

博士学位論文

グラフの高精度認識に関する研究

東北大学大学院 工学研究科

電気・通信工学専攻

宮崎 智

目次

第1章 序論	1
第2章 グラフを用いたパターン認識	3
2.1 準備	3
2.1.1 グラフ	3
2.1.2 部分グラフ	4
2.1.3 同型と部分グラフ同型	5
2.1.4 差分	6
2.1.5 共通グラフ	7
2.1.6 最大共通部分グラフ	7
2.1.7 最小共通スーパーグラフ	8
2.1.8 重み付き最大共通部分グラフ	9
2.1.9 重み付き最小共通スーパーグラフ	9
2.2 グラフ間の距離	12
2.2.1 エントロピーによる距離	12
2.2.2 編集系列による距離	13
2.3 重み付き最小共通スーパーグラフを用いたシルエット画像認識	16
2.3.1 シルエット画像のグラフ	16
2.3.2 シルエット画像とカテゴリの距離	23
2.3.3 評価実験	24
2.4 まとめ	29
第3章 集合を代表するグラフ	31
3.1 はじめに	31
3.2 集合を特徴付ける属性値の抽出	32
3.3 特徴属性値と頂点の対応関係	32
3.4 代表グラフの抽出	34
3.5 評価実験	37

3.5.1	実験で用いたデータ	37
3.5.2	代表グラフの抽出実験	42
3.5.3	認識実験	45
3.5.4	比較実験	47
3.6	まとめ	48
第 4 章	グラフの確率モデル	50
4.1	はじめに	50
4.2	確率モデル	50
4.3	確率モデルの構築	51
4.3.1	確率モデルの頂点との対応関係	51
4.3.2	確率モデルの頂点 (辺) の観測確率と属性値の確率	52
4.4	確率モデルからグラフの生成確率	54
4.5	確率モデルの評価実験	55
4.6	確率モデルによる高精度認識	62
4.6.1	AdaBoost 法	62
4.6.2	決定木	63
4.6.3	確率モデルを用いた高精度識別器	64
4.6.4	評価実験	66
第 5 章	結論	67
	参考文献	70
	謝辞	74
	研究業績	75

目次

2.1	部分グラフ同型	6
2.2	グラフの差分	7
2.3	重み付き最小共通スーパーグラフを得る手続き	11
2.4	グラフ集合の重み付き最小共通スーパーグラフを得る近似アルゴリズム	11
2.5	A-star アルゴリズムによる編集距離の計算	15
2.6	長方形の中心軸	17
2.7	中心軸の例	18
2.8	ノード, エッジの例	19
2.9	中心軸の画素に頂点, 辺を割り当てた例	22
2.10	ノイズを除去した例	22
2.11	シルエット画像, teddy, Misk, carriage, octopus	25
2.12	シルエット画像, fork, dog, sea_snake	26
2.13	学習データ数と認識率の関係	28
2.14	パラメータ a と RWMCS における認識率の変化	30
2.15	パラメータ a と RWMCS _t における認識率の変化	30
3.1	集合を代表するグラフを抽出するアルゴリズム	37
3.2	IAM Graph Database; Letter, 変形の強さは Low	39
3.3	IAM Graph Database; Letter, 変形の強さは Medium	39
3.4	IAM Graph Database; Letter, 変形の強さは High	40
3.5	飾り文字画像	40
3.6	大局的な構造を表す画像	41
3.7	飾り文字画像から抽出されたグラフ	41
3.8	代表グラフ, IAM Graph Database; Letter, Low	42
3.9	代表グラフ, IAM Graph Database; Letter, Medium	43
3.10	代表グラフ, IAM Graph Database; Letter, High	43
3.11	飾り文字画像から得た代表のグラフ	44

3.12 飾り文字画像における認識結果	46
3.13 IAM Graph Database における認識結果	46
3.14 認識率の比較. 横軸はデータセットの名前	49
3.15 計算時間の比較. 横軸はデータセットの名前	49
4.1 確率モデルの頂点集合と, 頂点集合と確率モデルの頂点集合との対応関係 を得る手順	52
4.2 グラフの頂点集合から確率モデルの頂点集合との対応関係を得る手順 . . .	53
4.3 確率モデルによる認識結果	56
4.4 確率モデルの性能と学習データ数の関係	57
4.5 認識された回数ごとのテストデータの割合. 学習データ数 10	59
4.6 認識された回数ごとのテストデータの割合. 学習データ数 20	59
4.7 認識された回数ごとのテストデータの割合. 学習データ数 30	60
4.8 認識された回数ごとのテストデータの割合. 学習データ数 40	60
4.9 確率モデルと代表グラフの比較	61
4.10 AdaBoost による強識別器の学習	63
4.11 強識別器による認識結果	66

表 目 次

2.1	正しいカテゴリに認識された数, 平均値と最高値, 最低値	27
2.2	RWMCS に対して最高値・最低値を得たデータセットにおける比較手法と の比較	28
4.1	データ形式	63

第1章

序論

グラフは、最も基本的なレベルにおいてある存在の集合とそれらを接続する集合によって定義される。その定義は普遍的であるため、幅広い分野のアプリケーションで、物体や対象となる概念の関係性を表現するために用いられている。例えば、ソフトウェア工学の分野において、システムの構成要素の独立性や相互関係を視覚化するために用いられる。関係データベースにおいてデータは複数のデータと関係（リレーション）を持つ。意思決定のプロセスであるフローチャートやダイアグラムといった概念もグラフを用いて表現することができる。システムの状態や条件分岐をノード、遷移する規則をエッジと定義することができる。World Wide Web もグラフの一例である。コンピュータネットワーク上に存在するウェブサイトは複雑なリンクを形成している。ネットワーク上の人間関係を表すソーシャルグラフも同様である。

グラフを用いて物体を表現する手法は、1970年代後半からパターン認識の分野で注目されている [2]。パターン認識は、観測されたパターンをあらかじめ定められたクラスの問題 (クラス) のひとつに対応させることによりパターンを識別する処理である。グラフ表現を用いるアプリケーションにおいて、グラフのパターン認識やクラスタリングは重要なタスクである。

近年、グラフのパターン認識に関する研究が盛んに行われている。 [3, 6, 14–17, 20, 27, 28, 30]。簡単にこれらの文献について述べると、文献 [6, 14, 15] はグラフから同一のグラフや部分グラフを検出する手法について述べられている。これらを発展させ、部分グラ

第1章 序論

フを用いることで文献 [6] はグラフ間の類似度を計測している。その他、グラフを他グラフへと変換する編集系列によるコストにより距離を計測する手法 [16,17] や、グラフのスペクトルを用いた手法 [27]、情報理論におけるエントロピーを用いた手法 [20] などがある。ここまで、グラフの距離を計測することに主眼を置いている文献について述べたが、文献 [3,28,30] はクラスを代表するグラフを学習することを主題としている。

本論文はクラスを表すグラフとモデルを用いて、グラフ認識する手法を提案する。提案したモデルを用いることにより、グラフを高精度に識別することができることを示す。

本論文は全5章で構成され、その内容は以下の通りである。第1章の序論において本研究の背景と目的について述べた。第2章では、グラフを用いてシルエット画像を認識する手法について述べる。第3章において、クラスを代表するグラフを抽出する手法を提案する。第4章では、確率的なモデルを構築する手法を提案し、提案したモデルを用いてグラフを高精度に識別することができることを示す。最後に第5章で本研究の成果と今後の課題についてまとめる。

第2章

グラフを用いたパターン認識

本章ではグラフの基本的な定義と概念を与え、グラフ間の距離を計測する手法について述べる。そしてグラフの概念を用い、シルエット画像のカテゴリを認識できることを示す。

2.1 準備

2.1.1 グラフ

グラフの基本的な定義と概念について述べる。 V を有限集合、 E を V のふたつの元からなる集合、つまり $E \subseteq V \times V$ である集合とすると、グラフは V と E の組により定義され、式(2.1)により表される。

$$G = (V, E) \quad (2.1)$$

V の元をグラフの頂点、 E の元をグラフの辺といい、 V を頂点集合、 E を辺集合という。

グラフの頂点集合の元の個数をグラフの位数といい、 $|G|$ と表記する。同様に辺集合の元の個数を辺数といい、 $\|G\|$ と表記する。グラフ G の頂点集合 V を $\mathcal{V}(G)$ と表記することにより参照する。同様に辺集合 E を $\mathcal{E}(G)$ により参照する。グラフのある頂点 v に対し、 v と接続する頂点の集合を v の近傍 (neighborhood) といい $N(v)$ と表記する。 $N(v)$ の頂点数を v の次数 (degree) といい、 $\deg(v)$ と表記する。

グラフを用いて表現するパターンに応じて、頂点集合や辺集合に情報を与えたい場合がある。頂点集合と辺集合に加える情報の例として、属性や重みを加えたグラフが考えられることから、以下に属性グラフ、重み付き属性グラフ、重み付きグラフを定義する。

第2章 グラフを用いたパターン認識

L_v をある頂点を取り得る属性の集合とし, L_e をある辺を取り得る属性の集合, つまり頂点間の関係を表す属性の集合とする. グラフ G の頂点集合から属性集合の写像 $\alpha: \mathcal{V}(G) \rightarrow L_v$, 辺集合から属性集合への写像 $\beta: \mathcal{E}(G) \rightarrow L_e$, が存在するとき, グラフ G を属性グラフといい, $H = (\mathbf{V}, \mathbf{E})$ を用いて, 式 (2.2) で定義する. 属性グラフは物体の構造を簡潔に記述することができる [13].

$$\begin{aligned} G &= (H, \alpha, \beta) \\ G &= (\mathbf{V}, \mathbf{E}, \alpha, \beta) \end{aligned} \quad (2.2)$$

属性グラフ G を拡張し, 重み付き属性グラフをも定義する. λ を頂点集合から実数への写像, $\lambda: \mathcal{V}(G) \rightarrow \mathbb{R}$, ϵ を辺集合から実数への写像, $\epsilon: \mathcal{E}(G) \rightarrow \mathbb{R}$ とすると, 重み付き属性グラフ G は属性グラフ $H = (\mathbf{V}, \mathbf{E}, \alpha, \beta)$ を用いて, 式 (2.3) で定義される.

$$\begin{aligned} G &= (H, \lambda, \epsilon) \\ G &= (\mathbf{V}, \mathbf{E}, \alpha, \beta, \lambda, \epsilon) \end{aligned} \quad (2.3)$$

頂点集合と辺集合, λ, ϵ の組から成るグラフを重み付きグラフといい, 重み付きグラフの定義を式 (2.4) により与える.

$$G = (\mathbf{V}, \mathbf{E}, \lambda, \epsilon) \quad (2.4)$$

2.1.2 部分グラフ

ふたつのグラフ G, G' について, $\mathcal{V}(G') \subseteq \mathcal{V}(G)$ かつ $\mathcal{E}(G') \subseteq \mathcal{E}(G)$ であるとき, G' は G の部分グラフであるといい, $G' \subseteq G$ と表記する. 属性グラフ, 重み付き属性グラフについての部分グラフは, 頂点集合と辺集合の部分集合と, それらから属性への写像, 重みへの写像の組として得ることができる. 属性グラフ $G = (\mathbf{V}, \mathbf{E}, \alpha, \beta)$ の部分グラフ $G_s = (\mathbf{V}_s, \mathbf{E}_s, \alpha_s, \beta_s)$ は,

$$\begin{aligned} \mathbf{V}_s &\subseteq \mathbf{V}, \\ \mathbf{E}_s &\subseteq \mathbf{E}, \\ \alpha_s(v) &= \begin{cases} \alpha(v) & \text{if } v \in \mathbf{V}_s, \\ \phi & \text{otherwise,} \end{cases} \\ \beta_s(e) &= \begin{cases} \beta(e) & \text{if } e \in \mathbf{E}_s, \\ \phi & \text{otherwise,} \end{cases} \end{aligned}$$

第2章 グラフを用いたパターン認識

となる. ϕ は定義なしである. 同様に重み付き属性グラフ $G = (\mathbf{V}, \mathbf{E}, \alpha, \beta, \lambda, \epsilon)$ の部分グラフ $G_s = (\mathbf{V}_s, \mathbf{E}_s, \alpha_s, \beta_s, \lambda_s, \epsilon_s)$ は以下のように定めることができる.

$$\begin{aligned}\mathbf{V}_s &\subseteq \mathbf{V}, \\ \mathbf{E}_s &\subseteq \mathbf{E}, \\ \alpha_s(v) &= \begin{cases} \alpha(v) & \text{if } v \in \mathbf{V}_s, \\ \phi & \text{otherwise,} \end{cases} \\ \beta_s(e) &= \begin{cases} \beta(e) & \text{if } e \in \mathbf{E}_s, \\ \phi & \text{otherwise,} \end{cases} \\ \lambda_s(v) &= \begin{cases} \lambda(v) & \text{if } v \in \mathbf{V}_s, \\ \phi & \text{otherwise,} \end{cases} \\ \epsilon_s(e) &= \begin{cases} \epsilon(e) & \text{if } e \in \mathbf{E}_s, \\ \phi & \text{otherwise.} \end{cases}\end{aligned}$$

2.1.3 同型と部分グラフ同型

ふたつのグラフ G の頂点集合から H の頂点集合への全単射 $f: \mathcal{V}(G) \rightarrow \mathcal{V}(H)$ が存在して, かつ,

$$(v_i, v_j) \in \mathcal{E}(G) \Leftrightarrow (f(v_i), f(v_j)) \in \mathcal{E}(H) \quad (2.5)$$

が成り立つとき, G と H は同型であるといい $G \simeq H$ と表記し, f を G から H への同型写像という. 同様に G と H の部分グラフ H_s があり, $G \simeq H_s$ であるとき, G は H に対して部分グラフ同型といい, G から H への部分グラフ同型写像が存在する. 部分グラフ同型の例を図 2.1 に示す. オレンジの破線は同型写像である. 図 2.1 において g は \hat{g} の部分グラフで, g は \hat{g} に対して部分グラフ同型である.

属性グラフや重み付き属性グラフにおける同型, 部分グラフ同型は頂点集合と辺集合が式 (2.5) の定義を満たせばよいものとし, 属性値や重みが同一である必要はないものとする. つまり, ふたつの属性グラフ $G_1 = (\mathbf{V}_1, \mathbf{E}_1, \alpha_1, \beta_1)$ と $G_2 = (\mathbf{V}_2, \mathbf{E}_2, \alpha_2, \beta_2)$ が同型であるとは, 全単射の写像 $f: \mathcal{V}(G) \rightarrow \mathcal{V}(H)$ が存在し, 式 (2.5) を満たすならば属性グラフ G_1 と G_2 は同型であるとする. また,

$$\alpha_1(v) = \alpha_2(f(v)) \quad (2.6)$$

$$\beta_1((v, v')) = \beta_2((f(v), f(v'))) \quad (2.7)$$

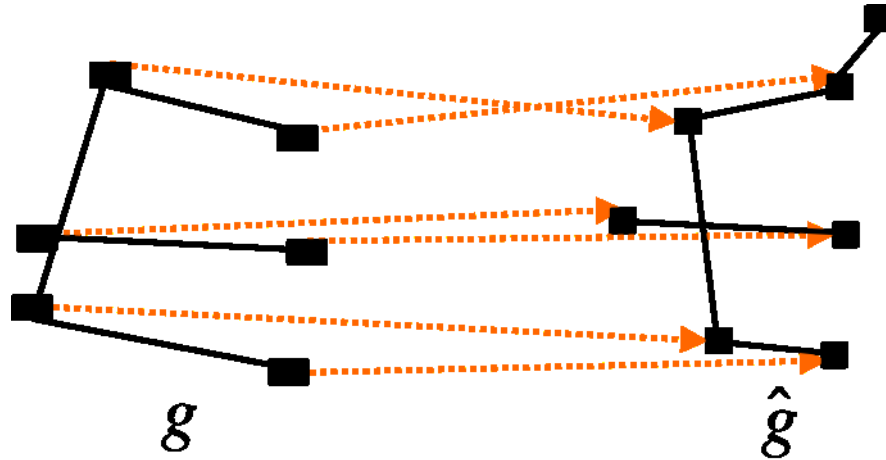


図 2.1: 部分グラフ同型

である必要はないものとする. ただし, $v, v' \in \mathcal{V}(G_1)$ である. 式 (2.6), 式 (2.7) を満たす場合, 属性グラフ G_1 と G_2 は完全同型という. ふたつの重み付き属性グラフについても同様に重みが一致する場合を完全同型であるという.

2.1.4 差分

グラフ G の部分グラフを G_s について, $G - G_s$ をグラフの差分といい, G から G_s を取り除いたグラフを表す. $(v, v') \in \mathcal{E}(G)$ について, $v \in \mathcal{V}(G) - \mathcal{V}(G_s)$ と $v' \in \mathcal{V}(G_s)$ の間を接続する辺の集合を $\text{emb}(G, G_s)$ とし, G から G_s を取り除くときは $\text{emb}(G, G_s)$ も取り除く. ふたつの重み付属性グラフ $G_s \subseteq G$ の差分は

$$\begin{aligned}
 G - G_s &= (\mathcal{V}(G - G_s), \mathcal{E}(G - G_s)) & (2.8) \\
 \mathcal{V}(G - G_s) &= \mathcal{V}(G) \setminus \mathcal{V}(G_s) \\
 \mathcal{E}(G - G_s) &= \mathcal{E}(G) \setminus \mathcal{E}(G_s) \setminus \text{emb}(G, G_s)
 \end{aligned}$$

差分の例を図2.2に示す. 図2.2において, G と部分グラフ $G_s \subseteq G$ の頂点集合はそれぞれ $\mathcal{V}(G) = \{1, 2, 3, 4, 5, 6\}$, $\mathcal{V}(G_s) = \{3, 4, 6\}$ で, 辺集合は $\mathcal{E}(G) = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 5\}, \{3, 6\}, \{4, 6\}\}$, $\mathcal{E}(G_s) = \{\{3, 6\}, \{4, 6\}\}$ である. $G - G_s$ の頂点集合は $\mathcal{V}(G - G_s) = \mathcal{V}(G) \setminus \mathcal{V}(G_s) = \{1, 2, 5\}$, 辺集合は $\mathcal{E}(G - G_s) = \mathcal{E}(G) \setminus \mathcal{E}(G_s) \setminus \text{emb}(G, G_s) = \{\{1, 2\}, \{2, 5\}\}$ となる. ここで $\text{emb}(G, G_s) = \{\{1, 3\}, \{2, 4\}\}$ である.

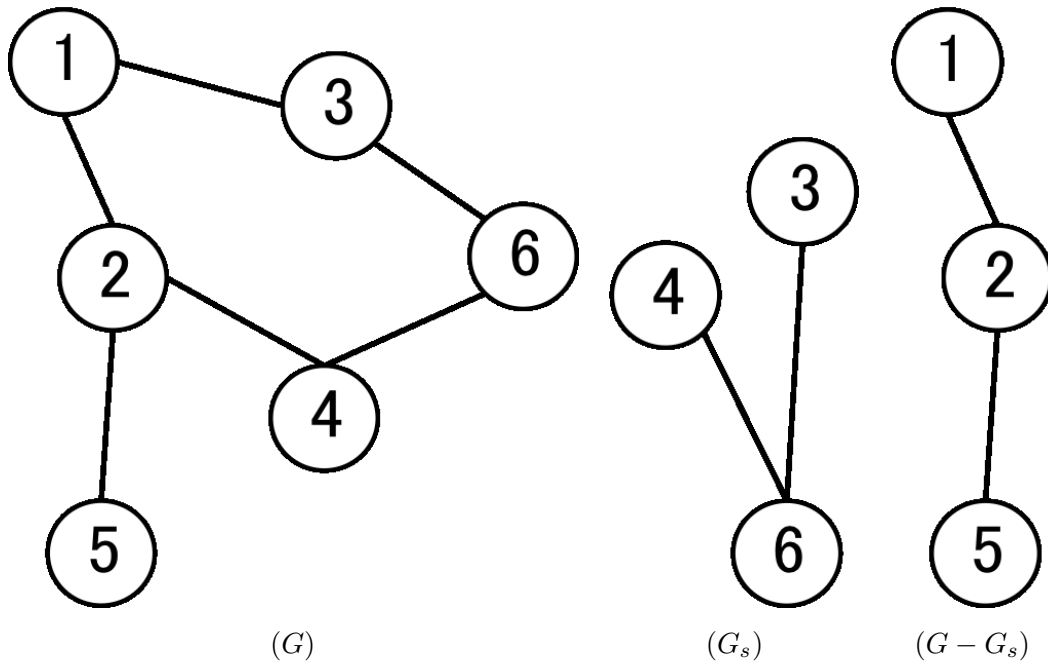


図 2.2: グラフの差分

2.1.5 共通グラフ

グラフを元とする集合をそれぞれ $G = \{g_1, g_2, \dots, g_n\}$, $H = \{h_1, h_2, \dots, h_m\}$ とする。あるグラフ t が $g_i \in G$ と同型で、かつ $h_j \in H$ と同型であるならば、グラフ t は G と H に共通なグラフという。 G と H について共通なグラフの集合を共通グラフ集合といい、以下のように定義する。

$$G \cap H = \{t \mid t \simeq g, h; g \in G, h \in H\}$$

2.1.6 最大共通部分グラフ

ふたつのグラフ G, G' の最大共通部分グラフ (maximum common subgraph, mcs) を定義する。 G の部分グラフを元とする集合を G_s , G' の部分グラフを元とする集合を G'_s とすると、 G_s と G'_s における共通グラフ集合は $H = G_s \cap G'_s = \{h_1, \dots, h_n\}$ となる。 H のグラフのうち頂点数最大のグラフを G と G' の最大共通部分グラフといい、 $\text{mcs}(G, G')$ と表記する。 $\text{mcs}(G, G')$ は式 (2.9) により得られる。

$$\text{mcs}(G, G') = \arg \max_{h \in \mathbf{H}} |h| \quad (2.9)$$

ふたつのグラフの最大共通部分グラフを検出する問題は NP 完全問題であることが知られている [7]. そのため近似アルゴリズムが提案されている. 本論文ではバックトラッキングアルゴリズムを用いた最大共通部分グラフを検出する手法 [4] について簡単に述べる.

