FOSE2010 ミニチュートリアル

データマイニング技術を応用したソフトウェア構築・保守支援

小林隆志(名大)

林晋平(東工大)

はじめに; About Us

小林隆志(発表者)

- 名古屋大学 大学院情報科学研究科 阿草・結縁研 所属
- 再利用,プログラム理解などの開発支援に興味あり
- データ工学の研究にも従事(2002年より)
  - マルチメディア情報検索,データマイニング応用など

林晋平

- 東京工業大学 大学院情報理工学研究科 佐伯研 所属
- ソフトウェア変更の分析・適用支援に興味あり
- 開発履歴を用いたリファクタリング支援の研究で学位取得(2008)

2

はじめに；ねらい

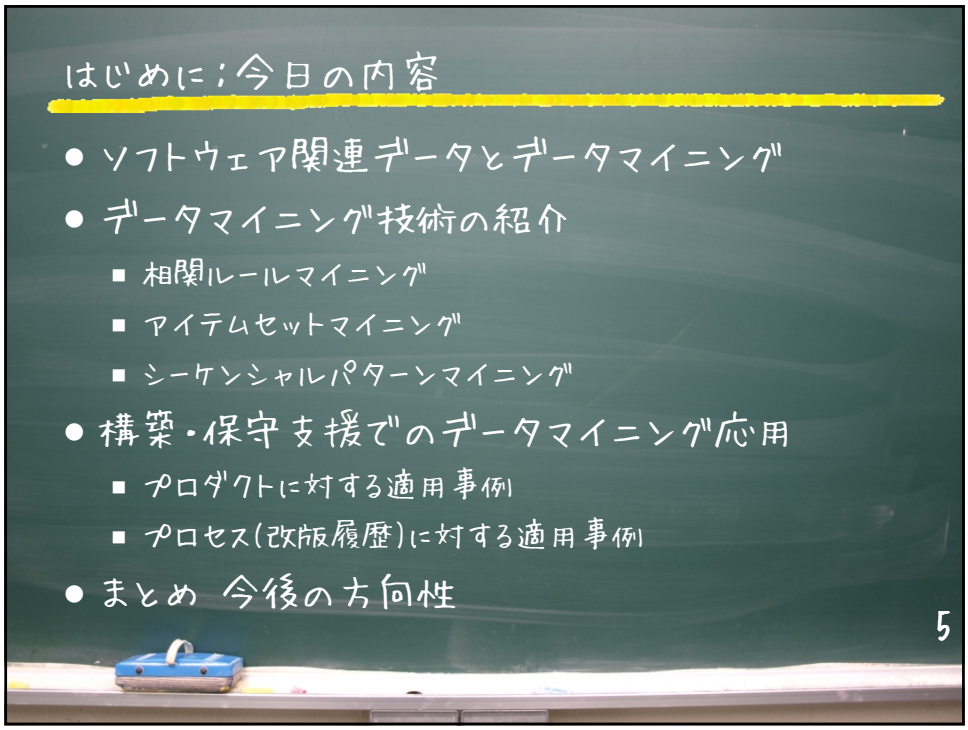
- データマイニング技術(の一部)に関して；
  - 聞いたことがある → 説明できる／利用できる
- SE分野におけるデータマイニング応用事例
  特に，構築・保守に関連する支援手法；
  - 聞いたことがある → 説明できる
    → 自身の研究に応用できる

