

GitHub上の活動履歴分析による開発者分類

尾上 紗野 畑 秀明 松本 健一

背景

- ・ 従来からソフトウェア開発者の活動に着目した研究が行われている
 - Iharaら：コミッターの過去の活動量から、コミッター予測モデルを構築した[1]
 - Zhouら：OSSプロジェクトのLong Term Contributors（LTC）の活動を分析し、開発者の活発さはプロジェクトの人気に左右されることを明らかにした[2]

[1] Ihara, A., Ohira, M. and Matsumoto, K.: An Analysis Method for Improving a Bug Modification Process in Open Source Software Development, Proc. IWPSEE vol '09, pp.135–144, ACM (2009).

[2] Zhou, M. and Mockus, A.: Does the Initial Environment Impact the Future of Developers?, Proc. ICSE'11, pp.271–280, ACM (2011).

背景

- ・ 開発者の性格分析に基づく研究が行われている
 - Sfetsosら：Myers-Briggs Type Indicator(MBTI)を用いて、ペアプログラミングでの相性の影響を調査, 良好なコミュニケーションを気づくことができる組み合わせを示した[3]
 - Sahhehら：人間性の個性を特徴づける5つの観点（適応性, 誠実性, 外向性, 同調性, 神経質性）から開発者のペアプログラミングの影響を調査[4]

[3] Sfetsos, P., Stamelos, I., Angelis, L. and Deligiannis, I.: An experimental investigation of personality types impact on pair effectiveness in pair programming, Empirical Softw. Eng., Vol.14, No.2, pp.187–226 (2009).

[4] Salleh, N., Mendes, E., Grundy, J. and Burch, G.S.J.: An empirical study of the effects of conscientiousness in pair programming using the five-factor personality model, Proc. ICSE '10, pp.577–586, ACM (2010).

先行研究の問題点

- ・ OSSプロジェクトの特徴や不特定多数の開発者の傾向を明らかにすることは可能であるが、開発者個人の特徴を明らかにすることはできない
- ・ 性格テストはコストがかかり、偏りのないデータを取得することが困難

目的

- ・ 開発者を分類する

- OSS開発者に参加する開発者の特徴を明らかにすることで, ソフトウェア工学における新しい観点の発見が期待される
- OSSプロジェクトの成功にはタイプの開発者がバランスよく参加する必要がある

アプローチとデータセット

- ・ **GitHubで分析**

- 活動履歴を用いることで、開発者個人に対する分析ができる
- データの取得が容易で偏りが無い

- ・ **ワード法で階層化クラスタリング**

- ・ **データセット**

- 活発なOSSプロジェクトで活動する開発者
 - ・ プロジェクト：homebrew, node
 - コミット数が9,500回を超えている
 - ・ 開発者：23人（homebrew：13人, node：10人）
 - コミット数が100を超える
 - ・ 2013/12/27に取得
 - 取得できる活動履歴は、直近の300件まで

