平成28年度　修士論文

圧縮ベースパターン認識に有用な新しい特徴量の抽出

電気通信大学大学院 情報システム学研究科
情報システム基盤学専攻

1553003　内野 太智

指導教員
古賀 久志 准教授
南 泰浩 教授
新谷 隆彦 准教授

平成29年1月26日

目 次

第1章

序論

1.1 研究背景と研究目的

多様な新しい種類のデータが大量に生成されるビッグデータの時代において，データを人手によらないで分類/認識する汎用的なパターン認識手法の重要性が高まっている．これに対して，一般的な統計的パターン認識手法では，パラメータ選択を含めモデルを人手で適切に設計する手間が大きく，新種のデータを分析するのは容易ではない．

一方，圧縮ベースパターン認識は，データ圧縮アルゴリズムを用いたパラメータフリーなパターン認識手法であり，1 次元に表現されたデータであれば何でも解析可能な汎用手法であるため，近年盛んに研究されている．圧縮ベースパターン認識は (1) 圧縮後のファイルサイズを利用する手法と (2) 圧縮時に生成される辞書 (以下，圧縮辞書) から特徴抽出する手法に大別される．前者の例としては，2 つのデータ x, y を結合したファイルを圧縮しそのサイズから，x と y の距離を測る Normalized Compression Distance (NCD) [2] が広く知られている．また，与えられたデータを複数の辞書で圧縮し，それぞれの辞書で求められた圧縮率を並べて特徴ベクトルとする Pattern Representation on Data Compression(PRDC) [5] もよく用いられる．後者の例としては，LZW 圧縮で生成される辞書をデータの要約とみなし，辞書間で類似度を計算する Normalized Dictionary Distance (NDD) [4] や単語の多重度も考慮した Normalized Multiset Distance (NMD) [1] が提案されている．

本研究は，圧縮ベースパターン認識に有用な特徴の探求を目的とする．具体的には，従来手法である PRDC と NMD を取り上げ，PRDC に対しては圧縮後のファ

イルから，NMD に関しては圧縮辞書からそれぞれ新しい特徴を抽出することで，認識精度を向上できることを示す．

本稿の構成は以下のようになる．2 章で既存手法を紹介する．3,4 章が提案手法であり，3 章では圧縮後のファイルからの特徴抽出による PRDC の改良手法，4 章では圧縮辞書からの特徴抽出による NMD の改良手法を説明する．5 章で実験評価を行い，6 章で結論を述べる．

1.2　本論文の構成

以下に本論文の構成を述べる．

第 2 章 圧縮ベースパターン認識の代表的な 2 種類の手法について述べる．また，それぞれの手法の問題点について言及する．

第 3 章 それぞれの手法の問題点を補う新たな特徴量の抽出について述べる．

第 4 章 本研究で提案した手法の評価実験をおこない，その有効性を示す．

第 5 章 本研究のまとめと今後の課題について述べる．

第2章

先行研究

2.1 圧縮ベースパターン認識

本章では，既存の圧縮ベースパターン認識手法を紹介する．

2.1.1 圧縮後のファイルサイズを利用する手法

Normalized Compression Distance(NCD)

NCD [(2)] は 2 つのオブジェクト x, y 間の距離を圧縮後のファイルサイズから算出する手法である．x, y 間の距離 NCD(x, y) は式 2·1 のように定義される．

$$\mathrm{NCD}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \tag{2·1}$$

ここで $C(x)$ はオブジェクト x を圧縮したときのファイルサイズ，また $C(xy)$ はオブジェクト x, y を連結した xy を圧縮したときのファイルサイズを表す．x, y が類似しているほど xy がよりコンパクトに圧縮できるので，NCD(x, y) は小さくなる．

PRDC

PRDC [(5)] では，圧縮率ベクトルと呼ばれる特徴量によりデータを表現して，分類や類似検索を実現する．PRDC はデータを特徴ベクトルとして扱うので，NCD 等の距離尺度と比べると，k-means 法や SVM(Support Vector Machine) などベクトルベースのパターン認識手法の利用に適する．

PRDC では，N 個の基底辞書集合 $B = \{D_1, D_2, \ldots, D_N\}$ を最初に定める．そして，オブジェクト x をそれぞれの辞書で圧縮した N 次元の圧縮率ベクトルとし

て x を表現する．圧縮率ベクトル $\boldsymbol{p}(B,x)$ はつぎのように定義される．

$$\boldsymbol{p}(D,x)=\left(\frac{l'_1}{l},\frac{l'_2}{l},\ldots,\frac{l'_N}{l}\right). \tag{2·2}$$

長さ l の入力データ x が与えられた時，x をそれぞれの基底辞書により圧縮すると，出力長 $l'_1,l'_2,\ldots,l'_N$ が得られる．それぞれの出力長を元のデータ長 l で割った圧縮率を並べたのが圧縮率ベクトルとなる．ここで出力長とは圧縮後のファイルサイズに他ならず，PRDC も圧縮後のファイルサイズを利用する手法と位置づけられる．

