アルゴリズムとプログラミング実践講座

http://akashi.ci.i.u-tokyo.ac.jp/mary/lectures/algorithm/

火曜　13:00 -- 14:30
I-REF 棟 6階　ヒロビー

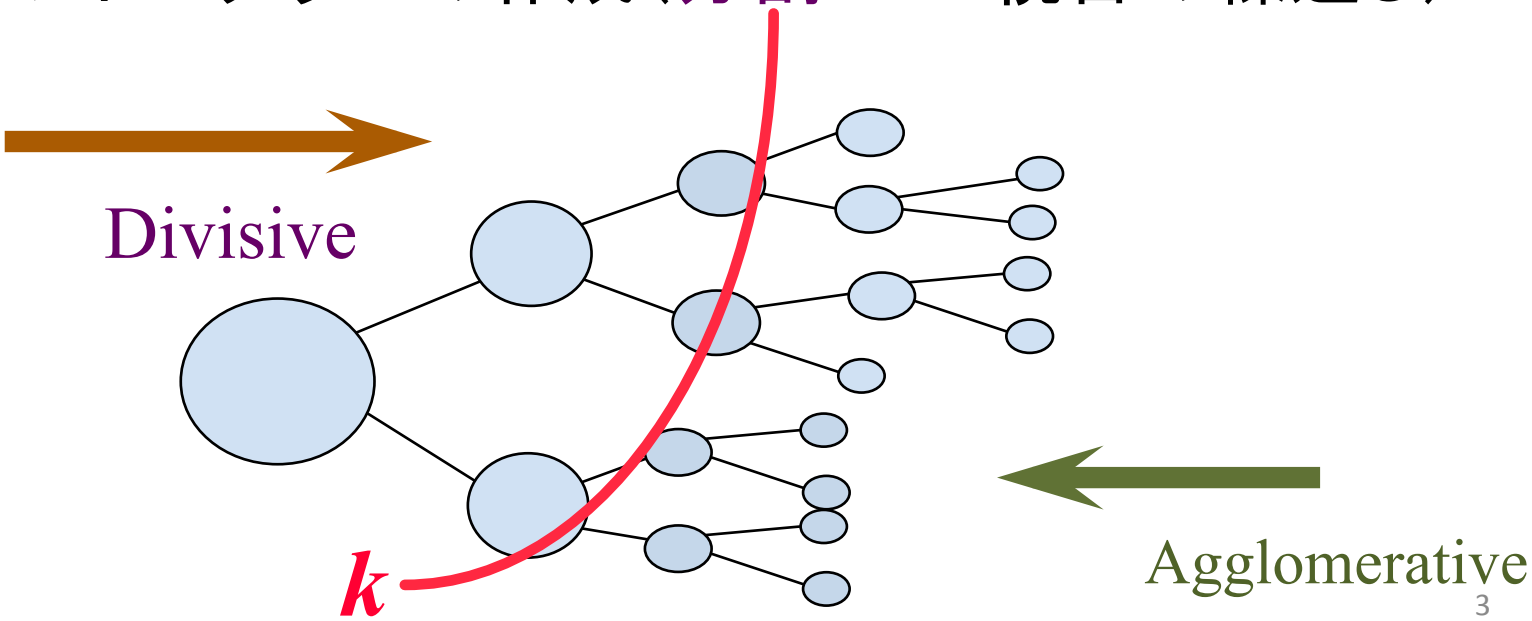
稲葉真理　with 浅井大史・手塚宏史

先週のトピック

クラスタリング

フラットクラスタリングと階層クラスタリング

- フラットクラスタリング
  - クラスタ間に階層構造がない
- 階層クラスタリング
  - デンドログラムの作成（分割　or　統合の繰返し）



二つのフラットクラスタリング

- Partition　Method

  [Kaufman, Rousseeuw]