開発者のデータ

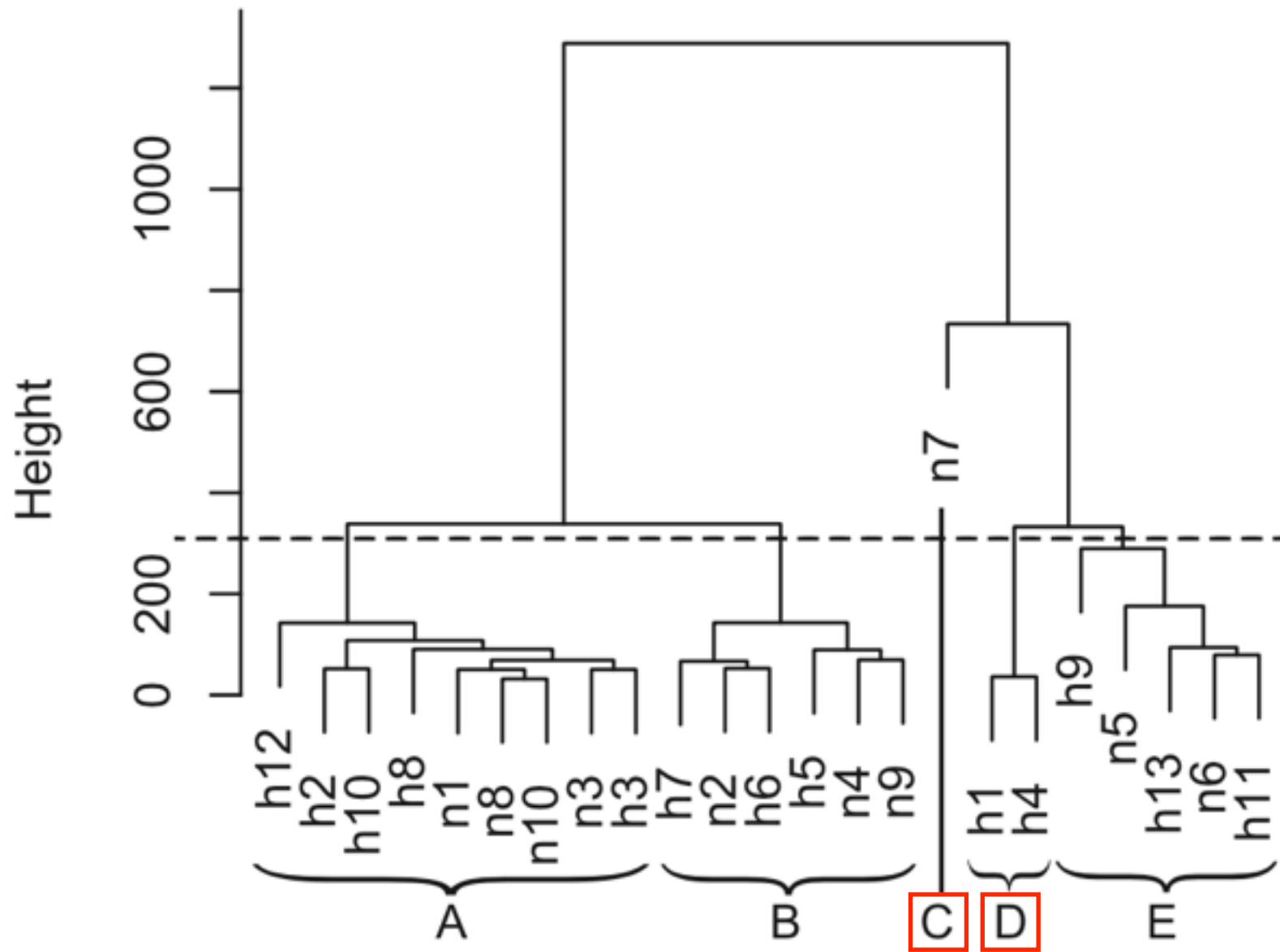
- ・ 収集するイベントは10種類

コーディング関係	Create, Delete, Fork, Push, PullRequest
コメント関係	CommitComment, PullRequestReviewComment, IssueComment
不具合報告	issue
リポジトリの動向を観察	watch