基底辞書集合 $B=\{D_1,D_2,\ldots,D_N\}$ は，データベースから選択した N 個のオブジェクト群 $\{T_1,T_2,\ldots,T_N\}$ を実際に圧縮して生成する．どのオブジェクトを利用するかはパターン認識性能に影響を与える因子である．(5) では，以下のような基底辞書集合の生成方法を提唱している．まず，N 個のオブジェクトをランダムに選んで暫定的な圧縮特徴空間を構築する．そして，暫定特徴空間上でデータベース内のオブジェクトをクラスタリングし，クラスタ代表となった N 個のオブジェクトから最終的な基底辞書集合を生成する．

2.1.2　辞書に基づく非類似度計算

2.1 章で述べた NCD では類似度を算出する際に毎回データを圧縮するために，類似度計算の計算量が大きい．そこで，予めデータを圧縮し，その過程で構成された圧縮辞書間で距離 (非類似度) を計算する手法が提案された．圧縮アルゴリズムとしては LZW が利用される．

LZW による辞書構築の流れを図 2–1 に示す．LZW アルゴリズムによる辞書の構築は，次の手順で行う．辞書には，長さ 1 の全ての文字が単語として登録されているものとする．

1. 辞書から，入力文字列に存在する最も長い単語 W を探す．
2. 入力文字列中の単語 W を W の符号に置き換える．
3. 辞書に，入力文字列中の W とその次の 1 文字を結合した単語を登録する．
4. 入力文字列中に符号に置き換えられていない文字が存在すれば，1 に戻る．

辞書による非類似度の例としては，Normalized Dictionary Distance (NDD) (4) がある．NDD では，辞書を登録された単語の集合として取扱う．$D(x)$ をオブジェ

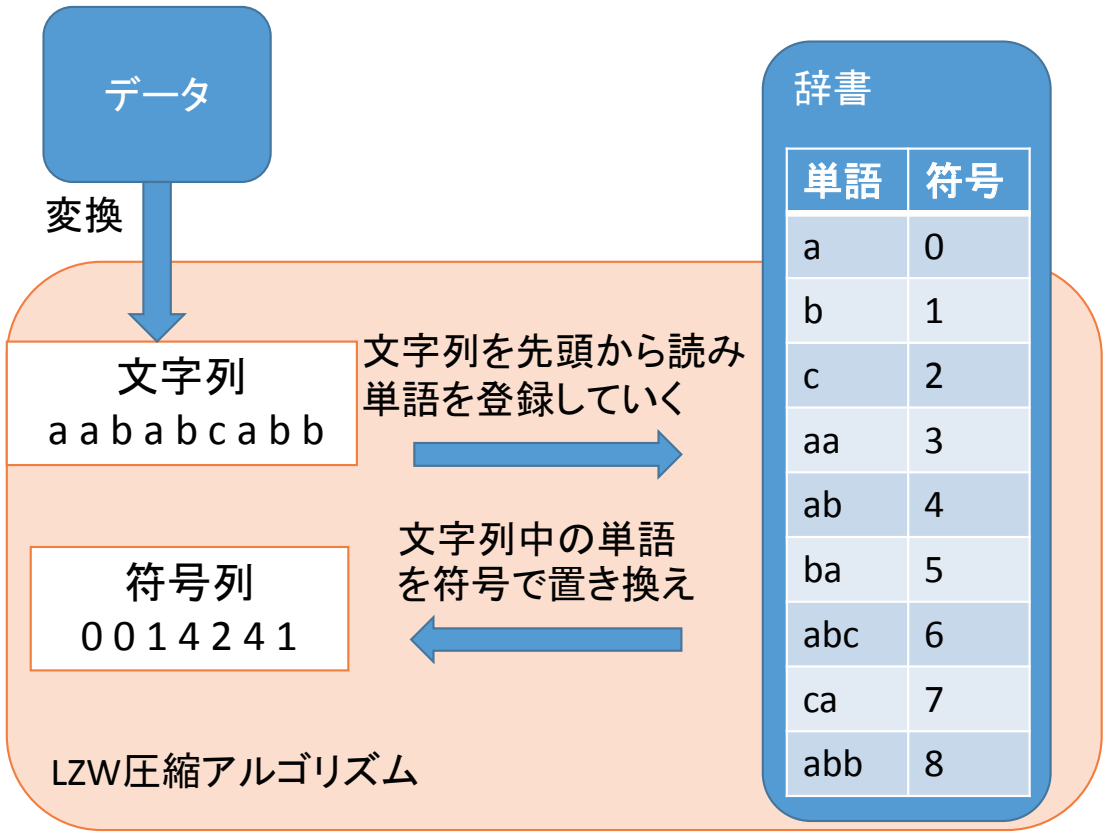

図 2–1: 辞書構築の流れ

クト x に対する圧縮辞書とする．x, y 間の距離 NDD(x, y) は式 (2·3) のように定義される．

$$\mathrm{NDD}(x, y) == \frac{\cup(D(x), D(y)) - \min\{|D(x)|, |D(y)|\}}{\max\{|D(x)|, |D(y)|\}} \tag{2·3}$$

