは，p_c と $p_{w,c}$ の二つのパラメータで値が決まる．

クラス c が与えられているときに，各単語 w が生起するかどうかを表す確率は，

$$p_{w,c}^{\delta_{w,x}}(1 - p_{w,c})^{1-\delta_{w,x}}$$

である．ここで，$\delta_{w,x}$ は単語 w が文章 x に出現したときに ”1 ”となり、そうでないときに ”0 ”となる．

文章 x の生起確率を式で表すと，

$$P(x|c) = \prod_{w \in V} p_{w,c}^{\delta_{w,x}} (1 - p_{w,c})^{1-\delta_{w,x}} \tag{3.5}$$

となる．よって多変数ベルヌーイモデルにおけるナイーブベイズ分類器は，

$$arg\max_{c} P(c)P(x|c) = arg\max_{c} p_c \prod_{w \in V} (p_{w,c}^{\delta_{w,x}} (1 - p_{w,c})^{1-\delta_{w,x}}) \tag{3.6}$$

を最大化するような c を出力する問題に落ち着く．こうすることで計算が困難だった $P(x|c)$ を $p_{w,x}$ の値の積で表すことが可能となる．$P(x|c)$ の算出のときは，単語の組み合わせを考える必要があったが，$p_{w,c}$ は 1 つの単語が生起するかどうかに注目しているため，計算が簡単となる．次に，この $p_{w,c}$ をどのように計算するかを述べていく．

多変数ベルヌーイモデルにおけるパラメータの最尤推定

最尤推定を用いて与えられてデータからパラメータをどのように推定していくかを述べていく．ここで，与えられたデータ D は、式 3.1 に示された形で与えられているとする．

求めるべきパラメータは，p_c と $p_{w,c}$ である．これを最尤推定で求めるので，

$$\begin{aligned}
\log P(D) &= \sum_{(x,c) \in D} \log P(x,c) \\
&= \sum_{(x,c) \in D} \left(\log p_c \prod_{w \in V} (p_{w,c}^{\delta_{w,x}} (1 - p_{w,c})^{1-\delta_{w,x}}) \right) \\
&= \sum_{(x,c) \in D} \left(\log p_c + \sum_{w \in V} (\delta_{w,x} \log p_{w,c} + (1 - \delta_{w,x}) \log (1 - p_{w,c})) \right) \\
&= \sum_{c} N_c \log p_c + \sum_{c} \sum_{w \in V} N_{w,c} \log p_{w,c} + \sum_{c} \sum_{w \in V} (N_c - N_{w,c}) \log (1 - p_{w,c})
\end{aligned}$$

を最大化することになる．ここで，未定義の変数についての解説を加える．

$$N_{w,c} \quad : \quad \text{クラス } c \text{ であり、} w \text{ を含むような訓練文書数} \tag{3.7}$$

$$N_c \quad : \quad \text{クラス } c \text{ であるような訓練文書数} \tag{3.8}$$

各文書は複数の異なる単語を含みえるので，$N_c = \sum_w N_{w,c}$ は通常は成立しない．なお，p_c については制約があるため，この最大化問題は，次のような制約付き条件最適化問題で表すことが出来る．

$$\begin{aligned}
max. \quad & \log P(D) \\
s.t. \quad & \sum_{c} p(c) = 1
\end{aligned}$$

この問題は，ラグランジュの未定乗数法（束縛条件の下で最適化を行う方法）により解くことが可能である．未定乗数 λ を導入して，次のようにラグランジュ関数 $L(\theta, \lambda)$ を定義する．

$$L(\theta, \lambda) = \log P(D) + \lambda\Big(\sum_c p_c - 1\Big) \tag{3.9}$$

ただし，θ は求めたいパラメータの集合 $(p_{w,c_{w\in V, c\in C}}, p_{c_{c\in C}})$ である．各パラメータに関する偏微分を計算してみると，

$$\frac{\delta L(\theta,\lambda)}{\delta p_{w,c}} = \frac{N_{w,c}}{p_{w,c}} - \frac{N_c - N_{w,c}}{1 - p_{w,c}}, \quad \frac{\delta L(\theta,\lambda)}{\delta p_c} = \frac{N_c}{p_c} + \lambda$$

となる．これらをそれぞれ ”0 ”として、$\sum_c p_c = 1$ と合わせて考えると、

$$p_{w,c} = \frac{N_{w,c}}{N_c}, \quad p_c = \frac{N_c}{\sum_c N_c} \tag{3.10}$$

となる．つまり，

$$p_{w,c} = \frac{\text{クラス } c \text{ に属する訓練文書のうち } w \text{ を含む文書数}}{\text{クラス } c \text{ に属する訓練文書数}}, \tag{3.11}$$

$$p_c = \frac{\text{クラス } c \text{ に属する訓練文書数}}{\text{訓練文書数}} \tag{3.12}$$

と推定していることとなる．多変数ベルヌーイモデルでは，パラメータ p_c，$p_{w,c}$ の推定に文書数を用いていることが分かる．

3.2.2 多項モデル（multinominal model）

導入

次に多項モデルの解説をする．多変数ベルヌーイモデルでは，各単語が生起するかしないかをモデル化した．多項モデルでは，文書中の各位置についてどんな単語が起こるかをモデル化する．

文書 x 内の単語数を $|x|$ で表すとする．多項モデルでは，語彙 V の中から 1 つの単語を選ぶ操作を $|x|$ 回繰り返すことで文書を作成する．つまり，$P(x|c)$ を多項分布でモデル化することとなる．

クラスが c であるとき，単語 w が選ばれる確率を $q_{w,c}$ で表す．つまり，W を単語とする確率変数 C をクラスの値とする確率変数とすると，$q_{w,c} = P(W = w|, C = c)$ である．文書 D 内で，単語 w がそれぞれ $n_{w,x}$ 回起こる確率は，次のように表すことが出来る．

$$\frac{\Big(\sum_w n_{w,x}\Big)!}{\prod_{w\in V} n_{w,x}!} \prod_{w\in V} q_{w,c}^{n_{w,x}}$$

$|x| = \sum_w n_{w,x}$ である．ただし，厳密には試行する回数を決定しなければならない．ここで，文書の長さはクラスに依存しないという仮定を設けて考えると，

$$p(x|c) = P\Big(K = \sum_w n_{w,x}\Big) \frac{\Big(\sum_w n_{w,x}\Big)!}{\prod_{w \in V} n_{w,x}!} \prod_{w \in V} q_{w,c}^{n_{w,x}} \tag{3.13}$$

となる．K は文書の長さを表す確率変数であり，$P(K = \sum_w n_{w,x})$ は長さが $\sum_w n_{w,x}$ であるような文書が起こる確率である．よって，多項モデルにおけるナイーブベイズ分類器は，

$$P(c)P(x|c) = p_c P(\sum_w n_{w,x}) \frac{\Big(\sum_w n_{w,x}\Big)!}{\prod_{w \in V} n_{w,x}!} \prod_{w \in V} q_{w,c}^{n_{w,x}} \tag{3.14}$$

を最大化するような c を出力する．なお，$p_c = P(c)$ である．

$$\begin{aligned} P(c)P(x|c) &= arg\max_c p_c P(\sum_w n_{w,x}) \frac{\Big(\sum_w n_{w,x}\Big)!}{\prod_{w \in V} n_{w,x}!} \prod_{w \in V} q_{w,c}^{n_{w,x}} \\ &= arg\max_c p_c \prod_{w \in V} q_{w,c}^{n_{w,x}} \end{aligned} \tag{3.15}$$

であるので，最大となる c を見つけるためには $p_c \prod_{w \in V} q_{w,c}^{n_{w,x}}$ さえ分かればいい．

多項モデルにおけるパラメータの最尤推定

データ D が式 3.1 の形で与えられているとして，多項モデルにおけるパラメータの推定の仕方を述べていく．求めるべきパラメータは p_c と $q_{w,c}$ であるので．$|x| = \sum_w n_{w,x}$ と表すと最尤推定により，

$$\begin{aligned} \log P(D) &= \sum_{(x,c) \in D} \log P(x,c) \\ &= \sum_{(x,c) \in D} \log \Big(\frac{P(|x|)|x|!}{\prod_{w \in V} n_{w,x}} p_c \prod_{w \in V} q_{w,c}^{n_{w,x}} \Big) \\ &= \sum_{(x,c) \in D} \log \frac{P(|x|)|x|!}{\prod_{w \in V} n_{w,x}!} + \sum_{(d,c) \in D} N_c \log p_c + \sum_{(x,c) \in D} \sum_{w \in V} n_{w,x} \log q_{w,c} \\ &= \sum_{(x,c) \in D} \log \frac{P(|x|)|x|!}{\prod_{w \in V} n_{w,x}!} + \sum_c N_c \log p_c + \sum_{(x,c) \in D} \sum_{w \in V} n_{w,x} \log q_{w,c} \end{aligned}$$

を最大化することになる．なお，N_c は多変数ベルヌーイモデルを解説する際に定義した式 3.8 と同じである．

多項モデルにおいては，$\sum_{c \in C} p_c = 1$ となる制約に加えて，任意の c について $\sum_{w \in V} q_{w,c} = 1$ となる制約がある．したがって，この最大化問題は次のような制約付き最適化問題で表すことが出来る．

$$\begin{aligned} max. \quad & \log p(D) \\ st. \quad & \sum_{c \in C} p_c = 1 \\ & \sum_{w \in V} q_{w,c} = 1; \forall c \in C \end{aligned}$$

これも多変数ベルヌーイモデル同様，ラグランジュ未定乗数法により解くことが可能である．未定乗数を $\beta c_{c\in C}$、γ を導入して（簡単のために $\beta c_{c\in C}$ を β で表す），次のようにラグランジュ関数 $L(\theta, \beta, \gamma)$ を定義する．

$$L(\theta, \beta, \gamma) = \log P(D) + \sum_{c\in C} \beta_c \Big(\sum_{c\in C} q_{w,c} - 1 \Big) + \gamma \Big(\sum_{c\in C} p_c - 1 \Big)$$

ここで，θ は求めたいパラメータの集合 $q_{w,c_{w\in V,c\in C}}$, $p_{c_{c\in C}}$ である．p_c やそれに対応する γ は多変数ベルヌーイモデルの場合と同じである．等式制約付きの凸計画問題（最大化問題）に対するラグランジュ未定乗数法では，$q_{w,c}$ に関する偏微分が 0 になれば良い．

これを偏微分すると，

$$\frac{\partial L(\theta, \beta, \gamma)}{\partial q_{w,c}} = \frac{n_{w,c}}{q_{w,c}} + \beta_c$$

となり，これが 0 となればよいので，$\sum_{w\in V} = 1$ と合わせて考えると，

$$q_{w,c} = \frac{n_{w,c}}{\sum_w n_{w,c}}$$

が得られる．なお，p_c に関しては多変数ベルヌーイモデルで述べた式 3.12 と同じである．つまり，$q_{w,c}$ は以下のように表すことが出来る．

$$q_{w,c} = \frac{(\text{クラス } c \text{ に属する訓練文書全体での } w \text{ の出現回数})}{(\text{クラス } c \text{ に属する訓練文書全体での全単語の出現回数})}$$

以上のように推定していることとなる．

3.2.3 2 つのモデルの比較

簡単にまとめると，多変数ベルヌーイモデルでの p_c と $p_{w,c}$，多項モデルでの p_c と $q_{w,c}$ を以下のように定義していることとなる．

$$p_c = \frac{\text{クラス } c \text{ に属する訓練文書数}}{\text{訓練文書数}}, \tag{3.16}$$

$$p_{w,c} = \frac{\text{クラス } c \text{ に属する訓練文書のうち } w \text{ を含む文書数}}{\text{クラス } c \text{ に属する訓練文書数}} \tag{3.17}$$

多変数ベルヌーイモデルにおけるパラメータ推定

$$p_c = \frac{\text{クラス } c \text{ に属する訓練文書数}}{\text{訓練文書数}}, \tag{3.18}$$

$$q_{w,c} = \frac{(\text{クラス } c \text{ に属する訓練文書全体での } w \text{ の出現回数})}{(\text{クラス } c \text{ に属する訓練文書全体での全単語の出現回数})} \tag{3.19}$$

多項モデルにおけるパラメータ推定

2つのモデル間で p_c の定義は同じであるが，$p_{w,c}$ と $q_{w,c}$ の定義は異なる．ここが2つのモデルの違いである．多変数ベルヌーイモデルでは，単語 w が文書 x で生起したか否かが分類に影響を与えていたのに対して，多項モデルでは，単語 w が文書 x で生起した回数が分類に影響を与えている．また，多変数ベルヌーイモデルでは，生起しなかった単語があった場合は $1-p_{w,c}$ を考慮して計算した．一方，多項モデルでは，生起しなかった単語については無視している．多変数ベルヌーイモデルは生起しなかったことをモデルに取り入れているのに対して，多項モデルは生起した単語にだけ着目しているのがわかる．