- ・ 300イベントを行った期間（開発者23人）
 - 最小日数は8日, 最多日数は659, 中央値は51

ワード法の分類

- ・ **活動の10種類と、活動期間を合わせた合計11種類のデータから分類**

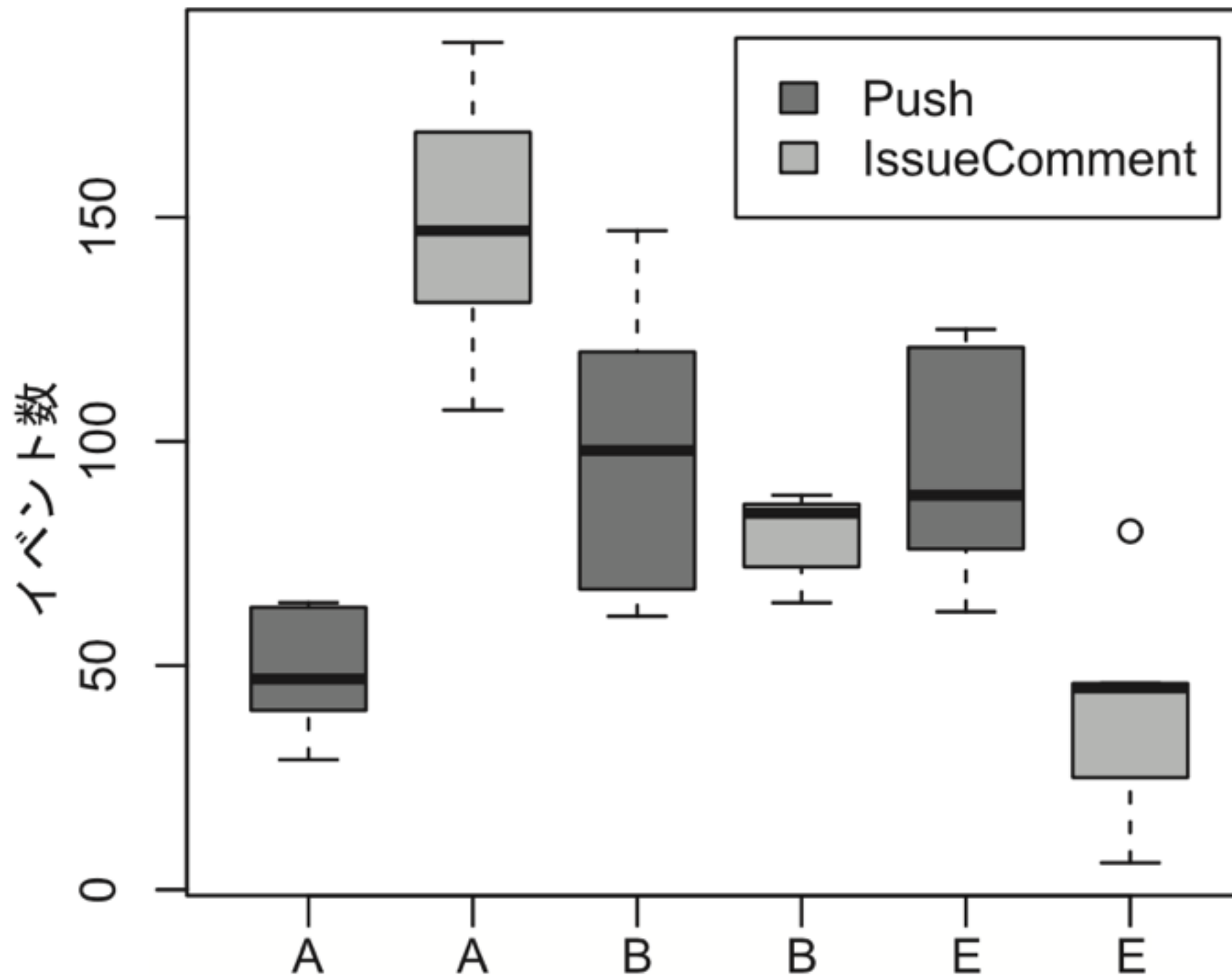


- ・ 外れ値
 - C（1人）は直近の活動期間が659日
 - D（2人）はほとんどプッシュしかしていない

図 1 開発者の階層化クラスタリング

結果

- 3種類（迅速・議論型, 迅速・総合型, 悠然・総合型）に分類



A：迅速・議論型

B：悠然・総合型

E：迅速・総合型

分類型の特徴と各でのプロジェクトの分布

- ・ **迅速・議論型**

- 活動期間の中央値が22日
- コメントが多い

- ・ **迅速・総合型**

- 活動期間の中央値が36日
- 偏りなく活動

- ・ **悠然・総合型**

- 活動期間の中央値が196日
- 偏りなく活動

- ・ **homebrew**

- 迅速・議論型：5人
- 迅速・総合型：3人
- 悠然・総合型：3人

- ・ **node**

- 迅速・議論型：4人
- 迅速・総合型：3人
- 悠然・総合型：2人

まとめ

- ・ **GitHubで活発なプロジェクトに所属する開発者にどのようなタイプがいるのかを分類した**
 - 3種類
 - ・ 迅速・議論型, 迅速・総合型, 悠然・総合型
 - OSSに参加する開発者の構成を調べることで, 成功しているOSSプロジェクトはどのような開発者で構成されているかを明らかにできる可能性がある

所感

- ・ やりたいこと, 先行研究の問題点がしっかりしていてわかりやすかった
- ・ 先行研究が参考になった

ワード法

・ クラスタ間の距離測定方法

(1) ウォード法

・ 2つのクラスターP,Qを結合したと仮定したとき、それにより移動したクラスターの重心とクラスター内の各サンプルとの距離の2乗和, $L(P \cup Q)$ と、元々の2つのクラスター内での重心とそれぞれのサンプルとの距離の2乗和, $L(P), L(Q)$ の差