$|D(x)|$ は $D(x)$ に含まれる単語数，$\cup(D(x), D(y))$ は $D(x)$ と $D(y)$ の和集合である．2 つの辞書が共通単語を多く含むほど $\cup(D(x), D(y))$ の要素数が小さくなり，辞書間距離は小さくなる．

辞書間距離の本質は，辞書を元データの要約として用いることで計算量を削減することにある．その一方で，辞書は元データから情報を捨てており，この点が辞書間距離の欠点である．そこで，Besiris らは辞書 $D(x)$ 内の各単語が元データ x 内に出現する回数を考慮した辞書間距離 Normalized Multiset Distance (NMD) [(1)] を提案した．NMD では，単語の出現回数を考慮した単語の多重集合間で距離計算をする．

文字列 x の辞書 $D(x) = \{w_1, w_2, ..., w_n\}$ を順序不同の n 個の単語の集合、w_i を i 番目に抽出された単語としたとき、その多重集合 $MS(x)$ は次のように定義される。

$$\begin{aligned} MS(x) &= \{D(x), m^x\} \\ &= \{(w_1, m_1^x), (w_2, m_2^x), ..., (w_n, m_n^x)\} \end{aligned} \tag{2·4}$$

ここで m_i^x は i 番目の単語 w_i の出現回数であり、m^x は x 中の単語出現回数を並べたリストである．$MS(x)$ の要素数 $|MS(x)|$ は次のように定義される．

$$|MS(x)| = \sum_{i=1}^{n} m_i^x \tag{2·5}$$

x, y 間の距離 NMD(x, y) は式 (2·6) で計算される．

$$\frac{|MS(x) \cup MS(y)| - \min(|MS(x)|, |MS(y)|)}{\max(|MS(x)|, |MS(y)|)} \tag{2·6}$$

式 (2·6) では, 2 つの多重集合 $MS(x) = \{D(x), m_x\}$, $MS(y) = \{D(y), m_y\}$ 間で非類似度を計算する式になっている．$MS(x)$ と $MS(y)$ の和集合 $MS(z) = \{D(z), m_z\} = MS(x) \cup MS(y)$ は

$$D(z) = D(x) \cup D(y) \tag{2·7}$$

$$m_z(w_i) = \max(m_x(w_i), m_y(w_i)) \tag{2·8}$$

として定義される。

[(1)] は NMD は NDD よりもパターン認識精度を向上させることを示した．

第3章

提案手法

3.1 圧縮後のファイルからの特徴抽出

本章では，圧縮後のファイルから特徴抽出を行う手法である PRDC に着目する．そして，圧縮後のファイルから別の特徴を新たに抽出することで PRDC を改良する．

本章ではまず，PRDC が使用する圧縮率は，テキスト内の単語頻度のみで決定され異なる単語間の関係を一切利用していない特徴であることを指摘する．そして，隣接する単語関係を表す新しい特徴量を提案する．その後で新しい特徴量を PRDC に組み込んだ手法を提案する．

3.1.1 圧縮率

PRDC が利用する圧縮率について考察する．PRDC では LZW アルゴリズムを用いて，基底辞書 D に単語とそれに対応する符号 (出力記号) のペアを登録する．符号の長さは元の単語長よりも小さい．そして，圧縮時には入力テキスト T を前方からスキャンし，T 内の単語を対応する符号に順次置き換えることで圧縮を実現する．一般に T を辞書 D を使って圧縮した時の圧縮率は

$$\begin{aligned} \text{圧縮率} &= \frac{\text{圧縮後のファイルサイズ}}{\text{元テキスト } T \text{ のサイズ}} \\ &= \frac{\text{圧縮後の符号数} * 1 \text{ 符号の bit 長}}{T \text{ の文字数} * 1 \text{ 文字の bit 長}} \\ &= \frac{\text{圧縮後の符号数}}{T \text{ の文字数}} * \alpha \end{aligned} \tag{3·1}$$

と記述できる．だたし，$\alpha = \frac{1 \text{ 符号の bit 長}}{1 \text{ 文字の bit 長}}$ を表す定数である．

ここで辞書 D に n 種類の単語 $(w_1, w_2, .., w_n)$ が登録されているとする．単語 w_i の文字数を l_i とおく．さらに T 内の単語 w_i の出現回数を m_i と記述する．この時，式 (3·1) における圧縮後の符号数は $\sum_{i=1}^{n} m_i$ である．一方，T の長さは $\sum_{i=1}^{n} m_i l_i$ になる．従って，式 (3·1) は次のように書き換えられる．

$$\begin{aligned} \text{圧縮率} &= \frac{\sum_{i=1}^{n} m_i}{\sum_{i=1}^{n} m_i l_i} * \alpha \\ &= 1 - (1 - \frac{\sum_{i=1}^{n} m_i}{\sum_{i=1}^{n} m_i l_i}) * \alpha \\ &= 1 - \frac{\sum_{i=1}^{n} m_i (l_i - 1)}{\sum_{i=1}^{n} m_i l_i} * \alpha \\ &= 1 - \frac{\sum_{i=1}^{n} m_i (l_i - 1)}{T \text{の長さ}} * \alpha \end{aligned} \tag{3·2}$$