3.2.4 ゼロ頻度問題の処理（スムージング）

ここで，ゼロ頻度問題について言及する．確率的言語モデルにおいて，ある単語 w の生起頻度によって出現確率を求めるとき，学習データに存在しない単語の出現確率は ”0 ”となってしまう．この問題をゼロ頻度問題といい，これを解消する方法をスムージング (smoothing) と呼ぶ．具体的には，最大事後確率推定という 0.00 に近い値となる確率が非常に小さいような事前分布を与えて確率分布を均すことを行う．多変数ベルヌーイモデルと多項モデルでは，ディリクレ分布を与えることでゼロ頻度問題を解消することが出来る．ここで，最大事後確率推定とディリクレ分布について解説する．

最大事後確率推定

最大事後確率推定（maximum a posteriori estimation）は，その頭文字をとって MAP 推定（MAP estimation）とも呼ばれている．あらかじめパラメータがどのような値をとりやすいかが分かっている場合，具体的には，パラメータ θ の確率分布 $P(\theta)$ が分かっている場合である．これをパラメータの事前確率分布（prior distribution）と呼ぶ．一方，データ D が与えられたときのパラメータ θ の確率分布 $P(\theta|D)$ を事後確率分布 (posterior distribution) と呼ぶ．MAP 推定では，事後確率 $P(\theta|D)$ が最大になるようにパラメータを決定する．

事後確率の最大化は次のような式に変形することができる．

$$\begin{aligned} arg\max_{\theta} P(\theta|D) &= arg\max_{\theta} \frac{p(\theta \cdot P(D|\theta)}{P(D)} && (3.20) \\ &= arg\max_{\theta} P(\theta) \cdot P(D|\theta) && (3.21) \end{aligned}$$

なお，式 3.20 の分母 $P(D)$ は最大化に関係ないので式 3.21 にて省略される．ここで，計算の簡単化のために対数をとる．

$$\log P(\theta) \cdot P(D|\theta) = \log P(\theta) + \sum_{x^{(i)} \in D} \log P(x^{(i)}|\theta) \quad (3.22)$$

これを最大化する θ を選ぶこととなる．

ディリクレ分布

ディリクレ分布は，$x_i \leq 0$ 、$\sum_i x_i = 1$ であるような $x = (x_1, x_2, \cdots, x_n)$ に対して確率を与える分布である．確率密度関数は，

$$p(\boldsymbol{x};\alpha) = \frac{1}{\int \prod_i x_i^{\alpha_i - 1} dx} \prod x_i^{\alpha_i - 1}$$

である．$\alpha_1, \cdots, \alpha_n$ という n 個のパラメータを持っている．大事な点は，確率密度が $\prod_i x_i^{\alpha_i - 1}$ に比例している点である．分母は積分すると ”1 ”になるように導入されている．

ディリクレ分布の性質を理解するため，$x = (x_1, x_2)$ として二次元ディリクレ分布を考える．パラメータ α については，$\alpha_1 = 2, \alpha_2 = 2$ とする．$\sum_i x_i = 1$ という制限から，$x_2 = 1 - x_1$ である．このとき，

$$p(\boldsymbol{x};\alpha) = \frac{1}{\int_0^1 x_1(1 - x_1) dx_1} x_1(1 - x_1)$$

である．また分母は，

$$\int_0^1 x_1(1 - x_1) dx_1 = \Big[\frac{1}{2}x_1^2 - \frac{1}{3}x_1^3\Big]_0^1 = \frac{1}{6}$$

であるので，

$$p(\boldsymbol{x};\alpha) = 6x_1(1 - x_1)$$

となる．x 軸を横にとってこのグラフを図示したものを図 3.1 に示す.

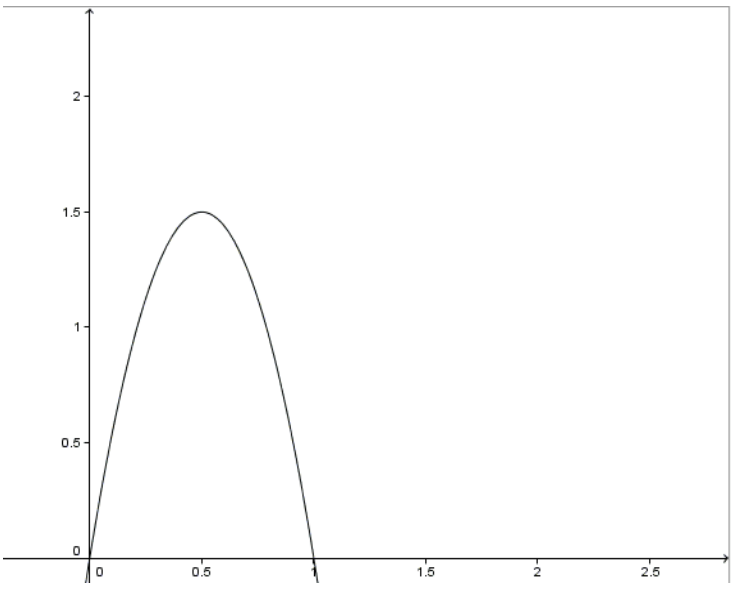

図 3.1: 二次元ディリクレ分布のグラフ

図 3.1 のグラフからも分かるように，中央が凸のグラフである．つまり，x_1 と x_2 の値が近いような x は比較的高い確率を保有しているが，$x_1 = 0.99$、$x_2 = 0.01$ のように偏った x は非

常に低い確率を持つ．いずれかの x_i が 0 もしくは 1 に近づくような x は確率が非常に小さくなる．このように，ディリクレ分布に従う確率変数は極端な値をとりにくいことで知られている．

この分布をナイーブベイズ分類器における 2 つのモデルに与えることで確率分布をならし，極端な値に対して補正を行う．

多変数ベルヌーイモデルにおける MAP 推定

多変数ベルヌーイモデルでは，式からもわかるように $p_{w,c}$ の値を積の形で表している．このため，ソース領域で出現しない単語がターゲットドメインに出現した場合，値が 0 となってしまう．この値が 0 とならないようにパラメータを推定することを考えていく．ここでディリクレ分布を与えることとなる．

多変ベルヌーイモデルにおける MAP 推定の目的関数となるのは，

$$\begin{aligned}
& \log P(\theta) + \log P(D) \\
= \; & \log\Big(\prod_c p_c^{\alpha-1}\Big) \times \Big(\prod_{w,c} (p_{w,c}^{\alpha-1}(1-p_{w,c})^{\alpha-1})\Big) + \sum_{(x,c)\in D} \log P(x,c) + (\text{定数}) \\
= \; & (\alpha-1)\sum_c \log p_c + (\alpha-1)\sum_{w,c} (\log p_{w,c} + \log(1-p_{w,c})) \\
& + \sum_{(x,c)\in D} \log\Big(p_c \prod_{w\in V} (p_{w,c}^{\delta_{w,x}}(1-p_{w,c})^{1-\delta_{w,x}})\Big) + (\text{定数})
\end{aligned}$$

であり，これを $\sum_c p(c) = 1$ となる制約のもとで最大化する．

最尤推定の場合とほとんど同じように計算が出来る．ラグランジュ関数は，

$$L(\theta,\lambda) = \log P(\theta) + \log P(D) + \lambda\Big(\sum_c p_c - 1\Big)$$

となる．偏微分を計算してみると，

$$\begin{aligned}
\frac{\partial L(\theta,\lambda)}{\partial p_c} &= \frac{(\alpha-1)}{p_{w,c}} - \frac{(\alpha-1)}{1-p_{w,c}} + \frac{N_{w,c}}{p_{w,c}} - \frac{N_c - N_{w,c}}{1-p_{w,c}}, \\
\frac{\partial L(\theta,\lambda)}{\partial p_c} &= \frac{(\alpha-1)}{p_c} + \frac{N_c}{p_c} + \lambda
\end{aligned}$$

となる．以上をそれぞれ 0 とおき，$\sum_s p_s = 1$ と合わせると，

$$p_{w,c} = \frac{N_{w,c} + (\alpha-1)}{N_c + 2(\alpha-1)}, \quad p_c = \frac{N_c + (\alpha-1)}{\sum_c N_c + \mid C \mid (\alpha-1)}$$

が得られる．ここで $|C|$ はクラスの数を表す．

ここで例として $\alpha = 2$ のときを考えてみると，

$$p_{w,c} = \frac{N_{w,c} + 1}{N_c + 2}, \quad p_c = \frac{N_c + 1}{\sum_c N_c + |C|}$$

となる．これはすべての w と c に対して生起回数に 1 を足して，パラメータを計算していることとなる．生起回数に 1 を足す手法は一般的に用いられる手法であり，ラプラススムージング (Laplace Smoothing) と呼ばれている．

多項モデルにおける MAP 推定

多項モデルにおいてもディリクレ分布で確率分布を均すことを考える．MAP 推定の目的関数となるのは，

$$\begin{aligned} & \log P(\theta) + \log P(D) \\ = & \Big(\prod_c p_c^{\alpha-1}\Big) \times \Big(\prod_{w,c} q_{w,c}^{\alpha-1}\Big) + \sum_{(x,c)\in D} \log P(x,c) + (\text{定数}) \\ = & (\alpha-1)\Big(\sum_c \log p_c + \sum_{w,c} \log q_{w,c}\Big) + \sum_{(x,c)\in D)} \log\Big(\frac{P(|x|)|x|!}{\prod_{w\in V} n_{w,x}!} p_c \prod_{w\in V} q_{w,c}^{n_{w,c}}\Big) + (\text{定数}) \end{aligned}$$

であり，これを $\sum_c p(c) = 1$、$\sum_w q_{w,c} = 1$ の制約の元で最大化する．

最尤推定の場合と同様にラグランジュ乗数 β と γ を導入し，ラグランジュ関数を，

$$L(\theta,\beta,\gamma) = \log P(\theta) + \log P(D) + \sum_{c\in C} \beta_c\Big(\sum_{w\in V} q_{w,c} - 1\Big) + \gamma\Big(\sum_c p_c - 1\Big)$$

となる．p_c と γ に関しては，多変数ベルヌーイモデルと同様である．$q_{w,c}$ について偏微分を計算してみると，

$$\frac{\partial L)(\theta,\beta,\gamma)}{\partial q_{w,c}} = \frac{(\alpha-1)}{q_{w,c}} + \frac{n_{w,c}}{q_{w,c}} + \beta_c$$

となる．これを 0 とおき $\sum_{w\in V} q_{w,c} = 1$ と合わせて考えると，

$$q_{w,c} = \frac{n_{w,c} + (\alpha-1)}{\sum_w n_{w,c} + |W|(\alpha-1)}$$

が得られる．ここで $|W|$ は単語の種類を表している．

ここで例として $alpha = 2$ を考えてみると，

$$q_{w,c} = \frac{n_{w,c}+1}{\sum_w n_{w,c} + |W|}, \quad p_c = \frac{N_c+1}{\sum_c N_c + |C|}$$

となる．多変数ベルヌーイモデルと同じようにすべての生起回数に 1 を加えて計算を行っていることとなる．

3.2.5 アンダーフロー対策

プログラムでナイーブベイズ分類器を実装する場合，注意しなければならいな点が 1 つある．多変数ベルヌーイモデルでの $p_{w,c}$ と多項モデルでの $q_{w,c}$ は非常に小さな値であり，なおかつ文書中にはたくさんの単語が出現するため乗算がアンダーフローを起こしてしまう可能性が出てくる．この問題を解決するために対数をとることで対処する．多変数ベルヌーイモデルの式 3.6 を変形すると，

$$arg\max_c p_c \prod_{w\in V} (p_{w,c}^{\delta_{w,x}}(1-p_{w,c})^{1-\delta_{w,x}}) = arg\max_c \Big(\log p_c \sum_{w\in V} \log p_{w,c}^{\delta_{w,x}}(1-p_{w,c})^{1-\delta_{w,x}}\Big)$$

となる．多項モデルの式 3.15 を変形すると，

$$arg\max_c \log\Big(p_c \prod_{w\in V} q_{w,c}^{n_{w,x}}\Big) = arg\max_c \Big(\log p_c \sum_{w\in V} \log q_{w,c}^{n_{w,x}}\Big)$$

となる．以上でアンダーフローを防ぐことが出来る．

3.3 サポートベクターマシン

3.3.1 概要

サポートベクターマシン [26] は線形二値分類器であり，クラス数が 2 つであるような分類問題に用いられる．2 つのクラスはそれぞれ正クラスと負クラスと呼ばれ，正クラスに属している事例を正例，負クラスに属している事例を負例という．訓練データが，

$$D = (\boldsymbol{x}^{(1)}, c^{(1)}), (\boldsymbol{x}^{(2)}, c^{(2)}), \cdots, (\boldsymbol{x}^{(|D|)}, c^{(|D|)}) \tag{3.23}$$