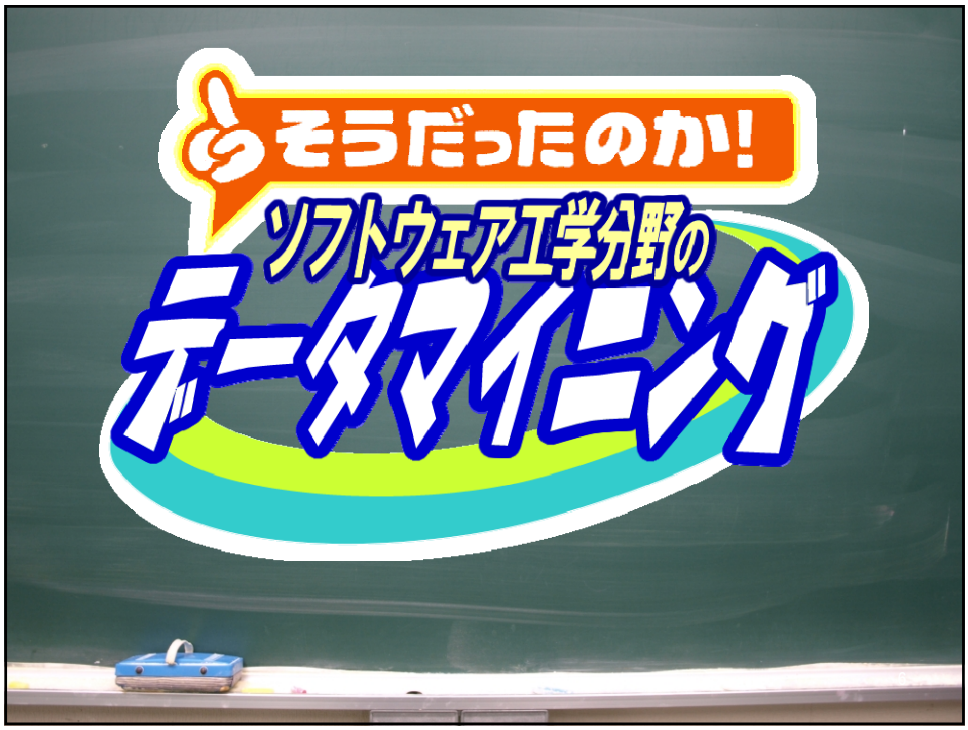
3

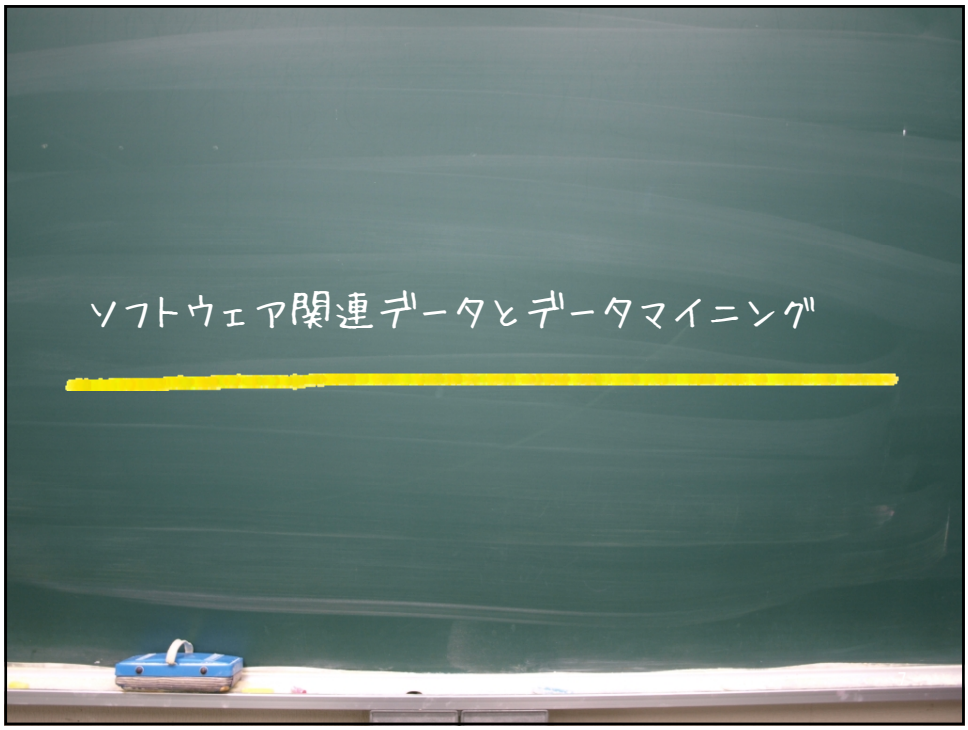
はじめに；お知らせ

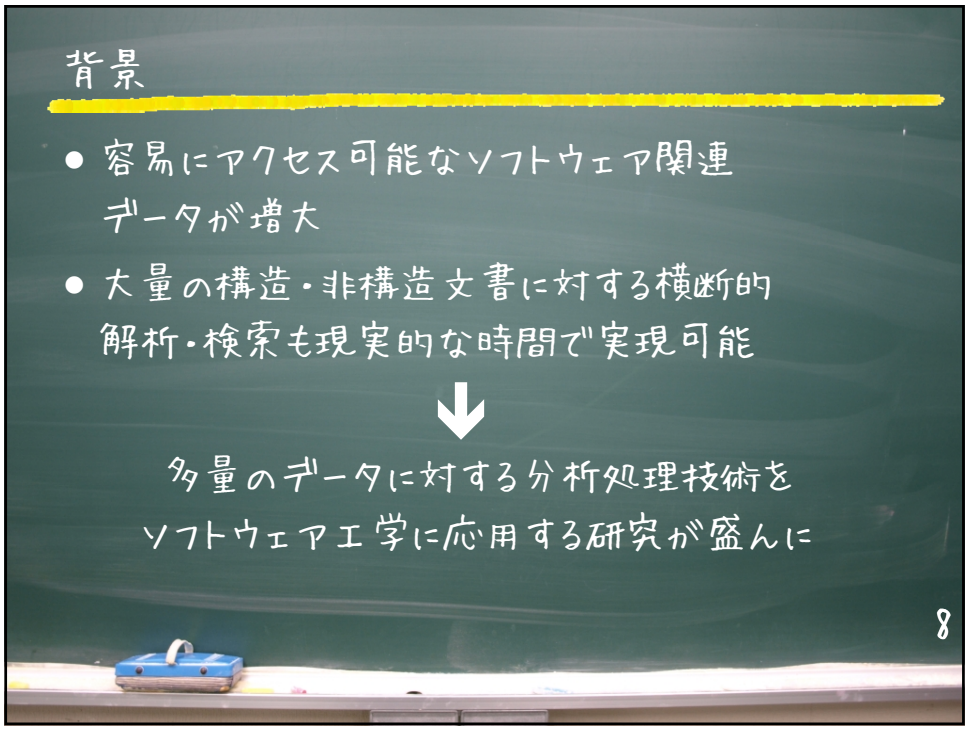
- 後半は以下の解説論文のダイジェスト版です
  - 小林隆志，林晋平：データマイニング技術を応用した
    ソフトウェア構築・保守支援の研究動向，コンピュータソフトウェア
    Vol.27，No.3（2010），pp.13-23 Aug 2010.
    http://www.jstage.jst.go.jp/article/jsssst/27/3/27_3_13/_article
    ※ 本チュートリアル中の参考文献の引用番号は論文と一致させてます
  - Now on Web
- MSR School in Asia があります
  - 当該分野の著名な研究者によるチュートリアル
    - Ahmed E. Hassan，Thomas Zimmermann，Sung kim
    - IWESEP2010（12/7-8，NAIST）の一部
  - 無料です．これは行くしかない！！

4









ソフトウェア関連データとは？

- 各開発工程で生成される多種多様なデータ
- 大きく以下の4つに分類される[38]
  - 構造情報
  - 動的情報
  - 語彙情報
  - プロセス情報

9

ソフトウェア関連データ（1）

- **構造情報：**
  - ソフトウェアの静的構造に関するもの
  - 抽象構文木パッケージ構造
  - コールグラフ/PDGなど構造要素間の関係
- **動的情報：**
  - ソフトウェアを実行した際に得られるもの
  - プロファイル情報
  - プログラムの実行時の振る舞いに関する情報

10

ソフトウェア関連データ（2）

- 語彙情報：
  - 設計意図や対象ドメインの抽象表現となりうる情報
  - 要求仕様書，設計書，マニュアル
  - プログラム中のコメント，識別子も含まれる
- プロセス情報：
  - 開発プロセスに関連するもの
  - 開発履歴，ファイル間の同時更新関係情報
  - バグレポート等

11

ソフトウェア関連データとソフトウェア工学

- 属人性の排除
  - ソフトウェア工学の大きなテーマ
  - 個人／組織内の暗黙知→形式知→活用　が有効
- 暗黙知をどのように発見・形式知化するのか？

↓

関連データに対するマイニング

12

データマイニング技術適用の流れ

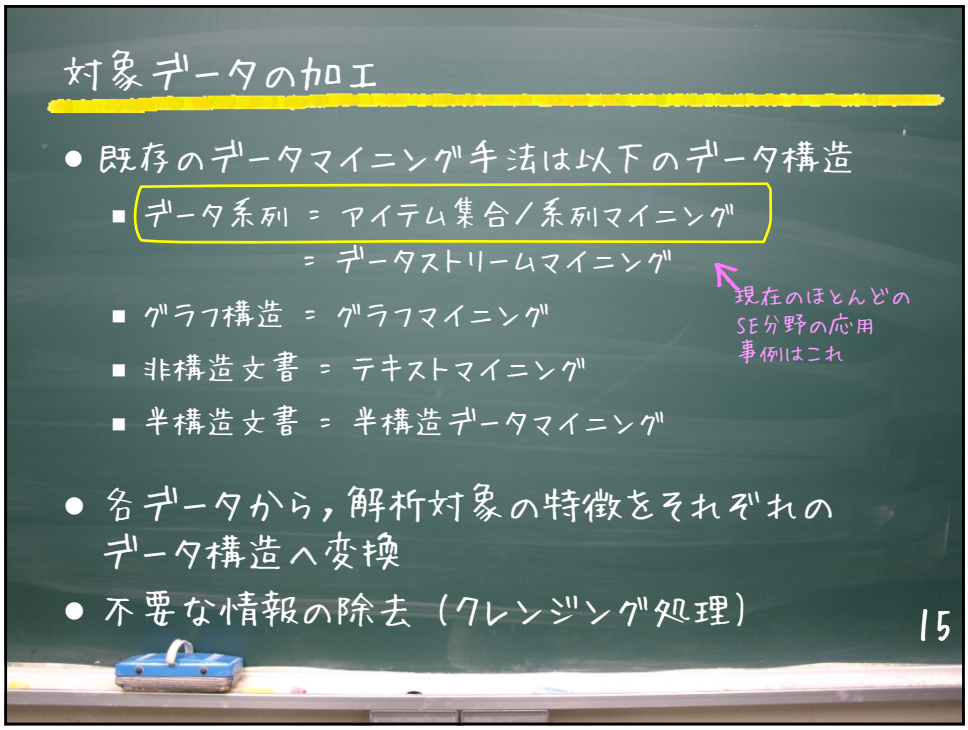
- 対象データの決定・収集
- 対象データの加工
- データマイニング技術の適用
  - 多くは既存技術．一部では専用の手法を開発も
- 発見した結果の利用
  - ルール・パターン→違反発見，入力支援

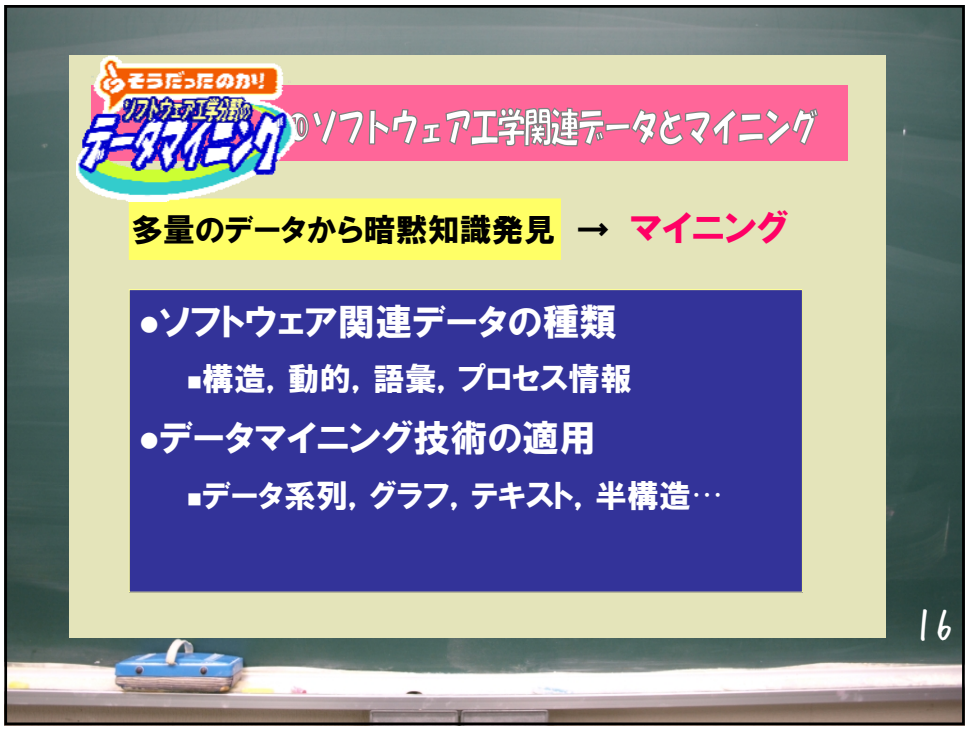
13

対象データの決定・収集

- 良い結果は正しいデータからしか得られない
  - マイニングは頻度に基づき発見
  - 質の悪いデータは精度・効率双方の面で悪
- 組織内外から随時発見・収集
  - 自前リポジトリ or クローリング
- Web上から逐次発見
  - 検索エンジンの結果を用いる
  - 目的の情報に近い情報のみでマイニングが可能

14





データマイニング技術の紹介

データマイニングとは？

- 対象データに関する有益な情報を
  データを解析して機械的に発見すること
- 有益な情報？
  - 対象データ間に存在する潜在的な相関性
  - データ間の関係に関する暗黙知
- 注意；データそのものを探すのは検索
  - 「大量のデータから条件に合うものを検出」は違う
  - 統計処理，情報分析・解析の一種

18

Mining is Voodoo on a Mountain of Data ???