$l_i - 1$ は w_i の符号化 1 回で削減される文字数であり，$m_i(l_i - 1)$ は，w_i の符号化により元のテキストから何文字削減できるかを表す．式 (3·2) が示すように圧縮率は各単語の出現頻度のみで決定される特徴量であり，T 内における異なる単語の関係は考慮しない．例えば，どの単語とどの単語が T 内で隣接して出現するかという情報は圧縮率には反映されない．

3.1.2　再圧縮による単語の隣接関係の抽出

前節で述べたように，基底辞書による圧縮率は T に各単語が何回含まれるかによって決定される特徴量である．しかし，T 内でどの単語が隣接して出現するかという情報を捨ててしまっている．そこで，本研究では T 内での単語の隣接関係から定まる新しい特徴量を圧縮後のファイルから抽出する．具体的には，T を辞書 D で圧縮後のファイル (符号列) を A_D^T とすると，A_D^T をもう一度再圧縮を行った時の圧縮率を特徴量として採用する．A_D^T の再圧縮では，辞書 D で圧縮するのではなく，A_D^T から構築した辞書で A_D^T 自身を圧縮する．すなわち，A_D^T の自己圧縮率を再圧縮率とする．

再圧縮率が T 内の単語の隣接関係から定まる特徴であることを例によって示す．図 3–1 は，テキスト T_1 と T_2 を再圧縮まで行った時の様子を示す．最下段が元のテキストであり，真ん中の段が辞書 D で圧縮後のファイル，最上段が再圧縮後の出力ファイルである．なお，再圧縮時には自己圧縮を行うため，T_1 の再圧縮と T_2 の再圧縮では異なる辞書が用いられている．

辞書 D で T_1, T_2 を圧縮後の符号化列はそれぞれ”ABCABC”，”ACBBAC” と異

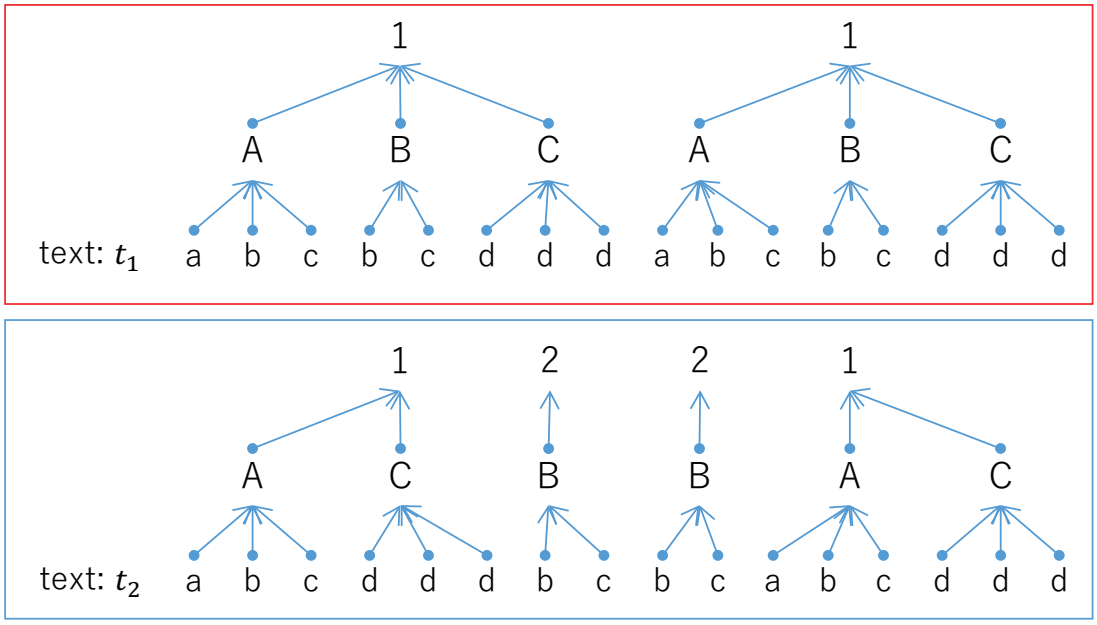

図 3–1: 圧縮後のファイルの再圧縮

なっている．しかし，圧縮率はどちらも $\frac{6}{16} = 0.375$ となって T_1 と T_2 は区別できない．これは符号語'A','B','C' の出現回数が同じであることが理由であり，3.1 節で述べたように単語の出現回数のみで決定される圧縮率の限界である．

一方，T_1 の再圧縮率は $\frac{2}{6}$，T_2 の再圧縮率は $\frac{4}{6}$ と異なる値になり，T_1 と T_2 を区別できる．この違いは T_1 と T_2 の単語順序の違いに起因する．例えば，T_1 の再圧縮率は符号語列'ABC' が T_1 内に 2 回出現した事実を反映する．このように再圧縮率は単語の隣接関係から定まる特徴である．