で与えられているとする．$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(|D|)}$ は事例の素性ベクトルを表しており，$c^{(1)}, c^{(1)}, \cdots, c^{(|D|)}$ は事例のラベルを表している．ここで正例のラベルは $+1$，負例のラベルは -1 である．線形分類器なので，分離平面の方向ベクトル $\boldsymbol{w}$ と切片 b をパラメータとして SVM は以下で定式化することができる．

$$f(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{x} - b \tag{3.24}$$

この関数を用いて事例 $\boldsymbol{x}$ を $f(\boldsymbol{x}) \geq 0$ ならば正クラス，$f(\boldsymbol{x}) < 0$ ならば負クラスに分類する．

SVM では，$\boldsymbol{w}, b$ の 2 つのパラメータを求める．この 2 つのパラメータはマージン最大化とよばれる手法で求めることができる．ここで以下のような 2 次元上の分離平面を考える．方向ベクトルを $\boldsymbol{w}$ ，切片を b と表すと分離平面は $\boldsymbol{w}\cdot\boldsymbol{x} = b$ を満たす点 $\boldsymbol{x}$ の集合となる．この分離平面に一番近い正例を $\boldsymbol{x}_+$，$\boldsymbol{x}_+$ から分離平面に向って引いた垂線と分離平面の交点を $\boldsymbol{x}_*$ とする．この $\boldsymbol{x}_+$ と $\boldsymbol{x}_*$ の距離をマージンといい，このマージンを最大にすることをマージン最大化という．正例と負例のどちらからもなるべく遠い位置で分離平面を構築することが望まれる．ここでマージンは以下で表すことができる．

$$|\boldsymbol{x}_+ - \boldsymbol{x}_*| \tag{3.25}$$

ここで $\boldsymbol{w}$ と $\boldsymbol{x}_+ - \boldsymbol{x}_*$ は同じ方向ベクトルなので，以下が成り立つ．

$$\boldsymbol{w}\cdot(\boldsymbol{x}_+ - \boldsymbol{x}_*) = |\boldsymbol{w}||\boldsymbol{x}_+ - \boldsymbol{x}_*| \tag{3.26}$$

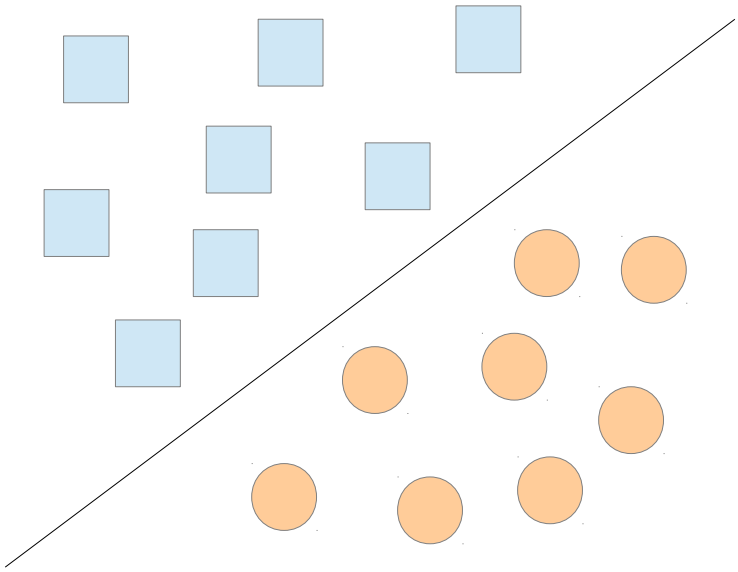

図 3.2: 訓練データの分布と分離平面の例

分離平面 $\boldsymbol{w}\cdot\boldsymbol{x}=b$ は，適切に定数倍することができれば $\boldsymbol{w}\cdot\boldsymbol{x}_+-b=1$ とできる．また $\boldsymbol{x}_*$ は分離平面上の点であるので，$\boldsymbol{w}\cdot\boldsymbol{x}_*=b$ は明確である．すなわち，

$$\begin{aligned}\boldsymbol{w}\cdot(\boldsymbol{x}_+-\boldsymbol{x}_*) &= \boldsymbol{w}\cdot\boldsymbol{x}_+-\boldsymbol{w}\cdot\boldsymbol{x}_* \\ &= (b+1)-b \\ &= 1 \end{aligned} \tag{3.27}$$

となる．つまり，式 (3.26) と式 (3.27) より，

$$\begin{aligned}|\boldsymbol{w}||\boldsymbol{x}_+-\boldsymbol{x}_*| &= 1 \\ |\boldsymbol{x}_+-\boldsymbol{x}_*| &= \frac{1}{|\boldsymbol{w}|}\end{aligned} \tag{3.28}$$

を導き出すことができる．$|\boldsymbol{x}_+-\boldsymbol{x}_*|$ はマージンを表している．つまり，分離平面のマージンは $1/|\boldsymbol{w}|$ で表せるということになる．マージン最大化の観点から，これを最大化すればいい．このままでは計算がしにくいので，計算簡単化のために $\boldsymbol{w}^2$ を最小化する問題に置き換える．

3.3.2 厳密制約下の SVM モデル

訓練事例を正しく分類することを考えていく．$y^{(i)}=+1$ であるような訓練事例は $\boldsymbol{w}\cdot\boldsymbol{x}^{(i)}-b\leq 1$ であれば良い．$y^{(i)}=-1$ の訓令事例は $\boldsymbol{w}\cdot\boldsymbol{x}^{(i)}-b\geq 1$ であれば良い．この 2 つの条件をまとめると，

$$y^{(i)}(\boldsymbol{w}\cdot\boldsymbol{x}^{(i)}-b)\leq 1$$

よって，これを制約とした最適化問題を解くこととなる．

$$
\begin{aligned}
min. \quad & \frac{1}{2}\boldsymbol{w}^2 \\
s.t. \quad & y^{(i)}(\boldsymbol{w}\cdot\boldsymbol{x}^{(i)} - b) - 1 \leq 0;\ \forall i.
\end{aligned}
$$

ここで，目的関数につけた係数 1/2 は計算を分かりやすくするためである．これは省略してもかまわない．

この不等式制約付き最適化問題は凸計画化問題であるので，ラグランジュ法を用いて解くこととする．ラグランジュ未定乗数 $\alpha_i(\leq 0)$ を導入するとラグランジュ関数は，

$$
L(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\boldsymbol{w}^2 - \sum_i \alpha_i \Big(y^{(i)}(\boldsymbol{w}\cdot\boldsymbol{x}^{(i)} - b) - 1 \Big)
$$

と表すことができる．これをそれぞれのパラメータで偏微分することで，

$$
\begin{aligned}
\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, b, \alpha) &= \boldsymbol{w} - \sum_i \alpha_i y^{(i)} \boldsymbol{x}^{(i)}, && (3.29) \\
\frac{\partial L(\boldsymbol{w}, b, \alpha)}{\partial b} &= \sum_i \alpha_i y^{(i)} && (3.30)
\end{aligned}
$$

となる．これらを 0 とおいて，

$$
\begin{aligned}
\boldsymbol{w}^* &= \boldsymbol{w} - \sum_i \alpha_i y^{(i)} \boldsymbol{x}^{(i)}, && (3.31) \\
\sum_i \alpha_i y^{(i)} &= 0 && (3.32)
\end{aligned}
$$

を得る．1 つ目の式 3.31 は，分離平面の方向ベクトル $\boldsymbol{w}^*$ は訓練事例ベクトルの線形和で表されることを意味している．この式 $\boldsymbol{w}^* = \sum_i \alpha_i y^{(i)} \boldsymbol{x}^{(i)}$ を分離平面の式 $\boldsymbol{w}\cdot\boldsymbol{x} = b$ に代入すると，

$$
f(\boldsymbol{x}) = \sum_i \alpha_i y^{(i)} \boldsymbol{x}^{(i)} \cdot \boldsymbol{x} - b \tag{3.33}
$$

となる．これで α_i と b を求められれば，分離平面を得ることができる．

そこで，これらの式を元のラグランジュ関数に代入することを考える．まず，$\boldsymbol{w}^* = \sum_i \alpha_i y^{(i)} \boldsymbol{x}^{(i)}$ を用いると，

$$
\begin{aligned}
L(\boldsymbol{w}^*, b, \alpha) &= \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} \boldsymbol{x}^{(i)} \cdot \boldsymbol{x}^{(j)} \\
&\quad - \sum_i \alpha_i y^{(i)} \Big(\sum_j \alpha_j y^{(j)} \boldsymbol{x}^{(j)} \cdot \boldsymbol{x}^{(i)} - b \Big) + \sum_i \alpha_i && (3.34) \\
&= -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} \boldsymbol{x}^{(i)} \cdot \boldsymbol{x}^{(j)} + b \sum_i \alpha_i y^{(i)} + \sum_i \alpha_i && (3.35)
\end{aligned}
$$

と変形することができる．ここで 2 つ目の式 3.32 より，$\sum_i \alpha_i y^{(i)} = 0$ なので，

$$\boldsymbol{w}^*, b, \alpha) = -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\cdot\boldsymbol{x}^{(j)} + \sum_i \alpha_i \tag{3.36}$$

となる．これで元々の変数 $\boldsymbol{w}$ と b がラグランジュ関数から消去された．

ここでラグランジュ関数の鞍点（saddle point）について説明を加える．ラグランジュ関数 $L(\boldsymbol{x}, \lambda)$ において，不等式 $L(\boldsymbol{x}^*, \lambda^*) \geq L(\boldsymbol{x}^*, \lambda^*) \geq L(\boldsymbol{x}, \lambda)$ を満たす点 $L(\boldsymbol{x}^*, \lambda^*)$ のことを鞍点という．これは $\boldsymbol{x}$ に関しては最大となり，λ に関しては最小となる点のことである．

今回のラグランジュ関数に置き換えて考えてみると，$L(\boldsymbol{w}, b, \alpha)$ を最大化する α_i を求めれば良い．ただし，$\sum_i \alpha_i y^{(i)} = 0, \alpha_i \leq 0$ の制約の下での最大化である．

最適解 α_i^* が求まれば，$\boldsymbol{w}^*$ が求まり，$\boldsymbol{x}_+$ を用いて，$b = \boldsymbol{w}^* \cdot \boldsymbol{x}_+ - 1$ として切片も求まる．

3.3.3 緩和制約下の SVM モデル

上記で導出した SVM は，実際のデータに対してはなかなかうまくいかない．原因としては，全ての訓令事例を正確に分類しなければならないと言う制約 $y(i)(\boldsymbol{w}\cdot\boldsymbol{x}^{(i)} - b)$ があるからである．訓練データの中には例外的な事例が存在することがあり得る．こうした事例によって分類平面が大きく影響を受けてしまう．極端なケースでは，訓練データが線形関数でうまく分類できずに，制約を満たす解が存在しないことにもなってしまう．

そこで，制約を少し緩めて考えることにする．新たな変数 $\xi_i(\leq 0)$ を導入して，

$$y^{(i)}(\boldsymbol{w}\cdot\boldsymbol{x}^{(i)} - b) - 1 \leq \xi_i \tag{3.37}$$

という制約に書き換える．ξ_i は，i 番目の訓練事例がうまく分けられない度合いを表す．つまり，x_i が小さいほうが良い．そこで，これを目的関数に加えることにする．新しい最適化問題はつぎのようになる．

$$min. \quad \frac{1}{2}\boldsymbol{w}^2 + C\sum_i \xi_i \tag{3.38}$$

$$s.t. \quad y^{(i)}(\boldsymbol{w}\cdot\boldsymbol{x}^{(i)} - b) \leq 1 - \xi_i;\ \forall i \tag{3.39}$$

ここで C は正の定数であり，この値が大きいほどきちんと分類できるようになる．逆にこの値が小さいと例外的な訓練事例をほぼ無視した分類器ができる．

この最適化問題をラグランジュ法を用いて解いてみる．ラグランジュ未定乗数 $\alpha_i(\leq 0)$ を導入すると，ラグランジュ関数は，

$$\begin{aligned} L(\boldsymbol{w}, b, \xi, \alpha, \beta) &= \frac{1}{2}\boldsymbol{w}^2 + C\sum_i \xi_i \\ &\quad - \sum_i \alpha_i\Big(y^{(i)}(\boldsymbol{w}\boldsymbol{x}^{(i)} - b) - 1 + \xi_i\Big) - \sum_i \beta_i\xi_i \end{aligned}$$

と表すことができる．これをそれぞれのパラメータで偏微分する．$\boldsymbol{w}$ と b については厳密制約下の場合と同じで，$\boldsymbol{w} = \sum_i \alpha_i y^{(i)}\boldsymbol{x}^{(i)}$ と $\sum_i \alpha_i y^{(i)} = 0$ が得られる．ξ_i に関しては，

$$\frac{\partial L(\boldsymbol{w}, b, \xi, \alpha, \beta)}{\partial \xi_i} = C - \alpha_i - \beta_i \tag{3.40}$$

であるので，$C = \alpha_i + \beta_i$ を得る．双対ラグランジュ関数は，

$$\begin{aligned} L(\boldsymbol{w}^*, b, \xi, \alpha, \beta) &= -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)} + \sum_i \alpha_i \\ &\quad + C\sum_i \xi_i - \sum_i \alpha_i\xi_i - \sum_i \beta_i\xi_i \end{aligned}$$