$|G| \leq |G'|$ であるふたつのグラフ G, G' があり, G の頂点集合から G' の頂点集合へのある写像を $f_i: \mathcal{V}(G) \rightarrow \mathcal{V}(G')$ とする. $\mathcal{V}(G)$ から $\mathcal{V}(G')$ への写像は $\mathcal{V}(G)$ と $\mathcal{V}(G')$ の組み合わせの数だけ存在し, $\mathcal{V}(G)$ から $\mathcal{V}(G')$ への写像の集合を $\mathbf{F} = \{f_1, \dots, f_k\}$ と表記する. ふたつのグラフの最大共通部分グラフは, \mathbf{F} を用いて以下の式で求めることができる.

$$\mathcal{E}(\text{mcs}(G, G')) = \{(v_i, v_j) \in \mathcal{E}(G) \mid (f(v_i), f(v_j)) \in \mathcal{E}(G')\} \quad (2.10)$$

$$\mathcal{V}(\text{mcs}(G, G')) = \{v \mid \mathcal{E}(\text{mcs}(G, G')) \text{ の端点 } \} \quad (2.11)$$

$$f = \arg \max_{f \in \mathbf{F}} \sum_{(v_i, v_j) \in \mathcal{E}(G)}^{\|G\|} \mathbb{I}((f(v_i), f(v_j)) \in \mathcal{E}(G')), \quad (2.12)$$

$$\mathbb{I}(p) = \begin{cases} 1 & \text{if } p \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \quad (2.13)$$

式 (2.12) により f を発見することができれば式 (2.10), 式 (2.11) から最大共通部分グラフを得ることができる. $\mathcal{V}(G)$ から $\mathcal{V}(G')$ への写像は $\mathcal{V}(G)$ と $\mathcal{V}(G')$ の組み合わせであるので, $\mathcal{V}(G)$ と $\mathcal{V}(G')$ の組み合わせから成る探索木をバックトラッキングアルゴリズムで探索し, 式 (2.12) を満たす f を得ることができる. バックトラッキングによる最大共通部分グラフの計算量は式 (2.14) となることが知られている [5].

$$|G'|! \cdot \left(\frac{1}{(|G'| - |G|)!} + \frac{1}{(|G'| - (|G| - 1))!} + \dots + \frac{1}{(|G'| - 1)!} \right) \quad (2.14)$$

2.1.7 最小共通スーパーグラフ

ふたつのグラフ G, H の最小共通スーパーグラフ (Minimum Common Supergraph, MCS) を定義する. G_s を G の部分グラフとすると, G は G_s のスーパーグラフであるという. G のスーパーグラフを元とする集合を \mathbf{G}^s , H のスーパーグラフを元とする集合を \mathbf{H}^s とすると, \mathbf{G}^s と \mathbf{H}^s における共通グラフ集合は $\mathbf{K} = \mathbf{G}^s \cap \mathbf{H}^s$ となる. \mathbf{K} のグラフ

第2章 グラフを用いたパターン認識

のうち頂点数最小のグラフを G と H の最小共通スーパーグラフといい、 $\text{MCS}(G, H)$ と表記する。 $\text{MCS}(G, H)$ は式 (2.15) により得られる。

$$\text{MCS}(G, H) = \arg \min_{k \in \mathbf{K}} |k| \quad (2.15)$$

また、最大共通部分グラフを用いてふたつのグラフの最小共通スーパーグラフを式 (2.16) により求めることができることが知られている [6]。 \cup_E は辺集合 E によりグラフを結合する操作である。

$$\begin{aligned} \text{MCS}(G, H) &= \text{mcs}(G, H) \cup_{E1} (G - \text{mcs}(G, H)) \cup_{E2} (H - \text{mcs}(G, H)) \quad (2.16) \\ E1 &= \text{emb}(G, \text{mcs}(G, H)) \\ E2 &= \text{emb}(H, \text{mcs}(G, H)) \end{aligned}$$

2.1.8 重み付き最大共通部分グラフ

重み付きグラフにおける最大部分グラフを重み付き最大部分グラフ (weighted maximum common subgraph, wmcs) という [3]。

ふたつの重み付きグラフを $G_1 = (\mathbf{V}_1, \mathbf{E}_1, \lambda_1, \epsilon_1)$ 、 $G_2 = (\mathbf{V}_2, \mathbf{E}_2, \lambda_2, \epsilon_2)$ とし、 G_1 と G_2 の部分グラフを元とする集合をそれぞれ \mathbf{G}_{1s} 、 \mathbf{G}_{2s} とする。 G_1 と G_2 の重み付き最大共通部分グラフは式 (2.17) と定義される。

$$\text{wmcs}(G_1, G_2) = \arg \max_{y \in \mathbf{G}_{1s} \cap \mathbf{G}_{2s}} |y| \quad (2.17)$$

$\text{wmcs}(G_1, G_2)$ から G_1 、 G_2 への対応関係が $f_1 : \mathcal{V}(\text{wmcs}(G_1, G_2)) \rightarrow \mathcal{V}(G_1)$ 、 $\zeta_1 : \mathcal{E}(\text{wmcs}(G_1, G_2)) \rightarrow \mathcal{E}(G_1)$ 、 $f_2 : \mathcal{V}(\text{wmcs}(G_1, G_2)) \rightarrow \mathcal{V}(G_2)$ 、 $\zeta_2 : \mathcal{E}(\text{wmcs}(G_1, G_2)) \rightarrow \mathcal{E}(G_2)$ とすると、 $\text{wmcs}(G_1, G_2)$ の重みは式 (2.18)、式 (2.19) となる。ただし $v, v' \in \mathcal{V}(\text{wmcs}(G_1, G_2))$ である。

$$\lambda(v) = \lambda_1(f_1(v)) + \lambda_2(f_2(v)) \quad (2.18)$$

$$\epsilon((v, v')) = \epsilon_1(\zeta_1((v, v'))) + \epsilon_2(\zeta_2((v, v'))) \quad (2.19)$$

2.1.9 重み付き最小共通スーパーグラフ

重み付きグラフにおける最小共通スーパーグラフを重み付き最小共通スーパーグラフ (Weighted Minimum Common Supergraph, WMCS) という [3]。グラフ G_1 、 G_2 のスー

第2章 グラフを用いたパターン認識

パーグラフを元とする集合を \mathbf{G}^{1S} , \mathbf{G}^{2S} とする. G_1 と G_2 の重み付き最小スーパーグラフを式 (2.20) と定義する. 重み付き最大共通部分グラフを用いると式 (2.21) と表せる. $E1 = \text{emb}(G_1, \text{wmcs}(G_1, G_2))$, $E2 = \text{emb}(G_2, \text{wmcs}(G_1, G_2))$ である. $\text{WMCS}(G_1, G_2)$ の重みは, $\text{wmcs}(G_1, G_2)$ と $G_1 - \text{wmcs}(G_1, G_2)$, $G_2 - \text{wmcs}(G_1, G_2)$, $E1$, $E2$ の重みとなる.

$$\begin{aligned} \text{WMCS}(G_1, G_2) &= \arg \min_{y \in \mathbf{Y}} |y| & (2.20) \\ \mathbf{Y} &= \mathbf{G}^{1S} \cap \mathbf{G}^{2S} \end{aligned}$$

$$\begin{aligned} \text{WMCS}(G_1, G_2) &= \text{wmcs}(G_1, G_2) \cup_{E1} (G_1 - \text{wmcs}(G_1, G_2)) \cup_{E2} \\ &\quad (G_2 - \text{wmcs}(G_1, G_2)) & (2.21) \end{aligned}$$

ここまでふたつの重み付きグラフの最小共通スーパーグラフについて述べてきた. 重み付きグラフを元とする集合 $\mathbf{G} = \{G_1, \dots, G_n\}$ の最小共通スーパーグラフが文献 [3] において論じられており, 式 (2.22) となる. \mathbf{G}^{iS} は $G_i \in \mathbf{G}$ のスーパーグラフを元とする集合である. グラフ集合の重み付き最小共通スーパーグラフは集合を代表するグラフとみなすことができる [3].

$$\begin{aligned} \text{WMCS}(\mathbf{G}) &= \arg \min_{y \in \mathbf{Y}} |y| & (2.22) \\ \mathbf{Y} &= \mathbf{G}^{1S} \cap \mathbf{G}^{2S} \cap \dots \cap \mathbf{G}^{nS} \end{aligned}$$

式 (2.20) の計算量は, 重み付きグラフの位数に応じて指数関数的に増大することから, 式 (2.22) を計算する近似アルゴリズムについて述べておく. \mathbf{G} の重み付き最小共通スーパーグラフを求める近似アルゴリズムは, 逐次的に式 (2.20) を適用するものである. 図 2.3 と図 2.4 に重み付き最小共通スーパーグラフを得る近似アルゴリズムを示す.


```

Procedure WMCS( $G_1, G_2$ )
input :  $G_1, G_2$  ;
output : WMCS( $G_1, G_2$ ) ;
begin
     $E1 = \text{emb}(G_1, \text{wmcs}(G_1, G_2))$ ;
     $E2 = \text{emb}(G_2, \text{wmcs}(G_1, G_2))$ ;
    WMCS( $G_1, G_2$ ) =  $\text{wmcs}(G_1, G_2) \cup_{E1}$ 
                         $(G_1 - \text{wmcs}(G_1, G_2)) \cup_{E2}$ 
                         $(G_2 - \text{wmcs}(G_1, G_2))$ ;
end Procedure
    
```

図 2.3: 重み付き最小共通スーパーグラフを得る手続き

```

Procedure WMCS( $\mathbf{G}$ )
input :  $\mathbf{G} = \{G_1, \dots, G_n\}$ ;
output : WMCS( $\mathbf{G}$ );
begin
    Initialize weights of vertices and edges  $\lambda(v) = 1, \epsilon(e) = 1$ ;
     $W = G_1$ ;
    for  $i = 2$  to  $n$ 
    begin
         $W = \text{WMCS}(W, G_i)$ 
    end for
    WMCS( $\mathbf{G}$ ) =  $W$ ;
end Procedure
    
```

図 2.4: グラフ集合の重み付き最小共通スーパーグラフを得る近似アルゴリズム

2.2 グラフ間の距離

グラフとグラフ間の距離をグラフにおける距離といい、今後グラフにおける距離を単に距離と表記する。本節ではグラフ間の距離を計測する手法について述べる。2.1.3節において述べた同型という概念は、同一のグラフ構造を検出する有用な手法である [14]。さらに部分グラフ同型は、あるグラフが他のグラフの一部であるかどうかを調べる事ができるため有用であると考えられている [14]。部分グラフをさらに拡張し、ふたつのグラフにおける最大共通部分グラフにより距離を計測する手法が考案されている [4,6,15]。本節では、平均情報量（エントロピー）を用いてグラフ距離を計測する手法 [3,10] と、一方のグラフを他方のグラフへ変換する編集系列を用いてグラフ距離を計測する手法 [16,17] について述べる。

2.2.1 エントロピーによる距離

グラフの頂点集合や辺集合に 0 以上 1 以下の実数、例えば重みや確率など、が割り当てられているグラフにおいてエントロピーを用いてグラフ距離を計算することができる。本論文では頂点集合や辺集合に重みを与えた重み付きグラフを定義しているが、重みの範囲は自然数である。重みの範囲を 0 以上 1 以下にする操作を行う必要がある。

重み付きグラフの集合を $\mathbf{G} = \{G_1, \dots, G_n\}$ とし、 \mathbf{G} の各元に対応する重み $\mathbf{Q} = \{q_1, \dots, q_n\}$ を割り当てる。ただし、 $q_k \geq \max(\lambda(\mathcal{V}(G_k)))$ かつ $q_k \geq \max(\epsilon(\mathcal{E}(G_k)))$ であるものとする。本論文においては、 G_k の頂点や辺の重みを q_k により除算することで、重みを 0 以上 1 以下の範囲にする。

重み付きグラフ G_k のエントロピーは式 (2.23) により定義される。 $v_i \in \mathcal{V}(G_k)$, $e_j \in \mathcal{E}(G_k)$ である。ふたつの重み付きグラフ G と H の距離は、 G と H の重み付き最小共通スーパーグラフのエントロピーと G との差分により得られ、式 (2.21) と式 (2.23) を用いて、式 (2.24) により定義される。ただし、 $\text{WMCS}(G_i, G_j)$ に対応する重みは $q = q_i + q_j$ とする。ここで式 (2.24) について $\Delta\text{Ent}(G_i, G_j) \neq \Delta\text{Ent}(G_j, G_i)$ であることを記しておく。また、 G_i と G_j が類似しているなら ΔEnt の値は小さくなる。

$$\text{Ent}(G_k) = - \sum_{i=1}^{|\mathcal{V}(G_k)|} \frac{\lambda(v_i)}{q_k} \log \left(\frac{\lambda(v_i)}{q_k} \right) - \sum_{j=1}^{|\mathcal{E}(G_k)|} \frac{\epsilon(e_j)}{q_k} \log \left(\frac{\epsilon(e_j)}{q_k} \right) \quad (2.23)$$

$$\Delta\text{Ent}(G_i, G_j) = \text{Ent}(\text{WMCS}(G_i, G_j)) - \text{Ent}(G_i) \quad (2.24)$$

2.2.2 編集系列による距離

一方のグラフを他方のグラフへ変換する編集系列により計算される距離をグラフ編集距離 (Graph Edit Distance) という [16,17,19].

基本的なグラフの編集操作を, 1) 挿入, 2) 置換, 3) 削除とすると, グラフ G から H への変換は編集操作を用いる. G の頂点, 辺をいくつか削除し, G の頂点, 辺を H の属性に置き換える. そして G に H の頂点と辺を挿入することにより G から H への変換が完了する. あるグラフからグラフへの編集操作の順番を編集系列 (Edit Path) といい, 編集操作を頂点や辺に対して行うコストを編集コスト (Edit Cost) という. 編集操作の編集コストが定められると, ふたつのグラフの編集距離は編集コストが最小となる編集系列により計算される [16].

グラフ $G_1 = (\mathbf{V}_1, \mathbf{E}_1, \alpha_1, \beta_1)$ から $G_2 = (\mathbf{V}_2, \mathbf{E}_2, \alpha_2, \beta_2)$ への編集距離について, 頂点 $u \in \mathbf{V}_1$ と頂点 $v \in \mathbf{V}_2$ における置換操作を $u \rightarrow v$ で表す. 置換操作 $u \rightarrow v$ により属性値も置換が行われ, $\alpha_1(u) \rightarrow \alpha_2(v)$ となる. 削除操作を $u \rightarrow \emptyset$, 挿入操作を $\emptyset \rightarrow v$ と表記する.

編集系列の i 番目の編集操作を p_i とすると, p_i は $p_i = \{u \rightarrow v\}$, $p_i = \{u \rightarrow \emptyset\}$, $p_i = \{\emptyset \rightarrow v\}$ のいずれかとなり, G_1 から G_2 への編集系列を $\mathbf{P} = \{p_1, p_2, \dots, p_l\}$ で表す. 編集操作 p_i のコストを計算する関数を cost とすると, \mathbf{P} の編集コストは式 (2.25) により得られる.

$$\text{COST}(\mathbf{P}) = \sum_{i=1}^{|\mathbf{P}|} \text{cost}(p_i) \quad (2.25)$$

編集系列 \mathbf{P} を元とする集合を $\mathbf{O} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n\}$ とする. \mathbf{O} は G_1 から G_2 へ変換するためのすべての編集系列とみなせ, G_1 と G_2 の編集距離は式 (2.26) により定義される.

$$\text{ED}(G_1, G_2) = \arg \min_{\mathbf{P}_i \in \mathbf{O}} \text{COST}(\mathbf{P}_i) \quad (2.26)$$

式 (2.26) によると, 編集距離は最小コストの編集系列を見つける必要がある. A-star アルゴリズム (A* アルゴリズム) により最小の編集系列を得るアルゴリズム [17] について述べておく.

A-star アルゴリズム [18] は経路探索アルゴリズムのひとつである. 探索木の節 n においてルートの節から節 n までのコストを $g(n)$, 節 n からゴール節までの推定コストを $h(n)$ とすると, 節 n を経由してゴール節へ至るコストは $g(n) + h(n)$ と表せる. $g(n)$ に関しては探索の過程で推定値を求めることができるが, $h(n)$ は推定することができない.

第2章 グラフを用いたパターン認識

そこでヒューリスティックなアルゴリズムにより, $h(n)$ に適当な推定値を与えて, 探索を進めるアルゴリズムを A-star アルゴリズムという.

A-star アルゴリズムを用いて, G_1 から G_2 へのグラフ編集距離を計算するアルゴリズムを図 2.5 に示す. 編集系列 \mathbf{P} において, 編集操作が行われた変換元のグラフの頂点の集合を $\mathcal{V}_s(\mathbf{P})$, 同様に編集操作が行われた変換先のグラフの頂点の集合を $\mathcal{V}_t(\mathbf{P})$ と表記する. 例えば, $\mathbf{P} = \{u_1 \rightarrow v_2, u_2 \rightarrow v_5\}$, $u \in \mathbf{V}_1 = \{u_1, u_2, u_3\}$, $v \in \mathbf{V}_2 = \{v_1, v_2, v_3, v_4, v_5\}$ とすると, $\mathcal{V}_s(\mathbf{P}) = \{u_3\}$, $\mathcal{V}_t(\mathbf{P}) = \{v_1, v_3, v_4\}$ である. 図 2.5 の 10 行目, $h(\mathbf{P})$ は, 編集系列 \mathbf{P} により変形した G_1 から G_2 へ必要な編集コストの予測値を計算する関数である.

```

Computation of graph edit distance by A* algorithm
Input :  $G_1$  and  $G_2$  ,  $\mathcal{V}(G_1) = \{u_1, \dots, u_{|G_1|}\}$  ,  $\mathcal{V}(G_2) = \{v_1, \dots, v_{|G_2|}\}$ 
Output : A minimum cost edit path from  $G_1$  to  $G_2$ 
         e.g.  $o = \{u_1 \rightarrow v_3, u_2 \rightarrow \emptyset, \dots, \emptyset \rightarrow v_6\}$ 

1: Initialize a set of all edit path  $O = \{\emptyset\}$ 
2: Construct edit path; Substitution
3:   for  $i = 1$  to  $|G_2|$ 
4:      $P_i = \{u_1 \rightarrow v_i\}$  ;  $O = O \cup P_i$ 
5:   end for
6: Construct edit path; Deletion
7:    $P_{|O|+1} = \{u_1 \rightarrow \emptyset\}$  ;  $O = O \cup P_{|O|+1}$ 
8: Loop
9:   Extract minimum edit path
10:   $P_{min} = \arg \min_{P \in O} COST(P) + h(P)$  ;  $O = O \setminus P_{min}$ 
11:  if  $G_1$  is transformed to  $G_2$  by  $P_{min}$  then
12:    Return  $P_{min}$  as the solution
13:  else
14:    if  $|\mathcal{V}_s(P_{min})| < |V_1|$  then
15:      Construct edit path; Substitution
16:      for  $w \in \mathcal{V}_t(P_{min})$ 
17:         $P' = P_{min} \cup \{u_{|\mathcal{V}_s(P_{min})|+1} \rightarrow w\}$ 
18:         $O = O \cup P'$ 
19:      end for
20:      Construct edit path; Deletion
21:       $P_{|O|+1} = P_{min} \cup \{u_{|\mathcal{V}_s(P_{min})|+1} \rightarrow \emptyset\}$  ;  $O = O \cup P_{|O|+1}$ 
21:    else
22:      Construct edit path; Insertion
23:       $P_{|O|+1} = P_{min} \cup \{\emptyset \rightarrow w\}$  ;  $w \in \mathcal{V}_t(P_{min})$  ;  $O = O \cup P_{|O|+1}$ 
24:    end if
25:  end if
26: end Loop

```

図 2.5: A-star アルゴリズムによる編集距離の計算

2.3 重み付き最小共通スーパーグラフを用いたシルエット画像認識

物体のシルエット画像は物体の基本的なカテゴリを表す図形で、基本レベルにおけるパターン認識に有用であることが知られている [21]。基本レベルにおける認識とは、シルエット画像を認識することを物体認識の第一段階と捉えることである。物体のシルエット画像を正しく認識することができれば、さらに詳細で的確な認識を行うことが可能となる。

シルエット画像の認識は、シルエットの形状を利用する手法 [22] やシルエットから得られる骨格 (Skeleton) や中心軸 (Medial Axis) を用いる手法 [23, 24] などがある。中心軸や骨格を利用することにより、シルエット画像の構造を抽出することができる。

重み付き最小共通スーパーグラフはグラフ集合を代表するグラフや、距離を計算するために用いられるなど、グラフを用いたパターン認識分野において有用である [3]。そこで本節では、シルエットの構造を表すグラフを抽出することで、シルエットをグラフとして扱う。グラフとグラフが認識されるカテゴリとの距離を重み付き最小共通スーパーグラフを用いて計算し、シルエット画像の認識を行う。

2.3.1 シルエット画像のグラフ

シルエット画像からグラフを抽出する。最初に、シルエット画像の骨格を中心軸変換 [24] により抽出する。次に、グラフの辺に辺の重要度を割り当て、重要度の低いものをノイズとして除去し、シルエットのグラフを抽出する。

中心軸変換

中心軸変換によりシルエットから中心軸を抽出する。シルエットの中心軸とは、シルエットの輪郭に少なくとも二点以上で内接する円の中心点の集合として定義される。中心軸の各点は8連結で連結しており、各点における内接円の半径の情報を保持している。長方形のシルエットにおける中心軸の例を図 2.6 に示す。

閉曲線から中心軸を求める古典的な手法は草焼きのイメージの細線化 [23] として表される。Kimia らはこの草焼きのイメージを閉曲線内で輪郭線上から湧き出る波面の発展方程式として表し、この発展方程式の特異点が中心軸に対応することを示した [25]。特異点を解析的に求めることは一般的に困難で、数値計算により特異点を求めることとなるため、手順が煩雑になるという問題があった。そこで、本論文では Siddiqi らが提案した、より簡便な方法 [24] を用いる。以下にその概要を記す。

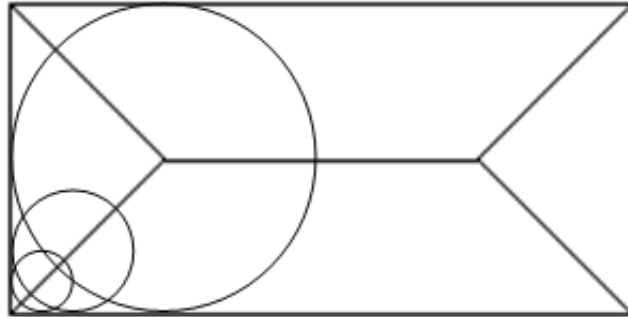


図 2.6: 中心軸の例

まず対象のシルエット上の各点に輪郭からの最短ユークリッド距離を割り当てる。距離を割り当てた画像を距離画像といい、 $D(x, y)$ とする。座標 (x, y) はシルエット画像内の点で、 $D(x, y)$ は輪郭からその点までの距離を表している。距離画像の各値 $D(x, y)$ は輪郭から距離が遠いほど値が大きくなる。輪郭から湧き出る流れの勾配は $\vec{F} = \nabla D$ 、 $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^T$ と表せる。平面上に流れがあるとき、シルエット上のある点 (x, y) における流れの平均変化率は式 (2.27) となる。 $(i, j) \neq (0, 0)$ 、 $N_{i,j}$ は点 (x, y) の 8 近傍の外向き単位ベクトルである。

$$E(x, y) = \frac{1}{8} \sum_{j=-1}^1 \sum_{i=-1}^1 N_{ij} \cdot F(x+i, y+j) \quad (2.27)$$

平均変化率はシルエットの中心軸に対応する点を除く任意の点で $E(x, y) = 0$ となる。これは微小領域内での流量の変化と輪郭上から溢れ出す流れ出す変化の和は一定となるからである。輪郭線上のすべての点を流速の湧き出し点として、そこから流体が流れ出して等速で広がっていくとすると、中心軸に対応する点で流れが衝突し、この地点は流れが流れて行くことができず、流れ込む地点となるので、 $E(x, y) < 0$ となる。 $E(x, y) < 0$ となる点が特異点であり、中心軸に対応する点となる。 $E(x, y) < 0$ となる点を単純に抽出するだけでは連結された中心軸は抽出できない。そこで以下の条件を適用して中心軸を抽出することで、連続した中心軸を抽出する。条件 1 におけるシンプルとは、その点を削除しても 8 連結の切断が発生しないことを言う

条件 1 点がシンプルでない、

条件 2 $\nabla \vec{F}$ が閾値以下、

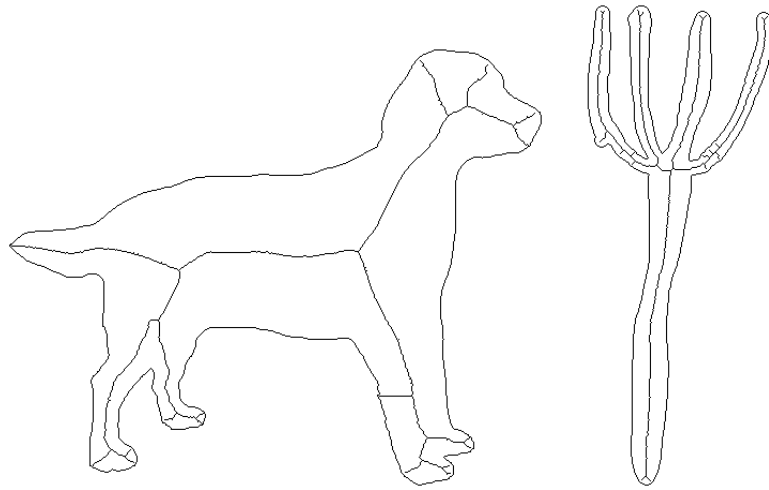


図 2.7: 中心軸の例

シルエット画像から中心軸を抽出した例を図 2.7 に示す。一般的に、中心軸変換は輪郭のノイズに対して影響を受けやすく、本来存在すべきでない中心軸が生じてしまう問題が知られている。対策として輪郭線にスムージングを施したり、閾値を厳密にすることである程度は回避できるが、完全に回避することができない。

中心軸からグラフへの変換

中心軸変換によって得た中心軸から、頂点集合 V とエッジ集合 E を得る手順について述べる。中心軸をグラフにするため、中心軸上の画素を頂点と辺にラベル付けする。式 (2.28) により、中心軸上の画素 (i, j) に頂点、辺を割り当てる。辺の両端の頂点は中心軸の連結関係を追跡することで得ることができる。

$$(i, j) = \begin{cases} \text{edge} & \text{if } \sum_{x=-1}^1 \sum_{y=-1}^1 p(i+x, j+y) = 3 \\ \text{vertex} & \text{otherwise} \end{cases} \quad (2.28)$$

$$p(i, j) = \begin{cases} 1 & \text{if } (i, j) \text{ is on the medial axis} \\ 0 & \text{otherwise} \end{cases}$$

式 (2.28) により得られる辺は頂点間を結ぶ中心軸の点の集合をひとつの辺とする。頂点と辺は画像上の座標、内接円の半径（つまり輪郭までの距離）の情報を有しているものとする。中心軸変換により、局所的に中心軸が集中する箇所があり、頂点の画素に頂

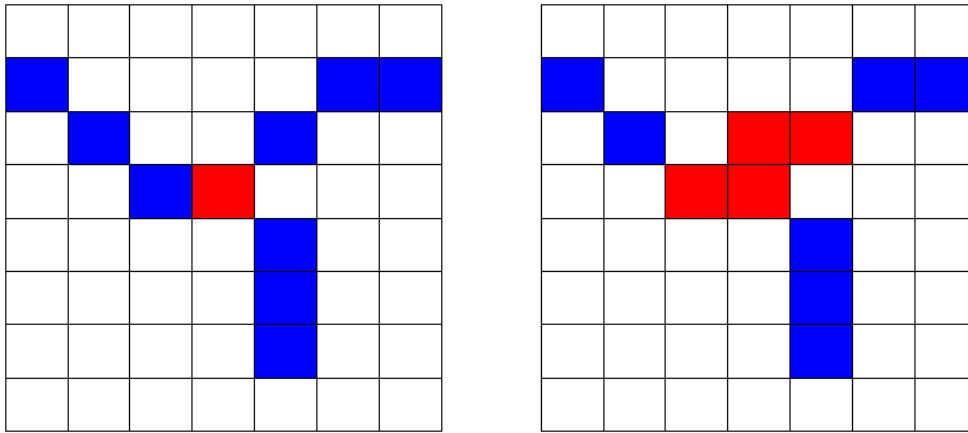


図 2.8: ノード, エッジの例

点の画素が隣接する場合がある。複数の頂点が隣り合った画素集合は1つの頂点とする。中心軸上の各画素に頂点、辺を割り当てた例を図 2.8 に示す。青い点の列がエッジ、赤の点の集まりがノードに対応する。同様にシルエット画像の中心軸上の画素に頂点と辺を割り当てた例を図 2.9 に示す。青線は辺、緑、赤点は頂点。図 2.9 の例では、輪郭の微妙な凸凹ノイズにより抽出される中心軸が多いため、頂点数が非常に多いグラフになってしまう。頂点数が多いグラフは重み付き最小共通スーパーグラフの計算や距離計算に悪影響を及ぼすことから、不要なエッジを除去する。

ノイズ除去

中心軸変換により得られたグラフの辺に対して、辺に対応するシルエット図形上の領域の面積を割り当てる。その上で、面積の小さい辺をノイズとして除去する [26]。

まず各辺に対して、頂点から見た辺の接続状態を示すラベルを与える。ラベルは以下の3つである。

- Even : 辺は頂点に対して対等に接続
- Under : 辺は頂点に従属している
- Over : 頂点は辺に従属している

辺にラベルを与えるために内接円の変化率を用いる。中心軸上の点を s とすると、 s における内接円の半径の変化率を式 (2.29) により定義できる。 s_i は中心軸上の各点、 r_i は

第2章 グラフを用いたパターン認識

s_i を中心とする内接円の半径である. \bar{s} , \bar{r} は n 点の平均である. n の値は実験的に定め, $n = 3$ とした.

$$a(s) = \frac{\sum_{i=1}^n s_i r_i - n \bar{s} \bar{r}}{\sum_{i=1}^n s_i^2 - n \bar{s}^2} \quad (2.29)$$

各頂点に注目し, それにつながる辺の画素の変化率 a の正負を用いてラベル付けする. 頂点から見て5画素までの範囲で $a < -t$ である画素が3以上あれば, 従属 (Under) しているとし, $a > +t$ の画素が3以上存在するならば, 被従属 (Over) であるとする. 従属, 被従属にも当てはまらない場合は対等 (Even) である. t は閾値で $t = 0.5$ と実験的に定めた.