- 「頻出するものは良いものであり再利用できる法則となる」という考え
  - 多くは頻出する組合せや構造を求める問題に帰着
- つまり当たり前の法則が見つかる
  - その中に“宝”も隠れている

(ビール,ナッツ),(ビール,ピザ)

(ビール,おむつ)

19

相関ルールマイニング: 相関ルールとは?

- 相関ルール (association rule)
  - 事象Xと事象Yに相関があることを示すルール
  - X→Y (X:前提部, Y:結論部)
- 例
  - (パン, バター) → ミルク
  - (17:00以降,おむつ) → ビール (Wall St. J 1992)
  - 風が吹く → 桶屋がもうかる ?
  - ブログに書かれる → 不幸になる ?

20

相関ルールマイニング：ルールの評価尺度

- 支持度(support)＝全データ中の割合(頻度)
  - 「風が吹いて桶屋が儲かった」がどれだけあるのか？
  - support(X) ＝ 全データ中にXが含まれる割合
- 確信度(confidence)＝条件付き確率
  - 「風が吹く」と必ず「桶屋は儲かる」のか？

$$\text{confidence} = P(Y \mid X) = \frac{\text{support}(X \rightarrow Y)}{\text{support}(X)}$$

$$\text{support}(X \rightarrow Y) = \text{support}(X \cup Y)$$

要素の集合なので和集合

- support計算 ＝ 頻出アイテム集合計算

21

頻出アイテム集合の発見

- データ系列から頻出アイテム集合Fを求める
  - I：アイテム集合，T:トランザクション集合
    - トランザクション tはデータ系列の単位に相当
  - 頻出 ＝ 最小支持度(minsup)以上の支持度

$$\mathrm{F} = \left\{ f \mid \text{support}(f) = \frac{|\{t \mid t \in T, f \subseteq t\}|}{|T|} \geq \text{minsup} \right\}$$

- 飽和(closed)頻出アイテム集合
  - 他の部分集合にならないもの

22

Aprioriアルゴリズム [4:Agrawal VLDB94]

- もっとも代表的なアルゴリズム
- 全ての組合せの頻度を段階的に計算
  - 要素1個の頻出集合→要素2個の候補集合
    →頻度計算→要素2個の頻出集合…
- 頻度のdownward closure propertyを利用
  - 最小支持度以下のものは組合せ計算しない
  - 例: $\sup(A) = 10$ $\sup(B) = 2$ なら $\sup(A \cup B)$ は2以下
- 無駄な計算をしないため効率が良い

23

Aprioriアルゴリズムの利用

- Rを使うととても簡単
  - arules パッケージを利用
  - 関数にデータと支持度，確信度等を与えるだけ

```
% library(arules)
% all <- read.transactions("all.csv",sep=" _,_ ",
                            rm.duplicates=TRUE)
% all.ap <- apriori(all,parameter=list(support=0.001,
                confidence=0.5, maxlen=3, minlen=2))
% t=capture.output(inspect(
            head(SORT(all.ap,by="support"),n=1000)))
% write(t,"result.txt")
```

24

そのほかのアイテム集合マイニング

- Apriori の改良
  - 候補集合作成の効率化，スキャン回数の削減
  - HPA-ELD 法[33: SIGMOD98]など並列化による高速化
- 候補集合を作らない方法
  - 頻出パターン集合(FP-Tree)を逐次更新する
  - Apriori より高速．FP-growth[13:SIGMOD2000] が有名
- 飽和頻出アイテム集合のみを求める手法
  - LCM[43]，FP-close[12]など実装が公開されている

25

頻出系列の発見 Sequential Pattern Mining (SPM)

- 集合から系列(sequence)へ拡張
- 系列の包含関係

系列 $\alpha = \langle a_1 a_2 \cdots a_n \rangle$ と $\beta = \langle b_1 b_2 \cdots b_m \rangle$ に対して

$\alpha \sqsubseteq \beta$ $a_1 \subseteq b_{i_1},\ a_2 \subseteq b_{i_2}, \ldots,\ a_n \subseteq b_{i_n}$

を満たす整数が $i_1 < i_2 < \cdots < i_n$ 存在するとき．

- 系列を包含する頻度が支持度

$$support(\alpha) = \frac{|\{\beta \mid \beta \in S \wedge \alpha \sqsubseteq \beta\}|}{|S|}$$

26

(飽和)頻出系列の例 [A1: Zaki, ML2001]より

SID	EID	Items
1	10	C D
1	15	A B C
1	20	A B F
1	25	A C D F
2	15	A B F
2	20	E
3	10	A B F
4	10	D G H
4	20	B F
4	25	A G H

対象データベース

SID: Sequence ID
EID: Event ID

Frequent Sequence	Support
<{A}> <{B}> <{F}>	1.0
<{D}>	0.5
<{B,F}>	1.00
<{A,B}> <{A,F}>	0.75
<{D},{A}> <{D},{B}> <{D},{F}> <{B},{A}> <{F},{A}>	0.50
<{D},{B,F}> <{B,F},{A}>	0.50
<{A,B,F}>	0.75
<{D},{F},{A}> <{D},{B},{A}>	0.50
<{D},{B,F},{A}>	0.50

minsup = 0.5 (2以上登場) として計算

例;⟨{D}{BF}{A}⟩は以下の部分系列をカウント

SID1=⟨{CD} {ABC} {ABF} {ACDF}⟩

SID4=⟨{DGH} {BF} {AGH}⟩

例;飽和系列は ⟨{A,B,F}⟩, ⟨{D},{B,F},{A}⟩

27

代表的なSPMアルゴリズム

- GSP [34: EDBT96]
  - 最初提案(Apriori拡張 [5: ICDE95])の改良
- SPADE [A1: Machine Learning2001] ← R用のlibraryがある
- Prefix/Postfixを用いて深さ優先探索
  - PrefixSpan [30: ICDE2001] → 多くの実装がある
- 速度はGSP < SPADE < PrefixSpanの順で高速[A2:TKDE04]
  - しかし，頻出部分系列が多いと遅くなる [A3: DMKD2007]
- 飽和頻出系列のみを高速に発見可能
  - BIDE [36: ICDE2004]