となるが，最後の 3 つの項は $C = \alpha_i + \beta_i$ を使うことで消去できるため，結局は厳密制約下の場合と同じになる．これを $\alpha_i \leq 0, \beta_i \leq 0, \xi_i \leq 0$ なる条件の下で最大化する．ただし，まず ξ_i は双対ラグランジュ関数に登場しないため，ひとまず後にまわすこととする．また，β_i も双対ラグランジュ関数には登場しないので，$\beta_i = C - \alpha_i$ が満たされていれば良い．結局，$\alpha_i \leq 0$ と合わせて $0 \geq \alpha_i \geq C$ の条件の下で，

$$L(\boldsymbol{w}^*, b, \xi, \alpha, \beta) = -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y^{(i)}y^{(j)}\boldsymbol{x}^{(i)}\boldsymbol{x}^{(j)} + \sum_i \alpha_i$$

を最大化することとなる．

b は $0 < \alpha_i < C$ の事例を 1 つ持ってきて，$b = \boldsymbol{w} \cdot \boldsymbol{x}^{(i)} - y^{(i)}$ とすることで計算できる．

3.3.4 関数距離

SVM は事例 $\boldsymbol{x}$ を，$f(\boldsymbol{x}) \leq 0$ なら正クラスに，$f(\boldsymbol{x}) < 0$ なら負クラスに分類する．この $f(\boldsymbol{x})$ を関数距離（functional distance）と呼ぶ．

分類する際，$f(\boldsymbol{x}) = 0.00001$ である $\boldsymbol{x}$ も，$F(\boldsymbol{x}) = 1000$ である $\boldsymbol{x}$ も正クラスに分類される．これはこれで良いが，前者のように関数距離 $f(\boldsymbol{x})$ が 0 に非常に近いものは，その分類結果が誤りであることが多い．また後者のように関数距離 $f(\boldsymbol{x})$ が非常に大きいものは，実際に正クラスに属していることが多い．負クラスについても同じことが言える．

このように，関数距離は分類結果の信頼度となっている．つまり，関数距離が大きい事例だけを集めれば，高い確率で正例となる事例を集めることができる．また関数距離が 0 に非常に近い事例を集めれば，分類結果が誤りである事例を集めることができる．誤り発見の重要な技術であると同時に，エラー分析の際にも利用できる．

3.3.5 SVM による多値クラスの分類問題

前述したように，SVM は二値分類器である．そのため 3 クラス以上のグループをもつ分類問題を扱うためには多値分類問題へ対応するように拡張する必要が出てくる．拡張する手法には以下の 2 つの代表的な手法が祖存在する．

1. one-versus-rest 法
2. ペアワイズ法

1 つ目の one-versus-rest 法は，各クラスごとに 1 つの分離平面を作成する．つまり対応するそれぞれのクラスに属するか属さないかを判定する分離平面を作成することとなる．すなわちクラス数の数によって作成する分離平面の数は変化する．n クラスの分類問題では n 個の分離平面が作成される．各分離平面において，対応するクラスに属するのであればそれ正例となり，属さないものは負例となる．今，1 つの事例に対して 1 つのクラスのみが対応しているとする．この時，もし 2 つ以上の分離平面において正例と判定された場合は関数距離の値が最も大きいクラスに事例を分類する．また正例が存在せず負例だけの場合は，関数距離の絶対値が最も小さいクラスに事例を分類する．

2 つ目のペアワイズ法は，クラス対ごとにどちらのクラスであるかを判定する分離平面を構築する．クラス数が C_1, C_2, C_3 の 3 つの場合は，C_1 と C_2 ，C_2 と C_3 ，C_1 と C_3 の 3 つの分離平面が作成される．つまり，クラスの数が n のときには ${}_nC_2$ 通りの分離平面を構築する．この分離平面で事例を分類し，多数決を取ることでクラスを予測する．C_i と C_j の分離平面で C_i に分類された場合は C_i に一票が入る．最も多く選ばれたクラスに分類する．

第4章　転移学習（transfer learning）

4.1　概要

転移学習（transfer learning）[9] は多くの機械学習の枠組みに当てはまり，統一した形式的な定義を唱えることは難しいとされている．転移学習という単語の他にも帰納学習（inductive transfer）や領域適応（domain adaptation），マルチタスク学習（multitask learning）など呼び方は多岐に渡るのもそのためである．形式的ではない定義ではあるが,「ある 1 つ以上のタスクから学習された知識を別のタスクの学習性能を向上させるために利用すること」ということが広く受け入れられている．つまり，あるタスクを効果的かつ効率的に解くために別の関連するタスクデータや学習結果を利用するということが転移学習であると言える．以下，本論文では転移学習に関してはこの定義を前提とて話を進めていく．

昨今，Web サービスなどの発展に伴ってネット上から大量のテキストデータを入手することがかなり容易になってきている．そのためラベル付きデータを必要としない教師なし学習の言語モデルは精度の向上がみられている．ラベル付きデータが必要ないという事は，人が手動でデータを作成する必要がないということである．この精度をさらに向上することができれば，費用や人的資源，時間などデータ作成にかかるコストを抑えることができ経済的効果は計り知れない．一般にこの問題を解決するための機械学習は 3 種類あり，半教師付き学習，能動学習，そして転移学習とされている．

本論文では，転移学習の中の領域適応および共変量シフト (covariate shift) に焦点をあてて問題解決を試みる．

4.2　領域適応

4.2.1　概要

従来，自然言語処理のタスクにおける機械学習では，分類器を構築するために使用される訓練データとその規則を適用するテストデータは同じ領域のコーパスから得たデータであることが前提であった．しかし，ある領域のデータのラベルを識別したいにも関わらず，別の領域のデータからしか識別規則を学習できないことも多い．そこで，ある領域の訓練データから識別規則を学習した分類器を，別の異なる領域のテストデータに適用できるようにチューニングすることが考えられる．このように，ある領域に属するデータから識別規則学習させて作った分類器を異なる領域に属するデータに適応する手法を領域適応（domain adaptation）という．識別規則を学習させるために利用するデータを訓練データ（training data），それが属する領域をソースドメイン (source domain)，学習された識別規則を適用するデータをテストデータ（test data），それが属する領域をターゲットドメイン (target domain) と呼ぶ．ここで領域とは，新聞や Yahoo!知恵袋などのデータが属するメディアとする．例として，ソース領域が新聞である訓練データから学習した分類器をターゲット領域が雑誌であるテストデータに適用する流れを図 4.1 に示す．

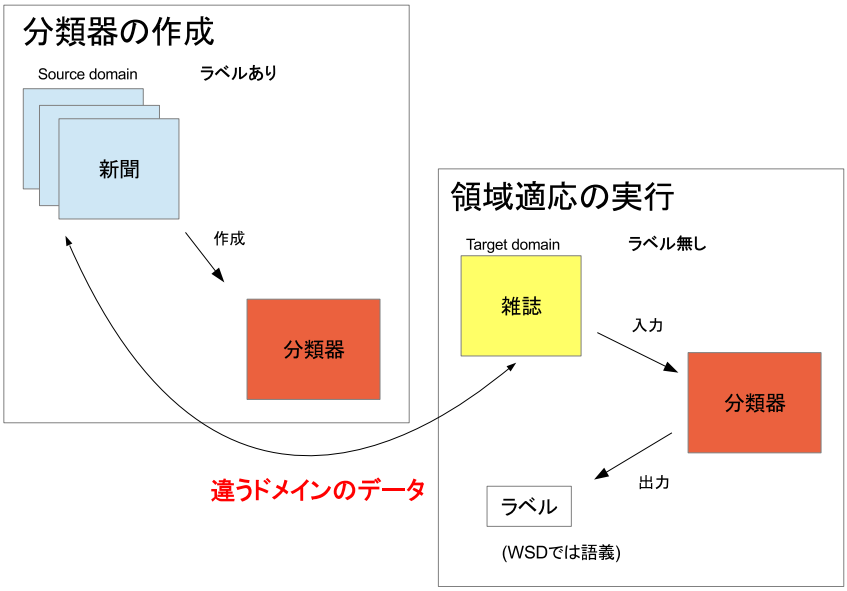

図 4.1: 領域適応時の機械学習

4.2.2 問題点

単語には多くの意味や正しい意味を識別する過程，または文脈上での意味が存在する．単語が属する領域が異なる場合，その語義の分布も異なることがほとんどである．これは WSD の領域適応の問題点である．

実際にソース領域の訓練データで Support Vector Machine(SVM) を学習させ，ターゲット領域のテストデータの語義を推定させた正解率と，ソース領域とターゲット領域が互いに同じ場合の正解率を表 4.1 に示す．なお，ソース領域とターゲット領域が同じ場合は，One-vs-Other 法の交差検定を用いて行う．

表から読み取れるように，ソース領域とターゲット領域が同じ場合と異なる場合とを比べると異なる場合では正解率が大幅に下がってしまっていることが分かる．これは前述したように，領域間での語義の比率が異なることによって起こる問題である．これを如何にして向上させるかが領域適応の研究における目的である．

4.3 関連研究

ここで，自然言語処理の分野で行われているさまざまな領域適応の研究のうち一部を紹介する．まず最も重要な研究である (Hal Daumé(2007))[6] の研究についてである．この研究によって自然言語処理における領域適応の研究の枠組みが明確化された．Daumé は，教師付き学習（supervised）の領域適応を行っている．この論文では訓練データとテストデータを合わせて，素性空間を「訓練データのみ」「テストデータのみ」「訓練データとテストデータ両方」の 3 倍にしてから通常の学習を行う実験を行っている．これは様々な教師付き学習に併用することが可能であり効果が高い，加えて実装が簡単である．マルチドメインへの拡張も容易であることも利点に挙げられている．筆者は (Hal Daumé(2010))[7] において文献 [6] を半教師付き学習

表 4.1: 領域適応による正解率の低下

ソース領域	ターゲット領域	正解率
Yahoo!知恵袋	Yahoo!知恵袋	82.04%
新聞		76.00%
書籍		77.40%
Yahoo!知恵袋	新聞	76.60%
新聞		84.29%
書籍		81.80%
Yahoo!知恵袋	書籍	80.60%
新聞		78.20%
書籍		83.27%

(semi-supervised) のために拡張している．拡張前の利点を引き継いだ上にターゲット領域のラベル無しデータを用いることで性能が向上する手法である．

WSD の領域適応を扱った研究としては古宮の研究が挙げられる [12][13][24]．これらは全て教師付き学習に属する研究である．（古宮 (2010)[24]）では，WSD について領域適応を行った場合，最も効果的な領域適応手法は訓練データとテストデータの性質により異なるとし，WSD の対象単語タイプ，訓練データ，テストデータの 3 つ組を 1 ケースで数え，そのケース毎にデータの性質から最も効果的な領域適応手法を決定木学習によって自動的に選択する手法について述べるとともに，どのような性質が効果的な領域適応手法の決定に影響を与えたかについて考察している．

(Agirre and Lacalle(2008))[1] は，半教師付き学習 (semi-supervised) についての WSD における領域適応を行っている．訓練データにターゲット領域のテストデータを加えて行列を作成し，特異値分解 (SVD) により素性圧縮して分類器を学習するというものである．また筆者らは (Agrrie and Lacalle(2009))[2] において同じ手法で教師付き学習 (supervised) の領域適応を行っている．

(Chan and Ng(2006))[3] は，EM アルゴリズムによる Prior(意味の割合) 推定により WSD の領域適応を行っている．また，筆者らは (Chan and Ng (2007))[4] でも，EM アルゴリズムによる Prior 推定を行っているが，そこでは Active-learning により用例をターゲット領域から足す教師付き学習 (supervised) の領域適応を行っている．Count-marging により重要文に重みをつけてから推定を行っている．

第5章　共変量シフト

前章において，語義曖昧性解消の領域適応ではソース領域とターゲット領域が同じである通常の語義曖昧性解消と比べると精度が悪化してしまう問題があると述べた．本論文では, この領域適応の問題が起こる原因を共変量シフトの問題 [16] であるとし，その解決を図っていく．

5.1　概要

共変量シフト（Coavariate Shift）とは，転移学習における精度悪化の原因と仮定される問題のことである．共変量シフトでは，以下の式が成り立つものと仮定している．

$$P_S(c|\boldsymbol{x}) = P_T(c|\boldsymbol{x}) \tag{5.1}$$

$$P_S(\boldsymbol{x}) \neq P_T(\boldsymbol{x}) \tag{5.2}$$

上記の式 (5.1) は，ある領域 S で出現した事例 $\boldsymbol{x}$ が他の領域 T で出現しても，その事例 $\boldsymbol{x}$ が示す意味は変わらないとすることを意味している．これは通常の自然言語処理のタスクでは成立している仮定だといえる．そして精度悪化の原因は式 (5.2) が表しているように領域 S の確率密度 $P_S(\boldsymbol{x})$ と領域 T の確率密度 $P_T(\boldsymbol{x})$ が異なることにより起こっていると仮定する．これが共変量シフトの問題であり，近年，研究が行われている．