$$\Delta = L(P \cup Q) - L(P) - L(Q)$$

が最小となるようなクラスター同士を結合する手法。

Δ の値を情報ロス量という。

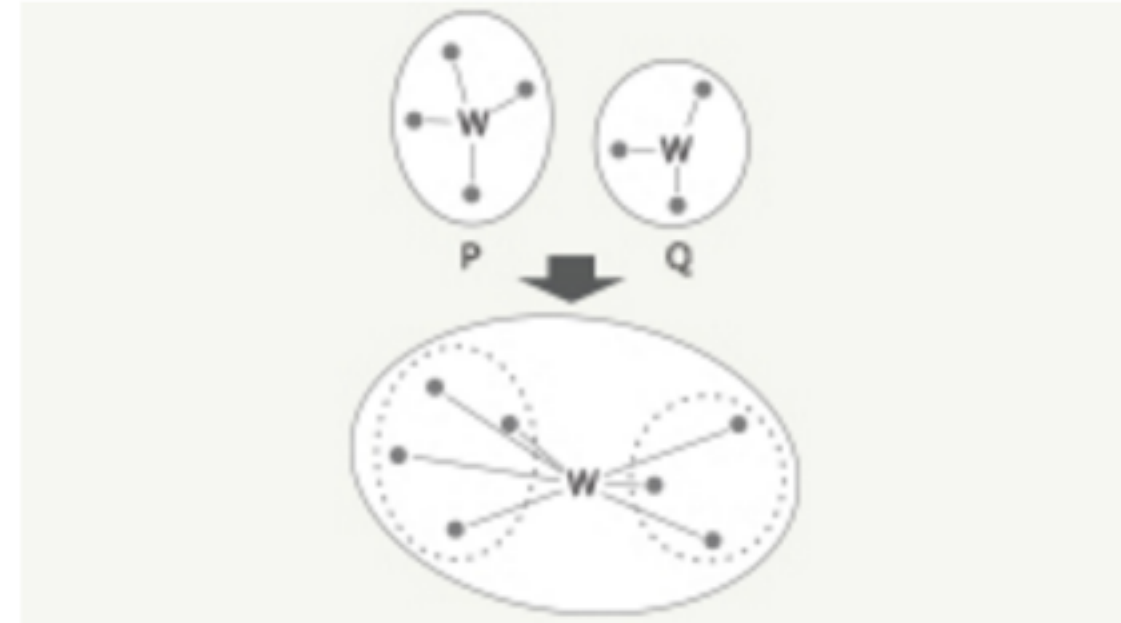


図15.ワード法のイメージ

・ 計算量が多いが, 分類感度が良いので, よく使われる