3.1.3　再圧縮率を利用した PRDC

本節では，前節で提案した再圧縮率を PRDC に組み込んだ手法を提案する．従来の PRDC では，テキスト x を N 個の基底辞書集合 $\{D_1, D_2, \ldots, D_N\}$ で圧縮し，その圧縮率を並べた N 次元特徴ベクトル (式 (2·2) で x を表現する．一方，提案手法では通常の圧縮率と再圧縮率を合わせた特徴ベクトルを構築する．基底辞書集合 $\{D_1, D_2, \ldots, D_N\}$ を用いた，データ x に対する再圧縮率を加えた圧縮率ベクト

ルは $\boldsymbol{rp}(D,x)$ は次のように定義される．

$$\boldsymbol{rp}(D,x) = \left(\frac{l'_1}{l}, \frac{l''_1}{l'_1}, \frac{l'_2}{l}, \frac{l''_2}{l'_2}, \ldots, \frac{l'_N}{l}, \frac{l''_N}{l'_N}\right) \tag{3·3}$$

- l:入力データ長
- l'_i: x を D_i で圧縮したときの出力長
- l''_i: x を D_i で圧縮した符号列を，再圧縮したときの出力長

である．1 つの基底辞書が 2 つの次元に対応するので，圧縮率ベクトルの次元数は $2N$ になる．この圧縮率ベクトルは以下の特徴を有する．

1. 通常の圧縮率により単語の頻度を考慮する．
2. 再圧縮率により単語の隣接関係を考慮する．

提案手法では単語の頻度だけでなく単語の隣接関係まで考慮したパターン認識を可能にする．この性質より，提案手法を HOPRDC (High-Order PRDC, 高階 PRDC) と名付ける．

3.2　辞書からの新しい特徴抽出

本章では，圧縮辞書から特徴抽出を行う手法である NMD を取り上げ，圧縮辞書からの新しい特徴抽出による NMD の改良手法を説明する．

3.2.1　NMD の課題

NDD や NMD などの辞書間距離では，辞書を元データの要約として用いることで，類似度計算の計算量を削減する．その一方で，元データから情報を捨てることが欠点である．NMD では各単語の出現回数を考慮し，NDD よりも捨てられる情報を削減した．しかし，NMD では辞書に登録された単語長を無視し，全単語は長さによらず均等の重みを持つものとして式 (2·6) の定義に従って距離計算を実施する．

単語長を無視することがパターン認識に与える影響を考えてみよう．図 3–2 左の画像は，上半分が単純な領域で，下半分が複雑な領域になっている．このような画像の現実例としては上半分が海で下半分が港湾都市であるケースが挙げられ

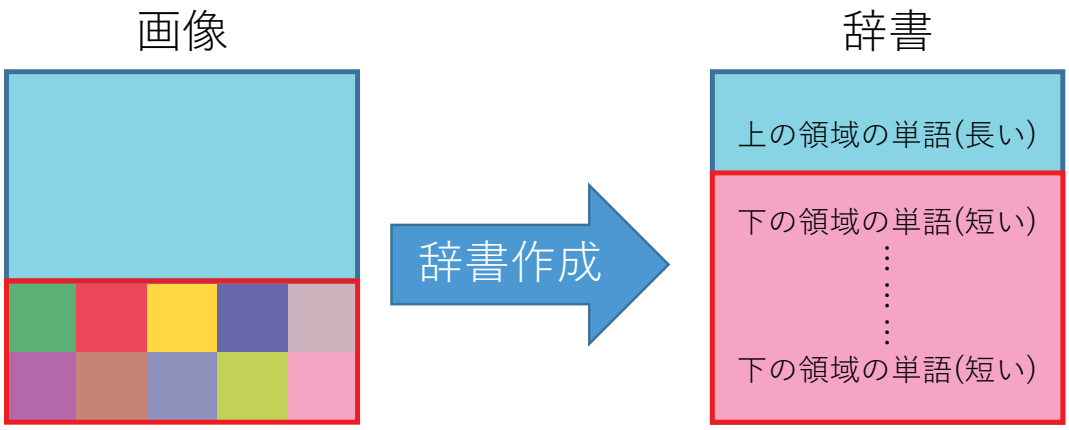

図 3–2: 単純な領域と複雑な領域に対する辞書内での単語の割合

る．この画像から辞書を作ると，上領域からは長い単語が少数生成され，下領域からは短い単語が多数生成される (図 3–2 右)．この時，全単語を均等に取り扱うということは，少数の単語を生成する上半分の領域を軽視することを意味する．距離計算においても上半分の領域が軽視されるので，類似検索の際，上半分が全く同じデータでも類似していると判定されないという問題が起こり得る．先述した現実例では，海を含む画像は海を含むという理由では類似していると判定されにくくなるが，これは人間の直観に反する．