5.2　共変量シフト下における領域適応

対象単語 w の語義の集合を C，また w の事例 $\boldsymbol{x}$ 内の w の語義を c と識別したときの損失関数を $l(\boldsymbol{x}, c, d)$ と表す．d は w の語義を識別する分類器である．$P_T(\boldsymbol{x})$ をターゲット領域上の分布とすれば，本論文のタスクにおける期待損失 L_0 は以下で表すことができる．

$$L_0 = \sum_{\boldsymbol{x},c} l(\boldsymbol{x}, c, d) P_T(\boldsymbol{x}, c) \tag{5.3}$$

また，$P_S(\boldsymbol{x}, c)$ をソース領域上の分布とすると以下が成立する．

$$L_0 = \sum_{\boldsymbol{x},c} l(\boldsymbol{x}, c, d) \frac{P_T(\boldsymbol{x}, c)}{P_S(\boldsymbol{x}, c)} P_S(\boldsymbol{x}, c) \tag{5.4}$$

ここで共変量シフトの仮定である，$P_T(c|\boldsymbol{x}) = P_S(c|\boldsymbol{x})$ を用いると，

$$\frac{P_T(\boldsymbol{x}, c)}{P_S(\boldsymbol{x}, c)} = \frac{P_T(\boldsymbol{x}) P_T(c|\boldsymbol{x})}{P_S(\boldsymbol{x}) P_S(c|\boldsymbol{x})} = \frac{P_T(\boldsymbol{x})}{P_S(\boldsymbol{x})} \tag{5.5}$$

となり，$w(\boldsymbol{x}) = \frac{P_T(\boldsymbol{x})}{P_S(\boldsymbol{x})}$ とおくと以下が成立する．

$$L_0 = \sum_{\boldsymbol{x},c} w(\boldsymbol{x}) l(\boldsymbol{x}, c, d) P_S(\boldsymbol{x}, c) \tag{5.6}$$

訓練データを $D = \{(\boldsymbol{x}_i, c_i)\}_{i=1}^N$ とし，$P_S(\boldsymbol{x}, c)$ を以下の経験分布関数，

$$P_N(\boldsymbol{x}_i, c) = P_N(\boldsymbol{x}_i) := \frac{1}{N}\sum_{i=1}^{N} 1 \quad (\boldsymbol{x}_i \in \boldsymbol{x})$$

を用いて近似することで以下を得る．

$$L_0 \approx \frac{1}{N}\sum_{i=1}^{N} w(\boldsymbol{x}_i) l(\boldsymbol{x}_i, c_i, d) \tag{5.7}$$

期待損失最小化の観点から考えると，共変量シフトの問題は以下の式 L_1 を最小にする d を求めればよいことが分かる．

$$L_1 = \sum_{i=1}^{N} w(\boldsymbol{x}_i) l(\boldsymbol{x}_i, c_i, d) \tag{5.8}$$

ここで，分類器 d に以下の事後確率最大化推定に基づく識別を考える．

$$d(\boldsymbol{x}) = \arg\max_c P_T(c|\boldsymbol{x}) \tag{5.9}$$

また期待損失として対数損失 $-\log P_T(c|\boldsymbol{x})$ を用いると，式 (5.8) は以下となる．

$$L_1 = -\sum_{i=1}^{N} w(\boldsymbol{x}_i) \log P_T(c_i|\boldsymbol{x}_i)$$

すなわち，分類問題の解決に $P_T(c|\boldsymbol{x}, \boldsymbol{\lambda})$ のモデルを導入するアプローチをとる場合，共変量シフト下での学習では確率密度比 $w(\boldsymbol{x}_i) = P_T(\boldsymbol{x}_i)/P_S(\boldsymbol{x}_i)$ を重みとした以下に示す重み付き対数尤度 $L(\boldsymbol{\lambda})$ を最大化するパラメータ $\boldsymbol{\lambda}$ を求める形となる．

$$L(\boldsymbol{\lambda}) = \sum_{i=1}^{N} w(\boldsymbol{x}_i) \log P_T(c_i|\boldsymbol{x}_i, \boldsymbol{\lambda})$$

モデルとしては，以下の式で示される最大エントロピー法がよく用いられる．

$$P_T(c|\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{\lambda})} \exp\left(\sum_{j=1}^{M} \lambda_j f_j(\boldsymbol{x}, c)\right)$$

$\boldsymbol{x} = (x_1, x_2, \cdots, x_M)$ が入力，c がクラスとなる．関数 $f_j(\boldsymbol{x}, c)$ は素性関数であり，実質 $\boldsymbol{x}$ の真のクラスが c のときに x_j を返し，そうでないときに 0 を返す関数に設定される．$Z(\boldsymbol{x}, \boldsymbol{\lambda})$ は正規化項であり，以下で表される．

$$Z(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{c \in C} \exp\left(\sum_{j=1}^{M} \lambda_j f_j(\boldsymbol{x}, c)\right)$$

そして $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_M)$ が素性に対応する重みパラメータとなる．

前述したように，共変量シフト下での確率密度比を利用した重み付け学習では，通常はロジスティック回帰や最大エントロピー法が用いられる．しかし，損失関数ベースの手法であれば重み付け学習の利用は可能である．本論文では，ロジスティック回帰や最大エントロピー法の

代わりにSVMを利用する．SVMは非常に高い識別精度を有することで知られており，重み付け学習による識別精度の値にも期待がもたれる．

ここでは，不均衡データに対するSVMの手法[?]を利用する．訓練データを$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ ($\boldsymbol{x}_i \in R^d$, $y_i \in \{-1, 1\}$) とするとき，SVMは通常，以下のからパラメータ$\boldsymbol{w}, b, \boldsymbol{\zeta}$を求めて識別器を学習する．

$$\min_{\boldsymbol{w},b,\boldsymbol{\zeta}} \left\{ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\zeta_i \right\}$$

ここで，

$$y_i(\boldsymbol{w}^T\phi(x_i) + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0$$

である．上記の式で$\boldsymbol{x}_i$に対してCの代わりに$w(\boldsymbol{x}_i)C$を用いることで，重み付き学習が可能となる[5]．

5.3 関連研究

共変量シフトを仮定した領域適応の研究としては，Jiangらの研究[8]と齋木らの研究[33]があげられる．Jiangは，手動で調整した確率密度比を重みとして利用し，モデルにはロジスティック回帰を利用している．また齋木は$P_T(\boldsymbol{x})$と$P_T(\boldsymbol{x})$をunigramでモデル化することで確率密度比を推定し，モデルには最大エントロピー法を用いている．ただし，どちらの研究もタスクは語義曖昧性解消ではない．さらにターゲット領域のラベル付きデータを利用しているため教師なし学習の手法ではない．また新納ら[28]は，語義曖昧性解消の領域適応に共変量シフトを仮定とした学習を行っている．そこではDaumé が提案した手法[6]も利用しているため教師なし学習とはいえない．

共変量シフトは，確率密度比を重みとした重み付け学習の一種とみなすことができる．Jiangらは識別精度を悪化させるデータをMisleadingデータとして訓練データから取り除いて学習させることで識別精度を向上させる手法を試みている[8]．これはMisleadingデータと判別されたデータの重みを0として学習しているとみなせるため重み付き学習の一種と判断することができる．吉田らはソース領域内の訓練データ$\boldsymbol{x}$がターゲット領域からみて外れ値とみなせる場合，$\boldsymbol{x}$をMisleadingデータとして訓練データから取り除いてから学習をしている[22]．ここでのタスクは語義曖昧性解消の教師なしの領域適応であるが，Misleadingデータの検出は困難であり，精度の改善はされていない．

第6章　確率密度比

前章で説明したように，共変量シフトの下での学習では確率密度比を重みとした重み付け学習を行う．そこで確率密度比の算出手法が重要となってくる．確率密度比は，$w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ の式で求めることが可能である．しかし，WSD をはじめとした実際の問題では $P_S(\boldsymbol{x})$ と $P_T(\boldsymbol{x})$ それぞれを計算して求めることは困難である．そこで，それぞれの領域の確率密度を推定する手法が存在する．確率密度比 $w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ の算出には大きく分けて 2 つの手法が存在する．1 つは $P_S(\boldsymbol{x})$ と $P_T(\boldsymbol{x})$ をそれぞれ推定し，直接その比を取る手法である．もう 1 つは，$w(\boldsymbol{x})$ を直接モデル化し，推定する手法である．ここでは，前者の手法としてベイズ規則を用いた確率密度比の算出 [28]，後者の手法として拘束なし最小二乗重要度適合法（unconstrained Least-Squares Importance Fitting, uLSIF）[10] を利用する．

6.1　NB 法

新納ら [28] は，ベイズ規則を用いて $P_S(\boldsymbol{x})$ と $P_T(\boldsymbol{x})$ を求め，その比を取ることによって確率密度比を算出している．

対象単語 w の用例 $\boldsymbol{x}$ の素性リストを $\{f_1, f_2, \cdots, f_n\}$ とする．求めたいのは領域 $R \in \{S, T\}$ 上の分布 $P_R(\boldsymbol{x})$ である．ここでは Naive Bayes に使われるモデルを用いることで，$P_R(\boldsymbol{x})$ の推定を試みる．Naive Bayes のモデルでは以下を仮定している．

$$P_R(\boldsymbol{x}) = \prod_{i=1}^{n} P_R(f_i) \tag{6.1}$$

領域 R 上のコーパス内の対象単語 w のすべての用例 $\boldsymbol{x}$ について素性リストを作成する．用例 $\boldsymbol{x}$ の数を $N(R)$，素性 f_i が出現する用例数を $n(R, f_i)$ とすると，MAP 推定のスムージングを用い，$P_R(f_i)$ は以下のように定義することができる [26].

$$P_R(f_i) = \frac{n(R, f_i) + 1}{N(R) + 2} \tag{6.2}$$

以上により，ソース領域 S の用例 $\boldsymbol{x}$ に対して，確率密度比 $w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ を計算することが可能となる．

$$w(\boldsymbol{x}) = \frac{P_T(\boldsymbol{x})}{P_S(\boldsymbol{x})} = \prod_{i=1}^{n} \left(\frac{n(T, f_i + 1)}{N(T) + 2} \cdot \frac{N(S) + 2}{n(S, f_i) + 1} \right) \tag{6.3}$$

6.2 uLSIF

ソース領域 S 内のデータを $\{\boldsymbol{x}_i^S\}_{i=1}^{N_S}$，ターゲット領域 T 内のデータを $\{\boldsymbol{x}_i^T\}_{i=1}^{N_T}$ とする．uLSIF では確率密度比 $w(\boldsymbol{x})$ を以下の式でモデル化している．

$$\begin{aligned} w(\boldsymbol{x}) &= \sum_{l=1}^{b} \alpha_l \psi_l(\boldsymbol{x}) \\ &= \boldsymbol{\alpha} \cdot \boldsymbol{\psi}(\boldsymbol{x}) \end{aligned}$$

ただし，$\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_b)$，$\boldsymbol{\psi}(\boldsymbol{x}) = (\psi(\boldsymbol{x}), \psi(\boldsymbol{x}), \cdots, \psi(\boldsymbol{x}))$ である．また α_l は正の実数値であり，$\psi_l(\boldsymbol{x})$ は基底関数と呼ばれるソース領域のデータ $\boldsymbol{x}$ から正の実数値への関数である．uLSIF では，自然数 b と基底関数 $\boldsymbol{\psi}(\boldsymbol{x})$ を定めた後にパラメータ $\boldsymbol{\alpha}$ を推定する手順をとる．

説明の都合上，b と $\boldsymbol{\psi}(\boldsymbol{x})$ が定まった後の $\boldsymbol{\alpha}$ の推定を先に説明することとする．$w(\boldsymbol{x})$ のモデルを $\hat{w}(\boldsymbol{x})$ とおくと，パラメータ α_l を推定するためには，$w(\boldsymbol{x})$ と $\hat{w}(\boldsymbol{x})$ の平均 2 乗誤差 $J_0(\boldsymbol{\alpha})$ を最小にするように $\boldsymbol{\alpha}$ を求める．$w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ に注意すると，$J_0(\boldsymbol{\alpha})$ は以下のように変形することができる．

$$\begin{aligned} J_0(\boldsymbol{\alpha}) &= \frac{1}{2}\int (\hat{w}(\boldsymbol{x}) - w(\boldsymbol{x}))^2 P_S(\boldsymbol{x}) d\boldsymbol{x} \\ &= \frac{1}{2}\int \hat{w}(\boldsymbol{x})^2 P_S(\boldsymbol{x}) d\boldsymbol{x} - \int \hat{w}(\boldsymbol{x}) w(\boldsymbol{x}) d\boldsymbol{x} P_S(\boldsymbol{x}) + \frac{1}{2}\int w(\boldsymbol{x})^2 P_S(\boldsymbol{x}) d\boldsymbol{x} \\ &= \frac{1}{2}\int \hat{w}(\boldsymbol{x})^2 P_S(\boldsymbol{x}) d\boldsymbol{x} - \int \hat{w}(\boldsymbol{x}) P_T(\boldsymbol{x}) d\boldsymbol{x} + \frac{1}{2}\int w(\boldsymbol{x})^2 P_S(\boldsymbol{x}) d\boldsymbol{x} \end{aligned}$$