  → 代表点はクラスタの要素から選ぶ

- k-means Method

  → 代表点はクラスタの平均を使う

  幾何的な性質を使うことが多い (Voronoi)

k-クラスタリングの　整数計画法モデル
Integer Programming model

$d_{ij} = d(x_i, x_j)$　x_i　と　x_j　の距離

$$y_j = \begin{cases} 1 & \text{点 } x_j \text{ が representative} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{点 } x_j \text{ によって、点} x_i \text{ が代表される} \\ 0 & \text{otherwse} \end{cases}$$

$$\min \sum \sum d_{ij} x_{ij}$$

s.t.　$\sum y_j = k$　（代表は、全部でk個）

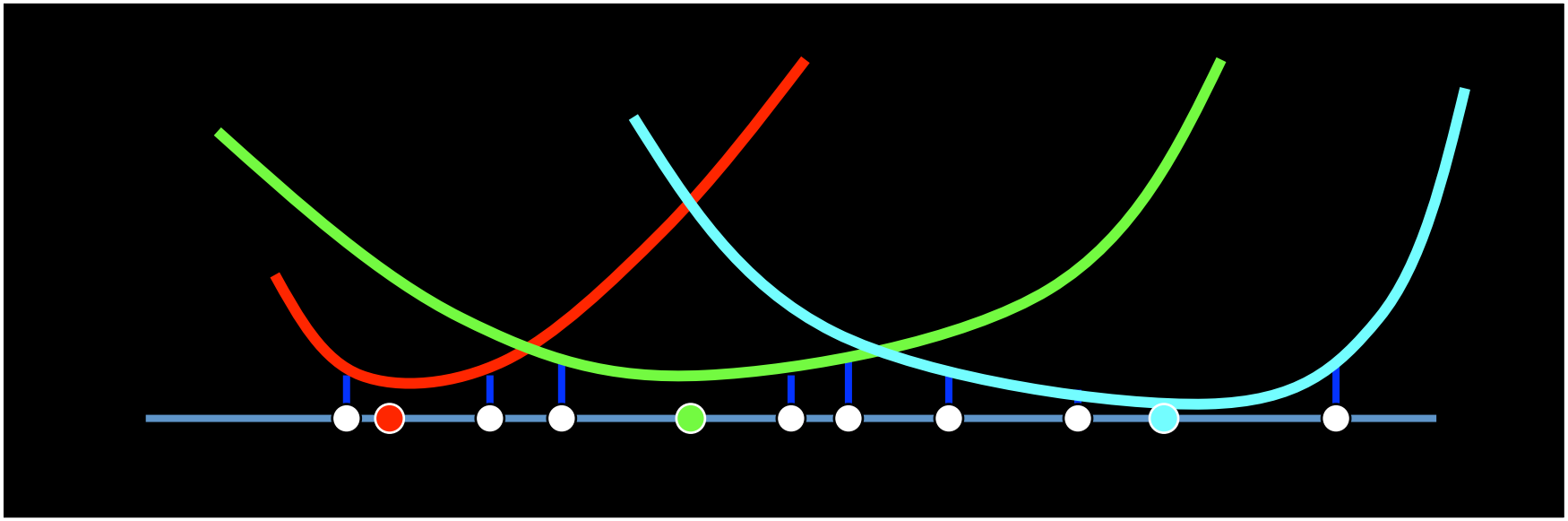
$\sum_{j=1..n} x_{ij} = 1$ (必ずどれかに代表されている)

$x_{ij} \leqq y_j$　（矛盾しないために）

加算的幾何クラスタリングの持つ良い構造

- 空間の分割と考えることができる。
- コスト関数が凸関数の場合

  → ポテンシャルファンクション



データ集計

- 人口調査　BC 3000 エジプト
- 計算機　→　データレコード、　データタプル
  ファイルはバイト列という抽象化は UNIXから
- Relational Data Base (1970 Codd)
- Association Rule Mining (1993 Agrawal)
  →　問い合わせ無しで　ルールを見つける