中心軸上の点 s のシルエット画像上の領域を $\text{region}(s)$ とする. 辺 e は n 個の点から成り立っているとし, e の点の集合を $S = \{s_1, s_2, \dots, s_n\}$ とする. 辺 e のシルエット上における領域は式 (2.30) により定義し, 辺 e に割り当てられる領域 $\text{Area}(e)$ は式 (2.31) と定義される. (e に属さない領域) は式 (2.32) により定義される. v は辺 e に端点の頂点である. (e に属さない領域) は式 (2.32) により得られる. n_u は頂点 v に対して Under で接続する辺の数で, 同様に n_o , n_e はそれぞれ v に対して Over, Even で接続する辺の数である. 式 (2.31) により辺に割り当てられた面積はシルエット全体の面積が1となるように正規化する. 辺に割り当てられた面積が小さい辺を取り除いた結果を図 2.10 に示す.

$$\text{REGION}(e) = \bigcup_{i=1}^{|S|} \text{region}(s_i) \quad (2.30)$$

$$\text{Area}(e) = \text{REGION}(e) - (e \text{ に属さない領域}) \quad (2.31)$$

$$(e \text{ に属さない領域}) = \begin{cases} \text{region}(v) \times \frac{1}{n_u} & e \text{ は } v \text{ に対して Under かつ,} \\ & \text{他の辺もすべて Under の場合} \\ \\ \text{region}(v) & e \text{ は } v \text{ に対して Under かつ,} \\ & \text{他の辺が Over もしくは Even の場合} \\ \\ \text{region}(v) \times \frac{1}{n_e + n_o} & e \text{ は } v \text{ に対して Over,} \\ & \text{もしくは Even である場合} \end{cases} \quad (2.32)$$

第 2 章 グラフを用いたパターン認識

これまでの処理により得られるシルエット画像のグラフを式 (2.33) とする. \mathbf{V} , \mathbf{E} , λ , ϵ は重み付きグラフ (式 (2.4)) と同じ定義とする. θ は辺に割り当てられた領域の面積を示す関数で, $\theta: e \rightarrow \mathbb{R}$, $e \in \mathbf{E}$ とする.

$$G = (\mathbf{V}, \mathbf{E}, \lambda, \epsilon, \theta) \quad (2.33)$$

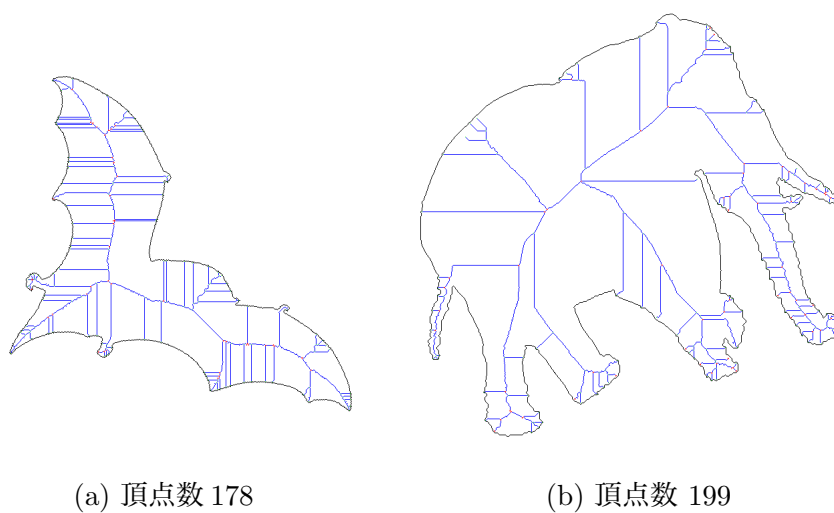


図 2.9: 中心軸の画素に頂点, 辺を割り当てた例

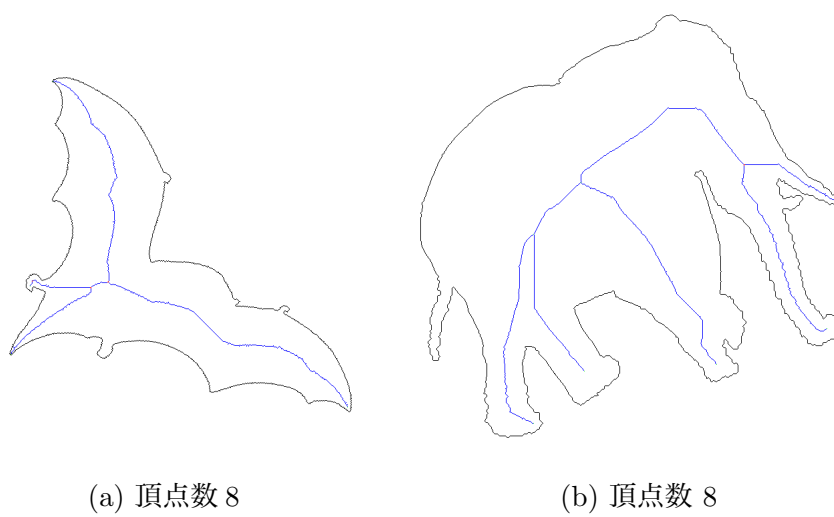


図 2.10: ノイズを除去した例

2.3.2 シルエット画像とカテゴリの距離

これまでシルエット画像のグラフを抽出することについて述べてきた。本節ではシルエット画像のカテゴリを認識するために、シルエット画像とカテゴリとの距離を定義する。

カテゴリを代表するグラフ

カテゴリを代表するグラフを重み付き最小共通スーパーグラフにより得る。グラフは式 (2.33) を用いるため、重み付き最小共通スーパーグラフの定義である式 (2.22) を拡張し、 θ を扱えるようにする。

カテゴリのシルエット画像から抽出されたグラフを元とする集合 $\mathbf{G} = \{G_1, \dots, G_n\}$ が与えられたとき、 \mathbf{G} の重み付き最小共通スーパーグラフは式 (2.22) により得ることができるが、 $\text{WMCS}(\mathbf{G})$ の θ は得られていない。 $G_i \in \mathbf{G}, i = 1, \dots, n$ は $\text{WMCS}(\mathbf{G})$ に対して部分グラフ同型であるので、 $\mathcal{V}(G_i)$ から $\mathcal{V}(\text{WMCS}(\mathbf{G}))$ への部分グラフ同型写像 f_i 、 $\mathcal{E}(G_i)$ から $\mathcal{E}(\text{WMCS}(\mathbf{G}))$ への写像 ζ_i が存在する。

$$f_i : \mathcal{V}(G_i) \rightarrow \mathcal{V}(\text{WMCS}(\mathbf{G})), i = 1, \dots, n \quad (2.34)$$

$$\zeta_i : \mathcal{E}(G_i) \rightarrow \mathcal{E}(\text{WMCS}(\mathbf{G})), i = 1, \dots, n \quad (2.35)$$

$\text{WMCS}(\mathbf{G})$ の頂点集合から G_i の頂点集合への対応関係 π_i とし、 f_i を用いて π_i を式 (2.36) のように求めることができる。同様に $\text{WMCS}(\mathbf{G})$ の辺集合から $G_i \in \mathbf{G}$ の辺集合への対応関係 z_i は式 (2.37) により求める。

$$\pi_i : v \in \mathcal{V}(\text{WMCS}(\mathbf{G})) \rightarrow \begin{cases} u \in \mathcal{V}(G_i) & \text{if } f_i(u) = v \\ \phi & \text{otherwise} \end{cases} \quad (2.36)$$

$$z_i : (v, v') \in \mathcal{E}(\text{WMCS}(\mathbf{G})) \rightarrow \begin{cases} (u, u') \in \mathcal{E}(G_i) & \text{if } \zeta_i((v, v')) \in \mathcal{E}(G_i) \end{cases} \quad (2.37)$$

$\text{WMCS}(\mathbf{G})$ の θ は、 $\mathbf{Z} = \{z_i \mid i = 1, \dots, n\}$ を用いて式 (2.38) により求める。

$$\theta((v, v')) = \text{median} \left(\bigcup_{i=1}^n \theta_i(z_i(v, v')) \right) \quad (2.38)$$

シルエット画像から抽出されたグラフにおいて、式 (2.22) と式 (2.38) により得られる最小共通スーパーグラフを RWMCS と表記する。カテゴリの学習データ \mathbf{G} における RWMCS をカテゴリを代表するグラフとして扱い、 $\text{RWMCS}(\mathbf{G})$ と表記し、 $\text{RWMCS}(\mathbf{G})$ をグラフとカテゴリとの距離を計算するために用いる。また、 RWMCS のある閾値以下の重みを持つ頂点と辺を除いた RWMCS を RWMCS_t と表記する。本論文では閾値を 2.5 とした。

カテゴリとの距離

R とグラフ H との距離を式 (2.24) を基に, 式 (2.39) と定義する. a はある定数で, $\Delta\text{Ent}(R, H)$ と $pn(R, H)$ を統合するためのパラメータである. 式 (2.23) のパラメータ q に関して, R に伴うパラメータを $q_R = |\mathbf{G}|$, テストデータのパラメータ $q = 1$ とする. H は各カテゴリを代表するグラフとの距離を式 (2.39) により計測し, 最近傍のカテゴリへと認識する.

$$d(R, H) = \Delta\text{Ent}(R, H) + a \cdot pn(R, H) \quad (2.39)$$

関数 $pn(R, H)$ は対応する R の辺と H の辺の妥当性を与える関数 $pn(R, H)$ である. $pn(R, H)$ を式 (2.40) により定義される. z は H の辺と R の辺の対応関係を表す.

$$pn(R, H) = \sum_{(v, v') \in \mathcal{E}(H)}^{||H||} |\theta_H((v, v')) - \theta_R(z((v, v')))| \quad (2.40)$$

2.3.3 評価実験

シルエット画像は MPEG7 CE-Shape-1 [29] に含まれる teddy, Misk, carriage, octopus, fork, dog, sea_snake の 7 カテゴリを用いた. 各カテゴリには 20 枚の画像が含まれ, 1 から 20 までの通し番号が付けられている. 以下の実験ではこの番号により画像を参照する. シルエット画像の例を図 2.11 と図 2.12 に示す. MPEG7 CE-Shape-1 データセットには穴のあいている画像も含まれている. 穴のある画像に対して中心軸変換により得られる軸は物体の骨格とならないため, 穴のあいているシルエット画像については輪郭線から内側を塗りつぶす加工を施した.

本節ではシルエット画像に対してグラフを抽出し, 2.3.2 節で述べた手法を用いて認識実験を行う. 各カテゴリにおいて, ランダムに 10 枚のシルエット画像を選択し学習データ, その他のシルエット画像をテストデータとし, 認識する実験を 20 回行った. 式 (2.39) における a は $a = 3$ とした.

比較手法として, グラフ編集距離による手法 [19] を用いる. 比較手法は, カテゴリ R の学習データを $\mathbf{G} = \{G_1, \dots, G_n\}$ とすると, 認識対象のグラフ H との距離は式 (2.41) により求める. $\text{ED}(H, G_i)$ は H から G_i へのグラフ編集距離を計算する関数である. H と各カテゴリとの距離を計算し, 最近傍のカテゴリへと認識する.

$$d(R, H) = \min(\text{ED}(H, G_i), \quad i = 1, \dots, n) \quad (2.41)$$

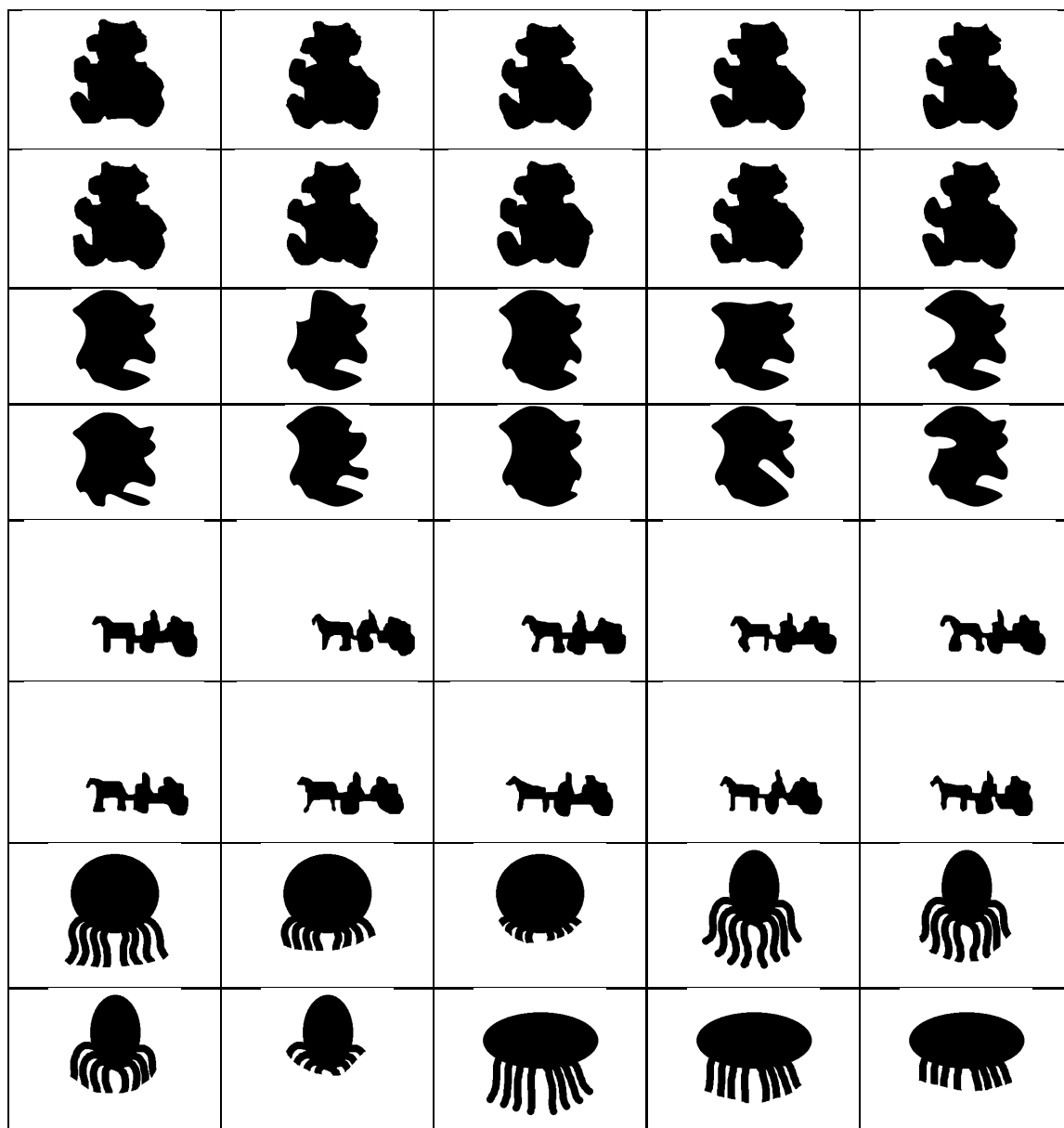


図 2.11: シルエット画像. teddy, Misk, carriage, octopus

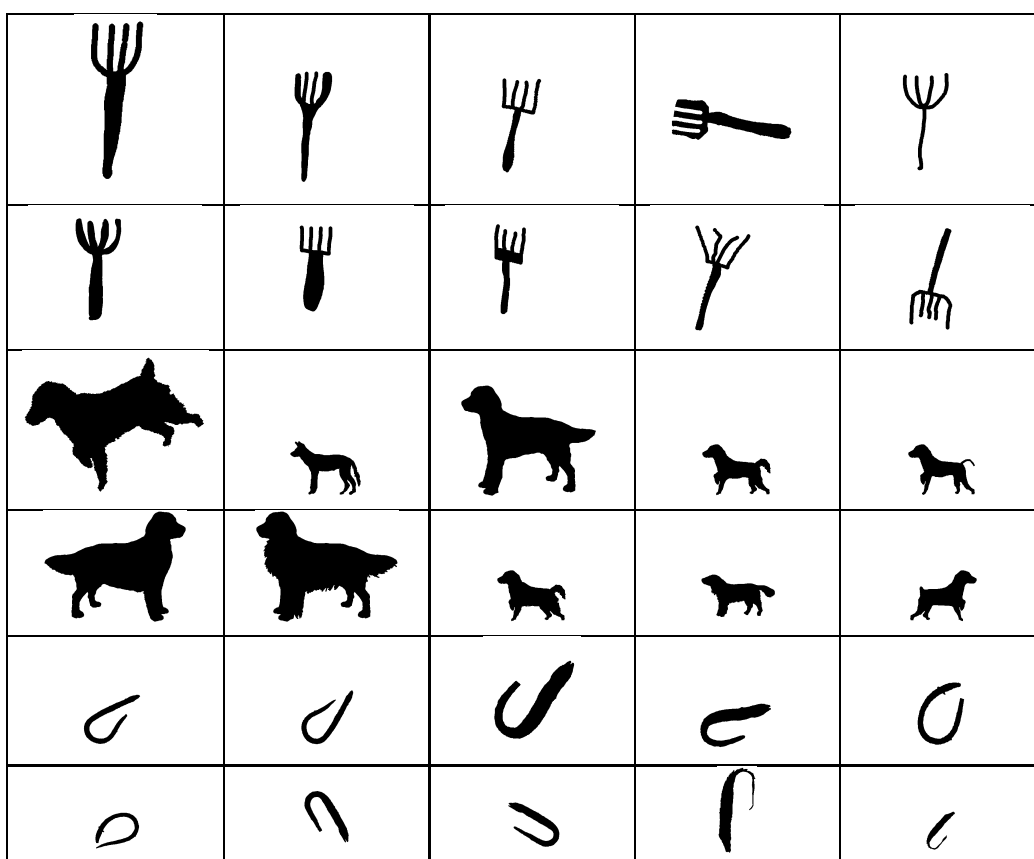


図 2.12: シルエット画像, fork, dog, sea_snake

第 2 章 グラフを用いたパターン認識

表 2.1: 正しいカテゴリに認識された数, 平均値と最高値, 最低値

		Misk	carriage	dog	fork	octopus	sea_snake	teddy	認識率
RWMCS	平均値	7.1	8.1	5.4	7.2	5.1	9.5	10	74.6(%)
	最高値	10	8	9	9	6	10	10	88.6(%)
	最低値	6	6	2	5	2	9	10	57.1(%)
RWMCS _t	平均値	6.1	8.6	6.5	8.8	7.2	9.5	9.8	80.5(%)
	最高値	7	9	9	9	8	10	10	88.6(%)
	最低値	1	10	7	8	4	9	10	70.0(%)
比較手法	平均値	5.8	5.3	6.6	7.7	6.8	9	9	71.6(%)
	最高値	8	5	7	9	7	9	9	77.1(%)
	最低値	3	5	6	6	6	9	9	62.9(%)

20 回の認識実験における認識率の平均値と最大値, 最小値を表 2.1 に示す. 表中の数字は 10 枚のテストデータ中, 正しいカテゴリに認識された数である. また, RWMCS を作成するために用いる学習データ数と認識率を図 2.13 に示す. 各学習データ数 (2~10 枚) においてランダムに学習データを選択して RWMCS を作成し, 認識する試行を 20 回行いその平均値を示している. すべての試行の中で, 学習データ 5 枚 (画像 8, 12, 3, 10, 15) を用いて WMCS を作成したとき, 最高の認識率 91.4% が得られ, 最低の認識率 57.1% が学習データ数 10 枚 (画像 7, 16, 20, 19, 18, 2, 6, 10, 17, 5) を用いた WMCS により得られた.

また, RWMCS が最良の結果となったデータセット A (学習データ; 画像 8, 12, 3, 10, 15, テストデータ; 画像 2, 4, 5, 6, 7, 11, 13, 16, 17, 18), RWMCS が最低の認識結果を得たデータセット B (学習データ; 画像 7, 16, 20, 19, 18, 2, 6, 10, 17, 5, テストデータ; 画像 1, 3, 4, 8, 8, 11, 12, 13, 14, 15) をそれぞれ用いた場合の認識結果を表 2.2 に示す. 表 2.2 から, RWMCS は比較手法と比較して, データセットのよっては認識率が大幅に向上することと, 認識率が下がる場合もその差はわずかであることが分かる.

実験結果から RWMCS を用いた手法は, 比較手法よりも多くシルエット画像を正しく認識しており, RWMCS の有効性を確認することができた. 表 2.1 の平均認識数から, 特に Misk や carriage などでの複雑な形状をもつ画像において, RWMCS は比較手法に対して約 13%~33%ほど認識数が向上した.