上記をまとめると，NMD のように全単語を均等に取り扱う手法では，元データにおいて単語が占める領域の割合と，各単語が類似度計算に与える影響の割合が乖離する．

3.2.2　重み付き NMD

前節で述べた課題に対して，本研究では単語ごとに重みを付けて類似度を計算する重み付き辞書間距離 WNMD (weighted NMD) を提案する．x, y 間の距離 $WNMD(x, y)$ は次のように定義される．

$$\frac{G(MS(x) \cup MS(y)) - \min(G(MS(x)), G(MS(y)))}{\max(G(MS(x)), G(MS(y)))}$$

$$G(MS(x)) = \sum_{i=1}^{n} g(w_i\text{の長さ}) \times m_i^x \tag{3·4}$$

ここで，$g(w_i\text{の長さ})$ は w_i の長さによって決定される単語の重みである．

関数 g は単語の長さに関して単調増加することが望ましい．本論文では n 文字の単語の重み $g(n)$ を，同一文字が n 回連続した最も単純な n 文字の文字列に対する最小記述長から定める．

文字列 $T = \underbrace{\mathrm{aa}\cdots\mathrm{a}}_{n}$ は，a が $\sqrt{n}$ 回連続した文字列 $A = \underbrace{\mathrm{aa}\cdots\mathrm{a}}_{\sqrt{n}}$ を符号語 (モデル) として，$\underbrace{\mathrm{AA}\cdots\mathrm{A}}_{\sqrt{n}}$ と表記した時に，

$$\text{記述長} = (\text{モデル } A \text{ の長さ}) + (A \text{ による } T \text{ の符号列長})$$

が最小になる．この時の記述長は $O(\sqrt{n})$ になることから，$g(n)$ を式 (3·5) とした．

$$g(n) = \sqrt{n}. \tag{3·5}$$

第4章

実験

2つの提案手法を実験により評価する．提案手法と既存手法を比較することで性能評価を実施する．

4.1 データセット

データセットには先行研究 (1) と同じ Corel image repository (3) の画像を用いる．このデータセットは [256 × 384] か [384 × 256] の画像1000枚からなり，100枚づつ10個のクラスに分けられている．その一部を図 4–1 に示す. 画像は2次元データなので，圧縮パターン認識に適用するには，1次元のテキストに変換する必要がある．そこで，既存研究 (1) と同様に，各画素の rgb 値をそれぞれ5段階に量子化して，125種類の文字で表現し，これらを水平スキャンで連結してテキストに変換した．

4.2 HOPRDC の評価結果

PRDC と HOPRDC を比較するためクラス分類実験を行う．データセットからランダムに10枚画像を選択し，各画像から1つずつ，合計10個の基底辞書を作成する．その後ランダムに890枚選択した画像を学習画像とし，残りの画像100枚をクエリとして用いる．クエリ画像の分類は次の手順で行う．

1. クエリ画像を圧縮率ベクトルに変換する
2. クエリ画像を k-NN 法でクラス分類する．クエリと最近の k 枚の学習画像を求め，その中で枚数が最も多いクラスに分類する．

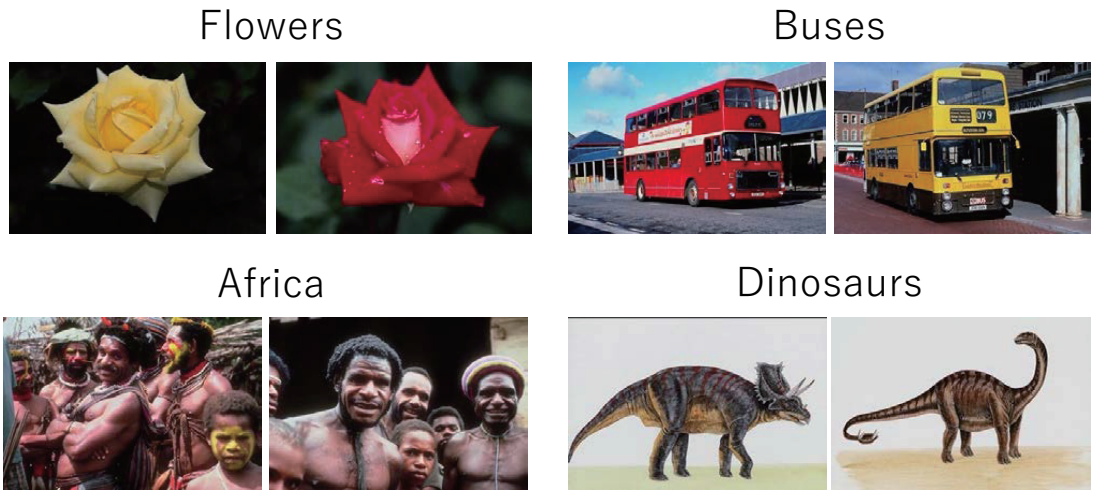

図 4–1: Corel データセットの一部

表 4–1: 全画像に対する判別率

PRDC	0.6393
HOPRDC	0.6798

今回の実験では k=5 とした．判別率 r は次のように計算する．

$$r = \frac{\text{正しく分類されたテスト画像数}}{\text{テスト画像数}} \tag{4·1}$$

本実験では基底辞書をランダムに選択するため，300 回の試行を行いその平均値を報告する．

表 4–1 は前テスト画像に対する判別率である．HOPRDC は PRDC に比べて 4% 程度判別率が向上した．図 4–2 はクラスごとの判別率である．提案手法では，Bus クラスや Flower クラスにおいて PRDC を大きく上回った．この 2 つのクラスのインスタンスには，同じ形で色が違うものが含まれる．PRDC では，単語の頻度のみを考慮しているため，オブジェクトの色に認識結果が大きく左右される．一方，提案手法では，再圧縮率により単語の隣接関係を考慮するため，オブジェクトの形も認識結果に影響を与える．これにより，これらのクラスで提案手法の精度が向上した．