28

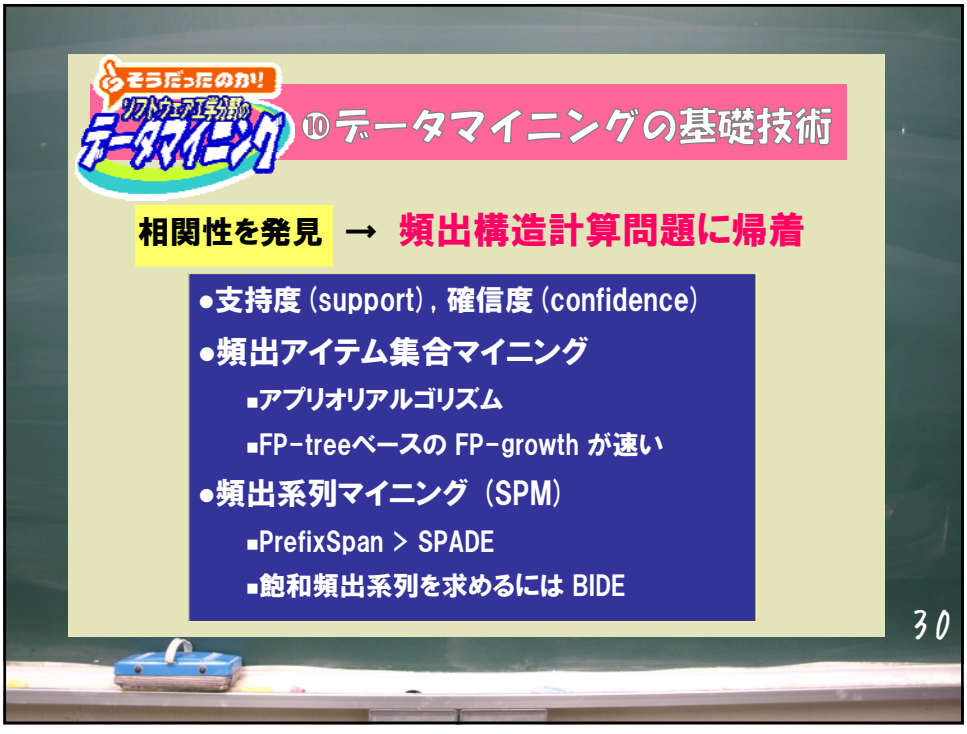
(追加スライド) その他のデータマイニング技術

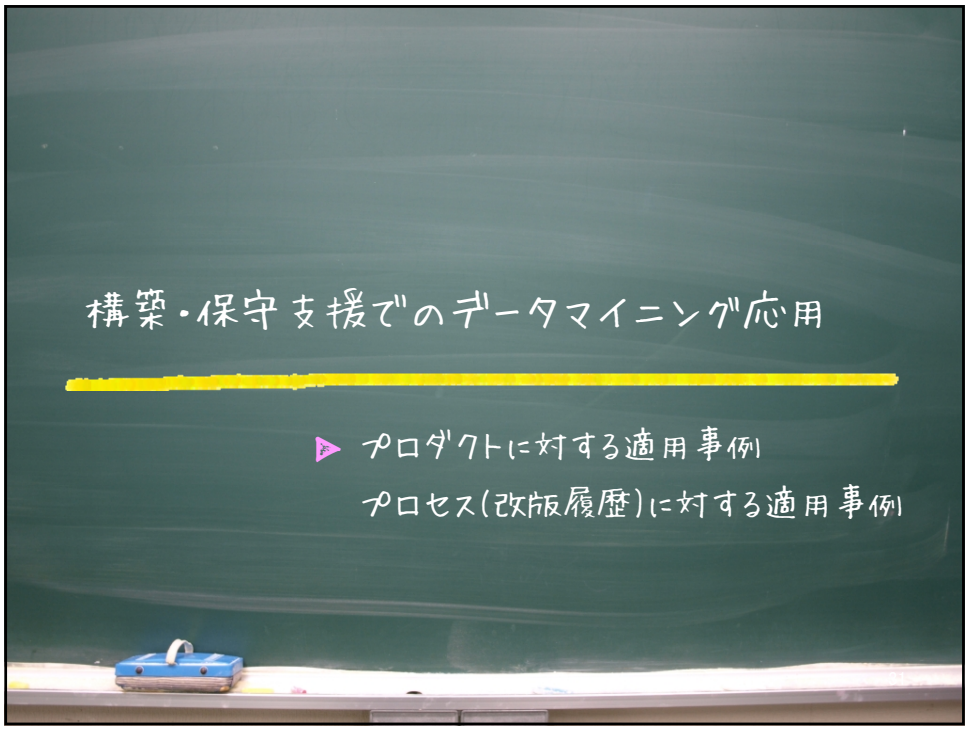
- 今回は，クラスタリング，クラス分類などのデータ解析手法は扱っていません．
- クラスタリング手法を利用するソフトウェア工学分野の研究事例は多数存在します．クラスタリング手法に関しては以下の[49,50]などを参考にしてください

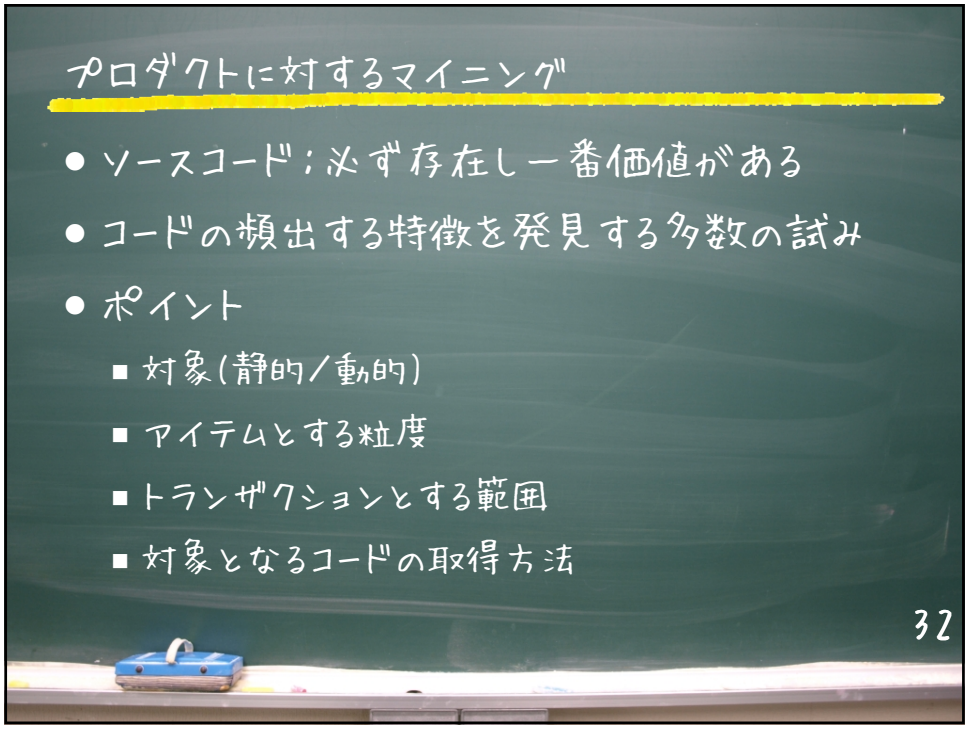
[49] 神嶌敏弘: データマイニング分野のクラスタリング手法(1)，人工知能学会誌，Vol. 18，No. 1 pp. 59-65.(2003)

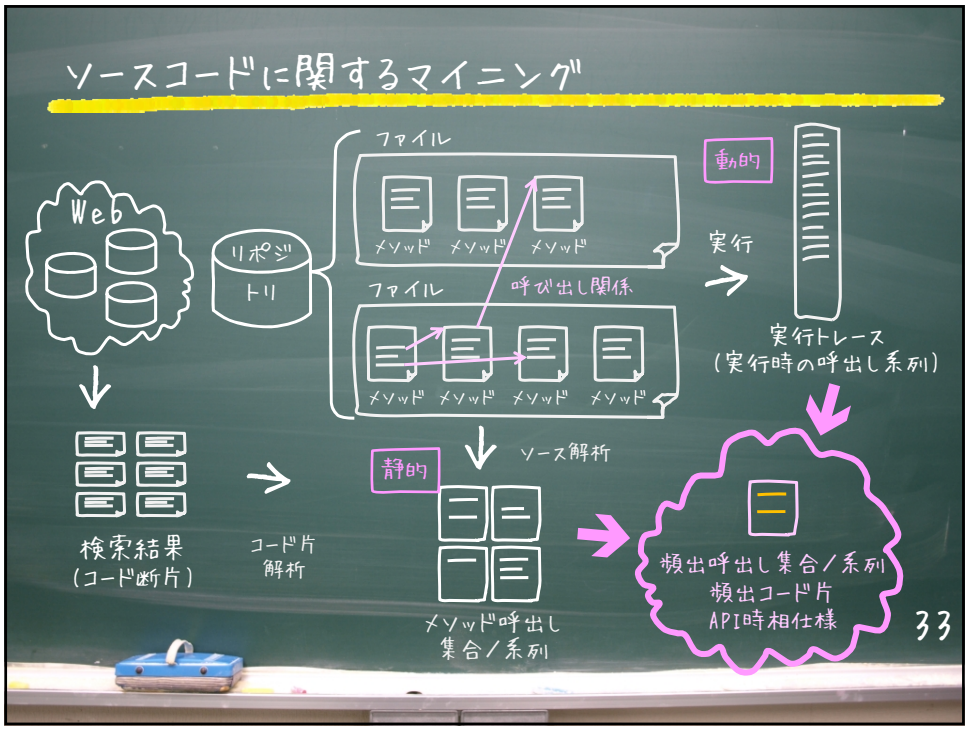
[50] 神嶌敏弘: データマイニング分野のクラスタリング手法(2)，人工知能学会誌，Vol. 18，No. 2 pp. 170-176.(2003)