第2章 グラフを用いたパターン認識

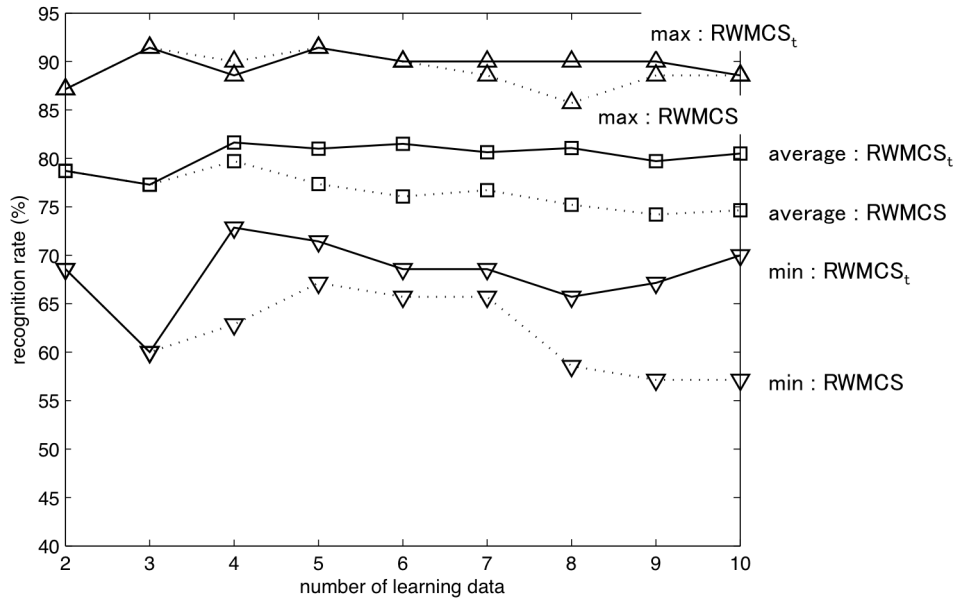


図 2.13: 学習データ数と認識率の関係

表 2.2: RWMCS に対して最高値・最低値を得たデータセットにおける比較手法との比較

		Misk	carriage	dog	fork	octopus	sea_snake	teddy	認識率
データセット A	RWMCS	9	10	7	10	8	10	10	91.4(%)
	比較手法	4	6	4	7	3	9	9	60.0(%)
データセット B	RWMCS	6	6	2	5	2	9	10	57.1(%)
	比較手法	3	5	6	6	6	9	9	62.9(%)

学習データ (画像 1, 2, 3, 4, 5, 6, 7, 8, 9, 10) から得られた RWMCS を用いて, テストデータ (画像 11, 12, 13, 14, 15, 16, 17, 18, 19, 20) を認識する実験を行った結果, dog カテゴリが最低の認識数 6 となった. dog カテゴリのテストデータの多くは fork や teddy カテゴリに誤認識していた. これは dog や fork, teddy カテゴリのテストデータはほぼ同じトポロジとなっていることが原因と考えられる. しかし, 式 (2.39) におけるパラメータ $a = 0.4$ とすると, 10 画像中 9 画像が正しく認識された. a が大きいとき, 同一カテゴリにおいて対応する辺の妥当性を与える項が小さくても, a が大きいため

第2章 グラフを用いたパターン認識

に式 (2.39) による距離が大きくなってしまふ。その結果異なるカテゴリとみなされてしまふことがある。一方で、 a を小さくすると、対応する辺の妥当性を与える項の影響を大きくしすぎなくすることができ、正しく認識される場合があると考えられる。つまり、式 (2.40) によって定義される対応する辺の妥当性の影響を適切に定めることができれば認識率を向上させることができるという可能性を示している。そこで、 a を変化させたときの認識結果を図 2.14 および図 2.15 に示す (学習データ数は 10 に固定)。 a の値が大きすぎると認識率が一定ないし下がる傾向にあり、対応する辺の妥当性が効かないことを示している。今回は a の値を固定して実験を行なったが、カテゴリに応じて適切な a の値を決めることができれば認識性能がさらに向上することが期待される。

本節における実験結果から、学習データから代表するグラフを抽出して、カテゴリとグラフの距離を計測する手法は、比較手法のように、個々の学習データを用いてカテゴリとグラフの距離を計測する手法と比較して良い性能を期待できる手法であると言えることができる。また、学習データを代表するグラフによって認識性能は変動することから (表 2.1 を参照)、学習データを代表するグラフの性能は認識性能に深い関係があることがわかる。

2.4 まとめ

本章では、グラフの基本的な概念とグラフ間の距離を計測する手法、そしてグラフを用いてシルエット画像認識を行う手法について述べた。

まず、グラフを頂点集合と辺集合により定義し、属性や重みによりグラフを拡張した、重み付きグラフ、属性グラフ、重み付き属性グラフを定義した。グラフの基本的な概念である部分グラフや同型、さらに最大共通部分グラフや最小共通スーパーグラフについて述べた。

グラフ間の距離を計測する手法については、エントロピーを用いたグラフ距離の計算手法、グラフをあるグラフへ変換するために必要な編集系列のコストにより距離を計測する手法に着いて述べた。

本章の最後である 2.3 節にて、カテゴリを代表するグラフを重み付き最小共通スーパーグラフにより求め、認識対象のグラフとカテゴリの距離を計測することにより、認識を行えることを示した。2.3.3 の最後にも述べたが、学習データを個々に用いるよりも、学習データを代表するグラフを求め、認識を行う方針は有効であること、代表のグラフは認識性能に深く関係することが分かった。

第2章 グラフを用いたパターン認識

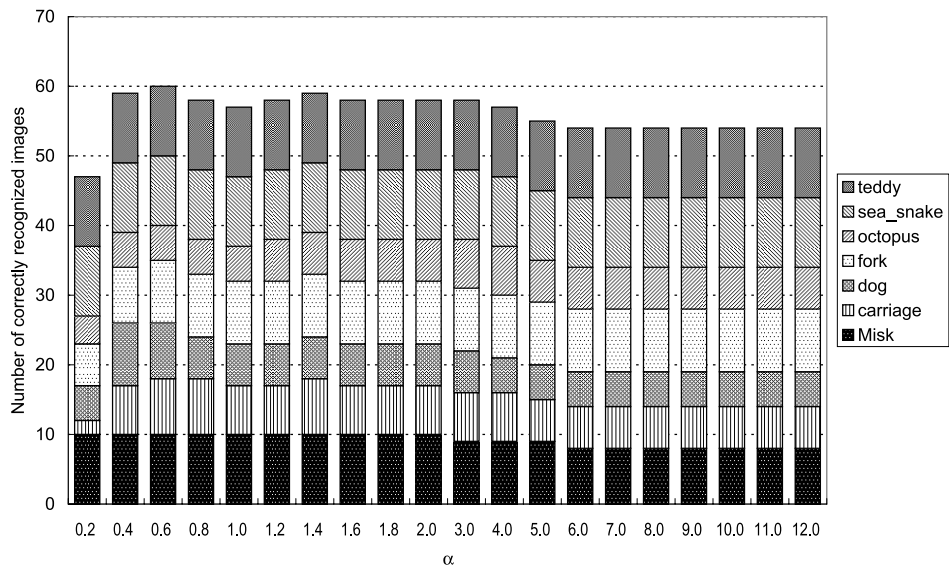


図 2.14: パラメータ a と RWMCS における認識率の変化

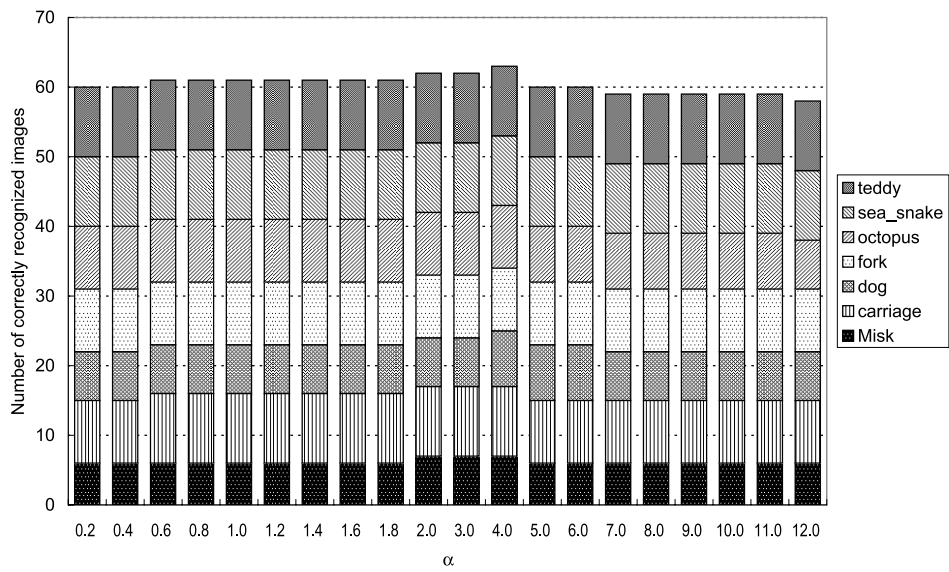


図 2.15: パラメータ a と RWMCS_t における認識率の変化

第3章

集合を代表するグラフ

3.1 はじめに

本章ではグラフの集合を代表するグラフを得る手法を提案する。2.3節で述べたように、グラフの集合を代表するグラフを用いた認識手法は有効であり、集合を代表するグラフの性能は認識性能に大きな影響を及ぼすことが分かっている。グラフの集合を代表するグラフを求めることについて述べている主な文献をあげると、重み付き最小共通スーパーグラフとする手法 [3] や最小記述長基準により代表するグラフを求める手法 [30] などがある。

しかしながら、最小記述長基準により代表のグラフを求める手法は、対象を木グラフとしているため適用できる範囲が限られている。例えば文字を表すグラフはループ構造を含むものがあるため適用できない。一方、重み付き最小共通スーパーグラフはループを持つグラフに適用することができるが、重み付き最小共通スーパーグラフは頂点と辺に関する情報のみから抽出されており、グラフが有するその他の有益な情報、例えば属性値などを無視している。

パターン認識において特徴抽出は重要なプロセスで、際立った特徴を抽出することが重要であるとされている [1]。グラフの集合を代表するグラフを求める際にも、集合の特徴を用いて代表グラフを抽出することができれば、集合の特徴を表すグラフを得ることが期待できる。

第3章 集合を代表するグラフ

本章で提案する代表のグラフを得る方針は、まず、集合を特徴付ける属性値を抽出する。得られた属性値を用いて、集合の各グラフの頂点にラベル付けを行う。ラベルを用いて集合のグラフを統合し、代表のグラフを抽出する。

3.2 集合を特徴付ける属性値の抽出

はじめに集合を特徴付けている属性値をクラスタリングにより抽出する。式 (2.3) により定義された重み付き属性グラフ $G_i = (\mathbf{V}_i, \mathbf{E}_i, \alpha_i, \beta_i, \lambda_i, \epsilon_i)$ を元とする集合を $\mathbf{G} = \{G_1, \dots, G_n\}$ とする。頂点 v がとりうる属性値 l を元とする集合を $\mathbf{L}_v = \{l_1, \dots, l_j, \dots\}$ とすると、 $\alpha : v \rightarrow \mathbf{L}_v$ である。ただし、属性値間の距離は $\mathcal{H} : \mathbf{L}_v \times \mathbf{L}_v \rightarrow \mathbb{R}$ により求めることができるものとする。

集合 \mathbf{G} の各グラフの頂点が有する属性値の集合を \mathbf{A} とする。

$$\mathbf{A} = \{\alpha_i(\mathcal{V}(G_i)) \mid i = 1, \dots, |\mathbf{G}|\}$$

\mathbf{A} に対して ISODATA 法 [8] を適用する。属性間の距離は関数 \mathcal{H} を用いるものとする。ISODATA 法により m 個のクラスタが得られ、それぞれのクラスタの重心を $\mathbf{K} = \{k_1, \dots, k_m\}$ とする。ただし、 $\mathbf{K} \subset \mathbf{L}_v$ である。そして、集合 \mathbf{G} を特徴付ける属性値を $k_i \in \mathbf{k}$ とし、 \mathbf{K} を特徴属性値集合、 k_i を特徴属性値と記述する。

3.3 特徴属性値と頂点の対応関係

特徴属性集合 $k \in \mathbf{K}$ とグラフ $G_i \in \mathbf{G}$ の頂点集合への対応関係を求める。

最初に、 $G_i \in \mathbf{G}$ の頂点から $k \in \mathbf{K}$ へ対応する写像 $X_i : \mathcal{V}(G_i) \rightarrow \mathbf{K}$ を求める。 X_i は式 (3.1) により得る。 X_i は頂点 $v \in \mathcal{V}(G_i)$ が ISODATA によりどのクラスタにクラスタリングされたかを示すものである。

$$X_i : v \in \mathcal{V}(G_i) \rightarrow \arg \min_{k \in \mathbf{K}} \mathcal{H}(k, \alpha_i(v)) \quad (3.1)$$

$\mathbf{X} = \{X_1, \dots, X_{|\mathbf{G}|}\}$ が得られると、特徴属性値 $k \in \mathbf{K}$ に対して、次数 x を持つ頂点 v の重要度を示す関数を式 (3.2) により定義する。 $G \in \mathbf{G}$ の頂点集合のうち k に割り当てられた頂点集合 $\mathbf{N}_k = \{v \in \mathcal{V}(G) \mid X(v) = k\}$ とし、 \mathbf{N}_k^x は \mathbf{N}_k の頂点のうち次数が x の頂点を元とする集合である。

第3章 集合を代表するグラフ

$$T_k(x) = \frac{1}{|\mathbf{G}|} \cdot \sum_{G \in \mathbf{G}} \frac{|N_k^x|}{|N_k|} \quad (3.2)$$

$$N_k = \{v \in \mathcal{V}(G) \mid X(v) = k\} \quad (3.3)$$

$$N_k^x = \{v \in N_k \mid \deg(v) = x\}$$

同様に特徴属性値 k と k' に割り当てられた頂点間を接続する辺の重要度を式 (3.4) により定義する. $\mathbf{E}_{k,k'}$ は k, k' に割り当てられた頂点を接続する辺の集合であり, $\mathbf{E}_{k,k'} \subseteq \mathcal{E}(G)$ である.

$$S(k, k') = \frac{1}{|\mathbf{G}|} \cdot \sum_{G \in \mathbf{G}} \frac{|\mathbf{E}_{k,k'}|}{\|G\|} \quad (3.4)$$

$$\mathbf{E}_{k,k'} = \{(v, v') \in \mathcal{E}(G) \mid (X(v), X(v')) = (k, k') \text{ or } (k', k)\}$$

グラフの頂点集合から特徴属性値への対応を求める. まず, 特徴属性集合 \mathbf{K} からグラフ G_i の頂点集合への対応関係 $C_i : \mathbf{K} \rightarrow \mathcal{V}(G_i)$ を式 (3.5) により求める. $k \in \mathbf{K}$ で, C_i の添字 i はグラフ G_i に関する対応関係であることを示している. N_k は式 (3.3) により得られる頂点集合で, $\deg(v)$ は頂点 v の次数, $N(v)$ は頂点 v に隣接する頂点を元とする集合である (2.1.1 節を参照). もし $N_k = \emptyset$ であるなら $C_i(k) = \phi$ とする. 式 (3.5) により特徴属性集合から各グラフ $G \in \mathbf{G}$ の頂点集合への対応関係が得られる. また, $G_i \in \mathbf{G}$ の頂点集合から \mathbf{K} への写像 X_i と, \mathbf{K} から $\mathcal{V}(G_i)$ への対応関係 C_i に関して, X_i の逆写像は C_i とならない点に留意したい.

頂点の次数と接続する辺の重要度によって, 頂点と特徴属性値への対応を式 (3.6) により定義し, 得られる対応を κ と表記する.

$$C_i : k \in \mathbf{K} \rightarrow \begin{cases} \arg \max_{v \in N_k} \left\{ T_k(\deg(v)) + \sum_{v' \in N(v)} S(X_i(v), X_i(v')) \right\} & \text{if } N_k \neq \emptyset \\ \phi & \text{otherwise} \end{cases} \quad (3.5)$$

$$\kappa : v \rightarrow \begin{cases} k & \text{if } C(k) = v \\ \phi & \text{otherwise} \end{cases} \quad (3.6)$$

3.4 代表グラフの抽出

グラフの集合 $G = \{G_1, \dots, G_n\}$ を代表するグラフを抽出する。代表するグラフは、学習データのすべての頂点と辺に関する情報を有するように設計する。グラフの和 (Graph union) はそのようなグラフを得るためのひとつの方法であるので、グラフの和という概念を用いて代表のグラフを抽出していく。

グラフの和は次のように定義される [31]。ふたつのグラフ $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ があり、 V_1 と V_2 は互いに素であるとする。このとき、 G_1 と G_2 の和 G を式 (3.7) により定義する。

$$\begin{aligned} G &= G_1 \cup G_2 & (3.7) \\ \mathcal{V}(G) &= V_1 \cup V_2 \\ \mathcal{E}(G) &= E_1 \cup E_2 \end{aligned}$$

単純にグラフの和を G に適用して得られるグラフは、 G とほぼ等しくなってしまう。そこで、グラフの和を求める際に、頂点や辺を統合しながらグラフの和を求めることにより、頂点や辺の情報をまとめたグラフを抽出する。 κ により、頂点が写像される特徴属性値 $k \in K$ が同じ頂点を統合する。

統合する頂点の対応関係

統合する頂点の対応関係 $\mathcal{M} : \mathcal{V}(G_1) \rightarrow \mathcal{V}(G_2)$ を式 (3.8) により得る。グラフ G_1, G_2 について $|G_1| \leq |G_2|$ であるとし、 G_1, G_2 の頂点集合 $\mathcal{V}(G_1), \mathcal{V}(G_2)$ から特徴属性集合 K への写像を X_1, X_2 、次数と接続する辺の重要度により得られる $\mathcal{V}(G_i)$ から K への対応を κ_i , $i = 1, 2$ とする。また、 $\mathcal{M} : \mathcal{V}(G_1) \rightarrow \mathcal{V}(G_2)$ の逆対応、 $\mathcal{M}^{-1} : \mathcal{V}(G_2) \rightarrow \mathcal{V}(G_1)$ は式 (3.9) となる。

$$\mathcal{M} : v \rightarrow \begin{cases} u & \text{if } \kappa_1(v), \kappa_2(u) \neq \phi \text{ かつ } \kappa_1(v) = \kappa_2(u) \\ u & \text{if } \kappa_1(v), \kappa_2(u) = \phi \text{ かつ } X_1(v) = X_2(u) \\ \phi & \text{otherwise} \end{cases} \quad (3.8)$$

$$\mathcal{M}^{-1} : u \rightarrow \begin{cases} v & \text{if } \mathcal{M}(v) = u \\ \phi & \text{otherwise} \end{cases} \quad (3.9)$$

式 (3.8) には、以下の三つの場合分けが存在している。

第3章 集合を代表するグラフ

- (1) 頂点 v と頂点 u が κ により同じ特徴属性値に対応する場合, v と u は統合する関係となる
- (2) v は κ により特徴属性値と対応しないが, X によって同じ特徴属性値へ対応する場合, u と v を統合する関係となる
- (3) 上記の三つに当てはまらない場合, v は統合されない

(1) に関しては一意に定めることができるが, (2) は X によって同じ特徴属性値と対応する $\mathcal{V}(G_1)$, $\mathcal{V}(G_2)$ の数の組み合わせが存在するため, 複数の対応関係 \mathcal{M} が存在する. そこで, \mathcal{M} によって統合される辺の数が最大となる \mathcal{M} を選ぶため, 式 (2.12) を \mathcal{M} に適用する. 統合する頂点は式 (3.10) により求めることができる. $\mathbf{F} = \{\mathcal{M}_1, \dots\}$ である.

$$\mathcal{M} = \arg \max_{\mathcal{M} \in \mathbf{F}} \sum_{(v, v') \in \mathcal{E}(G_1)}^{\|G_1\|} \mathbb{I}((\mathcal{M}(v), \mathcal{M}(v')) \in \mathcal{E}(G_2)) \quad (3.10)$$

頂点の組み合わせによりさまざまな \mathcal{M} を抽出することができるので, 2.1.6 で述べたバックトラッキングアルゴリズムを用いて式 (3.10) の \mathcal{M} を探索する. バックトラッキングアルゴリズムによる \mathcal{M} の探索の計算量は, 特徴属性集合の要素数の頂点間の組み合わせは一意に定まるため, 式 (2.14) を基に, 式 (3.11) となる.

$$(|G_2| - |\mathbf{K}|)! \cdot \left(\frac{1}{((|G_2| - |\mathbf{K}|) - (|G_1| - |\mathbf{K}|))!} + \dots + \frac{1}{(|G_2| - |\mathbf{K}| - 1)!} \right) \quad (3.11)$$

統合を用いたグラフの和

統合する頂点の対応関係 \mathcal{M} を用いてグラフ和を求める. グラフ G_1 , G_2 における対応 $\mathcal{M}: \mathcal{V}(G_1) \rightarrow \mathcal{V}(G_2)$ を用いて G_1 , G_2 の和 G を得る操作を

$$G = +(G_1, G_2, \mathcal{M}) \quad (3.12)$$

と表記する. \mathcal{M} から統合される辺の対応関係を $\hat{\mathcal{M}}$ とし, 式 (3.13) となる.

$$\hat{\mathcal{M}}: (v, v') \in \mathcal{E}(G_1) \rightarrow \begin{cases} (u, u') \in \mathcal{E}(G_2) & \text{if } \mathcal{M}(v) = u \wedge \mathcal{M}(v') = u' \\ \phi & \text{otherwise} \end{cases} \quad (3.13)$$

G の頂点集合は式 (3.14), 辺集合は式 (3.15) と定義する. 式 (3.14) の \mathbf{V}_1 は $\mathcal{V}(G_1)$ のうち統合されない頂点の集合, \mathbf{V}_2 は $\mathcal{V}(G_2)$ のうち統合されない頂点の集合, \mathbf{V}_3 は $\mathcal{V}(G_1)$ のうち統合される頂点の集合となる. 辺についても同様で式 (3.15) における右辺の \mathbf{E}_1

第3章 集合を代表するグラフ

は, $\mathcal{E}(G_1)$ のうち統合されない辺集合, \mathbf{E}_2 は $\mathcal{E}(G_2)$ のうち統合されない辺集合, \mathbf{E}_3 は $\mathcal{E}(G_1)$ のうち統合される辺集合である.

$$\mathcal{V}(G) = \mathbf{V}_1 \cup \mathbf{V}_2 \cup \mathbf{V}_3 \quad (3.14)$$

$$\mathcal{E}(G) = \mathbf{E}_1 \cup \mathbf{E}_2 \cup \mathbf{E}_3 \quad (3.15)$$

$$\mathbf{V}_1 = \{v \in \mathcal{V}(G_1) \mid \mathcal{M}(v) = \phi\}$$

$$\mathbf{V}_2 = \{u \in \mathcal{V}(G_2) \mid \mathcal{M}^{-1}(u) = \phi\}$$

$$\mathbf{V}_3 = \{v \in \mathcal{V}(G_1) \mid \mathcal{M}(v) \neq \phi\}$$

$$\mathbf{E}_1 = \{(v, v') \in \mathcal{E}(G_1) \mid \hat{\mathcal{M}}^{-1}((v, v')) = \phi\}$$

$$\mathbf{E}_2 = \{(u, u') \in \mathcal{E}(G_2) \mid \hat{\mathcal{M}}^{-1}((u, u')) = \phi\}$$

$$\mathbf{E}_3 = \{(v, v') \in \mathcal{E}(G_1) \mid \mathcal{M}((v, v')) \neq \phi\}$$

G の頂点と属性値と重みは式 (3.16), 式 (3.17), 辺の属性値と重みは式 (3.18), 式 (3.19) により求める. E は属性値の平均を計算する関数とし, G の頂点と G_1, G_2 の頂点との対応を $f_1: \mathcal{V}(G) \rightarrow \mathcal{V}(G_1)$, $f_2: \mathcal{V}(G) \rightarrow \mathcal{V}(G_2)$ とし, G の辺と G_1, G_2 の辺との対応を $\zeta_1: \mathcal{E}(G) \rightarrow \mathcal{E}(G_1)$, $\zeta_2: \mathcal{E}(G) \rightarrow \mathcal{E}(G_2)$ とする. $w \in \mathcal{V}(G)$, $v \in \mathcal{V}(G_1)$, $u \in \mathcal{V}(G_2)$ とする.

$$\alpha(w) = \begin{cases} \alpha_1(v) & \text{if } f_1(w) = v \wedge f_2(w) = \phi \\ \alpha_2(u) & \text{if } f_1(w) = \phi \wedge f_2(w) = u \\ E(\alpha_1(v), \alpha_2(u)) & \text{if } f_1(w) = v \wedge f_2(w) = u \end{cases} \quad (3.16)$$

$$\lambda(w) = \begin{cases} \lambda_1(v) & \text{if } f_1(w) = v \wedge f_2(w) = \phi \\ \lambda_2(u) & \text{if } f_1(w) = \phi \wedge f_2(w) = u \\ \lambda_1(v) + \lambda_2(u) & \text{if } f_1(w) = v \wedge f_2(w) = u \end{cases} \quad (3.17)$$

$$\beta((w, w')) = \begin{cases} \beta_1((v, v')) & \text{if } \zeta_1((w, w')) = (v, v') \wedge \zeta_2((w, w')) = \phi \\ \beta_2((u, u')) & \text{if } \zeta_1((w, w')) = \phi \wedge \zeta_2((w, w')) = (u, u') \\ E(\beta_1((v, v')), \beta_2((u, u'))) & \text{if } \zeta_1((w, w')) = (v, v') \wedge \zeta_2((w, w')) = (u, u') \end{cases} \quad (3.18)$$

```

1.  $G$  から特徴属性集合  $K$  を抽出する
2.  $\mathcal{U} = G_1$  とする
3. for  $i = 2$  to  $|G|$ 
4.     統合する頂点の対応関係  $\mathcal{M}$  を計算する.
5.      $\mathcal{U} = +(G_i, \mathcal{U}, \mathcal{M})$ 
6. end for
    
```

図 3.1: 集合を代表するグラフを抽出するアルゴリズム

$$\epsilon((w, w')) = \begin{cases} \epsilon_1((v, v')) & \text{if } \zeta_1((w, w')) = (v, v') \wedge \zeta_2((w, w')) = \phi \\ \epsilon_2((u, u')) & \text{if } \zeta_1((w, w')) = \phi \wedge \zeta_2((w, w')) = (u, u') \\ E(\epsilon_1((v, v')), \epsilon_2((u, u'))) & \text{if } \zeta_1((w, w')) = (v, v') \wedge \zeta_2((w, w')) = (u, u') \end{cases} \quad (3.19)$$

グラフの集合 G を代表するグラフは繰り返し $G \in G$ の和を求める事により抽出する。 G を代表するグラフを $\mathcal{U}(G)$ と表記する。 $\mathcal{U}(G)$ を得るアルゴリズムを図 3.1 に示す。

3.5 評価実験

本節では代表するグラフを抽出し、代表グラフを用いて認識する実験を行った。

3.5.1 実験で用いたデータ

実験には IAM Graph Database [9] と飾り文字画像のふたつの実験データを用いた。

IAM Graph Database

IAM Graph Database はグラフにおけるパターン認識・学習手法を評価する標準的なデータベースである。IAM Graph Database は様々な物体から抽出したグラフがあり、本実験では IAM Graph Database; Letter (文字のグラフ) を用いた。IAM Graph Database; Letter のグラフは重み付き属性グラフとして扱う。グラフの頂点は平面上の直交座標系の座標を属性値として有している。辺の属性値はない。今後、グラフを図示するとき、平面上の点は頂点を表すことに留意する。IAM Graph Database; Letter は変形の強さが異

第3章 集合を代表するグラフ

なる文字グラフのデータセット (Low, Medium, High) があり, それぞれ学習データ 750 個, テストデータ 750 個を有している. 文字グラフのクラスは, 直線で構成されているアルファベットの大文字で, A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z である. 各クラスは 50 個の学習データ, 50 個のテストデータを有している. 文字グラフの例を図 3.2, 図 3.3, 図 3.4 に示す.