3 項目の式は定数のため無視することができる．そのため $j_0(\boldsymbol{\alpha})$ を最小にするには，以下の $J(\boldsymbol{\alpha})$ を最小にすればよい．

$$J(\boldsymbol{\alpha}) = \frac{1}{2}\int \hat{w}(\boldsymbol{x})^2 P_S(\boldsymbol{x}) d\boldsymbol{x} - \int \hat{w}(\boldsymbol{x}) P_T(\boldsymbol{x}) d\boldsymbol{x}$$

$J(\boldsymbol{\alpha})$ を経験分布で近似した $\hat{J}(\boldsymbol{\alpha})$ は以下となる．

$$\begin{aligned} \hat{J}(\boldsymbol{\alpha}) &= \frac{1}{2N_S}\sum_{i=1}^{N_S} \hat{w}(\boldsymbol{x}_i^S)^2 - \frac{1}{N_T}\sum_{j=1}^{N_T} \hat{w}(\boldsymbol{x}_j^T) \\ &= \frac{1}{2}\sum_{l,l'=1}^{b} \alpha_l \alpha_{l'} \left(\frac{1}{N_S}\sum_{i=1}^{N_S} \psi_l(\boldsymbol{x}_i^S)\psi_{l'}(\boldsymbol{x}_i^S)) \right) - \sum_{l}^{b} \alpha_l \left(\frac{1}{N_T}\sum_{j=1}^{N_T} \psi_l(\boldsymbol{x}_j^T) \right) \\ &= \frac{1}{2}\boldsymbol{\alpha}^T \hat{H} \alpha - \hat{h}^T \boldsymbol{\alpha} \end{aligned} \tag{6.4}$$

ここで，$\hat{H}$ は $b \times b$ の行列であり，その l 行 l' 列の要素 $\hat{H}_{l,l'}$ は以下となる．

$$\hat{H}_{l,l'} = \frac{1}{N_S}\sum_{i=1}^{N_S} \psi_l(\boldsymbol{x}_i^S)\psi_{l'}(\boldsymbol{x}_i^S)$$

また $\hat{h}$ は b 次元のベクトルであり，その l 次元目の要素 $\hat{h}_l$ は以下である．

$$\hat{h}_l = \frac{1}{N_T}\sum_{j=1}^{N_T}\psi_l(\boldsymbol{x}_j^T)$$

$\hat{J}(\boldsymbol{\alpha})$ の最小値を求める際に正則化を行う．このとき付加する正則化項を L2 ノルムに設定し，$\boldsymbol{\alpha} > 0$ の条件を除外し，以下の最小化問題を解く．ここでパラメータ λ が導入されることに注意する．λ は基底関数を設定する際に決められる．

$$\min_{\boldsymbol{\alpha}}\left[\frac{1}{2}\boldsymbol{\alpha}^T\hat{H}\boldsymbol{\alpha} - \hat{h}^T\boldsymbol{\alpha} + \frac{\lambda}{2}\boldsymbol{\alpha}^T\boldsymbol{\alpha}\right]$$

この最小化問題は制約のない凸 2 次計画化問題であるため，唯一の大域解を得ることができる．その解は以下となる．

$$\tilde{\boldsymbol{\alpha}} = (\hat{H} + \lambda I_b)^{-1}\hat{h}^T \tag{6.5}$$

最後に，$\boldsymbol{\alpha} > 0$ の条件に合うように以下の調整をする．

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= (\max(0, \tilde{\alpha_1}), \max(0, \tilde{\alpha_2}), \cdots, \max(0, \tilde{\alpha_b})) \\ &= \max(0_b, \tilde{\boldsymbol{\alpha}})\end{aligned} \tag{6.6}$$

次にパラメータ b と基底関数 $\boldsymbol{\psi}(\boldsymbol{x})$ の設定であるが，まず，b については以下で設定する．

$$b = \min(100, N_T)$$

次にターゲット領域のデータから重複を許さずに b 個の用例をランダムに取り出す．それらの用例を $\{\boldsymbol{x}_{j=1}^T\}_{j=1}^b$ とおく．そして基底関数 $\psi_l(\boldsymbol{x})$ を以下のガウシアンカーネルで定義する．

$$\psi_l(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x}_l^T) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}_l^T||^2}{\sigma^2}\right)$$

以上より，確率密度比を求めるために残されているパラメータは正則化項の係数 λ とガウシアンカーネルの幅 σ の 2 つとなる．これらのパラメータはグリッドサーチの交差検定で求めていく．

まずはソース領域とターゲット領域のデータをそれぞれ交わりのない R 個の部分集合に分割する．それらの部分集合の中で r 番目の部分集合を除き，残りを結合した集合を作る．それらを新たなソース領域のデータとターゲット領域のデータとみなすこととする．そして λ と σ を適当な値に設定し，式 (6.5) と式 (6.6) より $\boldsymbol{\alpha}$ を求め，式 (6.4) より $\hat{J}(\boldsymbol{\alpha})^{(r)}$ の値を求める．r の値を 1 から R まで変化させることで R 個の $\hat{J}(\boldsymbol{\alpha})^{(r)}$ が求まる．それらを平均した値を λ と σ に対する $\hat{J}(\boldsymbol{\alpha})$ の値とする．次に λ と σ を変化させて，上記の手順で得られる $\hat{J}(\boldsymbol{\alpha})$ の値が最小になる $\hat{\lambda}$ と $\hat{\sigma}$ を求め，これを λ と σ の推定値とする．

6.3 WSD における確率密度比

WSD のタスクにおいて NB 法または uLSIF で算出される確率密度比の値は，実際の値よりも小さい数値を取る傾向が高く，学習で用いる際には少し上方修正をした値を用いた方が最終の識別精度が改善されることが多い．この問題は，以下の 2 点から生じると考えられている．

- ターゲット領域 T に $\boldsymbol{x}$ が含まれているかは確率的だが，ソース領域 S には必ず出現する．
- $P_S(\boldsymbol{x})$ を推定するために $\boldsymbol{x} \in S$ を用いるため，訓練データである $\boldsymbol{x}$ に過学習した結果 $P_S(\boldsymbol{x})$ は $P_T(\boldsymbol{x})$ に比べて高く見積もられてしまう．

このため，求めた確率密度比を上方修正するための手法が存在する．

6.3.1 上方修正の手法

杉山 [31] は，算出された確率密度比 $w(\boldsymbol{x})$ を p 乗 $(0 < p < 1)$ することで上方修正することを提案している．確率密度比 $w(\boldsymbol{x})$ が 1 以下である場合，$w(\boldsymbol{x})$ を p 乗すると上方に修正できることはそれらの比の対数を取ることで明らかである $(\log w(\boldsymbol{x}) < 0)$.

$$\log \frac{w(\boldsymbol{x})^p}{w(\boldsymbol{x})} = (p-1)\log w(\boldsymbol{x}) > 0$$

山田ら [19] は，以下で示される相対確率密度比 $w'(\boldsymbol{x})$ を確率密度比として用いることを提案している．

$$w'(\boldsymbol{x}) = \frac{P_T(\boldsymbol{x})}{\alpha P_S(\boldsymbol{x}) + (1-\alpha)P_T(\boldsymbol{x})}$$

ここで $0 < \alpha < 1$ である．相対確率密度比 $w'(\boldsymbol{x})$ は以下の変形から $w(\boldsymbol{x})$ を上方に修正しているとみなすことができる．

$$\begin{aligned} w'(\boldsymbol{x}) &= \frac{P_T(\boldsymbol{x})}{\alpha P_S(\boldsymbol{x}) + (1-\alpha)P_T(\boldsymbol{x})} \\ &= \frac{1}{\alpha + (1+\alpha)w(\boldsymbol{x})} w(\boldsymbol{x}) \\ &> \frac{1}{\alpha + (1-\alpha)} w(\boldsymbol{x}) \\ &= w(\boldsymbol{x}) \end{aligned}$$

確率密度比が 1 以上である場合，以上の 2 つの手法は確率密度比を下方修正する．つまり，正確に説明を設けるとすれば「確率密度比を 1 に近づける手法」と定義することができる．

6.3.2 NB 法における $P_S(\boldsymbol{x})$ の補正

新納らは，確率密度比を上方修正するために NB 法に改良を加えている [29]．そこでは，ソース領域 S とターゲット領域 T のデータを合わせたデータを新たにソース領域 S のデータとみなすことで $P_S(\boldsymbol{x})$ の補正を行っている．これはソース領域 S のスパース性を緩和させることを目的としている．確率密度比が真の値よりも低く推定されてしまう原因の 1 つは，$P_S(\boldsymbol{x})$ が実

際よりも高く見積もられてしまうことが原因だと考えられることは前述した．その原因がソース領域 S のスパース性と考え，緩和させるためにソース領域 S にデータを加えるとしている．ただし，このとき追加するデータはソース領域 S に類似している領域のデータであることが望まれる．WSD の領域適応の場合，ソース領域 S とターゲット領域 T は完全に異なることはなく，比較的類似しているため，追加するデータはターゲット領域 T のデータでもよいと考えられる．

この手法により作成された新たなソース領域 S を $S+T$ とする．このことを踏まえると $P_S(\boldsymbol{x}) > P_{S+T}(\boldsymbol{x})$ が成り立つこととなる．$P_{S+T}(\boldsymbol{x})$ を求める式を以下に示す．

$$\begin{aligned} P_{S+T}(f_i) &= \frac{n(S+T,f_i)+1}{N(S+T)+2} \\ &= \frac{n(S,f_i)+n(T,f_i)+1}{N(S)+N(T)+2} \end{aligned}$$

この不等式が成立すれば，従来の確率密度比の算出式 $w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ は $w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_{S+T}(\boldsymbol{x})$ に更新され，上方修正されることとなる．この手法は，確率密度比が 1 以上であるかどうかという点に関しては考慮しない．つまり，確率密度比が 1 以上の場合でも上方修正してしまうことがあることに注意する．

以上を考慮すると，NB 法を改良した式は以下となる．

$$w(\boldsymbol{x}) = \frac{P_T(\boldsymbol{x})}{P_{S+T}(\boldsymbol{x})} = \prod_{i=1}^{n} \left(\frac{n(T,f_i)+1}{N(T)+2} \cdot \frac{N(S)+N(T)+2}{n(S,f_i)+n(T,f_i)+1} \right)$$

以上の式より，算出した確率密度比を重みとして用いる．

第7章　実験

7.1　実験説明

本実験では，BCCWJ コーパスの Yahoo!知恵袋（OC），書籍（PB），新聞（PN）の 3 つのメディアをそれぞれ異なる領域として利用した．SemEval-2 の日本語 WSD のタスクではこれらの領域のコーパスの一部に語義タグを付けたデータを公開しており，そのデータを利用することとする．この 3 つの領域からある程度の頻度で出現する多義語 16 単語を対象単語 w として，それぞれの単語に対して領域適応の実験を行う．対象単語に関する情報を表 7.1 に示す。

表 7.1: 対象単語の情報

対象単語 $\boldsymbol{w}$	OC での頻度	OC での語義数	PB での頻度	PB での語義数	PN での頻度	PN での語義数
言う	666	2	1114	2	363	2
入れる	73	2	56	3	32	2
書く	99	2	62	2	27	2
聞く	124	2	123	2	52	2
子供	77	2	93	2	29	2
時間	53	2	74	2	59	2
自分	128	2	308	2	71	2
出る	131	3	152	3	89	3
取る	61	7	81	7	43	7
場合	126	2	137	2	73	2
入る	68	4	118	4	65	3
前	105	3	160	2	106	4
見る	262	5	273	6	87	3
持つ	62	4	153	3	59	3
やる	117	3	156	4	27	2
ゆく	219	2	133	2	27	2

領域適応の種類としては，OC → PB，OC → PN，PB → OC，PB → PN，PN → OC，PN → PB の計 6 種類となる．つまり，合計で $16 \times 6 = 96$ 通りの実験を行うこととなる．分類器の学習に使用する素性としては，(e0) 対象単語 w の表記，(e1) 対象単語 w の品詞，(e2)$w_{(-1)}$ の表記，(e3)$w_{(-1)}$ の品詞，(e4)$w_{(+1)}$ の表記，(e5)$w_{(+1)}$ の品詞，(e6) 対象単語 w の前後 3 単語までの自立語の表記，(e7)e6 の分類語彙表の番号の 4 桁と 5 桁の 8 種類を使用した．ここで $w_{(-1)}$ は対象単語 w の直前の単語，$w_{(+1)}$ は対象単語 w の直後の単語とする．

次に，対象単語 w に関するソース領域 S からターゲット領域 T への領域適応の実験について

説明する．本実験では，教師なし手法を用いて領域適応を行う．ソース領域 S の訓練データのみを用いて，手法 A により分類器を学習させ対象単語 w に対する正解率を求める．16 種類の対象単語 $(w_1, w_2, \cdots, w_{16})$ に対する正解率の平均，つまりマクロ平均をソース領域 S からターゲット領域 T に対する手法 A の正解率とする．結果，手法 A に関して 6 種類の領域適応の正解率が得られることとなる．それらの平均を手法 A の平均正解率とし，その値が手法 A 評価値となる．

7.2 実験結果

各手法としては以下を試す.