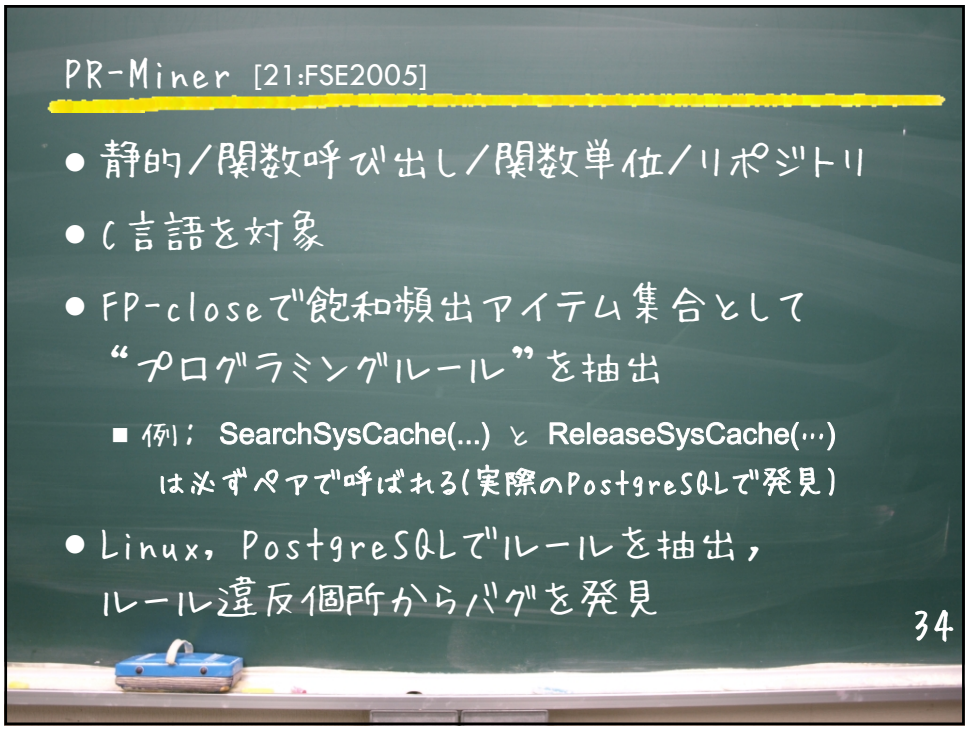
29

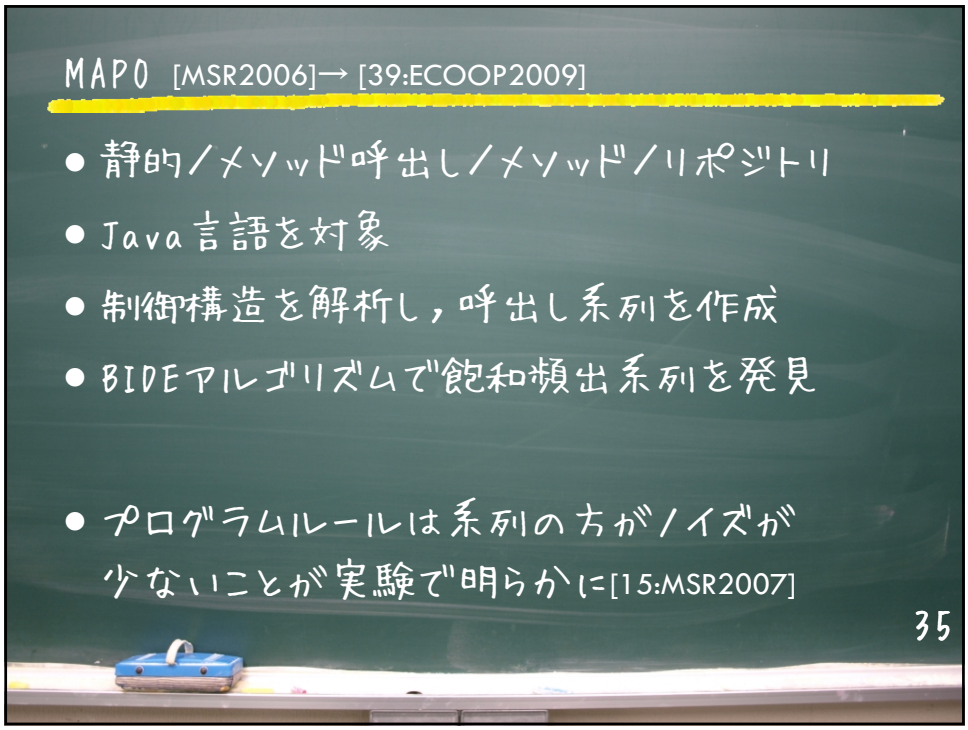


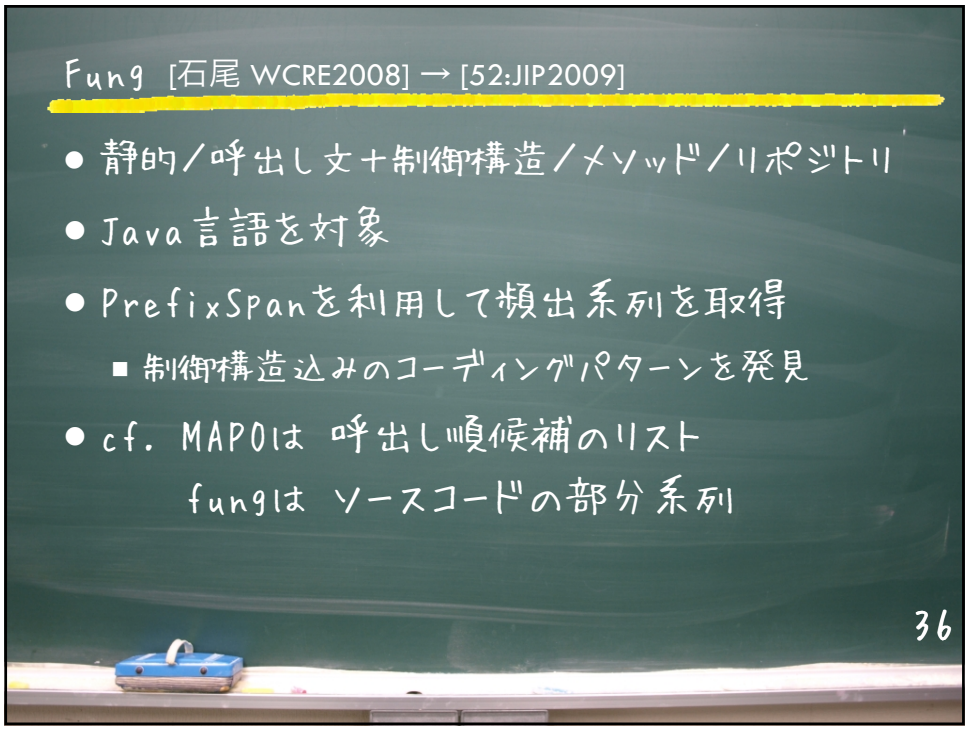












CAR-Miner [A4: ICSE2009]

- 静的／メソッド呼出し+α／メソッド／検索
- Java言語対象
- 特定のメソッドの例外処理パターンを発見

  $(FC^1\ FC^2 \ldots) \wedge FT \Rightarrow (FE^1 \ldots)$

  $(FC^1\ FC^2 \ldots)$ という呼出しパターンに合致し，
  FTを呼び出すメソッドでは，
  $(FE^1 \ldots)$ という例外処理が無くてはいけない
- FTで検索した結果のtry-catchを解析し
  例外処理部にマークを付けてSPM

37

CAR-Miner [A4: ICSE2009] 例

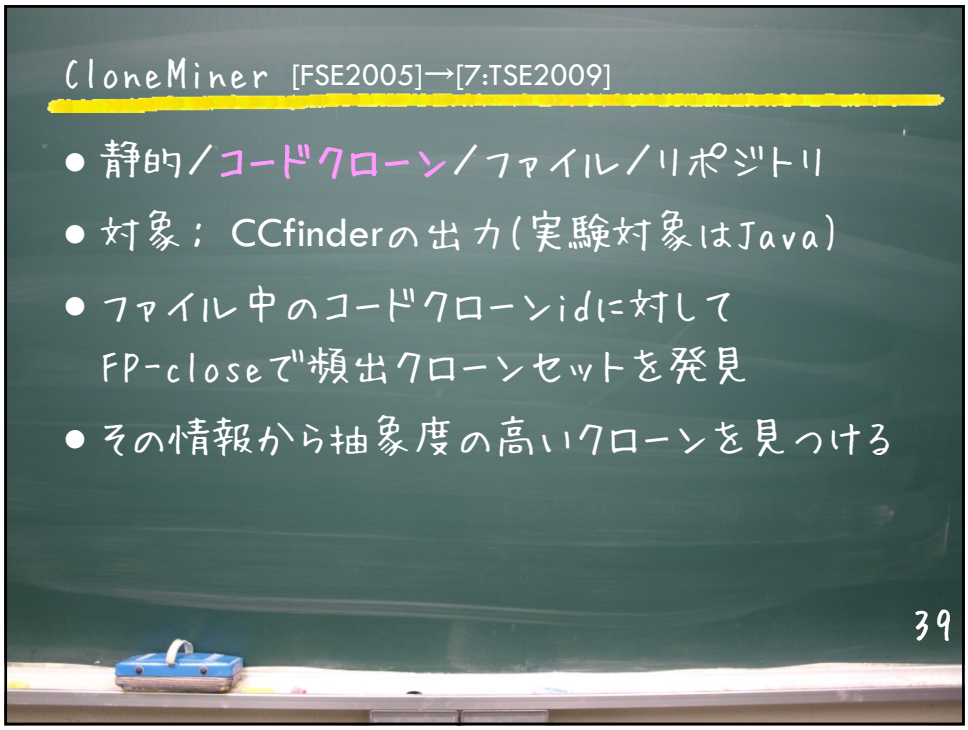
```
1.1: ...
1.2: OracleDataSource ods = null; Session session = null;
     Connection conn = null; Statement statement = null;
1.3: logger.debug("Starting update");
1.4: try {
1.5:     ods = new OracleDataSource();
1.6:     ods.setURL("jdbc:oracle:thin:scott/tiger@ 192.168.1.2:1521:catfish");
1.7:     conn = ods.getConnection();
1.8:     statement = conn.createStatement();
1.9:     statement.executeUpdate("DELETE FROM table1")
1.10:    connection.commit(); }
1.11:    catch (SQLException se) {
1.12:        if (conn != null) { conn.rollback(); }
1.13:        logger.error("Exception occurred"); }
1.14: finally {
1.15:    if(statement != null) statement.close();
1.16:    if(conn != null) conn.close();
1.17:    if(ods != null) ods.close();
1.18: }
```