ビッグデータ

- 限りなくすべてのデータを扱う
- 量さえあれば、精度は重要ではない
- 原因と結果を求める因果関係からの脱却

データマイニング

- 予測モデルの構築
  - 決定木の構築、ベイズ分類、ニューラルネット、サポートベクターマシン、CAEP
  - 数値予測モデル（重回帰分析）
- 特徴パターンの発見
  - バスケット解析
  - 逸脱発見（Deviation Detection)
- クラスタリング

Data Preprocess

- 普通、集めただけでは使えない
  - データが足りない
  - データが間違ってる。
  - データが矛盾してる

- データマイニングにおいて、 データのプリプロセス (preparation, cleaning, transformation) が、全作業の90%をしめる。

Data Cleaning

- 欠けてるデータの補完
  - それっぽいデータをいれる (EM法, ベイジアン、decision tree, 他の値の平均)
- おかしなデータをはじく
- データの矛盾を解消する
- 冗長なデータを整理する

バスケット解析

したいこと

顧客の買い物データがあるとき、

品物A と 品物 B は、一緒に良く売れる（など）

→ 可能性の列挙は時間がかかりすぎ

→ エキスパート（人間）に頼ると、常識的

→ 季節、天気など、他の要因も考慮。

→ 数値属性の扱いが困難 （e.x. 30歳以上）

バスケット解析

- $I = \{i_1, i_2, \ldots, i_m\}$: アイテム集合
- t : transaction　ある買物のアイテム集合 $t \subseteq I$.
- T: transaction 集合 $T = \{t_1, t_2, \ldots, t_n\}$.

- $sup = \Pr(X \cup Y)$　　*支持度　＞＝　minsup*
- $conf = \Pr(Y \mid X)$　　確信度 >= minconf

となるような　X, Y　の組を列挙する

例

t1: 牛肉、鶏肉、ミルク
t2: 牛肉、チーズ
t3: チーズ、ブーツ
t4: 牛肉、鶏肉、チーズ
t5: 牛肉、鶏肉、洋服、チーズ、ミルク
t6: 鶏肉、洋服、ミルク
t7: 鶏肉、洋服、ミルク

- Transaction data
- 条件:

  minsup = 30%

  minconf = 80%
- frequent *itemset*:

  {鶏肉、洋服、ミルク}　[sup = 3/7]
- Association rules

  洋服 → 鶏肉、ミルク　[sup = 3/7, conf = 3/3]

  … …

  洋服、鶏肉 → ミルク,　[sup = 3/7, conf = 3/3]

EM法(Expectation Maximization) 期待値最大化法

- k-means 法の一般化 （反復法）
  重心(centroid) にあたるものが、パラメータ
- mixture-resolving (混合推定）
- 「隠れ変数」があると仮定、「観測可能なデータ」から　分布パラメータを推定。
- 尤度を最大化するクラスタリングが良い

Initialize 適当な初期化

Expectation step　パラメータを使い、クラスターの割当を決める。

Maximization step　現在のクラスタリング尤度が最大になるように、パラメータを再計算。　→　Expectation step に

text mining
ベクトル空間分類

Contiguity Hypothesis （連続性仮説）

→ 同じクラスの文書は、連続した領域にあり、
他のクラスの領域とは重ならない。

用語集合V として、|V|次元ベクトル

- 出現するかしないかで $\{0,1\}^{|V|}$
- **tf-idf** の重みを持った $\mathbf{R}^{|V|}$

コサイン距離

- $$cosine(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\| \mathbf{d}_j \| \times \| \mathbf{q} \|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}$$

レポートの〆切について

第三回・第四回は6月22日（日）〆切ます。

〆切後、いくつかのプログラムを　講義のweb ページで公開する予定です。

第九回　課題

第三回・第四回のレポートは
6/22（日）で〆切ます。

レポート・成績について

- ほぼ毎回、プログラミング課題を出題する予定
  - 効率の良い計算機実験のためのツールを使ってみる
  - アルゴリズムの実装
  - ライブラリの利用・・・　など