飾り文字画像のグラフ

飾り文字画像はアルファベットの大文字 A Z に対して, 文字ごとに 10 種類の飾りを加えた画像で, 260 枚の画像を用いた. そのうち 130 枚を学習データ, その他 130 枚をテストデータとして用いた. 飾り文字画像からグラフを抽出する手法について簡単に述べる. まず, マルチスケール解析に基づく手法 [32] により, 飾り文字画像を文字の大局的な構造を表す画像に変換する. マルチスケール解析に基づく手法は, 飾り文字画像にぼかしを与えた画像の尾根という特徴 [34] の部分のうち, 濃淡値が高い尾根部分は文字の大局的な構造を表し, 濃淡値が小さい尾根部分は文字に加えられた飾りにより生じる局所的な文字構造を表すことを利用し, 原画像からストロークの突出している箇所など重要な構造情報を抽出する. 抽出された重要な構造を大局的な構造という. 次に, 大局的な構造を表す画像からグラフを抽出する. ストロークを辺, ストロークの交点や端点, 大きく曲がっている変曲点を頂点とする. 大局的な構造を表す画像に細線化処理 [33] を施すことにより得られる線分からグラフを抽出する. 細線化手法 [33] は上と左方向, 下と右方向から並列に細線化を行うもので, 輪郭に対する中心線が細線化画像として得られる. 飾り文字画像と大局的な構造を表す画像, 飾り文字画像から得られたグラフをそれぞれ, 図 3.5 と図 3.6, 図 3.7 に示す.

第3章 集合を代表するグラフ

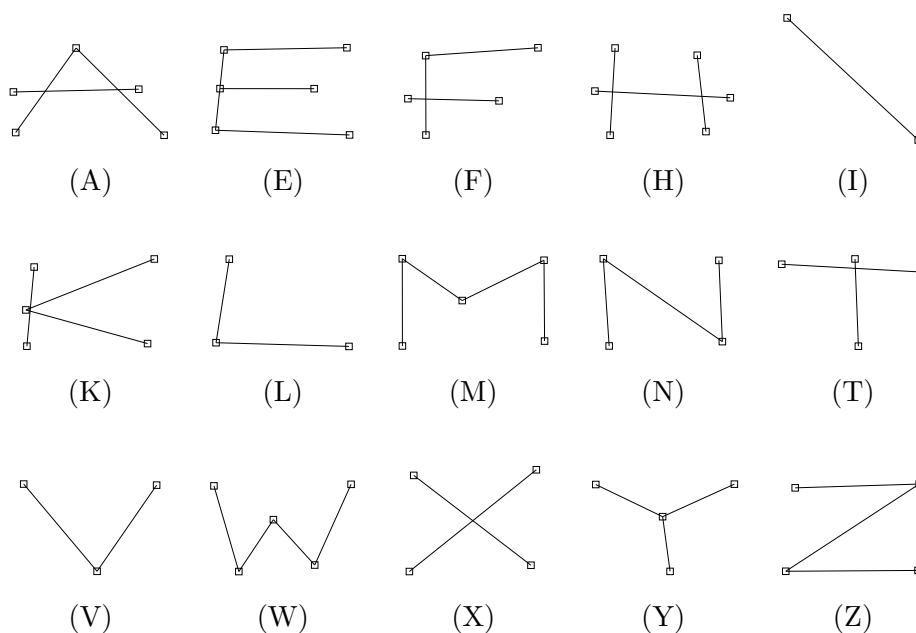


図 3.2: IAM Graph Database; Letter. 変形の強さは Low

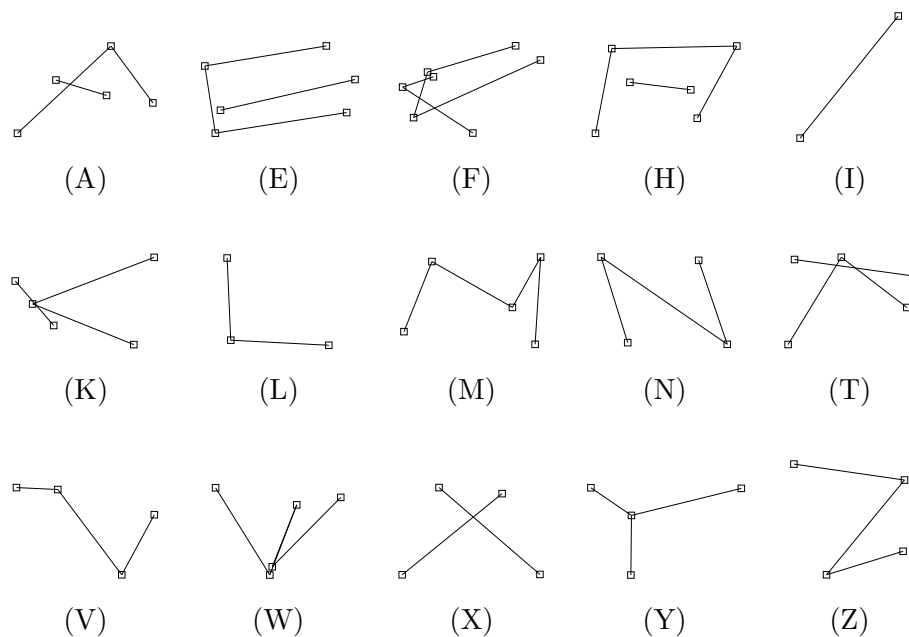


図 3.3: IAM Graph Database; Letter. 変形の強さは Medium

第3章 集合を代表するグラフ

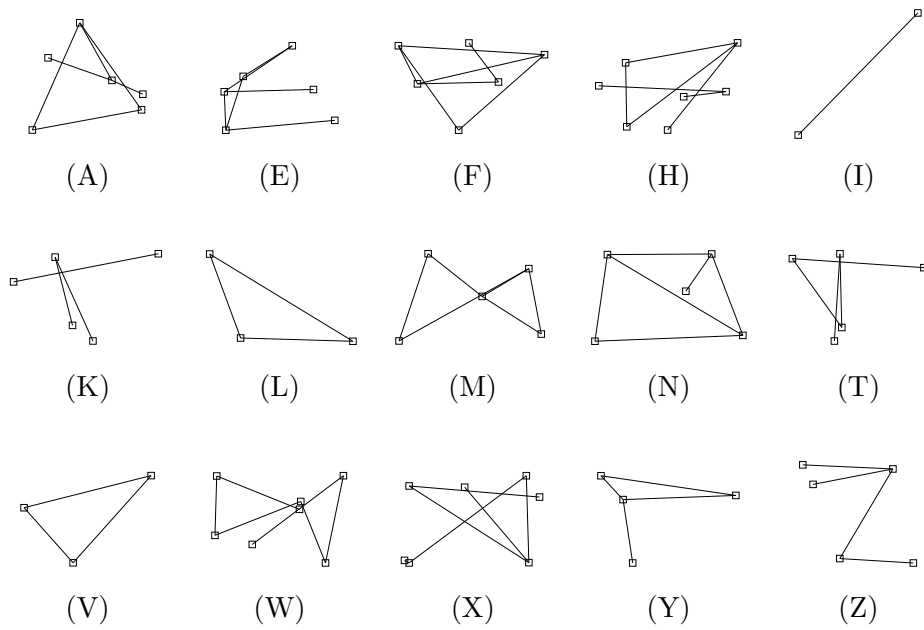


図 3.4: IAM Graph Database; Letter. 変形の強さは High

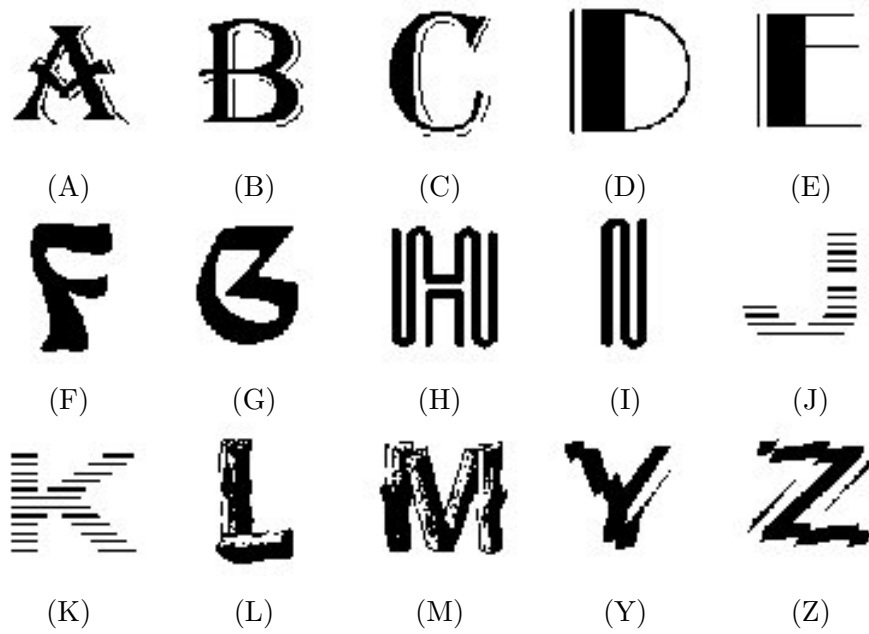


図 3.5: 飾り文字画像

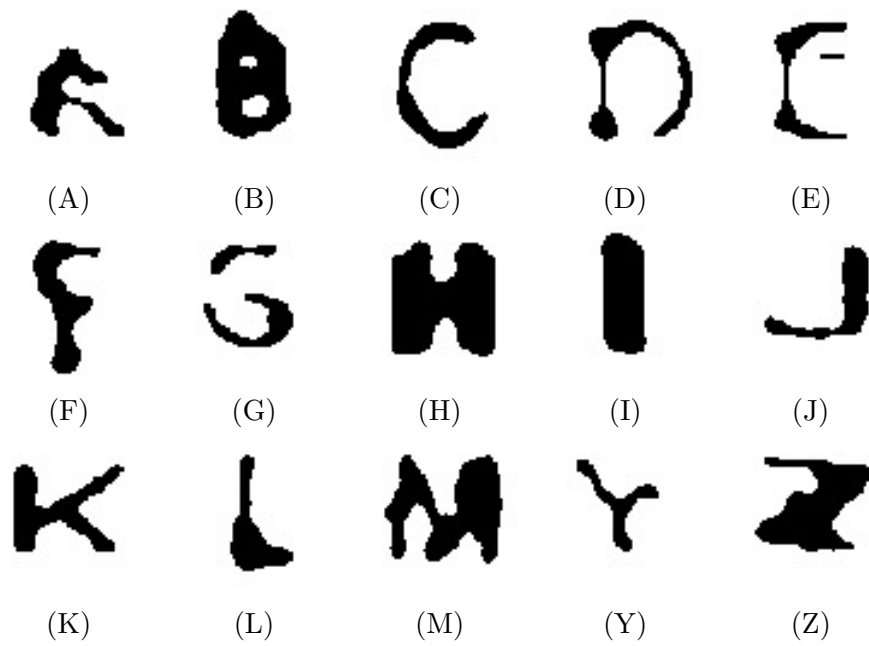


図 3.6: 大局的な構造を表す画像

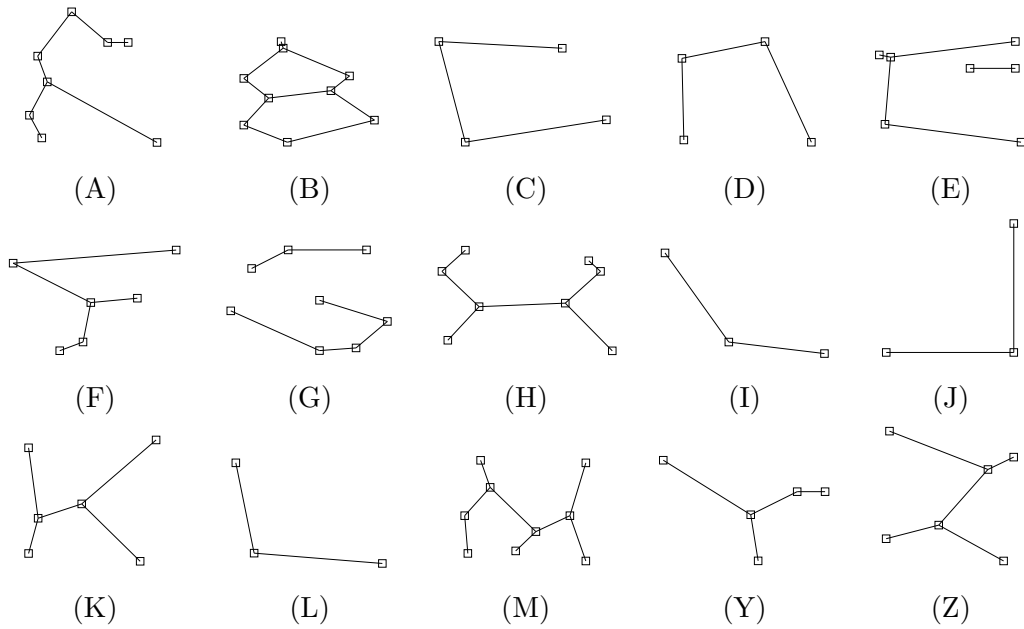


図 3.7: 飾り文字画像から抽出されたグラフ

3.5.2 代表グラフの抽出実験

IAM Graph Database; Letter と飾り文字画像のグラフについて、文字のクラスごとに学習データを作成し、代表グラフを抽出する。図3.1に示したアルゴリズムにより代表のグラフを抽出するが、抽出アルゴリズムは逐次的に処理を行うものなので、学習データに適用する順番によって異なる代表グラフが得られる。そこで、学習データをランダムに並び替え代表グラフを抽出する。IAM Graph Database; Letter については100回の抽出実験を行い、飾り文字画像については10回の抽出実験を行った。また、得られた代表のグラフの重みについて、閾値以下の重みの頂点と辺を除去した。本実験ではIAM Graph Database; Letter に対して閾値は5、飾り文字画像に対して閾値を1とした。

抽出された代表のグラフを示す。IAM Graph Database; Letter の Low, Medium, High をそれぞれ図3.8, 図3.9, 図3.10に示す。また、飾り文字画像から得られた代表のグラフを図3.11に示す。代表のグラフはさまざまな構造をしており、特にIAM Graph Database; Letter において、変形のレベルが強くなるに従って、代表グラフのグラフ構造が複雑になっていることが分かる。

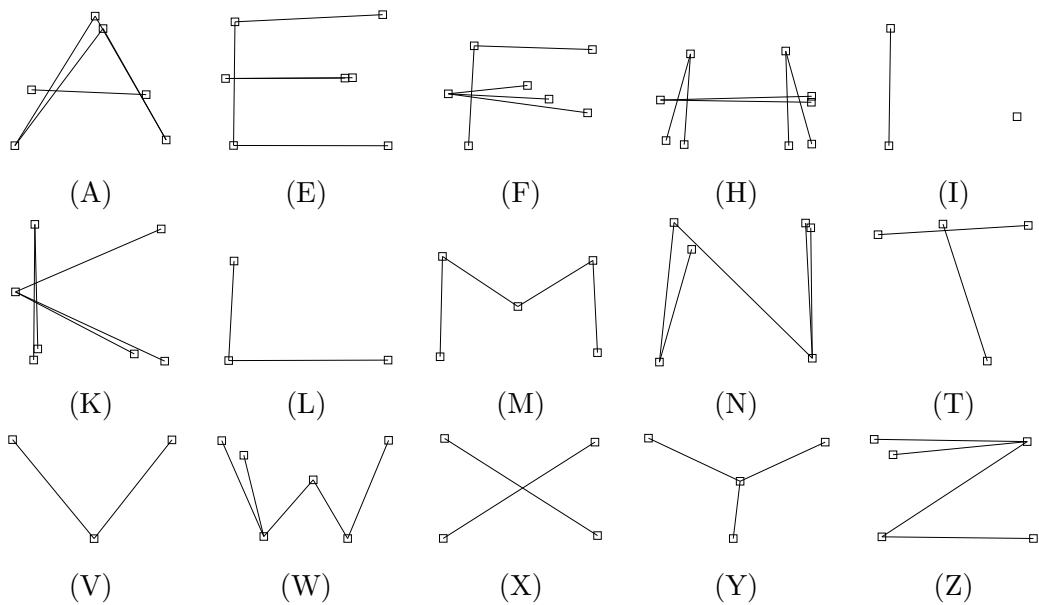


図 3.8: 代表グラフ. IAM Graph Database; Letter, Low

第3章 集合を代表するグラフ

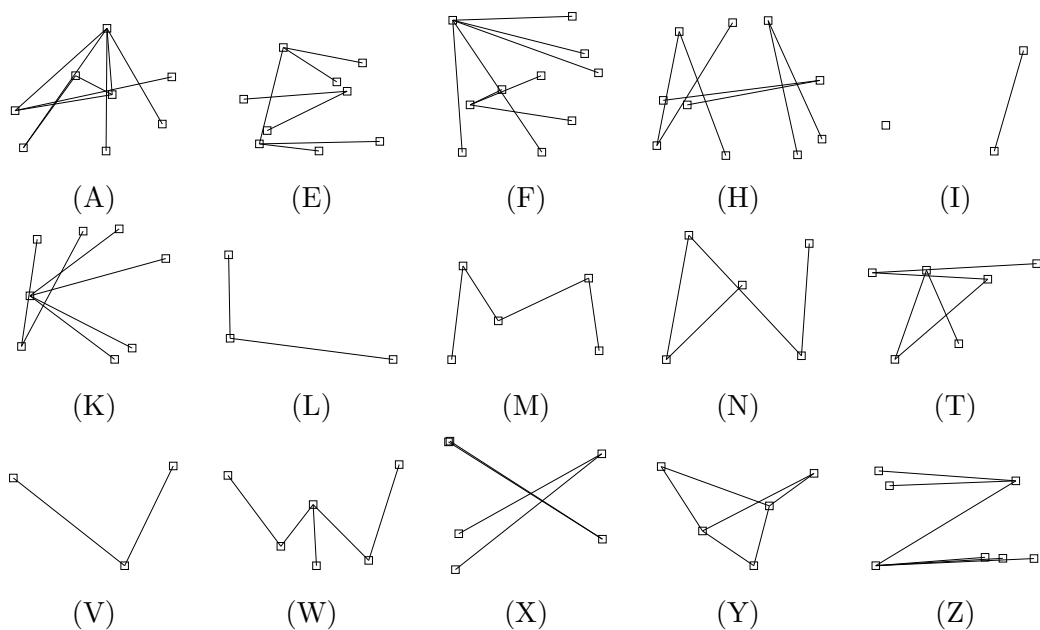


図 3.9: 代表グラフ. IAM Graph Database; Letter, Medium

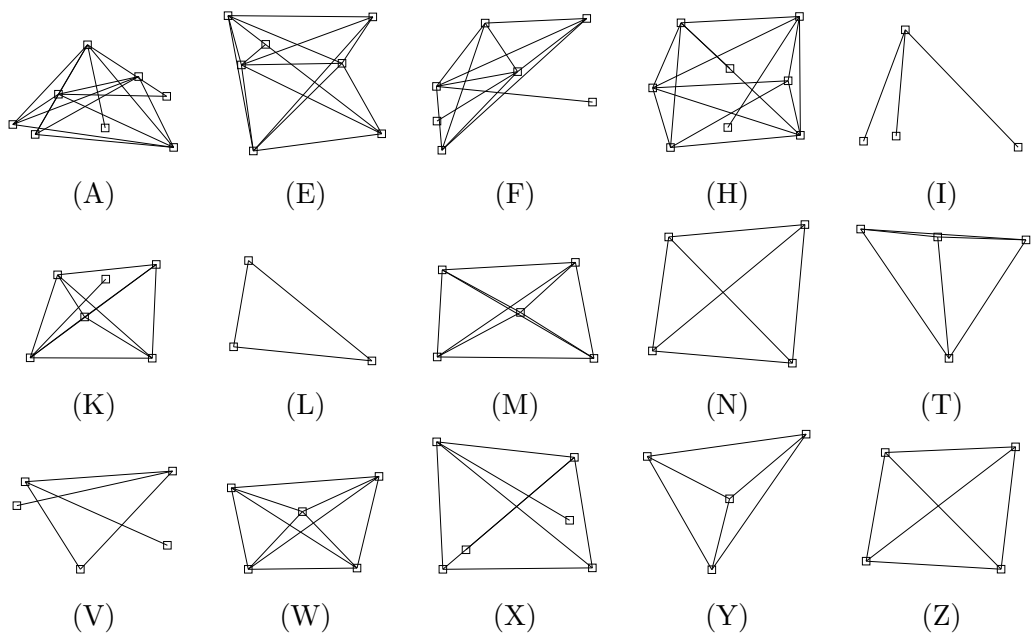


図 3.10: 代表グラフ. IAM Graph Database; Letter, High

第3章 集合を代表するグラフ

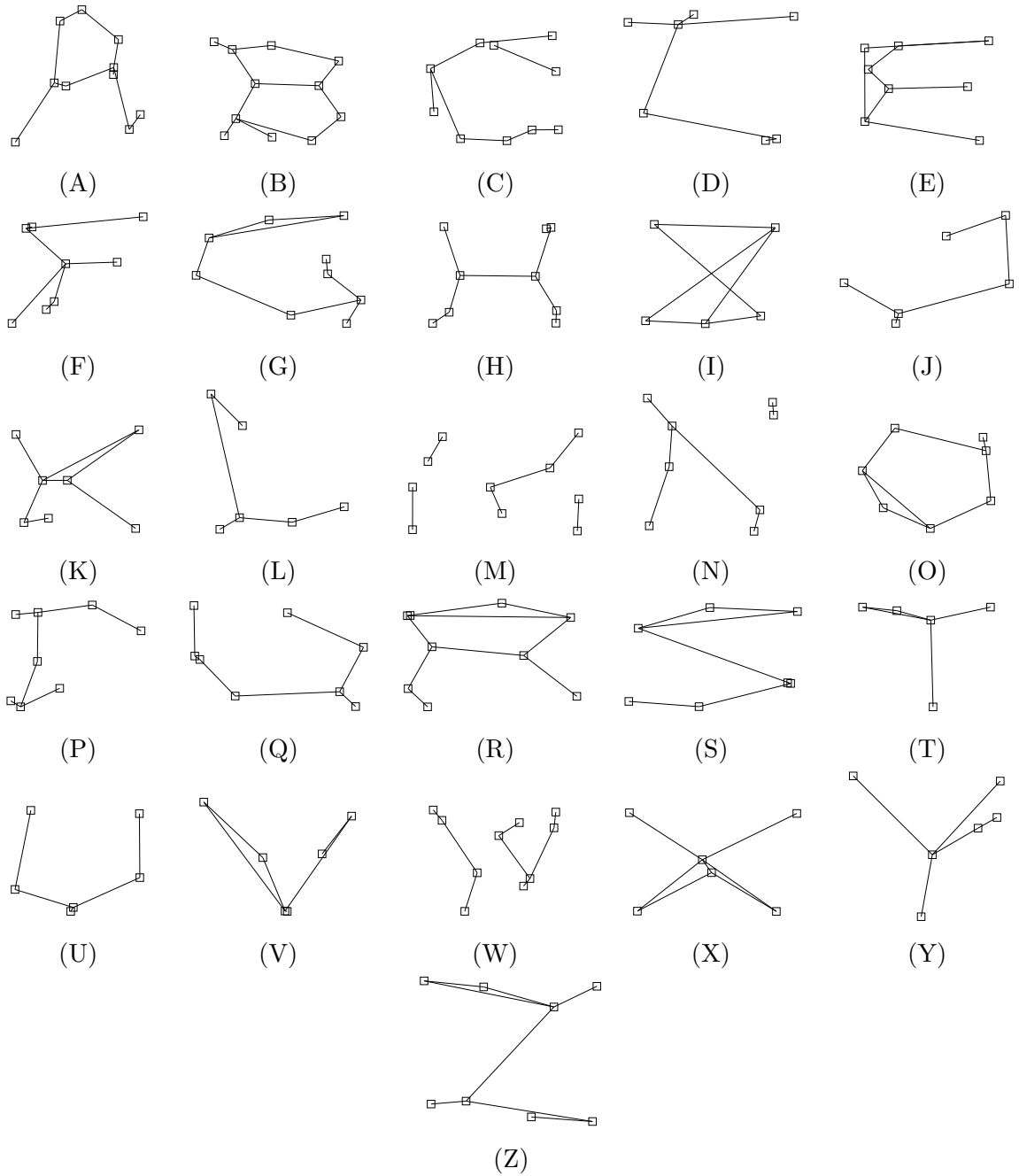


図 3.11: 飾り文字画像から得た代表のグラフ

第3章 集合を代表するグラフ

3.5.3 認識実験

本節では、文字を代表するグラフを用いた認識実験を行う。代表のグラフとテストデータとの距離を計測し、認識に用いる。代表のグラフを \mathcal{U} 、認識対象のグラフ G とすると、 \mathcal{U} と G の距離は、式 (2.24)、式 (3.12) を用いて式 (3.20) により定義する。 Q は \mathcal{M} により対応する辺の整合度を計算する関数である。 \angle はふたつの辺が作る角度とする。式 (3.23) と式 (3.24) は辺の長さを表している。

$$d(\mathcal{U}, G) = \text{Ent}(\mathcal{U}(G, \mathcal{U}, \mathcal{M})) - \text{Ent}(\mathcal{U}) + Q(G, \mathcal{U}, \mathcal{M}) \quad (3.20)$$

$$Q(G, \mathcal{U}, \mathcal{M}) = \sum_{(v, v') \in \mathcal{E}(G)} \hat{Q}((v, v'), G, \mathcal{U}, \mathcal{M}) \quad (3.21)$$

$$\hat{Q}((v, v'), G, \mathcal{U}, \mathcal{M}) = \begin{cases} l_i^2 l_j^2 - 2l_i l_j \cos(\angle) & \text{if } (\mathcal{M}(v), \mathcal{M}(v')) \in \mathcal{E}(\mathcal{U}) \\ l_i^2 & \text{otherwise} \end{cases} \quad (3.22)$$

$$l_i = \mathcal{H}(\alpha(v), \alpha(v'))^2 \quad (3.23)$$

$$l_j = \mathcal{H}(\alpha_{\mathcal{U}}(\mathcal{M}(v)), \alpha_{\mathcal{U}}(\mathcal{M}(v')))^2 \quad (3.24)$$