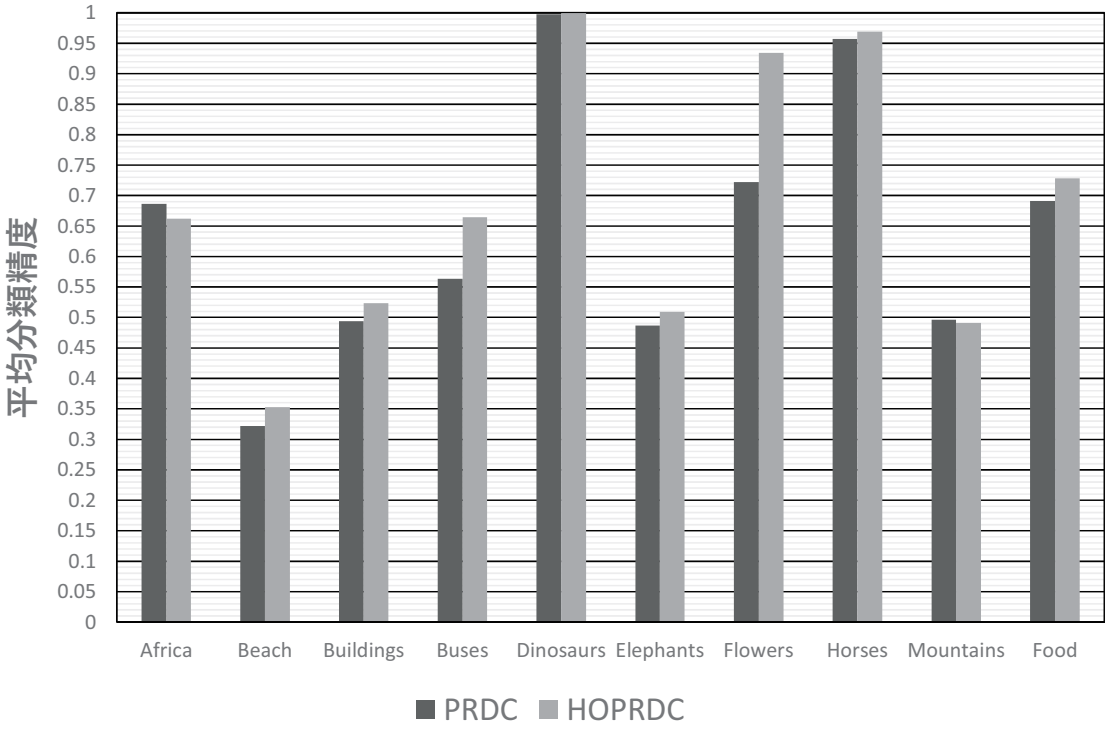

図 4–2: クラス毎の判別率（クエリの数は 100 枚，基底辞書の個数は 10 個，試行回数は 300 回）

4.3　WNMD の評価結果

NMD と WNMD を比較するため，類似画像検索を行う．データベースには 1000 枚の画像全てを登録する．クエリはこの中の 1 枚を使用する．クエリとデータベースの全画像間で類似度計算を行い，類似度が高い上位 100 枚の画像を検索結果とする．クエリ画像 Q に対する検索結果は式 (4·2) に示す適合率で評価する．分母の 100 は検索結果の画像数であるが，1 クラスあたりの画像数とも一致する．

$$P(Q) = \frac{N_r}{100} \tag{4·2}$$

ここで N_r は検索結果の中でクエリ画像 Q と同じクラスの画像の枚数である．全ての画像をクエリとして使い，1000 回類似画像検索を行った平均適合率によりアルゴリズムの性能を評価する．

表 4–2 に全画像に対する平均適合率を示す．WNMD は NMD と比べて 1.6%程度平均適合率が向上した．図 4–3 に NMD と WNMD のクラスごとの平均適合率を示す．. 提案手法では Africa クラスや，Dinosaurs クラスにおいて NMD を大き