1. 重みを考慮しない手法（重みを 1 で固定）（Base）
2. NB 法により確率密度比を算出し，それを重みとした手法 (NB)
3. NB 法で算出した確率密度比を p 乗 ($p = 0.2$) し，重みとした手法 (P-NB)
4. ソース領域 S とターゲット領域 T を合わせたデータを新たなソース領域 S のデータとして NB 法により確率密度比を算出し，それを重みとした手法 (NB-ST)
5. uLSIF により算出した確率密度比を重みとした手法 (uLSIF)
6. uLSIF により算出した確率密度比を p 乗 ($p = 0.3$) したものを重みとした手法 (P-uLSIF)

P-NB と P-uLSIF の手法では，確率密度比を p 乗した値を重みとして利用している．その際，パラメータ p の値が重要となってくる．ここでは，p を 0.1 から 0.9 の範囲で 0.1 ずつ変化させて一番よい正解率を出したものを p の値として設定した．P-NB では $p = 0.2$，P-uLSIF では $p = 0.3$ に設定している．

全ての手法において，分類器の学習手法としては SVM を用いる．カーネルは線形カーネルを利用した．また，実行ツールとしては scikit-learn にて提供されている SVM[1] 使用した．実験の結果を表 7.2 に示す.

表 7.2: 各手法の正解率

手法	OC → PB	OC → PN	PB → OC	PB → PN	PN → OC	PN → PB	平均正解率
Base	0.7172	0.7006	0.7008	0.7173	0.7123	0.7174	0.7109
NB	0.7039	0.6900	0.6894	0.7128	0.7022	0.6928	0.6985
P-NB	0.7191	**0.7027**	0.7033	0.7195	0.7180	0.7208	0.7139
NB-ST	0.7175	0.7026	0.7025	0.7198	**0.7221**	**0.7253**	**0.7150**
uLSIF	0.6924	0.6969	0.6840	0.6896	0.6884	0.6952	0.6911
P-uLSIF	**0.7203**	0.7025	**0.7052**	**0.7222**	0.7161	0.7203	0.7144

結果から，領域適応の種類ごとに最適な手法は異なっているが，平均正解率としては，NB-ST が最も高い値を示している．重み付き SVM において，ソース領域 S とターゲット領域 T を合わせたデータを新たなソース領域 S とみなして算出した確率密度比の効果を確認することができた．P-NB は NB よりも高い平均正解率を，P-uLSIF は uLSIF よりも高い平均正解率を示していることから，上方修正の手法が有効であったことがわかる．

[1] http://scikit-learn.org/stable/modules/svm.html

第8章　考察

8.1　確率密度比の補正

実験結果から，確率密度比を上方修正する際に確率密度比を p 乗する手法が有効であることが確認できた．ここでは，NB，uLSIF において p の値を 0.1 から 0.9 まで 0.1 刻みで変化させたときの正解率の変化を確認する．結果を表 8.1 および表 8.2 に，平均正解率の推移を図 8.1 と図 8.2 に示す．表中の括弧内は p の値を示している．

表 8.1: NB 法における確率密度比の補正

NB 法	OC → PB	OC → PN	PB → OC	PB → PN	PN → OC	PN → PB	平均正解率
NB	0.7039	0.6900	0.6894	0.7128	0.7022	0.6928	0.6985
NB(0.1)	0.7164	0.7008	0.7008	0.7170	0.7112	0.7163	0.7104
NB(0.2)	**0.7191**	**0.7027**	**0.7033**	**0.7195**	**0.7180**	0.7208	**0.7139**
NB(0.3)	0.7067	0.6954	0.6981	0.7165	0.7169	**0.7209**	0.7091
NB(0.4)	0.7108	0.6976	0.6982	0.7172	0.7165	0.7189	0.7099
NB(0.5)	0.7129	0.6980	0.6969	0.7163	0.7142	0.7163	0.7091
NB(0.6)	0.7141	0.6985	0.6978	0.7179	0.7164	0.7173	0.7103
NB(0.7)	0.7105	0.6944	0.6961	0.7173	0.7144	0.7144	0.7079
NB(0.8)	0.7080	0.6922	0.6944	0.7159	0.7055	0.7005	0.7027
NB(0.9)	0.7053	0.6908	0.6918	0.7143	0.7046	0.6987	0.7009

表 8.2: uLSIF における確率密度比の補正

uLSIF	OC → PB	OC → PN	PB → OC	PB → PN	PN → OC	PN → PB	平均正解率
uLSIF	0.6924	0.6969	0.6840	0.6896	0.6884	0.6952	0.6911
uLSIF(0.1)	0.7158	0.7008	0.7016	0.7186	0.7133	0.7182	0.7114
uLSIF(0.2)	0.7183	**0.7025**	0.7035	0.7208	0.7151	0.7196	0.7133
uLSIF(0.3)	**0.7203**	**0.7025**	**0.7052**	**0.7222**	**0.7161**	**0.7203**	**0.7144**
uLSIF(0.4)	0.7165	0.6989	0.7011	0.7180	0.7126	0.7178	0.7108
uLSIF(0.5)	0.7104	0.6951	0.6967	0.7153	0.7107	0.7163	0.7074
uLSIF(0.6)	0.7026	0.6868	0.6860	0.7033	0.7010	0.7078	0.6979
uLSIF(0.7)	0.7035	0.6851	0.6833	0.6990	0.6965	0.7017	0.6948
uLSIF(0.8)	0.7092	0.6953	0.6877	0.7001	0.6957	0.7018	0.6983
uLSIF(0.9)	0.7026	0.6936	0.6869	0.6945	0.6919	0.6985	0.6947

p の値がいかなる場合でも，ベースとなる NB または uLSIF よりも結果は向上している．このことから，算出された確率密度比を p 乗 $(0 < p < 1)$ した値を新たな確率密度比とみなして

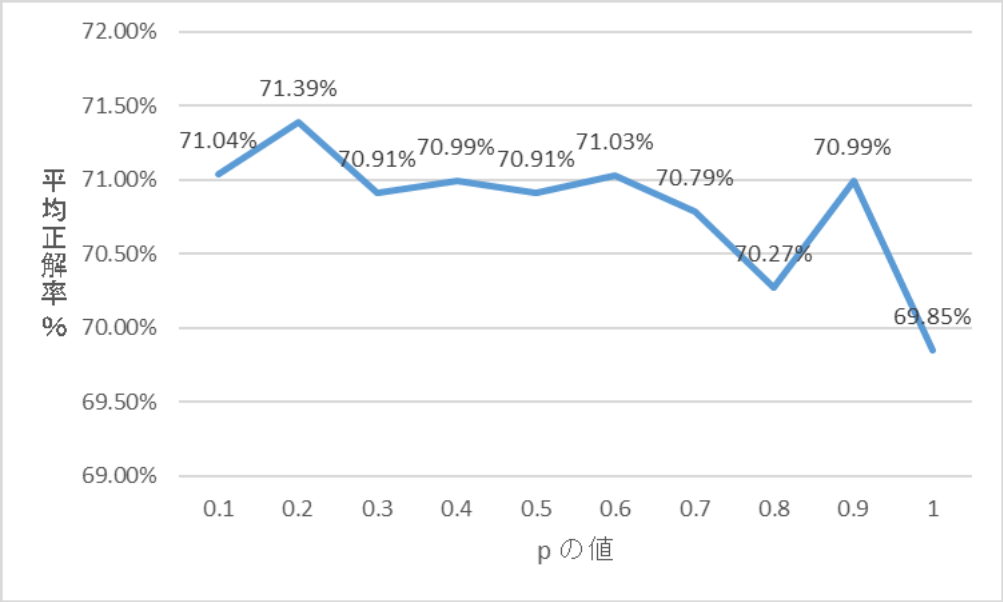

図 8.1: p 乗による NB の上方修正と平均正解率

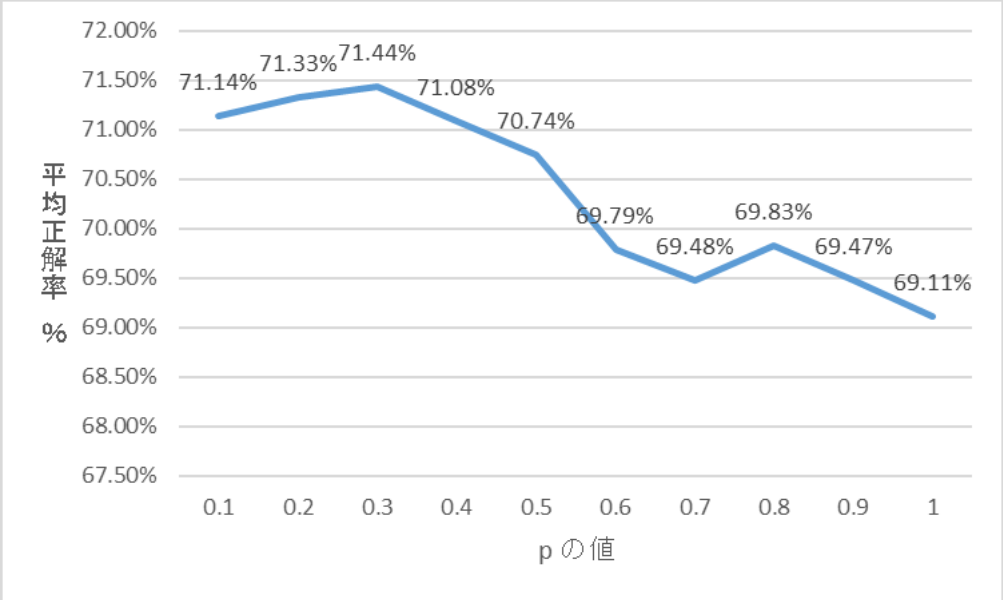

図 8.2: p 乗による uLSIF の上方修正と平均正解率

重みとして適用する効果がうかがえる．今回の実験の場合は，NB 法において $p = 0.2$，uLSIF において $p = 0.3$ の場合で最も高い平均正解率を示した．

また，NB-ST の手法を適用する際に，同じように確率密度比を p 乗する手法を試した場合も確認する．結果を表 8.3 に，平均正解率のグラフを図 8.3 に示す．

表 8.3: NB-ST における確率密度比の補正の併用

NB-ST	OC → PB	OC → PN	PB → OC	PB → PN	PN → OC	PN → PB	平均正解率
NB-ST	0.7175	0.7026	0.7025	**0.7198**	**0.7221**	**0.7253**	**0.7150**
NB-ST(0.1)	0.7166	0.7009	0.7008	0.7173	0.7123	0.7174	0.7109
NB-ST(0.2)	0.7164	0.7009	0.7008	0.7170	0.7116	0.7168	0.7106
NB-ST(0.3)	0.7173	0.7019	0.7024	0.7184	0.7124	0.7174	0.7116
NB-ST(0.4)	0.7183	0.7025	0.7023	0.7189	0.7135	0.7184	0.7123
NB-ST(0.5)	**0.7186**	0.7030	**0.7031**	0.7197	0.7165	0.7216	0.7137
NB-ST(0.6)	0.7167	0.7021	0.7027	0.7196	0.7188	0.7228	0.7138
NB-ST(0.7)	0.7179	**0.7033**	0.7024	0.7194	0.7203	0.7241	0.7146
NB-ST(0.8)	0.7156	0.7011	0.7015	0.7186	0.7206	0.7240	0.7136
NB-ST(0.9)	0.7152	0.7013	0.7016	0.7187	0.7214	0.7244	0.7138