認識実験においてテストデータと文字の距離は式 (3.20) を用いて測定する。まず、飾り文字画像のグラフについて認識実験を行った。学習データをランダムに並び替えて抽出した代表グラフは 10 パターンあり、テストデータは正解文字との距離が N 位以内であれば正解とした。 $N = 1, \dots, 5$ 位までの認識率の平均、最低値、最高値を図 3.12 に示す。 $N = 1, \dots, 5$ における平均認識率は、25.3, 37.6, 46.2, 52.5, 58% となった。

同様に IAM Graph Database のグラフを用いて認識実験を行った。ランダムに並び替えて抽出した代表グラフの数は 100 である。テストデータを最近傍の代表グラフの属する文字へと認識する。認識率の平均値、最低値、最高値を図 3.13 に示す。Low, Medium, High の各データセットにおける平均認識率は 94.2, 78.8, 80.1% であった。

第3章 集合を代表するグラフ

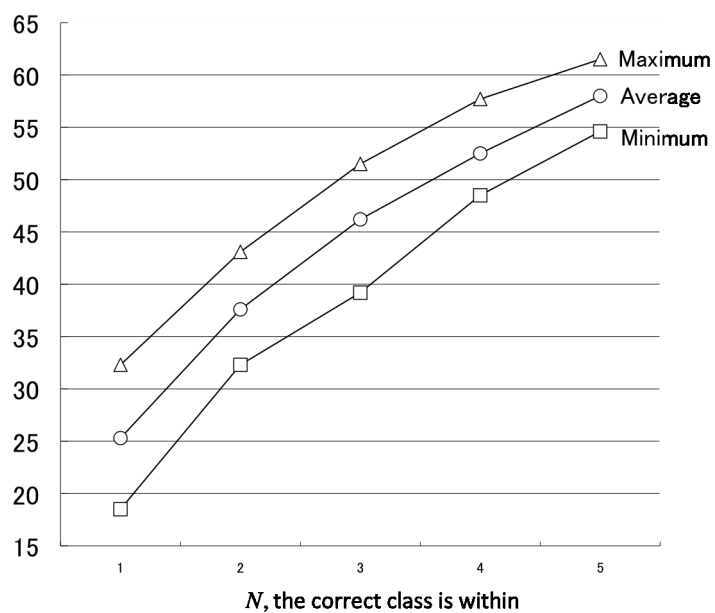


図 3.12: 飾り文字画像における認識結果

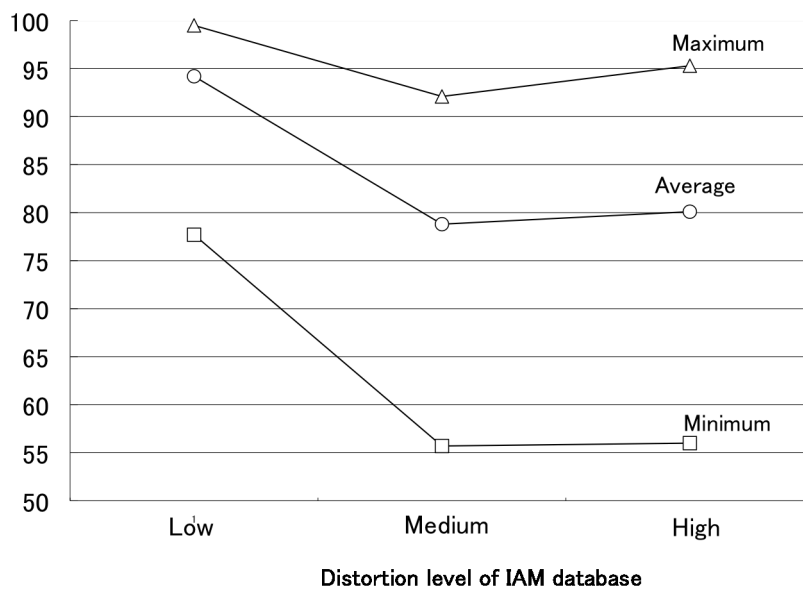


図 3.13: IAM Graph Database における認識結果

3.5.4 比較実験

代表グラフの性能に関して WMCS と比較を行う。IAM Graph Database と飾り文字画像のグラフについて、図 2.4 に示したアルゴリズムにより WMCS を求める。WMCS を抽出する際に、最大共通部分グラフを求める必要がある。最大共通部分グラフは頂点 $v \in \mathcal{V}(G_1)$ と頂点 $u \in \mathcal{V}(G_2)$ との組み合わせを、バックトラッキングアルゴリズムにより探索する。WMCS の計算には莫大な計算時間が必要で、100 パターンや 10 パターンの WMCS を計算することが困難であるから、あらかじめ探索の前に

$$\mathcal{H}(\alpha_1(v), \alpha_2(u)) > t$$

となる頂点の組み合わせは除くようにした。本論文では実験的に $t = 0.3$ と定めた。テストデータと WMCS との距離を、式 (2.24) と辺の整合性を考慮して式 (3.25) により計測した。 $f: \mathcal{V}(G) \rightarrow \mathcal{V}(\text{WMCS})$ とし、式 (3.27) の l_i, l_j は式 (3.23), 式 (3.24) により求めるものとした。テストデータは WMCS との距離が最近傍の WMCS の属する文字と認識した。

$$d(\text{WMCS}, G) = \Delta\text{Ent}(\text{WMCS}, G) + Q'(\text{WMCS}, G, f) \quad (3.25)$$

$$Q'(\text{WMCS}, G, f) = \sum_{(v, v') \in \mathcal{E}(G)} \hat{Q}'((v, v'), \text{WMCS}, G, f) \quad (3.26)$$

$$\hat{Q}'((v, v'), \text{WMCS}, G, f) = \begin{cases} l_i^2 l_j^2 - 2l_i l_j \cos(\angle) & \text{if } (f(v), f(v')) \in \mathcal{E}(\text{WMCS}) \\ l_i^2 & \text{otherwise} \end{cases} \quad (3.27)$$

WMCS と提案手法の認識率を比較したものを図 3.14 に示す。図中の Proposed method は本章で述べてきた手法により得た結果で、Decorative characters は飾り文字画像である。WMCS を用いた場合、飾り文字画像、Low, Medium, High における認識率はそれぞれ、32.8, 94, 64.3, 67.6% となった。同様に提案手法を用いた場合、飾り文字画像、Low, Medium, High における認識率は、25.3, 94.2, 78.8, 80.1% を得た。飾り文字画像における認識率は WMCS, 提案法ともに 30% 前後と高いとは言えない認識率となった。これは飾り文字画像のグラフは変形が激しいたえであると考えられる。同一文字から得たグラフであっても飾りが異なれば、抽出されるグラフに共通した属性や構造は見つけることが困難であることが分かった。飾り文字画像からグラフを抽出する手法を再検討する必要がある。一方で、IAM Graph Database においては WMCS, 提案法ともに

第3章 集合を代表するグラフ

良好な認識率を得た。Low においては両手法ともに 90%以上の認識率を得た。Medium, High において、提案法は WMCS を上回っており、提案法の優位性を示している。

各データセットにおいて代表グラフ，WMCS を求めた計算時間を図 3.15 に示す。飾り文字画像，Low, Medium, High における WMCS の計算時間はそれぞれ，3.4, 1.8, 12.5, 4.6 (秒)であった。提案法を用いた場合の飾り文字画像，Low, Medium, High における計算時間はそれぞれ，0.1, 0.2, 0.3, 0.2 (秒)であった。提案法は WMCS と比較して，飾り文字画像において約 34 倍，IAM Graph Database において 25 倍，処理時間を高速化することができたことが分かった。

3.6 まとめ

本章はグラフの集合を代表するグラフを抽出する手法について述べた。グラフの集合から，特徴を表す属性値を抽出し，頂点が同じ特徴の属性値であるものを統合することにより，代表するグラフを構築した。

本章で提案した手法は，グラフにおけるパターン認識・学習手法を評価する標準的なデータベースである IAM Graph Database において，おおよそ 80%の認識率を得た。また代表のグラフを得る既存手法である WMCS と比較して，提案法が優れていることが実験結果から明らかとなった。さらに，提案法は WMCS と比較して高速に代表グラフを計算することができることを実験により示した。

第3章 集合を代表するグラフ

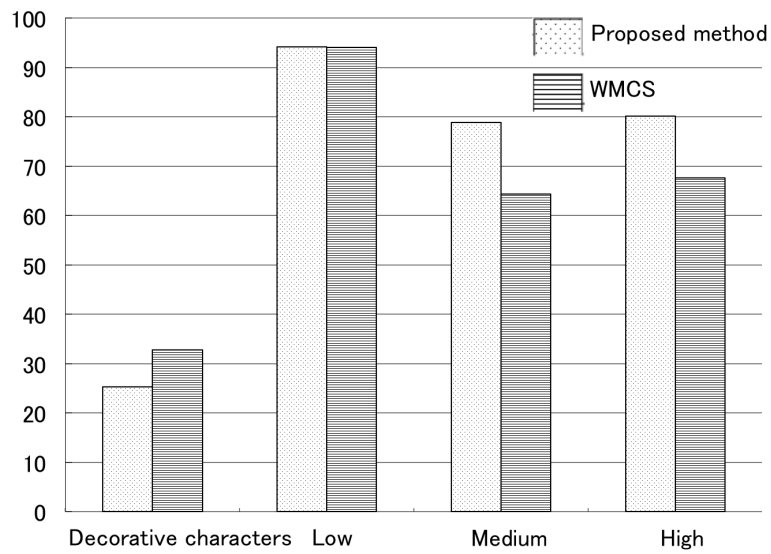


図 3.14: 認識率の比較. 横軸はデータセットの名前

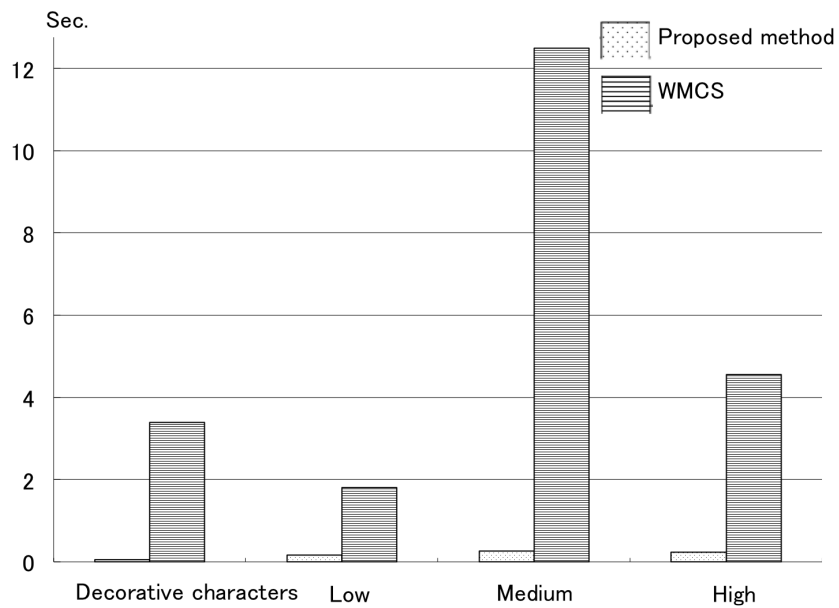


図 3.15: 計算時間の比較. 横軸はデータセットの名前

第4章

グラフの確率モデル

4.1 はじめに

本章ではグラフの集合を記述する確率モデルを構築する。グラフが確率モデルから生成される尤度関数を定義し、認識に用いる。

確率的にグラフを与えるものとしてランダムグラフがある [10, 11, 20]。ランダムグラフとは、頂点集合と辺集合が何らかのランダムなメカニズムに従って生成されるグラフである。例えば、 N 個の頂点に確率 p で辺が挿入されるものである。

ランダムグラフの頂点が生成される確率や、辺が挿入される確率を用いれば集合を記述することができる。また、頂点や辺が有する属性値も何らかの確率分布で表現すれば、さらに集合の詳細を捉えた確率モデルとなる。本章では集合の確率モデルを構築する手法を提案し、グラフがそのモデルから生成される尤度関数を定義する。

また、確率モデルを用いてグラフを高精度に認識することができることを示す。

4.2 確率モデル

確率モデルを \mathcal{R} と表記する。 \mathcal{R} は式 (2.1) により定義されるグラフ構造を有し、頂点や辺が、集合のサンプルにおいて観測される確率もまた有するものとする。さらに頂点や辺がとりうる属性値を確率分布により記述する。集合の確率モデル \mathcal{R} を式 (4.1) によ

第4章 グラフの確率モデル

り定義する.

$$\mathcal{R} = (\mathbf{V}, \mathbf{E}, \theta, \hat{\theta}, P, \hat{P}) \quad (4.1)$$

確率モデルの頂点と辺は、観測される確率を有しており、頂点の観測確率を θ 、辺の観測される確率を $\hat{\theta}$ とする. $\theta : \mathcal{V}(\mathcal{R}) \rightarrow \{x \mid 0 \leq x \leq 1\}$, $\hat{\theta} : \mathcal{E}(\mathcal{R}) \rightarrow \{x \mid 0 \leq x \leq 1\}$ である. また頂点と辺における属性値の確率分布を P , \hat{P} と表記する. 頂点, 辺がとりうる属性値の集合をそれぞれ \mathbf{L}_v , \mathbf{L}_e とすると, $P : \mathcal{V}(\mathcal{R}) \times \mathbf{L}_v \rightarrow \{x \mid 0 \leq x \leq 1\}$, $\hat{P} : \mathcal{E}(\mathcal{R}) \times \mathbf{L}_e \rightarrow \{x \mid 0 \leq x \leq 1\}$ である. 例えば $P(v, l_v)$ は, 頂点 $v \in \mathcal{V}(\mathcal{R})$ が属性値 l_v を有する確率を示す.

4.3 確率モデルの構築

グラフの集合 $\mathbf{G} = \{G_1, \dots, G_n\}$ の確率モデル \mathcal{R} を構築する手法を提案する. G_i は属性グラフで, G_i に割り当てられた重みを式 (4.2) により表す. \mathbb{R}^+ は正の実数である.

$$W : G_i \rightarrow \mathbb{R}^+ \quad (4.2)$$

4.3.1 確率モデルの頂点との対応関係

本節では、確率モデル \mathcal{R} の頂点集合と、 \mathbf{G} の頂点と $\mathcal{V}(\mathcal{R})$ の対応関係を求める.

\mathbf{G} のグラフ G の頂点集合から頂点をいくつか抽出し、確率モデルの頂点とする. 確率モデルの頂点集合に頂点を与えたグラフを基準グラフと呼ぶ. 基準グラフを用いて、グラフの頂点集合から確率モデルの頂点集合への写像を求める. \mathbf{G} の頂点集合から \mathcal{R} の頂点集合への写像を求める手順を図 4.1 に示す. ただし、基準グラフは以下のように選択する.

$$G_j = \arg \min_{G_j \in \mathbf{G}} \left\{ \frac{1}{|\mathbf{G}|} \cdot \sum_{G_i \in \mathbf{G}} \text{ED}(G_j, G_i) \right\} \quad (4.3)$$

図 4.1 の処理を終えると、確率モデルの頂点集合 $\mathcal{V}(\mathcal{R}) = \{u_1, \dots\}$, $G_i \in \mathbf{G}$ の頂点集合から $\mathcal{V}(\mathcal{R})$ への写像である $f_i \in \mathbf{F}$, $\mathcal{V}(\mathcal{R})$ へ頂点を与えたグラフの集合である \mathbf{H} , H_j の頂点集合から $\mathcal{V}(\mathcal{R})$ への写像 $l_j \in \mathbf{L}$ を得ることができる. \mathbf{H} は \mathcal{R} の基準グラフであり, \mathbf{L} を基準グラフ写像と表記する.

K から基準グラフの集合 $\mathbf{H} = \{H_1, \dots, H_m\}$ への編集系列 $P_{k,j}, j = 1, \dots, m$, $\mathbf{L} = \{l_1, \dots, l_m\}$ を用いて、グラフ K の頂点集合から $\mathcal{V}(\mathcal{R})$ への対応 f を求める手順を図 4.2 に示す.

第4章 グラフの確率モデル

入力.	$\mathcal{V}(\mathcal{R}) = \emptyset, \mathbf{H} = \emptyset, \mathbf{F} = \{f_1, \dots, f_n\}$
出力.	$\mathcal{V}(\mathcal{R}), \mathbf{H}, \mathbf{F}, \mathbf{l}$
1.	初期化. $j = 1, \mathcal{V}(\mathcal{R}) = \emptyset, \mathbf{H} = \emptyset, f_i : \mathcal{V}(G_i) \rightarrow \phi, i = 1, \dots, n, \mathbf{l} = \emptyset.$
2.	最初の基準グラフ $G_j \in \mathbf{G}$ を選ぶ.
3.	$\mathcal{V}(\mathcal{R}) = \mathcal{V}(\mathcal{R}) \cup \{v \in \mathcal{V}(G_j) \mid f_j(v) = \phi\}$
4.	$l_j : \mathcal{V}(G_j) \rightarrow \mathcal{V}(\mathcal{R})$ を以下の式により求める $l_j : v \in \mathcal{V}(G_j) \rightarrow \begin{cases} w \in \mathcal{V}(\mathcal{R}) & \text{if } f_j(v) = \phi \\ \phi & \text{if } f_j(v) \neq \phi \end{cases}$
5.	G_j から G_i への編集系列 $\mathbf{P}_{j,i}$ を得る.
6.	$\mathbf{P}_{j,i}$ で置換された頂点間を対応させる $f_i : u \in \mathcal{V}(G_i) \rightarrow l_j(v \in \mathcal{V}(G_j)) \text{ if } (f_i(u) = \phi) \wedge (\exists \{v \rightarrow u\} \in \mathbf{P}_{j,i})$
7.	if $\neg \exists u \in \mathcal{V}(G_i); f_i(u) = \phi$, then stop
8.	else $\mathbf{H} = \mathbf{H} \cup G_j, j = j + 1.$
9.	新しい基準グラフ $G_j \in \mathbf{G}; G_j \notin \mathbf{H}$ を選び, goto 3

図 4.1: 確率モデルの頂点集合と、頂点集合と確率モデルの頂点集合との対応関係を得る手順

4.3.2 確率モデルの頂点 (辺) の観測確率と属性値の確率

\mathcal{R} の頂点集合から \mathbf{G} の頂点集合への対応 \hat{f} より θ を計算することができる.

$$\hat{f}_i : \mathcal{V}(\mathcal{R}) \rightarrow \mathcal{V}(G_i), \quad i = 1, \dots, n \quad (4.4)$$

$$\hat{\zeta}_i : \mathcal{E}(\mathcal{R}) \rightarrow \mathcal{E}(G_i), \quad i = 1, \dots, n \quad (4.5)$$

頂点 $v \in \mathbf{V}$ が観測される確率 θ は, \mathbf{G} とグラフに割り当てられた重み W , 式 (4.4) の対応 \hat{f} を用いて式 (4.6) により定義する. \mathbb{I} は式 (2.13) である. 同様に辺 (v, v') が観測される確率 $\hat{\theta}$ は式 (4.5) を用いて, 式 (4.7) により定義する.

$$\theta(v) = \frac{\sum_{G_i \in \mathbf{G}} W(G_i) \cdot \mathbb{I}(\hat{f}_i(v) \in \mathcal{V}(G_i))}{\sum_{G_i \in \mathbf{G}} W(G_i)} \quad (4.6)$$

$$\hat{\theta}((v, v')) = \frac{\sum_{G_i \in \mathbf{G}} W(G_i) \cdot \mathbb{I}(\hat{\zeta}_i((v, v')) \in \mathcal{E}(G_i))}{\sum_{G_i \in \mathbf{G}} W(G_i)} \quad (4.7)$$

第4章 グラフの確率モデル

入力. $K, \mathcal{V}(\mathcal{R}), \mathbf{H}, l$ 出力. $f: \mathcal{V}(K) \rightarrow \mathcal{V}(\mathcal{R})$ 1. $f: \mathcal{V}(K) \rightarrow \mathcal{V}(\mathcal{R})$ の初期化. $f: \mathcal{V}(K) \rightarrow \emptyset$. 2. for $j = 1$ to m 3. K から H_j への編集系列 $\mathbf{P}_{k,j}$ を求める. 4. 以下の式により f を更新する. $f: v \in \mathcal{V}(K) \rightarrow \begin{cases} l_j(u \in \mathcal{V}(H_j)) & \text{if } (f(v) = \phi) \wedge (\exists \{v \rightarrow u\} \in \mathbf{P}_{k,j}) \\ \phi & \text{otherwise} \end{cases}$ 5. end for

図 4.2: グラフの頂点集合から確率モデルの頂点集合との対応関係を得る手順

次に頂点 $v \in \mathcal{V}(\mathcal{R})$ において, v が属性値 l をとる確率を求める. まず, 頂点 v と対応する頂点の属性値を元とする集合を \mathbf{X}_v とし, 式 (4.8) により求める. \mathbf{G} に割り当てられた重み $W(\mathbf{G})$ を昇順に並べたとき, $G_i \in \mathbf{G}$ の重み $W(G_i)$ の順位を $r(G_i)$ とする. 集合の要素 X を N 個複製する操作を式 (4.9) と定義する. 式 (4.9) により得られる属性値の集合 \mathbf{X}_v は, G_i の重みを考慮した集合となっている.

$$\mathbf{X}_v = \bigcup_{G_i \in \mathbf{G}} \left\{ \alpha_i \left(\hat{f}_i(v) \right) \diamond r(G_i) \right\} \quad (4.8)$$

$$X \diamond N = \underbrace{\{X, X, \dots, X\}}_{N \text{ times}} \quad (4.9)$$

頂点 v における属性値の確率分布を \mathbf{X}_v から抽出する. L_v が連続値である場合, 正規分布を適用する. \mathbf{X}_v における平均値を μ_v , 分散を σ_v とする. 頂点 v が属性値 l となる確率は式 (4.10) となる. 一方で, L_v が離散値である場合, 頂点 v において属性値 l が観測される確率は式 (4.11) により求める.