- ３回以上、レポートのファイルとプログラムのソースを添付し、メールで提出のこと
  - E-mail: algorithm2014@edu.jar.jp
  - サブジェクト 「アルゴリズムとプログラム実践講座・レポート」
  - 学生証番号と名前は、 メールの本文にも書いてください。
  - プログラムやレポートは（見本として）公開することがあります。適宜、作者名や　コピーライトをいれておいてください。公開不可の場合は、プログラムの冒頭にその旨、コメントをいれておいてください。
  - 質問・作問提案も歓迎 （作問については採用の場合は別途加点）
  - サンプルプログラムは「初心者向け」です。 **上級者は無視してください。**

推奨環境など

- Linux, Mac,（Windows+Cygwin）
- 仮想マシン環境（VMware, VirtualBox, Parallels)
  - 余裕があれば、いろいろな組み合わせを試して比較してみると面白いと思います

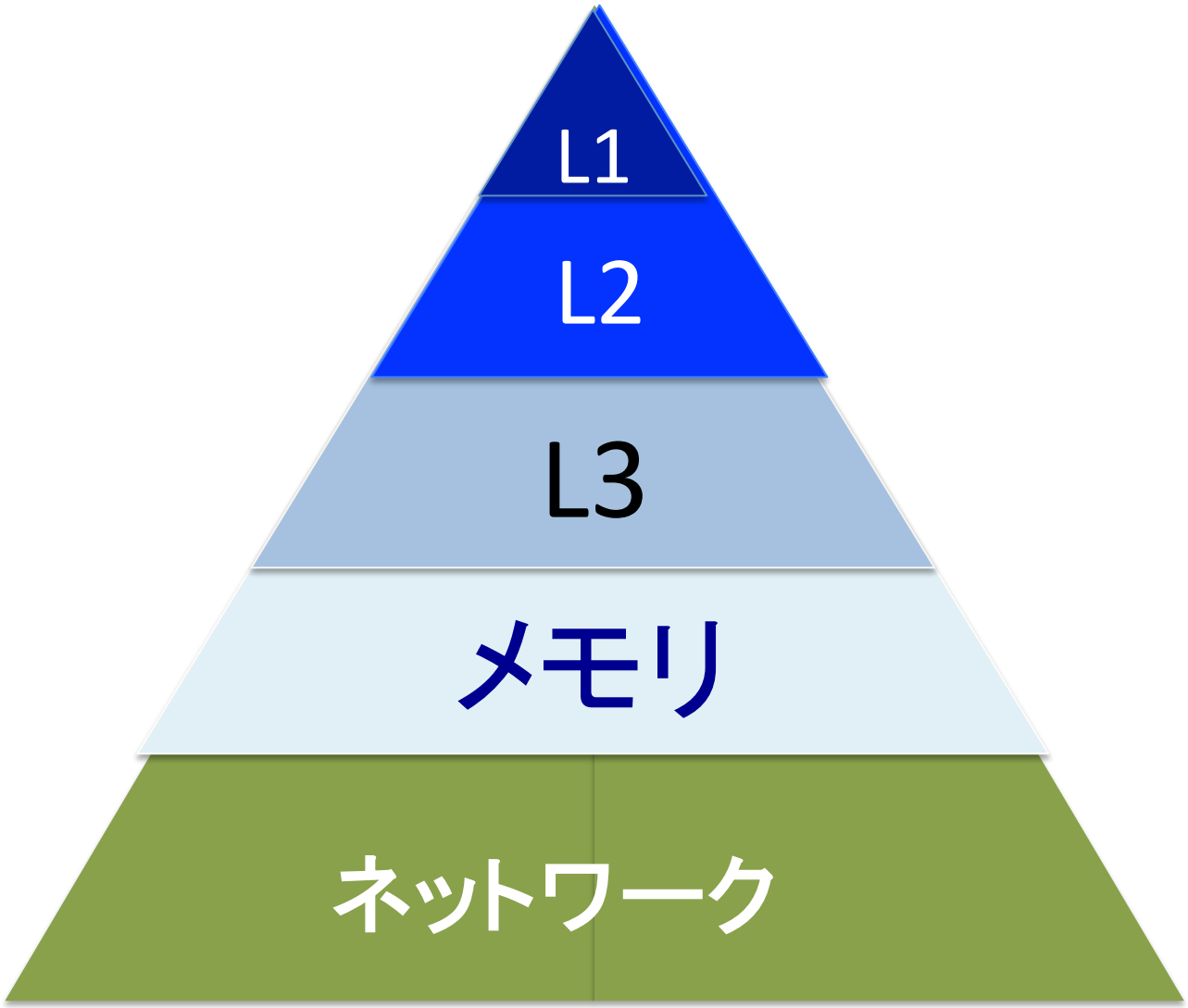
- 言語
  - 自由。ただし、一般的でない言語については、上記いずれかのOS 上にインストール可能なもの