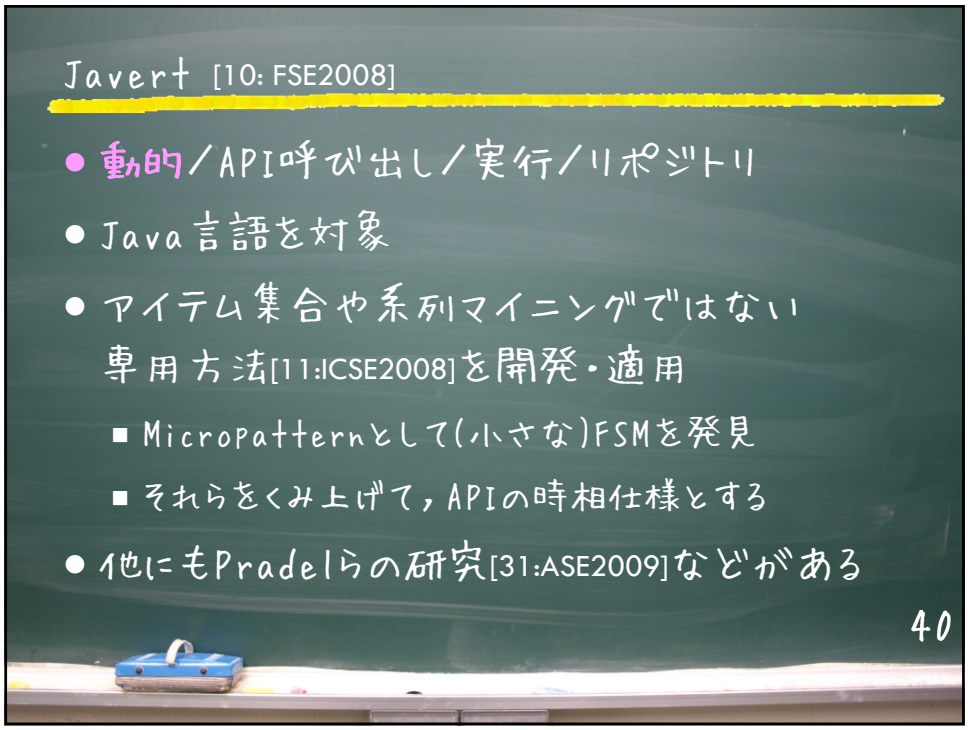
<{FC1},{FC2}>

{FT}

<{FE1} >

38


CloneMiner [FSE2005]→[7:TSE2009]
● 静的／コードクローン／ファイル／リポジトリ
● 対象； CCfinderの出力(実験対象はJava)
● ファイル中のコードクローンidに対して
FP-closeで頻出クローンセットを発見
● その情報から抽象度の高いクローンを見つける
39


Javert [10: FSE2008]
● 動的／API呼び出し／実行／リポジトリ
● Java言語を対象
● アイテム集合や系列マイニングではない
専用方法[11:ICSE2008]を開発・適用
■ Micropatternとして(小さな)FSMを発見
■ それらをくみ上げて，APIの時相仕様とする
● 他にもPradelらの研究[31:ASE2009]などがある
40

RAPID [A5: ASE2008]

- 動的／バグとの関連度／実行／リポジトリ
- 対象：C言語
- 正常／失敗トレースから各行の失敗との相関を Tarantula [A6: ASE2005]で算出
- 閾値以上の行からなる系列群対して BIDE を適用し飽和頻出系列を作成
- 欠陥個所候補のランキングを行う

41

ソースコードに対するその他のマイニング応用

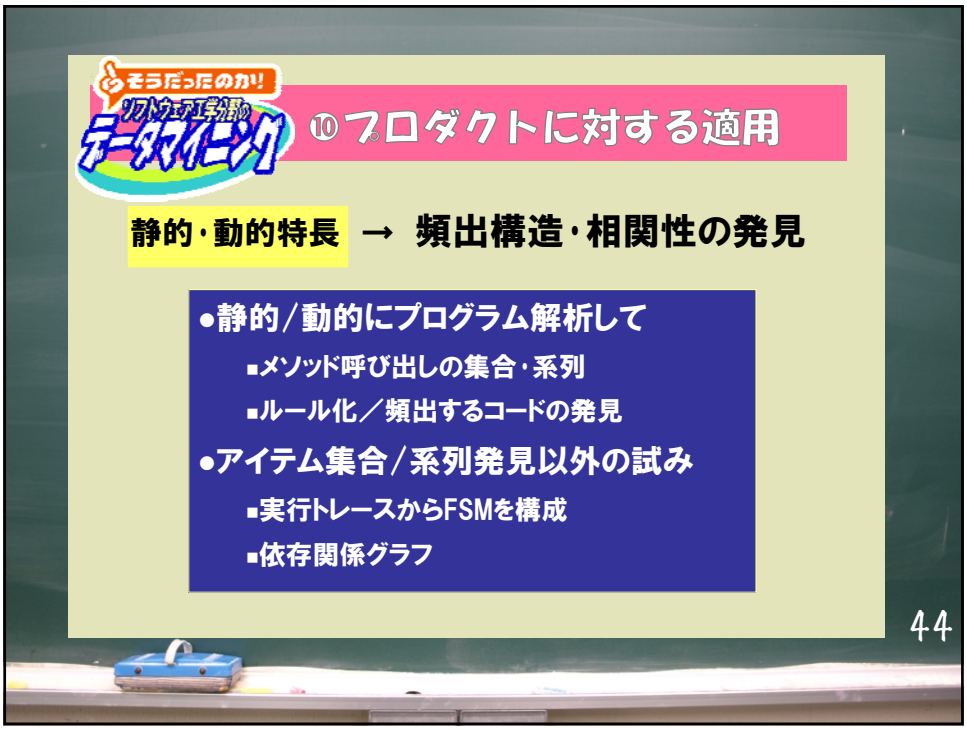
- プログラム構造・依存関係に対するマイニング
- Prospector [24: PLDI2005]
- Xsnippet [32: OOPSLA2006]
  - 型情報に着目し，“インスタンスを作成するために必要な型”という関係でグラフを作成する
    - X のインスタンスはY,Zのインスタンスを引数に作成
  - 元となる型と必要な型を入力すると，グラフの探索により，経由すべき型のリストが提示される