$$P(v, l) = N(l \mid \mu_v, \sigma_v) \quad (4.10)$$

$$P(v, l) = \frac{|\{X \in \mathbf{X}_v \mid X = l\}|}{|\mathbf{X}_v|} \quad (4.11)$$

辺 $(v, v') \in \mathcal{E}(\mathcal{R})$ についても同様に, 式 (4.12) により $\mathbf{X}_{v,v'}$ を求め, L_e が連続値の場合は式 (4.13) により確率を求め, L_e が離散値の場合, 式 (4.14) により属性値 l を観測する確率を計算する.

$$\mathbf{X}_{v,v'} = \bigcup_{G_i \in \mathbf{G}} \left\{ \beta_i \left(\hat{\zeta}_i(v, v') \right) \diamond r(G_i) \right\} \quad (4.12)$$

$$\hat{P}((v, v'), l) = N(l \mid \mu_{v,v'}, \sigma_{v,v'}) \quad (4.13)$$

$$\hat{P}((v, v'), l) = \frac{|\{X \in \mathbf{X}_{v,v'} \mid X = l\}|}{|\mathbf{X}_{v,v'}|} \quad (4.14)$$

4.4 確率モデルからグラフの生成確率

本節ではグラフ $K = (\mathbf{V}, \mathbf{E}, \alpha, \beta)$ が確率モデル $\mathcal{R} = (\mathbf{V}, \mathbf{E}, \theta, \hat{\theta}, P, \hat{P})$ から生成される確率 $\Pr(K \mid \mathcal{R})$ を求める。集合 \mathbf{G} の確率モデル \mathcal{R} と、 \mathcal{R} の基準グラフ \mathbf{H} 、基準グラフ写像 l は、4.3.1 節と 4.3.2 節で述べた手順により得られているものとする。また、 K の頂点集合から \mathcal{R} の頂点集合への写像を $f: \mathcal{V}(K) \rightarrow \mathcal{V}(\mathcal{R})$ とし、 f を図 4.2 に示した手順で求める。また、 $\mathcal{V}(\mathcal{R})$ から $\mathcal{V}(K)$ への写像 $\hat{f}: \mathcal{V}(\mathcal{R}) \rightarrow \mathcal{V}(K)$ は以下とする。辺の対応関係を $\hat{\zeta}: \mathcal{E}(\mathcal{R}) \rightarrow \mathcal{E}(K)$ とする。

$$\hat{f}: u \in \mathcal{V}(\mathcal{R}) \rightarrow \begin{cases} v & \text{if } f(v) = u \\ \phi & \text{otherwise} \end{cases}$$

まず、 \mathcal{R} から K の頂点が抽出される確率を求める。 $\mathcal{V}(K)$ と対応する $\mathcal{V}(\mathcal{R})$ は観測される確率と、それ以外は観測されない確率の総乗により求める。 \mathcal{R} から K の頂点集合と属性値が得られる確率を式 (4.15) により定義する。

$$\Pr(\mathcal{V}(K) \mid \hat{f}, \mathcal{R}) = \prod_{u \in \mathcal{V}(\mathcal{R})} \Pr(\mathcal{V}(K) \mid u, \hat{f}, \mathcal{R}) \quad (4.15)$$

$$\Pr(\mathcal{V}(K) \mid u, \hat{f}, \mathcal{R}) = \begin{cases} \theta(u) \cdot P(u, \alpha(\hat{f}(u))) & \text{if } \hat{f}(u) \in \mathcal{V}(K) \\ 1 - \theta(u) & \text{otherwise} \end{cases} \quad (4.16)$$

第4章 グラフの確率モデル

同様に \mathcal{R} から K の辺集合が抽出される確率を求める。 \mathcal{R} から K の辺集合と属性値が得られる確率を式 (4.17) により求める。

$$\Pr(\mathcal{E}(K) \mid \hat{\zeta}, \mathcal{R}) = \prod_{(u, u') \in \mathcal{E}(\mathcal{R})} \Pr(\mathcal{E}(K) \mid (u, u'), \hat{\zeta}, \mathcal{R}) \quad (4.17)$$

$$\Pr(\mathcal{E}(K) \mid (u, u'), \hat{\zeta}, \mathcal{R}) = \begin{cases} \hat{\theta}((u, u')) \cdot \hat{P}((u, u'), \beta(\hat{\zeta}((u, u')))) & \text{if } \hat{\zeta}((u, u')) \in \mathcal{E}(K) \\ 1 - \hat{\theta}((u, u')) & \text{otherwise} \end{cases} \quad (4.18)$$

\mathcal{R} において K が観測される確率を式 (4.15) と式 (4.17), を用いて, 式 (4.19) により定義する。

$$\Pr(K \mid \hat{f}, \hat{\zeta}, \mathcal{R}) = \Pr(\mathcal{V}(K) \mid \hat{f}, \mathcal{R}) \times \Pr(\mathcal{E}(K) \mid \hat{\zeta}, \mathcal{R}) \quad (4.19)$$

4.5 確率モデルの評価実験

確率モデルを構築する実験を行い, 確率モデルを用いて認識する実験を行った結果を示す。実験に用いたデータは, 3.5.1 節で述べた, IAM Graph Database; Letter (文字のグラフ) から Low, Medium, High の三つのデータベースを用いた。

最初に, 認識対象のグラフと確率モデルとの距離について述べる。学習データ $\mathbf{G} = \{G_1, \dots, G_n\}$ の確率モデル $\mathcal{R}(\mathbf{G})$ とし, 認識対象のグラフを H とする。 $\mathcal{R}(\mathbf{G})$ の頂点集合と H の頂点集合の対応関係, $\mathcal{R}(\mathbf{G})$ の辺集合と H の辺集合の対応関係を以下とする。

$$\begin{aligned} \hat{f} &: \mathcal{V}(\mathcal{R}(\mathbf{G})) \rightarrow \mathcal{V}(H) \\ \hat{\zeta} &: \mathcal{E}(\mathcal{R}(\mathbf{G})) \rightarrow \mathcal{E}(H) \end{aligned}$$

式 (4.19) により H が $\mathcal{R}(\mathbf{G})$ から生成される確率 $\Pr(H \mid \hat{f}, \hat{\zeta}, \mathcal{R}(\mathbf{G}))$ を求めることができる。生成される確率の対数を H と $\mathcal{R}(\mathbf{G})$ の距離とする。 H と $\mathcal{R}(\mathbf{G})$ の距離は式 (4.20)

第4章 グラフの確率モデル

により求める.

$$\begin{aligned} d(\mathcal{R}(\mathbf{G}), H) &= \log \left(\Pr(H \mid \hat{f}, \hat{\zeta}, \mathcal{R}(\mathbf{G})) \right) & (4.20) \\ &= \log \left(\Pr(\mathcal{V}(K) \mid \hat{f}, \mathcal{R}(\mathbf{G})) \right) + \log \left(\Pr(\mathcal{E}(K) \mid \hat{\zeta}, \mathcal{R}(\mathbf{G})) \right) \\ &= \sum_{u \in \mathcal{V}(\mathcal{R})} \log \left(\Pr(u \mid \hat{f}, \mathcal{R}(\mathbf{G})) \right) + \\ &\quad \sum_{(u, u') \in \mathcal{E}(\mathcal{R})} \log \left(\Pr((u, u') \mid \hat{\zeta}, \mathcal{R}(\mathbf{G})) \right) \end{aligned}$$

4.3節に述べてた手法により確率モデルを抽出する. 各クラスの学習データは50個のグラフから構成され, 各クラスの確率モデルを求めた. 確率モデルとグラフの距離を計算し, グラフを最近傍の確率モデルの属するクラスへと認識する. 各クラスの確率モデルを元とする集合を $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_m\}$ とすると, グラフ H が認識されるクラスは

$$\mathcal{R}_i = \arg \min_{\mathcal{R}_i \in \mathcal{R}} d(\mathcal{R}_i, H)$$

となる. 認識実験の結果を図4.3に示す. データセット Low, Medium, High での認識率はそれぞれ 94.6, 85.5, 84.2% となった.

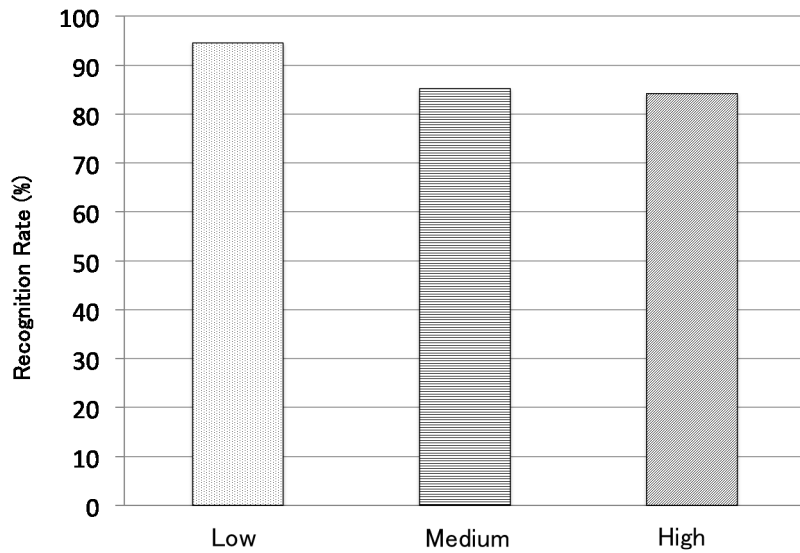


図 4.3: 確率モデルによる認識結果

第4章 グラフの確率モデル

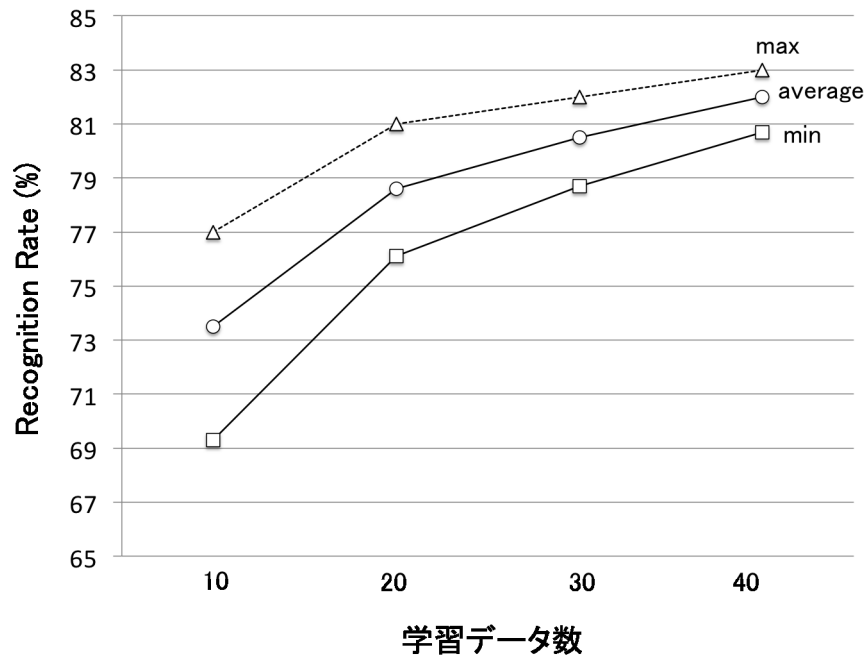


図 4.4: 確率モデルの性能と学習データ数の関係

次に、学習データによって確率モデルの性能がどう変化するかを確認する実験を行った。ランダムにグラフを選ぶことによって10個、20個、30個、40個の大きさの学習データを抽出する。大きさや学習データの元が異なるデータセットから確率モデルを構築し、認識実験を行う。この試行を10回繰り返し、認識率の平均値、最高値、最低値を図4.4に示す。図4.4から、確率モデルは多くの学習データを用いたほうが認識性能が良いことが分かる。

学習データをランダムに選び大きさの異なる学習データを構築していたので、同じ学習データ数から確率モデルでも試行の度に異なる確率モデルが得られている。そこで10回の試行により、テストデータが正しく認識された回数を計測し、認識回数によりテストデータを分類した。学習データ数が10、20、30、40個における実験結果をそれぞれ図4.5、図4.6、図4.7、図4.8に示す。学習データ数10、20、30、40個での、7回以上認識されたテストデータの割合は、それぞれ67.2、73.1、76.2、80%であった。大部分のテストデータが、さまざまな確率モデルによって認識されていることが分かる。一方、認識された回数が少ないテストデータは、ある確率モデルによって認識されたが、他の確率モデルでは認識できなかったことを示している。これは学習データの選び方によって、一

第 4 章 グラフの確率モデル

部のデータに有効な確率モデルを作ることができるという可能性を示している。

最後に、確率モデルを用いた認識結果と第三章で述べた代表グラフを用いた認識結果を比較する。比較結果を図 4.9 に示す。確率モデルは low, medium, high のデータセットすべてにおいて代表グラフの認識結果を上回った。特に medium においては、大きく性能が向上していることが伺える。

第4章 グラフの確率モデル

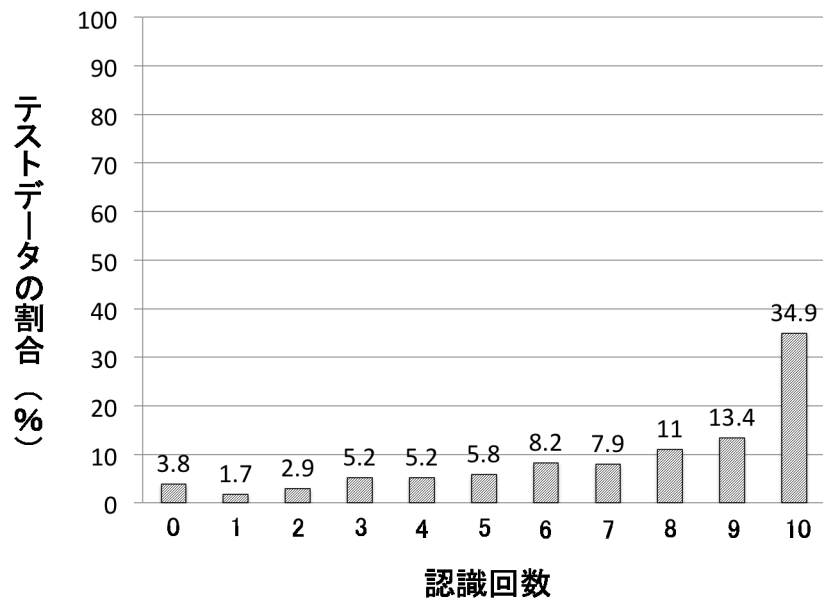


図 4.5: 認識された回数ごとのテストデータの割合, 学習データ数 10

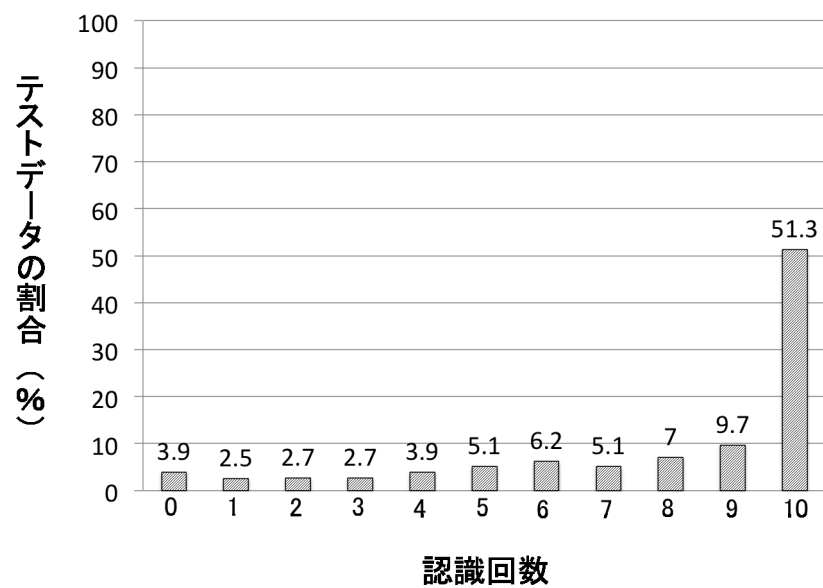


図 4.6: 認識された回数ごとのテストデータの割合, 学習データ数 20

第 4 章 グラフの確率モデル

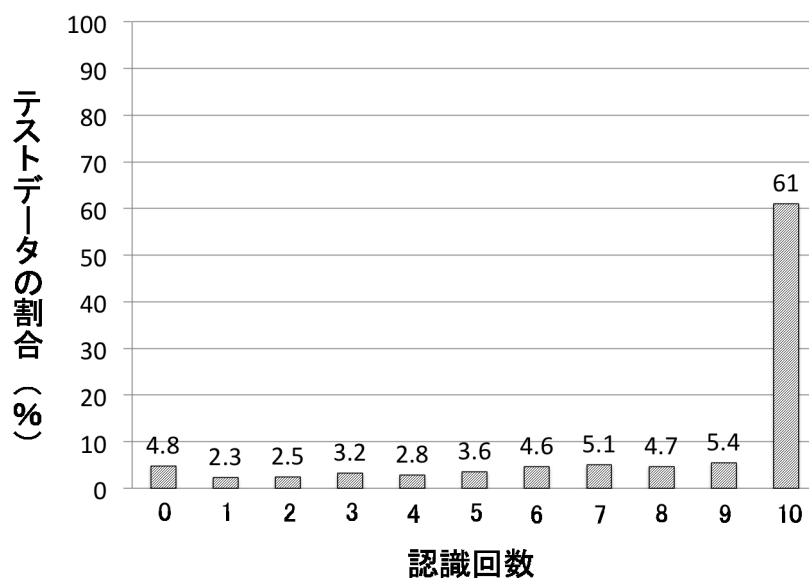


図 4.7: 認識された回数ごとのテストデータの割合, 学習データ数 30

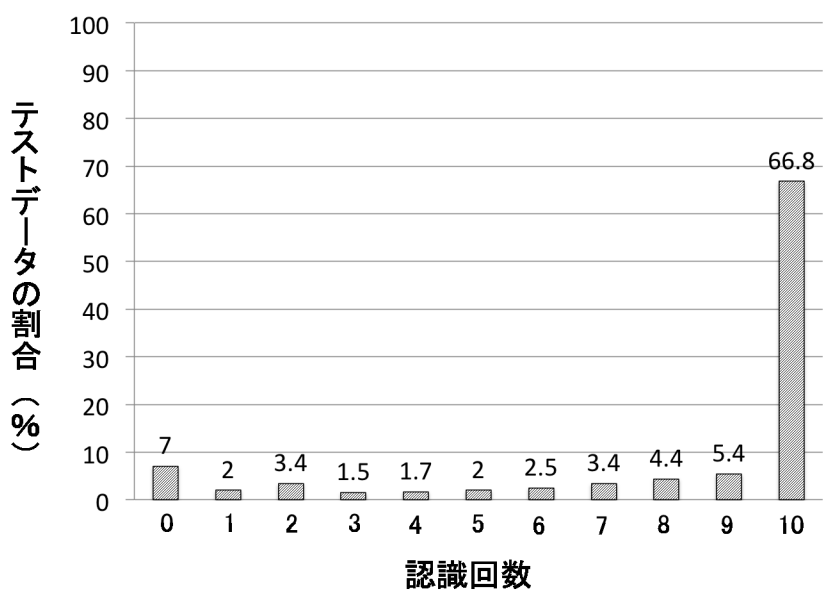


図 4.8: 認識された回数ごとのテストデータの割合, 学習データ数 40

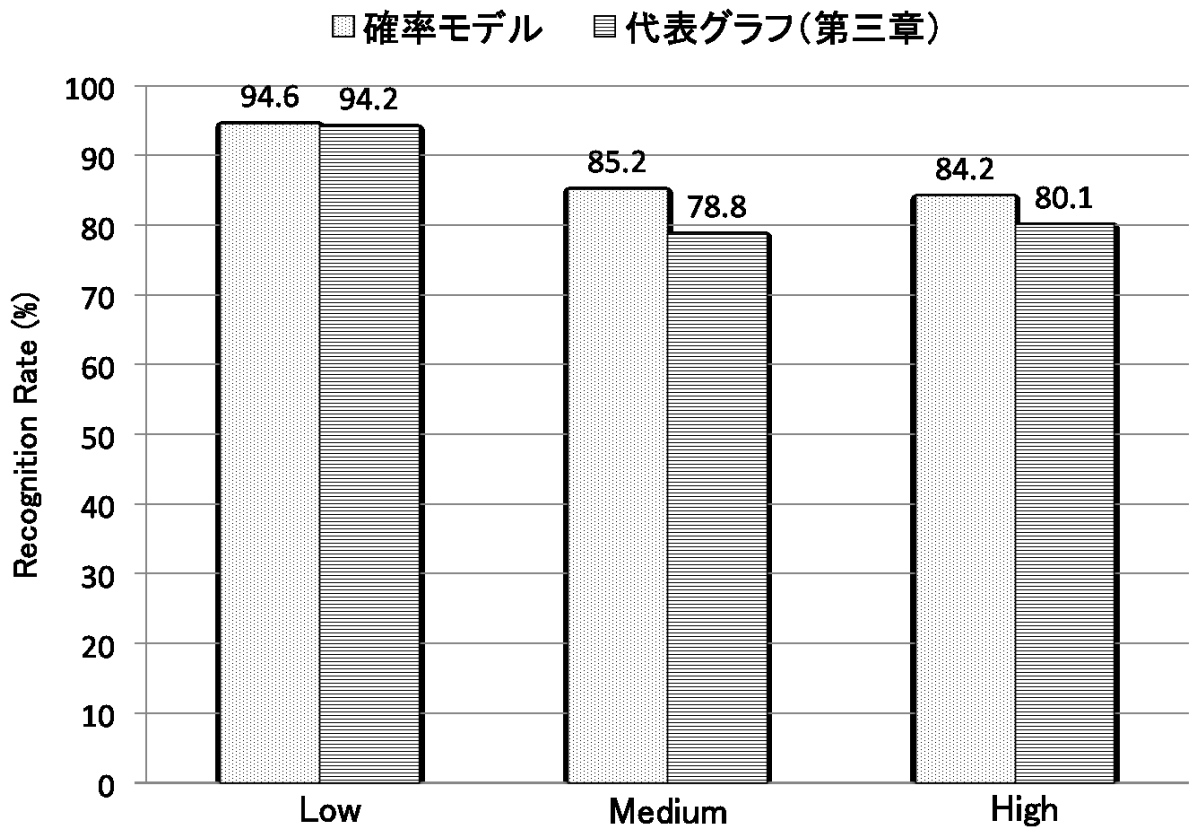


図 4.9: 確率モデルと代表グラフの比較

4.6 確率モデルによる高精度認識

学習データを代表するグラフを抽出する手法を第三章で提案し、本章では学習データの確率モデルを構築する手法を提案した。代表グラフ、確率モデルを用いた認識実験の結果、両手法ともに平均認識率 80%程度の性能を得ている。特に変形の少ないデータセットにおいては 95%程度と高い認識性能を得た一方で、IAM Graph Database のように変形が大きいデータセット Medium と High に対する認識性能が低下していることが分かる。

4.5 節の実験結果から、確率モデルは学習データによって認識性能が変化することが分かった。一部の学習データから得られる確率モデルの性能は、すべての学習データを用いて得られる確率モデルの性能と異なる。言い換えれば、一部の学習データをすべての学習データからランダムに抽出し、性能の異なる確率モデルを得ることができる。

ここでパターン認識におけるブースティングという概念について述べておく。ブースティングは、“一連の弱い弱識別器をまとめることで強い識別器をつくることができるか”という疑問から始まる。ブースティングとは、あまり性能のよくない学習アルゴリズムをもとに、性能のよい学習アルゴリズムを構成する技法である。ここでいう性能は、学習アルゴリズムが出力する認識精度の善し悪しのことである。

本節では、ランダムに選んだ学習データの部分集合から得られる確率モデルを集めて、ブースティングの代表的な手法である AdaBoost の枠組みにより識別器を構成する手法を提案する。得られた識別器は変形の大きなデータセットに対しても有効であることを評価実験により確認し、グラフの高精度認識を行えることを示す。

4.6.1 AdaBoost 法

本節では、最も基本となる AdaBoost 法 [35] について簡単に説明する。AdaBoost 法はアンサンブル学習のアルゴリズムの中で最も代表的で成功したアルゴリズムのひとつである。このアルゴリズムでは、学習用の事例それぞれに対してその重要度が定義されていて、識別が難しい事例の重要度を増加させることにより、その事例に対する重点的な学習を実現している。観測特徴ベクトルを $x_i \in \mathbb{R}^d$ 、そのクラスラベルを $y_i \in \{1, -1\}$ とする。学習データ $\mathcal{D} = \{(x_i, y_i) \mid i = 1, \dots, N\}$ が与えられたとき、AdaBoost は強識別器 $H : \mathbb{R}^d \rightarrow \{1, -1\}$ を学習する。

$$H(x_i) = \text{sign} \left(\sum_{m=1}^M \alpha^{(m)} h^{(m)}(x_i) \right) \quad (4.21)$$

1. Initialize the observation weights $w_i = \frac{1}{N}$
2. For $m = 1$ to M
 - (a) Fit a classifier $h^{(m)}$ to \mathcal{D} with weights
 - (b) Compute

$$err^{(m)} = \sum_{i=1}^N w_i \mathbb{I}(h^{(m)}(x_i) \neq y_i) / \sum_{i=1}^N w_i$$
 - (c) Compute

$$\alpha^{(m)} = \log\left(\frac{1 - err^{(m)}}{err^{(m)}}\right)$$
 - (d) Set

$$w_i \leftarrow w_i \cdot \exp(\alpha^{(m)} \cdot \mathbb{I}(y_i \neq h^{(m)}(x_i)))$$
 - (e) Re-normalize weights

図 4.10: AdaBoost による強識別器の学習

No.	x_1	x_2	\cdots	x_p	y
1	x_{11}	x_{12}	\cdots	x_{1p}	y_1
2	x_{21}	x_{22}	\cdots	x_{2p}	y_2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
i	x_{i1}	x_{i2}	\cdots	x_{ip}	y_i
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
n	x_{n1}	x_{n2}	\cdots	x_{np}	y_n

表 4.1: データ形式

ここで, $h^{(m)} : \mathbb{R}^d \rightarrow \{1, -1\}$ は m 番目の弱識別器である. $\alpha^{(m)} \in \mathbb{R}$ はその識別重みで, 信頼度という. M は弱識別器の総数である. AdaBoost の学習を図 4.10 に示す. w は学習データの重要度である.