クラスタリングの比較

たとえば「プロジェクト・グーテンベルグ」や「青空文庫」から、書籍のデータを、適当な個数収集し、特徴空間に文書を写像し、クラスタリングアルゴリズムを適用する。

このとき、特徴空間への写像方法・近似度の定義、あるいはクラスタリングアルゴリズムを変え、比較考察せよ。

コアが沢山ある場合は？



主として計算科学用
並列計算のボトルネック

- 密行列積：演算ボトルネック
- 疎行列：メモリボトルネック
- FFT： バンド幅ボトルネック

LSI コンテスト in 沖縄

LSIデザインコンテスト

参加資格： 大学･高専生による3人以下のチーム

2012 年 第１５回　課題：　FFT

http://www.lsi-contest.com/2012/shiyou_3-1.html

DFT (離散フーリエ変換）

ここでは密度汎関数ではない

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad k = 0, \ldots, N-1$$

N＝4のとき

$$X[0] = W_4^0 x[0] + W_4^0 x[1] + W_4^0 x[2] + W_4^0 x[3]$$

$$X[1] = W_4^0 x[0] + W_4^1 x[1] + W_4^2 x[2] + W_4^3 x[3]$$

$$X[2] = W_4^0 x[0] + W_4^2 x[1] + W_4^4 x[2] + W_4^6 x[3]$$

$$X[3] = W_4^0 x[0] + W_4^3 x[1] + W_4^6 x[2] + W_4^9 x[3]$$

こうなると高速

$$W_N^{k+N} = W_N^{k}$$
$$W_N^{k+\frac{N}{2}} = -W_N^{k}$$

（周期性の利用）

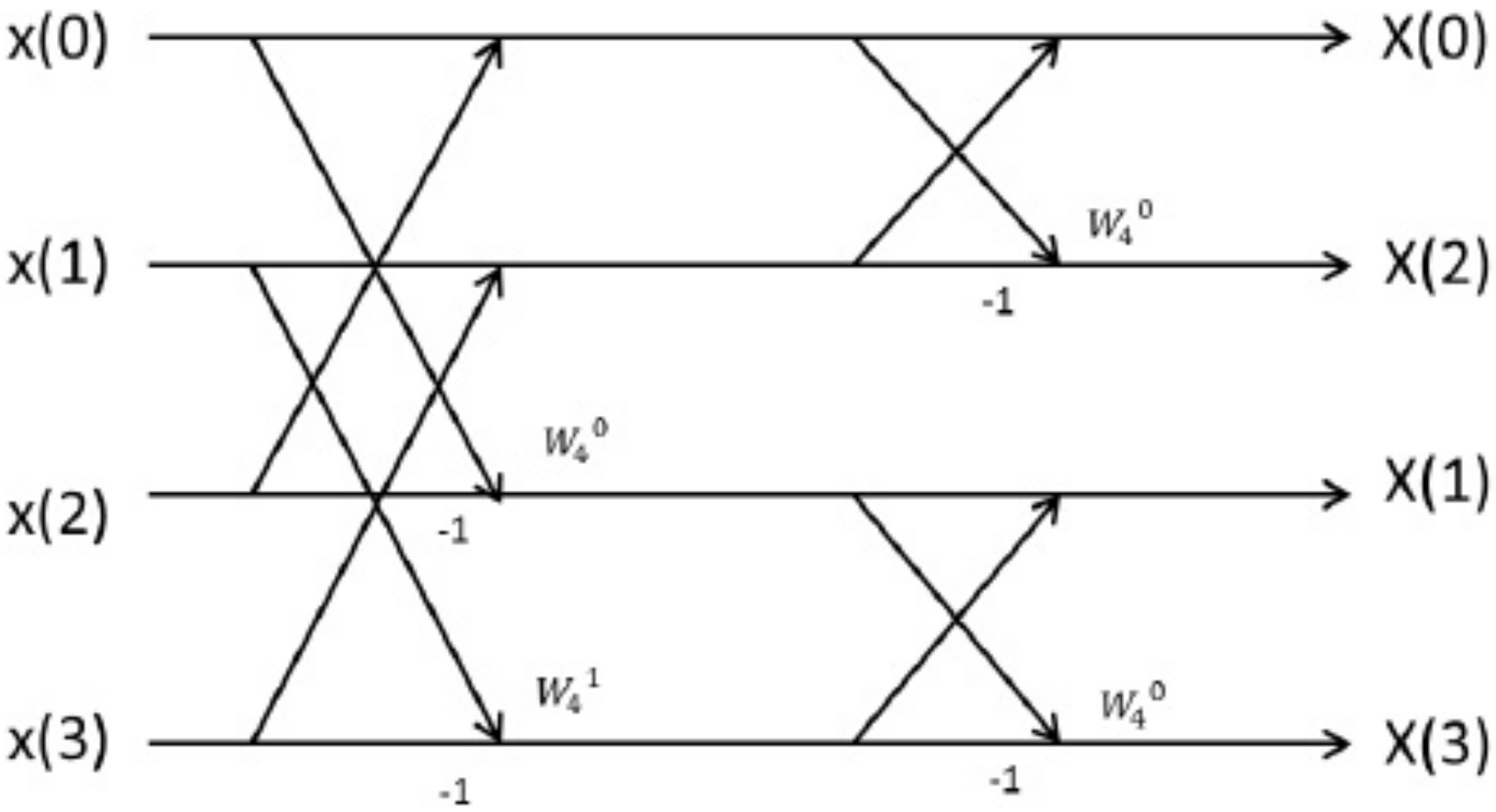
$$X[0] = W_4^0 x[0] + W_4^0 x[1] + W_4^0 x[2] + W_4^0 x[3]$$
$$X[1] = W_4^0 x[0] + W_4^1 x[1] - W_4^0 x[2] - W_4^1 x[3]$$
$$X[2] = W_4^0 x[0] - W_4^0 x[1] + W_4^0 x[2] - W_4^0 x[3]$$
$$X[3] = W_4^0 x[0] - W_4^1 x[1] - W_4^0 x[2] + W_4^1 x[3]$$