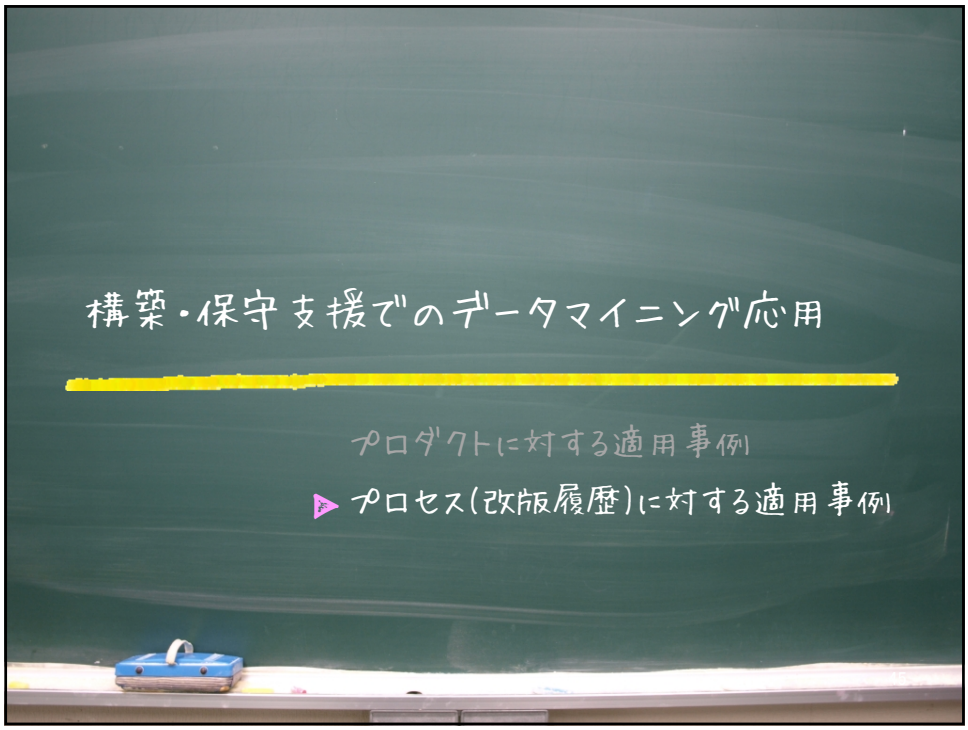
42

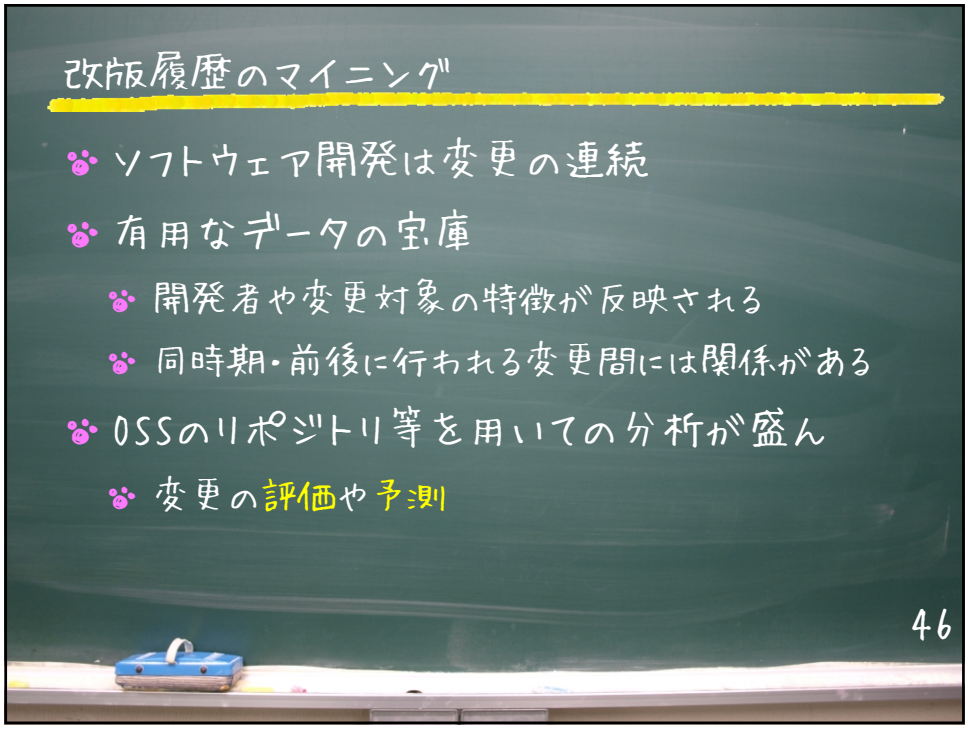
ソースコードに対するその他のマイニング応用

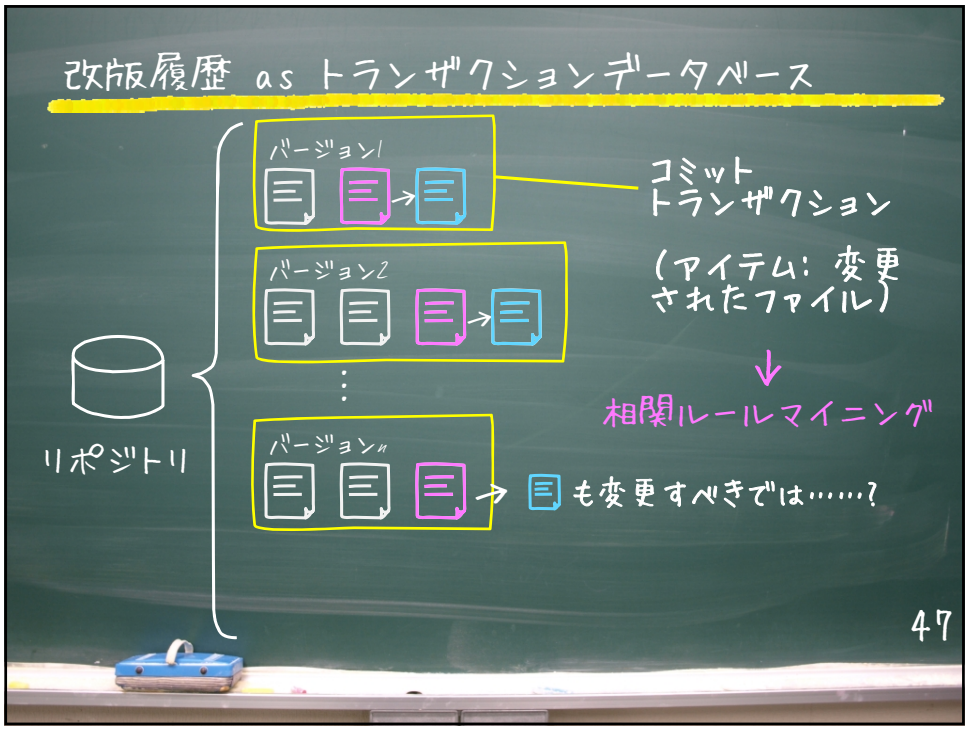
- PARSEWeb [35: ASE2007]
  - キーワードに関連するコード断片を Google Code Searchで発見
  - 不完全なコードの解析は本来不可能
    - コード断片を経験則を多用して解析し
  - 呼び出し系列を作成し，2つの型間での頻出呼び出し系列を提示

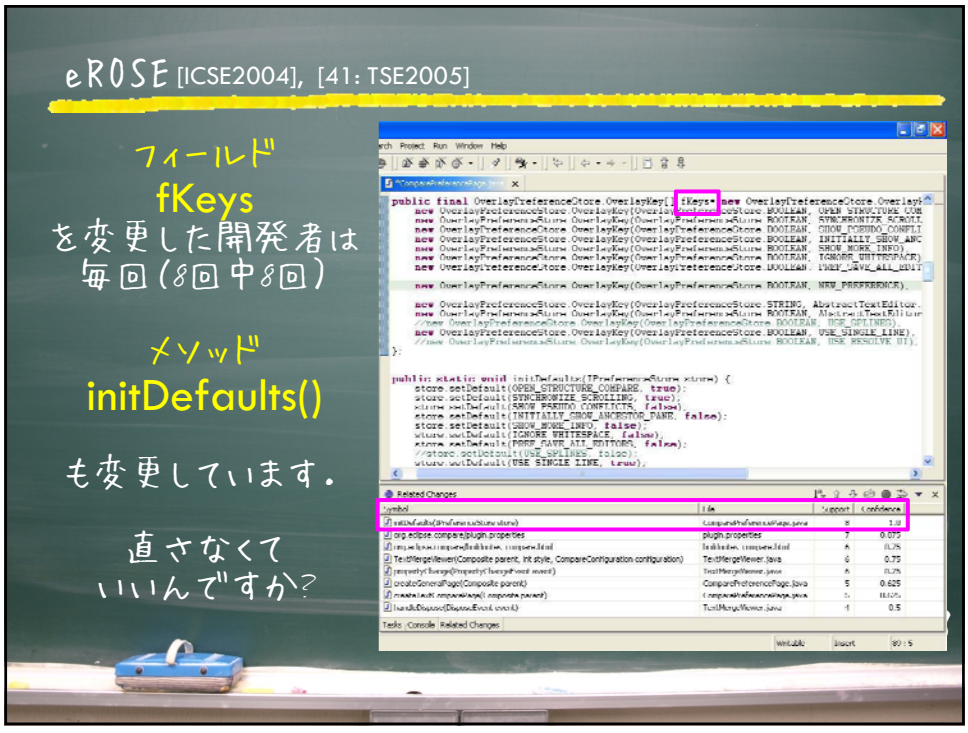
43









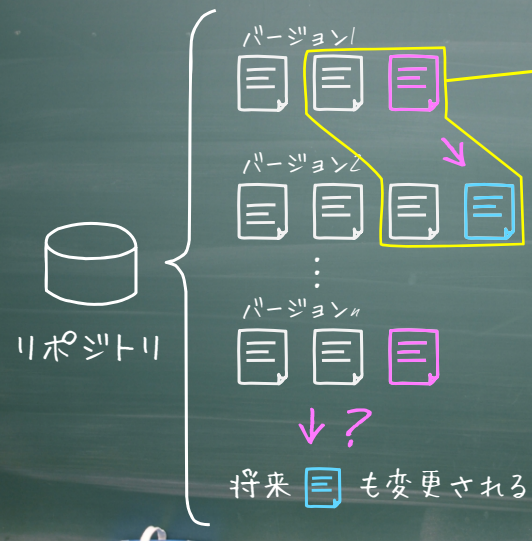


例えばどんな情報が得られる?