ワード法を使った階層クラスタ分析

- ・ 似ている組み合わせから, クラスタにしていく
- ・ 最終的には, 右図のような樹形図 (デンドログラム) を作成する

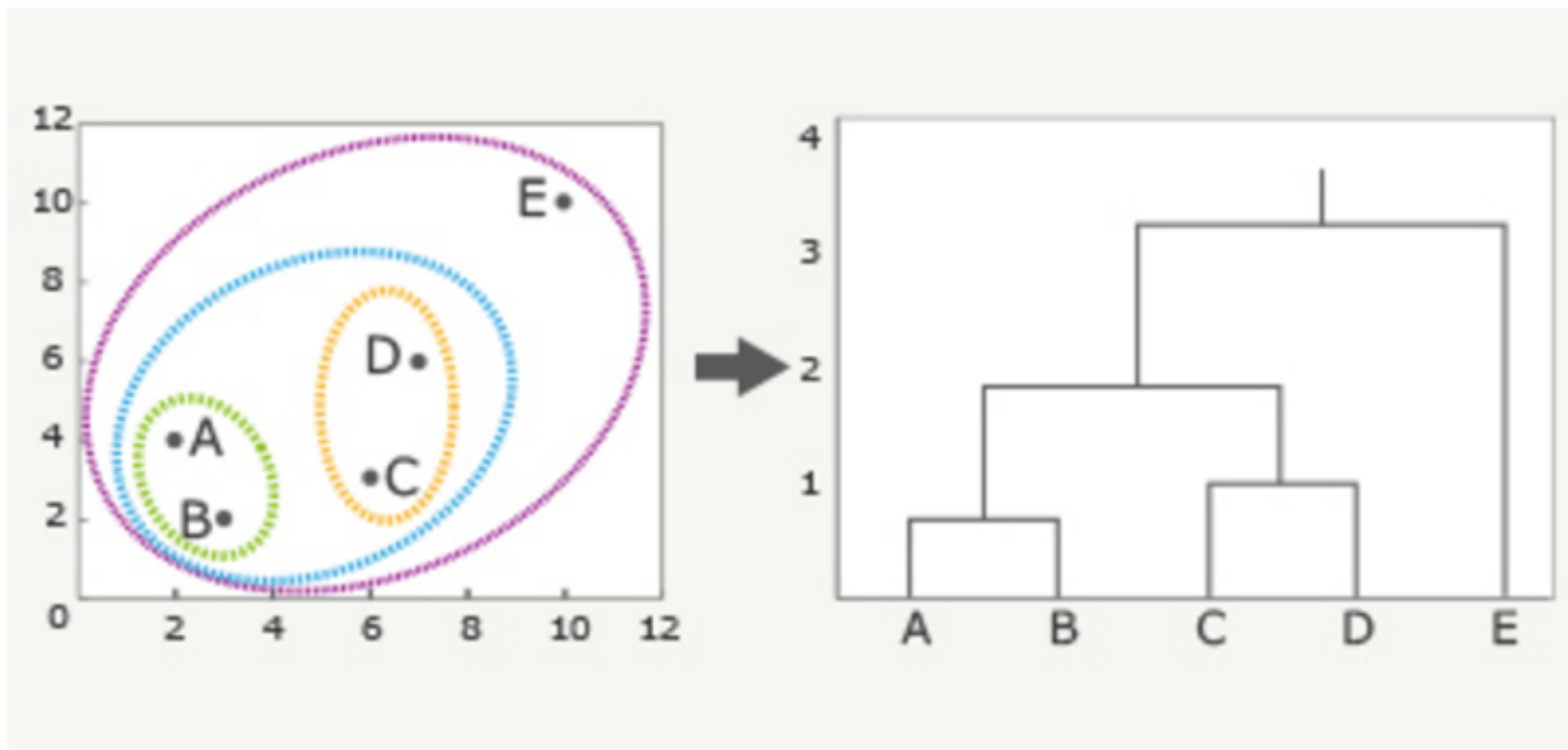


図10.階層クラスタのイメージ