式变形

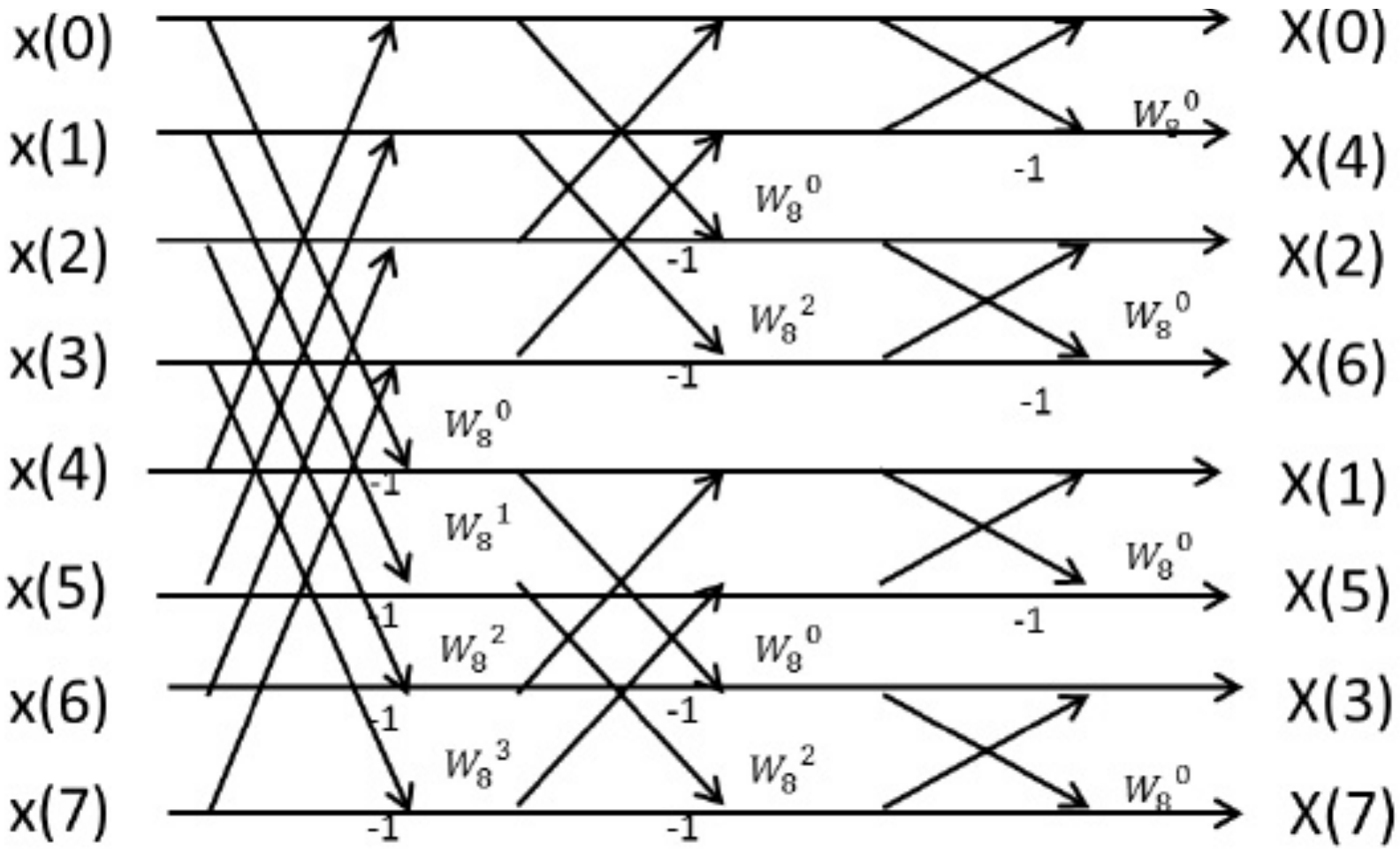
$$\begin{bmatrix} X[0] \\ X[2] \\ X[1] \\ X[3] \end{bmatrix} = \begin{bmatrix} A & A \\ B & -B \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

$$\begin{bmatrix} X[0] \\ X[2] \\ X[1] \\ X[3] \end{bmatrix} = \begin{bmatrix} A & O \\ O & B \end{bmatrix} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

ハードウェア化



ハードウェア化



Control Flow Graph

```
1  int m(int x, int y) {
2    while (x > 10) {
3      x -= 10;   // x = x - 10;
4      if (x == 10) {
5        break;
6      }
7    }
8    x = square(x);
9    if (y < 20 && x%2 == 0) {
10     y += 20;   // y = y + 20;
11   }
12   else {
13     y -= 20;   // y = y - 20;
14   }
15   return 2*x + y;
16 }
```