4.6.2 決定木

AdaBoost により繰り返し得られる弱識別器に決定木を用いる.

解析するデータの形式を表 4.1 とする. ここで y は目的変数, x_1, x_2, \dots, x_p は説明変数である. 決定木は目的変数の違いをできるだけ際立たせるような 2 分割分類ルールを用いる.

第4章 グラフの確率モデル

第一段階として、全サンプルの2分割を考える。ある説明変数 x_i に着目して2分割することを考える。分割方法 $G_1 = \{x \mid x_i > a\}$, $G_2 = \{x \mid x_i \leq a\}$ によりグループ G_1 とグループ G_2 に分割することができる。 a は閾値で、 a の選び方は、サンプル数 n なので、たかだか $n - 1$ 通りである。

CART アルゴリズムは分割する閾値を選択する指標として Gini インデックスを用いている。 Gini インデックスはもともと所得格差を表す指標として考案され、様々な分野で用いられている。 CART アルゴリズムにおいては、集合のデータが属するクラスの偏りを示す指標とみなすことができる。例えば、あるデータがクラス A である確率を P_A 、クラス B となる確率を P_B であるとする、 Gini インデックスは以下のように求めることができる。

$$\begin{aligned} \text{Gini} &= 1 - (P_A^2 + P_B^2) \\ &= 1 - (P_A^2 + (1 - P_A)^2) \\ &= 2P_A(1 - P_A) \end{aligned}$$

すべての説明変数とそれぞれの説明変数の分割方法に対して Gini インデックスにより評価し、評価値が最も大きくなる説明変数と分割ルールにより全サンプルを G_1 と G_2 に分割する。次に、 G_1 と G_2 をそれぞれふたつに分割する。分割の考え方は第一段階と同じである。

以上の操作を繰り返すことにより分割が進んでいく。分割を停止するための条件は以下のお m ののが考えられる。 1) 最終グループのデータ数が最大値となる、 2) 各最終グループに含まれるデータ数が最小値となる、 3) Gini インデックスの値が下限となる。

4.6.3 確率モデルを用いた高精度識別器

AdaBoost の枠組みによって確率モデルを用いた高精度識別器をつくる。

学習データを $\mathbf{G} = \{G_1, \dots, G_n\}$ とする。 n は学習データ数である。学習データの重みを $W : \mathbf{G} \rightarrow \mathbb{R}^+$ 、学習データの所属するクラスを $Y : \mathbf{G} \rightarrow \mathbf{C}$ とする。ただし、 $\mathbf{C} = \{c_1, \dots, c_k\}$ は認識するクラス c_i を元とする集合で、クラス数は k である。

以下に AdaBoost 法により識別器を学習するアルゴリズムを示す。グラフの集合を \mathbf{X} とする。弱識別器を $h_{c_i} : \mathbf{X} \rightarrow \{c_i, 0\}$ とし、弱識別器を統合した強識別器を $H_{c_i} : \mathbf{X} \rightarrow \{c_i, 0\}$ とする。 h_{c_i} 、 H_{c_i} は \mathbf{X} がクラス c_i に所属しているかどうかを識別する。 0 は所属していないことを示す。

アルゴリズム (1) で学習データの重みを初期化し、 (2) は \mathbf{G} からサンプリングした集合から弱識別器を学習している。弱識別器の学習は T 回繰り返される。この繰り返しのこ

第4章 グラフの確率モデル

とをブースティングという。\$T\$ 回の繰り返しにより、\$T\$ 個の弱識別器が得られる。(3)において、\$T\$ 個の弱識別器を統合し、強識別器が構成される。また、弱識別器には決定木を用いた [12].

(1) 重みの初期化 $W(G) = \frac{1}{n}$

(2) for $t = 1$ to T

(a) G からランダムにグラフを選び、サンプリングした学習データ $S = \{S_1, \dots, S_m\}$ を得る。 $S \subseteq G$ である。 S の重みとクラスは W と Y により得られるものとする。

(b) $S_{c_i} = \{S \in S \mid Y(S) = c_i\}$ から確率モデル \mathcal{R}_{c_i} を求める。すべてのクラスの確率モデルを $\mathcal{R} = \{\mathcal{R}_{c_1}, \dots, \mathcal{R}_{c_k}\}$ とする。

(c) \mathcal{R} から弱識別器 h^t を作る。

(d) h^t の重み付き誤り率 err_t を計算する。

$$err_t = \sum_{G_i \in G} W(G_i) \cdot \mathbb{I}(h^t(G_i) \neq Y(G_i)) \quad (4.22)$$

(e) h^t の信頼度 γ_t を計算する。

$$\gamma_t = \log \left(\frac{1 - err_t}{err_t} \right) \quad (4.23)$$

(f) 重みの更新

$$W(G_i) \leftarrow W(G_i) \cdot \exp(\gamma_t \cdot \mathbb{I}(h^t(G_i) \neq Y(G_i))) \quad (4.24)$$

(3) 強識別器 H の出力。クラス c かその他のクラス 0 かを出力する。

$$H(G) = \arg \max_{\{c, 0\}} \sum_{t=1}^T \left(\gamma_t \cdot \hat{\mathbb{I}}(h^t(G) = c) \right) \quad (4.25)$$

$$\hat{\mathbb{I}}(p) = \begin{cases} 1 & \text{if } p \text{ is true} \\ -1 & \text{otherwise} \end{cases} \quad (4.26)$$

第4章 グラフの確率モデル

4.6.4 評価実験

提案法によりグラフの強識別器を構築し、認識実験を行った。実験に用いたデータは4.5節でも用いた、IAM Graph Database; Letter の Low, Medium, High のデータベースを使用した。

提案法により各クラスを識別する強識別器を構築した。IAM Graph Database; Letter は15クラスあるので、15個の強識別器を構築した。すべての強識別器によりテストデータを自クラスか他クラスかを識別し、正しく識別できた結果を図4.11に示す。認識結果によると、構造モデル、平均モデルの性能が落ちていた変形の大きいデータセットに対して、識別器による認識結果は96.5% (high) と高い認識率を得ることができた。

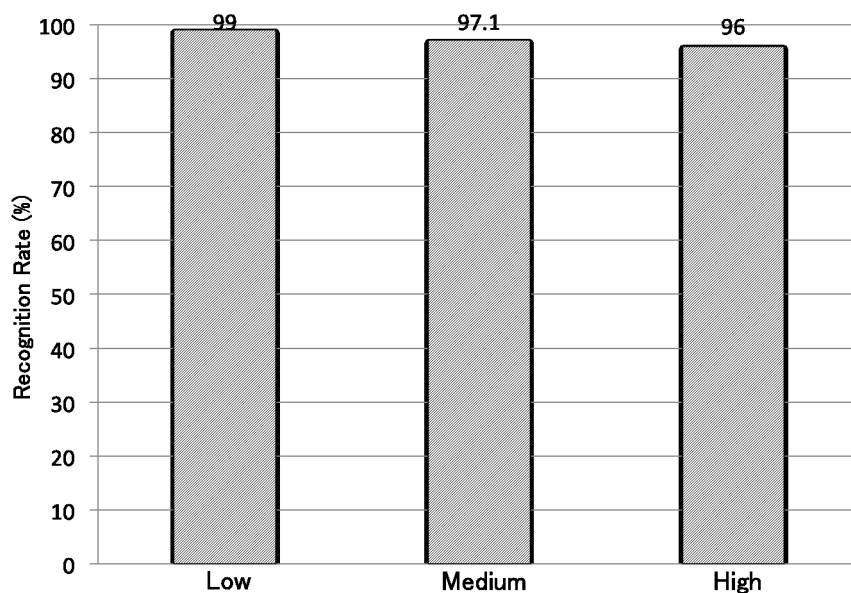


図 4.11: 強識別器による認識結果

第5章

結論

グラフは物体や対象の関係性を簡潔に記述することができる。近年、パターン認識やコンピュータビジョンの分野において、グラフを用いたアプリケーションの有用性が注目されている。

様々なアプリケーションにおいて、パターンを識別することは重要なタスクである。パターン認識の一般的な手法である k 平均法は、識別クラスの代表値を更新しながらクラスタリングを行う。また、混合正規分布によるデータのモデル化は洗練された手法である。以上に述べたような一般的な認識、学習手法はベクトルに対して適用することができるが、グラフに対しては容易に適用できず、グラフのパターン認識は未だ発展の途上である。本論文はグラフのパターン認識に対して、高精度な認識手法を提案した。

第1章は序論である。

第2章では、グラフの基本的な定義と概念、グラフ間の距離を計測する手法について述べ、グラフを用いてシルエット画像をカテゴリ分類する手法を提案した。提案手法は、中心軸変換によりシルエット画像の構造を表すグラフを抽出することで、シルエット画像をグラフとして扱った。重み付き最小共通スーパーグラフの概念を用いて、学習データからカテゴリを代表するグラフを抽出した。得られたカテゴリのグラフとシルエット画像のグラフとの距離を計測し、得られた距離をカテゴリとシルエット画像との距離として認識に用いた。シルエット画像に対して提案手法を用いて認識実験を行った結果、提案手法はおおよそ 80% の認識率を得た。学習データを個々に用いてカテゴリとシルエッ

第5章 結論

ト画像のグラフとの距離を求める手法と比較して、提案手法の認識率が8%程度上回ったことを確認した。

第3章では、属性値を考慮してグラフの集合を代表するグラフを抽出する手法を提案した。提案手法は、属性値と頂点の次数との関係式、属性値と頂点間に接続する辺との関係式を定義し、グラフの集合における特徴的な頂点を検出した。同じ特徴を有する頂点を統合しながら、学習データのグラフの和を求めることで代表グラフを抽出した。クラスの代表グラフとグラフの距離計測することにより、グラフとクラスの距離を計測し認識した。パターン認識、学習法を評価する標準的なデータベースである IAM Graph Database の文字グラフと、飾り文字画像から得られるグラフに対して提案手法を用いて認識実験を行った。飾り文字画像に対して提案手法は認識率 25%程度となった。これは飾り文字画像から得られるグラフは、同一クラスのグラフとの間でも変形が激しく、クラスの特徴的な属性値を検出することが困難であったためであると考えられる。IAM Graph Database で、変形の程度が異なるデータセットに対する提案手法の認識率は 94%、79%、80%であった。代表グラフを得る従来手法である重み付き最小共通スーパーグラフと比較して、提案手法は 0.2%から 14.5%ほど従来手法を上回った。計算時間においても提案手法は従来手法と比べて 9 倍から 42 倍高速に代表グラフを求めることができることがわかった。

第4章では、グラフの集合を記述する確率モデルを構築する手法を提案した。確率モデルはグラフ部分と統計部分から構成され、頂点集合と辺集合から構成されるグラフ部分と、各頂点と辺が観測される確率、各頂点と辺がとりうる属性値の確率分布から構成される統計部分である。頂点と辺が観測される確率は学習データとの対応関係により求めた。頂点と辺がとる属性値の確率分布も同様であった。グラフが確率モデルから観測される確率を計算することができ、確率モデルとグラフの距離は確率の対数により計算することができりことを示した。IAM Graph database に対して確率モデルを構築し、認識実験を行った結果、変形の程度が異なるデータセットに対してそれぞれ、94%、86%、84%であった。これは第三章で提案した手法を用いた認識結果を上回っており、確率モデルの有効性を確認した。また、ランダムに学習データを選び、確率モデルを構築、認識実験を行った結果、最大約 17%程度、認識性能が異なる確率モデルを得ることがわかった。さまざまな性能を有する確率モデルを用いることにより、アンサンブル学習を適用することができた。さまざまな認識性能を有する確率モデルを用いて弱識別器を構築し、代表的なアンサンブル学習である AdaBoost により、弱識別器を統合、強識別器を構築した。IAM Graph Database に対して強識別器を用いて認識実験を行った。認識率は 99%から 96%となった。これは確率モデルと比較し、5%から 12%程度、認識率

第5章 結論

が向上し、確率モデルと AdaBoost を組み合わせることにより、グラフを高精度に認識することができたことを確認した。

今後の課題として、提案手法を評価する実験に使用したグラフは文字のグラフであるため、文字以外のグラフに対して評価実験を行い、提案手法の有効性を確認する必要がある。確率モデルの頂点と辺が観測される確率は、確率モデルの頂点と対応する頂点を有する学習データの数を数えることで求めている。最尤推定法の枠組みで頂点と辺が観測される確率をもとめることができれば、集合をさらに表す確率モデルを構築することができると考えられる。また、飾り文字画像から提案手法に適したグラフを抽出することも今後の課題である。

参考文献

- [1] R. O. Duda, P. E. Hart and D. G. Stork, "Pattern Classification," A Willey Interscience Publication, 2nd Edition, 2000.
- [2] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty Years of Graph Matching in Pattern Recognition," International Journal of Pattern Recognition and Artificial Intelligence, Vol.18, No.3, pp.265-298, 2004.
- [3] H. Bunke, P. Foggia, C. Guidobaldi, and M. Vento, "Graph Clustering Using the Weighted Minimum Common Supergraph," Graph Based Representations in Pattern recognition, pp. 235-246, 2003.
- [4] J. J. McGREGOR, "Backtrack Search Algorithms and the Maximal Common Subgraph Problem," Software Practice and Experience, Vol. 12, pp. 23-34, 1982.
- [5] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, and M. Vento, "A Comparison of Algorithms for Maximum Common Subgraph on Randomly Connected Graphs," In T. Caelli, A. Amin, R. Duin, M. Kamel, and D. de Ridder, editors, Structural, Syntactic, and Statistical Pattern Recognition in Proc. Joint IAPR Int. Workshops SSPR 2002 and SPR 2002, volume 2396 of Lecture Notes in Computer Science, pp. 123-132, Springer, 2002.
- [6] H. Bunke, X. Jiang, A. Kandel, "On the Minimum Common Supergraph of two Graphs," Computing 65, No. 1, pp. 1325, 2000.
- [7] M.R. Garey, D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman & Co, New York, 1979.
- [8] J.T. Tou and R.C. Gonzalez, "Pattern Recognition Principles", Addison-Wesley, 1974.

参考文献

- [9] K. Riesen, and H. Bunke, "IAM graph database repository for graph based pattern recognition and machine learning," In N. da Vitoria Lobo et al., editor, Structural Syntactic, and Statical Pattern Recognition, LNCS 5342, pp.287-297, 2008.
- [10] Bollobas, B., "Random Graph," Academic Press, 1985.
- [11] Wong, A. K. C. and Ghahraman, H. E., "Random Graph: Structural-contextual dichotomy," IEEE Trans. Pattern Anal. Machine Intell., vol.PAMI-2, no.4, pp.341-348, 1980.
- [12] Breiman, L., J. Friedman, R. Olshen, and C. Stone, "Classification and Regression Trees," Boca Raton, FL: CRC Press, 1984.
- [13] W. H. Tsai and K. S. Fu, "Error-correcting Isomorphism of Attributed relational graphs for pattern analysis," IEEE Trans. Syst. Man. Cybern., vol. SMC-9, pp.757-768, 1979.
- [14] Ullman, J. R., "An Algorithm for Subgraph Isomorphism," J. Assoc. Comput. Mach. 23, pp.31-42, 1976.
- [15] G. Levi, "A Note on The Derivation of Maximal Common Subgraphs of Two Directed or Undirected Graphs," Calcolo 9, pp.341-354, 1972.
- [16] A. Sanfeliu, K. Fu, "A Distance Measure Between Attributed Relational Graphs for Pattern Recognition," IEEE Trans. on Systems, Man, and Cybernetics vol.13 no.3, pp.353-363, 1983.
- [17] M. Neuhaus, K. Riesen and H. Bunke, "Fast Suboptimal Algorithms for the Computation of Graph Edit Distance," Structural, Syntactic, and Statistical Pattern Recognition Lecture Notes in Computer Science, vol.4109 pp.163-172, 2006.
- [18] P. Hart, N. Nilsson, B. Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions of Systems, Science, and Cybernetics vol.4, pp.100-107, 1968.
- [19] P. N. Klein, T. B. Sebastian, and B. B. Kimia, "Shape Matching Using Edit-Distance: An Implementation," Proc. Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-01), pp.781-790, 2001.

参考文献

- [20] A. K. C. Wong and M. L. YOU, "Entropy and Distance of Random Graphs with Application to Structural Pattern Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.7, no.5, pp.599-609, 1985.
- [21] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Bream, "Basic Objects in Natural Categories," *Cognitive Psychology*, vol.8, no.3, pp.382-439, 1976.
- [22] S. Belongie, J. Malik and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.24, no.4, pp.509-522, 2002.
- [23] H. Blum, "Biological Shape and Visual Science," *Journal of Theoretical Biology*, vol.38, pp.205-287, 1973.
- [24] K. Siddiqi, S. Bouix, A. Tannenbaum, and S.W. Zucker, "Hamilton-Jacobi Skeleton," *Int. J. Comput. Vis.*, vol.48, no.3, pp.215-231, 2002.
- [25] B. Kmia, A. Tannebaum, and S. Zucker, "Shapes, Shocks, and Deformations i: The Components of Two-Dimensional Shape and The Reaction-Diffusion Space," *International Journal of Computer Vision*, Vol. 15, pp.189-224, 1995.
- [26] K. Kageyama, S. Omachi, H. Aso, "Structurization of Silhouette Images Using Medial Axis Transforms," *IEICE Trans.*, Vol. J90-D, No.6, pp. 1568-1578, 2007.
- [27] B. Luo, R.C. Wilson, E. R. Hancock, "Spectral Feature Vectors for Graph Clustering," *Proceeding of the Joint IAPR International Workshops SSPR and SPR*, pp.83-93. 2002.
- [28] T. Horvath, S. Worbel, and U. Bohnbeck, "Relational Instance-Based Learning with Lists and Terms," *Machine Learning*, vol. 43, pp.53-80, 2001.
- [29] L. J. Latecki, R. Lakaemper, and U. Eckhardt, "Shape Descriptors for Non-rigid Shapes with a Single Closed Contour," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2000)*, pp.424-429, 2000.
- [30] A. Torsello and E.R. Hancock, "Learning Shape-Classes Using a Mixture of Tree-Unions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.28, no.6, pp.954-967, 2006.

参考文献

- [31] F. Harary, "Graph Theory," Addison-Wesley, Reading, MA., 1994.
- [32] S. Omachi, M. Inoue, and H. Aso, "Structure Extraction from Decorated Characters using Multiscale Images," IEEE Trans. Pattern Anal. Mach. Intell., vol.23, no.3, pp.315-322, 2001.
- [33] Lam, L., Seong-Whan Lee, and Ching Y. Suen, "Thinning Methodologies - A Comprehensive Survey," IEEE Trans. Pattern Anal. Mach. Intell., Vol.14, No.9, pp.869-885, 1992.
- [34] R. M. Haralic, L.T. Watson, and T. J. Laffey, "The Topographic Primal Sketch," International Journal of Robotics Research, vol.2, no.1, pp.50-72, 1983.
- [35] Freund, Y., Schapire, R., "A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting," Journal of Computer and System Sciences vol.55, no.1, pp.119-139, 1997.

謝辞

本研究を進めるにあたり、ゼミやさまざまな機会を通じて御指導を頂きました、東北大学大学院工学研究科教授 大町真一郎先生に深く感謝を申し上げます。この大町研究室で研究する機会を与えてくださったことに感謝いたします。大町先生から研究を進めていく上でたくさんの御指導と、激励を頂きました。忍耐強く研究を見守ってくださったことに厚く、心より感謝申し上げます。本論文をまとめるにあたり東北大学大学院工学研究科教授 伊藤彰則先生、東北大学大学院工学研究科教授 篠原歩先生から御指導いただきました。ここに深く感謝申し上げます。また、たくさんの御指導と助言を頂きました日本大学工学部教授 阿曾弘具先生と、東北大学大学院工学研究科助教 菅谷至寛先生に深く感謝申し上げます。

大町研究室の先生方をはじめ、先輩、後輩、そして同期のみなさんと過ごすことのできた3年間に特別な感謝を申し上げます。みなさんと学生生活を共有することができたことに感謝しています。みなさんがいたからこそ、この素晴らしい日々がありました。深く感謝申し上げます。

また、私に勉強する機会と生活を与えてくださっている父母、家族に深い感謝と特別な気持ちを感じております。深く感謝を申し上げます。

研究業績

学術論文

- 宮崎智, 大町真一郎, 阿曾弘具,
“重み付き最小共通スーパーグラフを用いたシルエット画像認識,”
画像電子学会誌, Vol.38, No.5, pp.640-647, 2009.
- Tomo Miyazaki, Shinichiro Omachi,
“Representative Graph Generation for Graph-Based Character Recognition,”
画像電子学会論文誌 (投稿中 [条件付採択])

国際会議 (査読あり)

- Tomo Miyazaki, Shinichiro Omachi,
“Fast Method for Extracting Representative Graph from Decorative Character Images,”
Proceedings of 2010 2nd IEEE International Conference on Network Infrastructure and Digital Content, pp.219-223, 2010.
- Tomo Miyazaki, Shinichiro Omachi,
“Extraction of Decorative Character Images, ”
Proceedings of the First CJK Joint Workshop on Pattern Recognition, Vol.2, pp.944-948, 2009.
- Tomo Miyazaki, Shinichiro Omachi, Hiroto Aso,
“Extraction of Structure of Silhouette Images by Weighted Minimum Common

Supergraph,”

Second Korea-Japan Joint Workshop(KJPR2007), pp.57-61, October, 2007.

国際会議 (査読なし)

- Tomo Miyazaki, Shinichiro Omachi,
“Extracting Representative Graph of Decorative Character Images by Random Method, ”
Proceedings of the 3rd Student Organizing International Mini-Conference on Information Electronics Systems (SOIM-GCOE10), F3L-1, pp.67-68, October 2010.
- Tomo Miyazaki, Shinichiro Omachi,
“Iterative Extraction of Representative Graph Using Common Features from Decorative Character Images,”
Proceedings of the 2nd Student Organizing International Mini-Conference on Information Electronics Systems (SOIM-GCOE09), F4L-2, pp.85-86, 2009.
- Tomo Miyazaki, Shinichiro Omachi, Hirotomo Aso,
“Silhouette Image Recognition, ”
Proceedings of the Third Korea-Japan Joint Workshop on Pattern Recognition (KJPR2008) , pp.13-14, 2008.
- Tomo Miyazaki, Shinichiro Omachi, Hirotomo Aso,
“Structure Extraction from Silhouette Images by Weighted Minimum Common Supergraph,”
Proceedings of the 1st Student Organizing International Mini-Conference on Information Electronics Systems (SOIM-GCOE08), F3L-1, pp.75-76, 2008.
- Tomo Miyazaki, Shinichiro Omachi, Hirotomo Aso,
“Extraction of Structure of Shapes Using Weighted Minimum Common Supergraph, ”
Proceedings of China-Korea-Japan Graduates Workshop on Electronic Information, pp.49-50, 2008.

国内会議（査読なし）

- Tomo Miyazaki, Shinichiro Omachi,
“Structure Extractio from Multiple Decorative Character Images,”
平成 21 年度電気関係学会東北支部連合大会講演論文集, 2A・11, p.23, 2009.
- 宮崎智, 大町真一郎, 阿曾弘具,
“共通拡大グラフによる飾り文字画像群からの文字構造抽出,”
画像の認識・理解シンポジウム（MIRU2009）講演論文集, pp.506-512, 2009.
- 宮崎智, 大町真一郎, 阿曾弘具,
“重み付き最小共通スーパーグラフによるシルエット画像の構造抽出, ”
画像の認識・理解シンポジウム（MIRU2008）講演論文集, pp.1408-1413, 2008.

受賞

- IEEE Sendai Section Student Awards 2009, The Best Paper Prize.