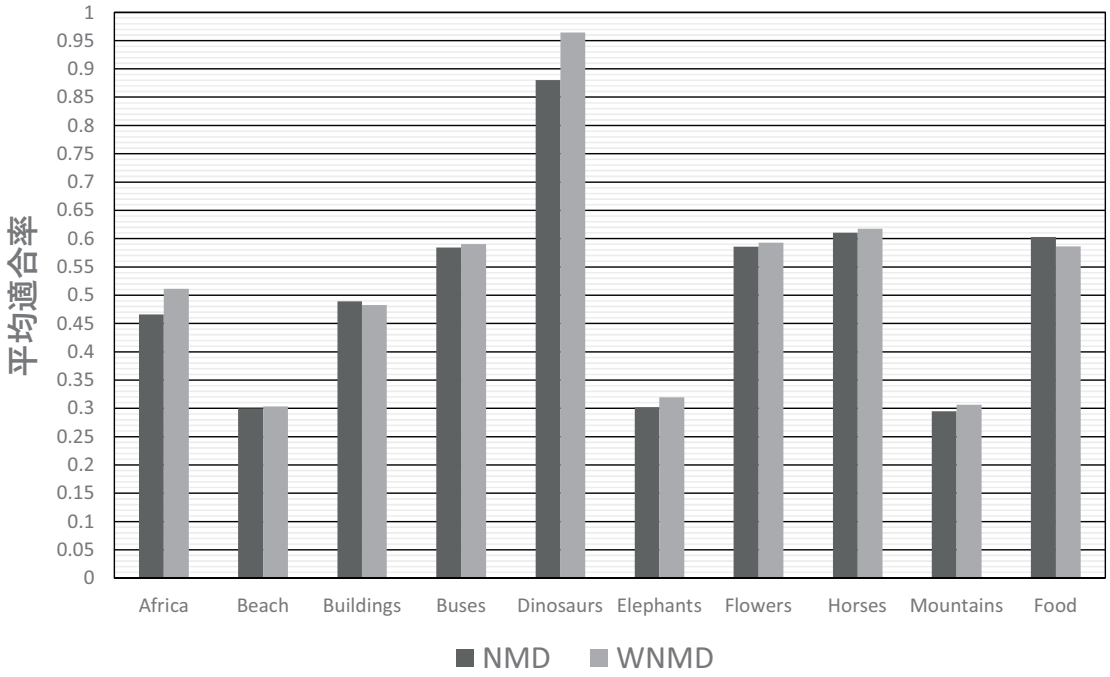

図 4–3: NMD と提案手法の平均適合率の比較（クエリはデータセット中の全ての画像）

表 4–2: 全画像に対する平均適合率

NMD	0.5086
WNMD	0.5249

く上回った．この 2 つのクラスでは，同色で単純な領域が広い面積を占める画像が多い．Africa クラスには人物を映した画像が多く，肌色の単純な領域が大きい．Dinosours クラスでは，背景や恐竜の肌が少ない色で表現されており，単純である．NMD では単純領域の特徴を軽視するのに対し，WNMD では，単語に長さに応じた重みをつけることで，これらのクラスで重要な特徴となる単純な領域を重視して類似計算を行うことができ，精度が向上した．

第5章

まとめ

本稿では, 圧縮ベースパターン認識手法である PRDC と NMD を取り上げ, PRDC に対しては圧縮後のファイルから，NMD に関しては圧縮辞書からそれぞれ新しい特徴を抽出することで，認識精度を向上できることを示した．

本論文では，まず PRDC が使用する圧縮率はテキスト内の単語頻度のみで決定され，異なる単語間の関係を一切利用していない特徴である事を指摘した．そして本研究では，単語の隣接関係を考慮するため，基底辞書で圧縮したテキストをもう一度圧縮することで得られる再圧縮率を特徴量として利用した．その結果，判別率は 4%程度向上し，特に色が異なるが形が同じオブジェクトを多く含むクラスで認識精度の向上を確認した．

NMD のような辞書間距離では，辞書を元データの要約として用いることで，類似度計算の計算量を削減する．その一方で，元データから情報を捨てることが欠点である．NMD では，全単語を長さによらず均等の重みを持つものとして距離計算を実施するため，単語の長さ情報を捨てる．本研究では，単語帳により重みを付け類似度を計算する辞書間距離，WMND を提案した．その結果，平均適合率は 1.6%程度向上し，中でも単純な領域が多いクラスに対して分類精度が向上した．

今後の課題としては，画像以外のデータセットへの適用が挙げられる．本稿で抽出した特徴量は，画像に対して有効であることを示したが，その他のデータに対しても有効であるかは，調査する必要がある．

謝辞

ありがとう

参考文献

[1] D. Besiris and E. Zigouris. Dictionary-based color image retrieval using multiset theory. *Journal of Visual Communication and Image Representation*, 24(7):1155 – 1167, 2013.

[2] R. Cilibrasi and P.M.B. Vitanyi. Clustering by compression. *IEEE Trans. Inf. Theory 51*, pages 1523–1545, 2005.

[3] Corel Corp. Corel stock photo library. Ontario, Canada.

[4] A. Macedonas, D. Besiris, G. Economou, and S. Fotopoulos. Dictionary based color image retrieval. *Journal of Visual Communication and Image Representation*, 19(7):464–470, October 2008.

[5] T. Watanabe, K. Sugawara, and H. Sugihara. A new pattern representation scheme using data compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):579–590, May 2002.

図一覧

表一覧