- open メソッドを直したら close も直す
- *.c と *.h を同時に直す
- *.py を直したら *.c も直す (Python インタプリタ)
- 2つの(関連)ドキュメントを同時に直す

49

改版履歴 as 系列データベース



50

SPminer[18:MSR 2006], [17: ICPC 2007] によるパターン例

- インタフェース(*.h) → 実装クラス(*.cpp)
- UI定義(*.rc) → ソースコード(*.cpp)
- ソースコード(*.cpp, *.h) → ビルドファイル (Makefile)
- ソースコード → ドキュメント
  - 追跡性の回復にも利用可能

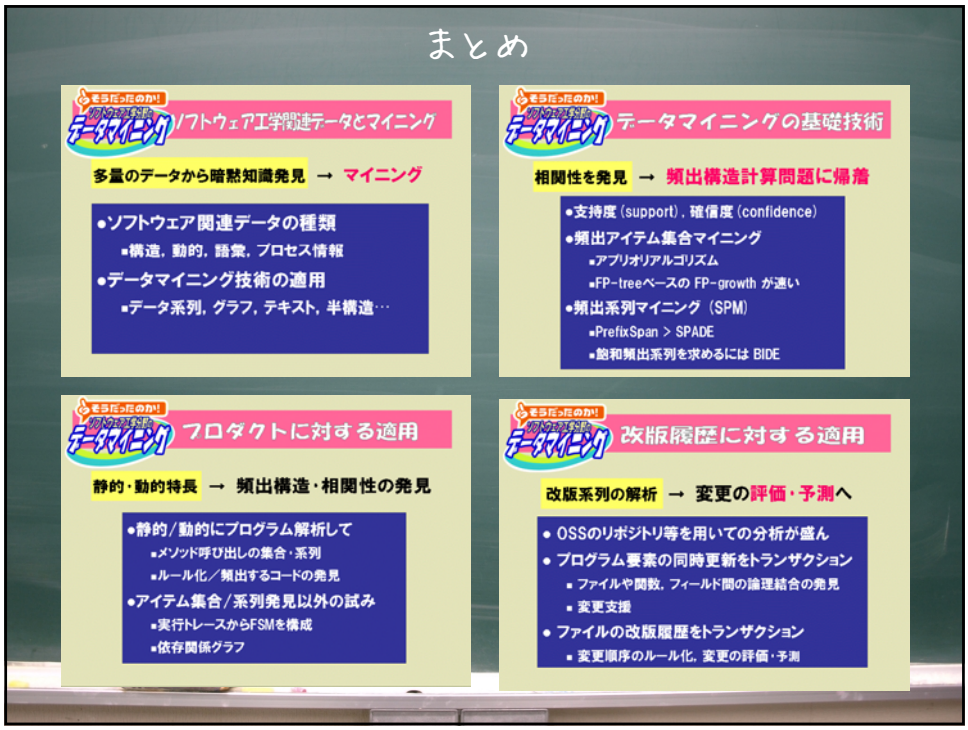
51

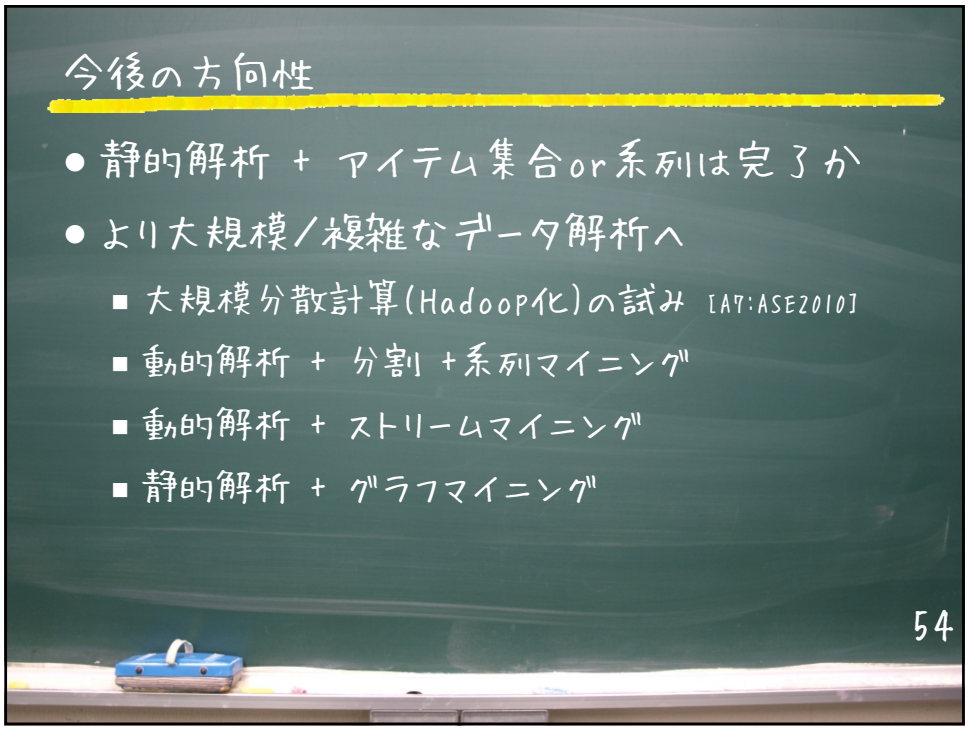
そうだったのか! ソフトウェア工学的データマイニング ⑩改版履歴に対する適用

改版系列の解析 → 変更の評価・予測へ

- OSSのリポジトリ等を用いての分析が盛ん
- プログラム要素の同時更新をトランザクション
  - ファイルや関数，フィールド間の論理結合の発見
  - 変更支援
- ファイルの改版履歴をトランザクション
  - 変更順序のルール化，変更の評価・予測

52


まとめ
そうだったのか! ソフトウェア工学 データマイニング
ソフトウェア工学関連データとマイニング
多量のデータから暗黙知識発見 → マイニング
●ソフトウェア関連データの種類
▪構造，動的，語彙，プロセス情報
●データマイニング技術の適用
▪データ系列，グラフ，テキスト，半構造…
そうだったのか! ソフトウェア工学 データマイニング
データマイニングの基礎技術
相関性を発見 → 頻出構造計算問題に帰着
●支持度（support），確信度（confidence）
●頻出アイテム集合マイニング
▪アプリオリアルゴリズム
▪FP-treeベースの FP-growth が速い
●頻出系列マイニング（SPM）
▪PrefixSpan > SPADE
▪飽和頻出系列を求めるには BIDE
そうだったのか! ソフトウェア工学 データマイニング
プロダクトに対する適用
静的・動的特長 → 頻出構造・相関性の発見
●静的/動的にプログラム解析して
▪メソッド呼び出しの集合・系列
▪ルール化/頻出するコードの発見
●アイテム集合/系列発見以外の試み
▪実行トレースからFSMを構成
▪依存関係グラフ
そうだったのか! ソフトウェア工学 データマイニング
改版履歴に対する適用
改版系列の解析 → 変更の評価・予測へ
● OSSのリポジトリ等を用いての分析が盛ん
● プログラム要素の同時更新をトランザクション
▪ ファイルや関数，フィールド間の論理結合の発見
▪ 変更支援
● ファイルの改版履歴をトランザクション
▪ 変更順序のルール化，変更の評価・予測


今後の方向性
● 静的解析 + アイテム集合or系列は完了か
● より大規模/複雑なデータ解析へ
■ 大規模分散計算(Hadoop化)の試み [A7:ASE2010]
■ 動的解析 + 分割 +系列マイニング
■ 動的解析 + ストリームマイニング
■ 静的解析 + グラフマイニング
54

追加参考文献

[A1] Mohammed J. Zaki, SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning Journal, 42(1/2):31-60. Jan/Feb 2001

[A2] J. Pei et al. : Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. IEEE TKDE, Vol. 16, No. 11, Nov 2004

[A3] J. Han, H. Cheng, D. Xin, X. Yan: Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery, Vol. 14, No. 1, 2007

[A4] S. Thummalapenta, T. Xie: Mining Exception-Handling Rules as Sequence Association Rules, In Proc. ICSE2009

[A5] H. Hsu, J. Jones, A. Orso. Rapid: Identifying bug signatures to support debugging activities. In Proc. ASE2008

[A6] J. Jones, M. J. Harrold. Empirical evaluation of the Tarantula automatic fault-localization technique. In Proc. ASE2005

[A7] W. Shang, B. Adams, A. Hassan: An Experience Report on Scaling Tools for Mining Software Repositories Using MapReduce. In Proc. ASE2010

55

謝辞

- スライド作成にあたり以下を利用しました
  どうもありがとうございました
- 黒板風PowerPointテンプレート
  By 鳴門教育大学 地域連携センター藤原さん
  http://rcse4.naruto-u.ac.jp/works/modules/blog/details.php?bid=22
- 日本語かなフォント「ふい字」
  http://hp.vector.co.jp/authors/VA039499/

56