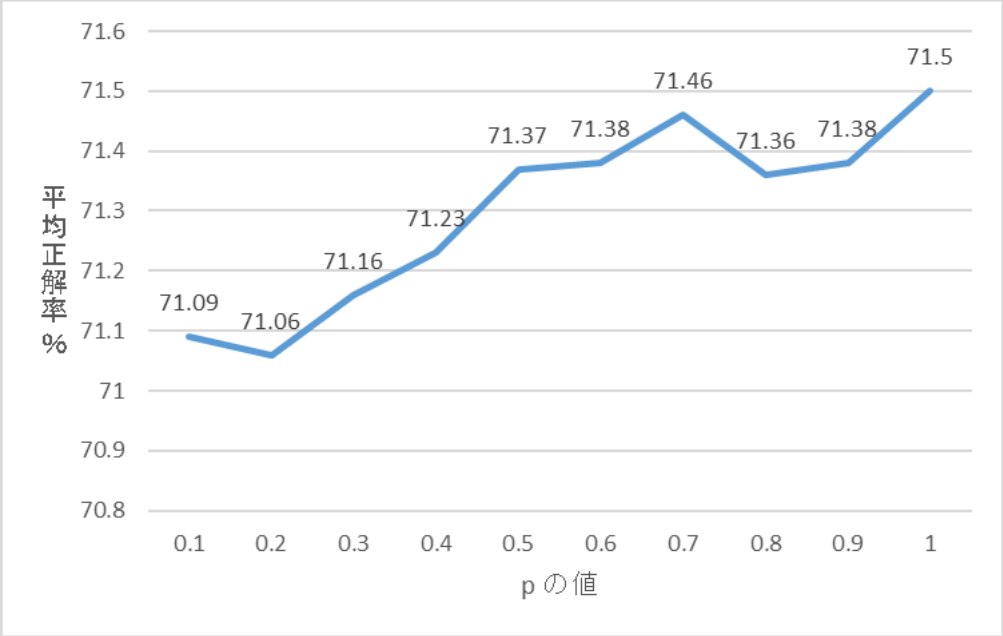

図 8.3: p 乗による NB-ST の上方修正と平均正解率

NB-ST に対して確率密度比を p 乗した場合，平均正解率の向上はみられなかったが，各領域適応に関しては有効な結果も確認された．OC から PB と PB から OC の領域適応に関しては，$p = 0.5$ に，OC から PN の領域適応に関しては，$p = 0.7$ に設定した際に最も高い値を記録した．重み付き SVM において，NB-ST の手法で算出した確率密度比を p 乗して上方修正する効果は確認することができなかった．

8.2 $P_T(\boldsymbol{x})$ の補正

新納の NB 法では，ソース領域とターゲット領域のコーパスを合わせたものを新しいソース領域のコーパスとすることで $P_S(\boldsymbol{x})$ を補正し，確率密度比を上方修正することを試みている．この手法によって $P_S(\boldsymbol{x})$ が下方修正され，確率密度比 $w(\boldsymbol{x}) = P_T(\boldsymbol{x})/P_S(\boldsymbol{x})$ が上昇する．そこで，ここでは $P_T(\boldsymbol{x})$ に補正をかけることで確率密度比を修正し，その効果があるかどうかを確認する．

$P_T(\boldsymbol{x})$ を補正するために，スムージングの値を $P_S(\boldsymbol{x})$ を算出する際よりも多く取る方法を試すこととする．$P_T(\boldsymbol{x})$ を算出する式を以下とする．

$$P_T(\boldsymbol{x}) = \frac{n(T, f_i) + 2}{N(T) + 4}$$

以上により算出した確率密度比を重みとして SVM に適用する手法を T-NB とする．また，NB-ST に上記の手法を適用して $P_T(\boldsymbol{x})$ を補正したものを T-NB-ST とする．結果を表 8.4 に，各手法の平均正解率のグラフを図 8.4 に示す．

表 8.4: $P_T(\boldsymbol{x})$ の補正による実験結果

手法	OC → PB	OC → PN	PB → OC	PB → PN	PN → OC	PN → PB	平均正解率
Base	0.7172	0.7006	0.7008	0.7173	0.7123	0.7174	0.7109
NB	0.7039	0.6900	0.6894	0.7128	0.7022	0.6928	0.6985
P-NB	**0.7191**	0.7027	**0.7033**	0.7195	0.7180	0.7208	0.7139
NB-ST	0.7175	0.7026	0.7025	**0.7198**	**0.7221**	**0.7253**	**0.7150**
T-NB	0.7127	0.7009	0.7017	0.7179	0.7195	0.7210	0.7123
T-NB-ST	0.7188	**0.7028**	0.7026	0.7187	0.7130	0.7183	0.7124

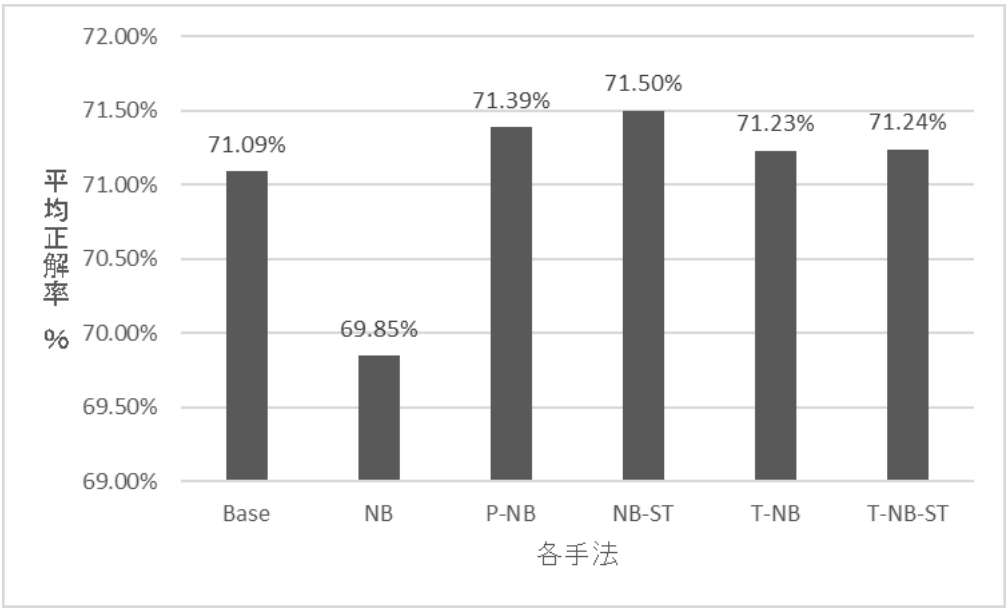

図 8.4: NB 法をベースとした確率密度比の補正手法の平均正解率

結果から，スムージングを調整し，$P_T(\boldsymbol{x})$ を補正することで平均正解率を向上させることはできなかった．しかし，Base や NB の平均正解率よりは高い値を示している．また，T-NB-ST では，$P_S(\boldsymbol{x})$ と $P_T(\boldsymbol{x})$ の両方を補正し，確率密度比を上方修正することを試みたが平均正解率の向上は見られなかった．だが，OC から PN への領域適応では，どの手法よりも高い正解率を示していることからある程度効果が得られる場合もあることがわかる．

第9章　おわりに

本論文では，WSD の教師なし領域適応に対して，その問題に共変量シフトを仮定して確率密度比による重み付け学習を行った．確率密度比の算出手法に，新納によって提案された NB 法とその確率密度比を修正する手法を採用し，重み付き SVM で語義判定の精度が改善するかどうかの確認実験を行った．比較手法としては uLSIF や算出された確率密度比を p 乗する手法の効果も確認し比較した．実験の結果，修正手法を用いた NB 法の正解率が最も高かった．uLSIF により算出した確率密度比を用いた重み付け学習では，そのままの値を重みとして適用した場合は効果がなかったが，p 乗した値を用いた場合，精度の改善がみられた．また修正手法を用いた NB 法と p 乗の手法を合わせて用いる実験も試したが，精度の改善はみられなかった．新納によって提案された手法は，ソース領域側の事前分布 $P_S(\boldsymbol{x})$ を修正することで確率密度比を上方に修正する手法であったが，$P_T(\boldsymbol{x})$ を修正することで確率密度比を上方修正する手法で算出した値を重みとして適用する実験も試みたが，新納の手法を用いた場合よりも精度が改善されることはなかった．

今後の課題としては，より適切な確率密度比の算出手法と修正手法の考案が挙げられる．

謝辞

本研究を進めるにあたり、多大なご指導とご協力を頂いた新納浩幸准教授、佐々木稔教員、古宮嘉那子教員に心から感謝いたします。また、日常の議論を通じて多くの知識や示唆を頂いた新納研究室の皆様に感謝します。

参考文献

[1] E. Agirre and O. L. de Lacalle. On robustness and domain adaptation using svd dor word sense disambiguation. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pp. 17–24, 2008.

[2] E. Agirre and O. L. de Lacalle. Supervised domain adaptation for wsd. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pp. 42–50, 2009.

[3] Yee Seng Chan and Hwee Tou Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *COLING-ACL-2006*, pp. 89–96. Association for Computational Linguistics, 2006.

[4] Yee Seng Chan and Hwee Tou Ng. Domain adaptation with active learning for word sense disambiguation. In *ACL-2007*, Vol. 45, p. 49, 2007.

[5] C. Cortes and V. Vapnik. Support-vector networks. In *Machine learning*, Vol. 20, pp. 273–297, 1995.

[6] Daumé III, Hal. Frustratingly Easy Domain Adaptation. In *ACL-2007*, pp. 256–263, 2007.

[7] Daumé III, Hal. Frustratingly Easy Semi-Supervised Domain Adaptation. In *ACL-2010*, p. 2359, 2010.

[8] Jing Jiang and Chengxiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL-2007*, pp. 264–271, 2007.

[9] Toshihiro Kamishima. Transfer learning (in japanese). *The Japanese Society for Artificial Intelligence*, Vol. 25, No. 4, pp. 572–580, 2010.

[10] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, Vol. 10, pp. 1391–1445, 2009.

[11] Hironori Kikuchi and Hiroyuki Shinnou. Domain Adaptation for Word Sense Disambiguation under the Problem of Covariate Shift. 情報処理学会自然言語処理研究会報告, pp. NL–212–4, 2013.

[12] Kanako Komiya and Manabu Okumura. Automatic Determination of a Domain Adaptation Method for Word Sense Disambiguation using Decision Tree Learning. In *IJCNLP-2011*, pp. 1107–1115, 2011.

[13] Kanako Komiya and Manabu Okumura. Automatic Domain Adaptation for Word Sense Disambiguation Based on Comparison of Multiple Classifiers. In *PACLIC-2012*, pp. 75–85, 2012.

[14] Shinsuke Mori. Domain adaptation in natural language processing (in japanese). *The Japanese Society for Artificial Intelligence*, Vol. 27, No. 4, pp. 365–372, 2012.

[15] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 22, No. 10, pp. 1345–1359, 2010.

[16] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, Vol. 90, No. 2, pp. 227–244, 2000.

[17] Anders Sogaard. *Semi-Supervised Learning and Domain Adaptation in Natural Language Processing*. Morgan & Claypool, 2013.

[18] Masashi Sugiyama and Motoaki Kawanabe. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation* . MIT Press, 2011.

[19] Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. Relative density-ratio estimation for robust distribution comparison. *Neural Computation*, Vol. 25, No. 5, pp. 1370–1370, 2011.

[20] 菊池裕紀, 新納浩幸. ulsif による重み付き学習を利用した語義曖昧性解消の領域適応. 第 5 回コーパス日本語学ワークショップ, pp. 63–70, 2014.

[21] 菊池裕紀, 新納浩幸, 佐々木稔, 古宮嘉那子. ベイズ規則による確率密度比の推定を用いた語義曖昧性解消の領域適応. 言語処理学会第 21 回年次大会, p. to appear, 2015.

[22] 吉田拓夢, 新納浩幸. 外れ値検出手法を利用した misleading データの検出. 第 5 回コーパス日本語学ワークショップ, pp. 49–56, 2014.

[23] 玉垣隆幸, 白井清昭. 読解支援システムのための語義曖昧性解消に関する研究. 言語処理学会第 9 回年次大会, pp. 481–484, 2003.

[24] 古宮嘉那子, 奥村学. 語義曖昧性解消のための領域適応手法の決定木学習による自動選択. 自然言語処理, Vol. 19, No. 3, pp. 143–166, 2012.

[25] 胡寅駿, 谷田泰郎. Wikipedia を用いた語義曖昧性解消のための辞書の自動構築. 言語処理学会第 20 回年次大会, pp. P39–42, 2014.

[26] 高村大也. 言語処理のための機械学習入門. コロナ社, 2010.

[27] 新納浩幸, 菊池裕紀, 佐々木稔, 古宮嘉那子. ulsif を用いた事例への重み付けによる語彙曖昧性解消の領域適応. 情報処理学会自然言語処理研究会, pp. NL–218–2, 2014.

[28] 新納浩幸, 佐々木稔. 共変量シフトの問題としての語義曖昧性解消の領域適応. 自然言語処理, Vol. 21, No. 1, pp. 61–79, 2014.

[29] 新納浩幸, 佐々木稔. 共変量シフト下における語義曖昧性解消の教師なし領域適応. 自然言語処理, Vol. 21, No. 5, pp. 1011–1035, 2014.

[30] 杉山一成, 奥村学. 用例のクラスタリング結果を利用した語義曖昧性解消. 言語処理学会第14回年次大会, pp. P564–567, 2008.

[31] 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.

[32] 杉山将. 密度比に基づく機械学習の新たなアプローチ. 統計数理, Vol. 58, No. 2, pp. 141–155, 2010.

[33] 齋木陽介, 高村大也, 奥村学. 文の感情極性判定における事例重み付けによるドメイン適応 (情報抽出・評判分析). 情報処理学会研究報告. 自然言語処理研究会報告, Vol. 2008, No. 33, pp. 61–67, 2008.