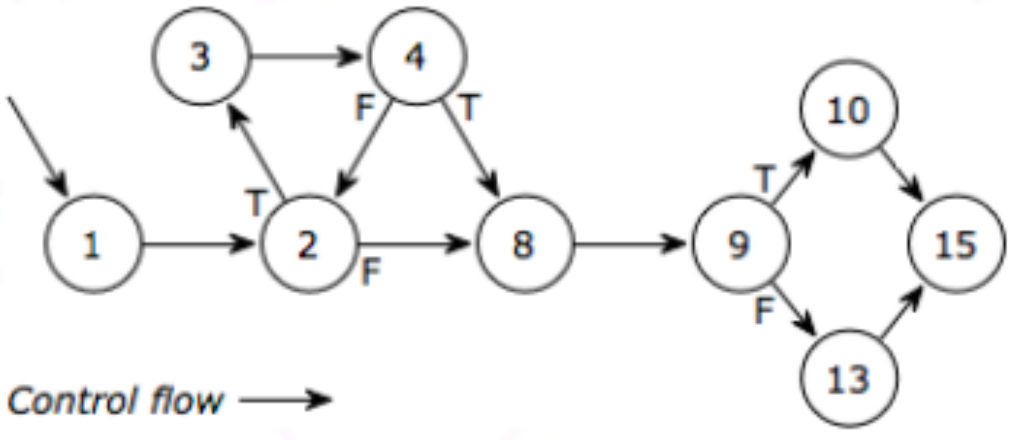


Figure 1. Control flow graph of m()

http://www.thomasalspaugh.org/pub/fnd/dataFlow.html

Data Flow Graph

```
1  int m(int x, int y) {
2    while (x > 10) {
3      x -= 10;   // x = x - 10;
4      if (x == 10) {
5        break;
6      }
7    }
8    x = square(x);
9    if (y < 20 && x%2 == 0) {
10     y += 20;   // y = y + 20;
11   }
12   else {
13     y -= 20;   // y = y - 20;
14   }
15   return 2*x + y;
16 }
```

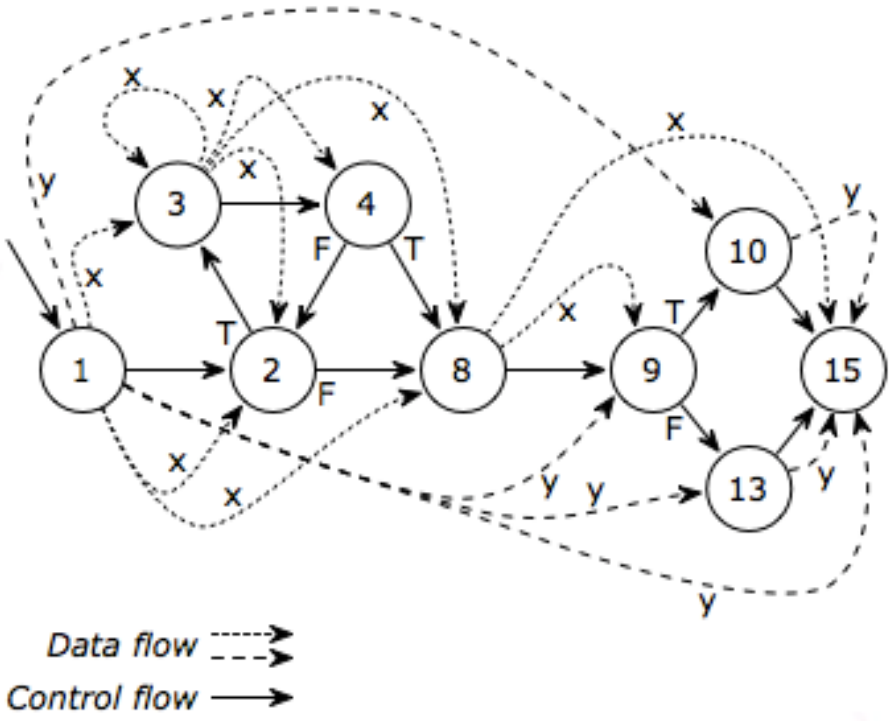
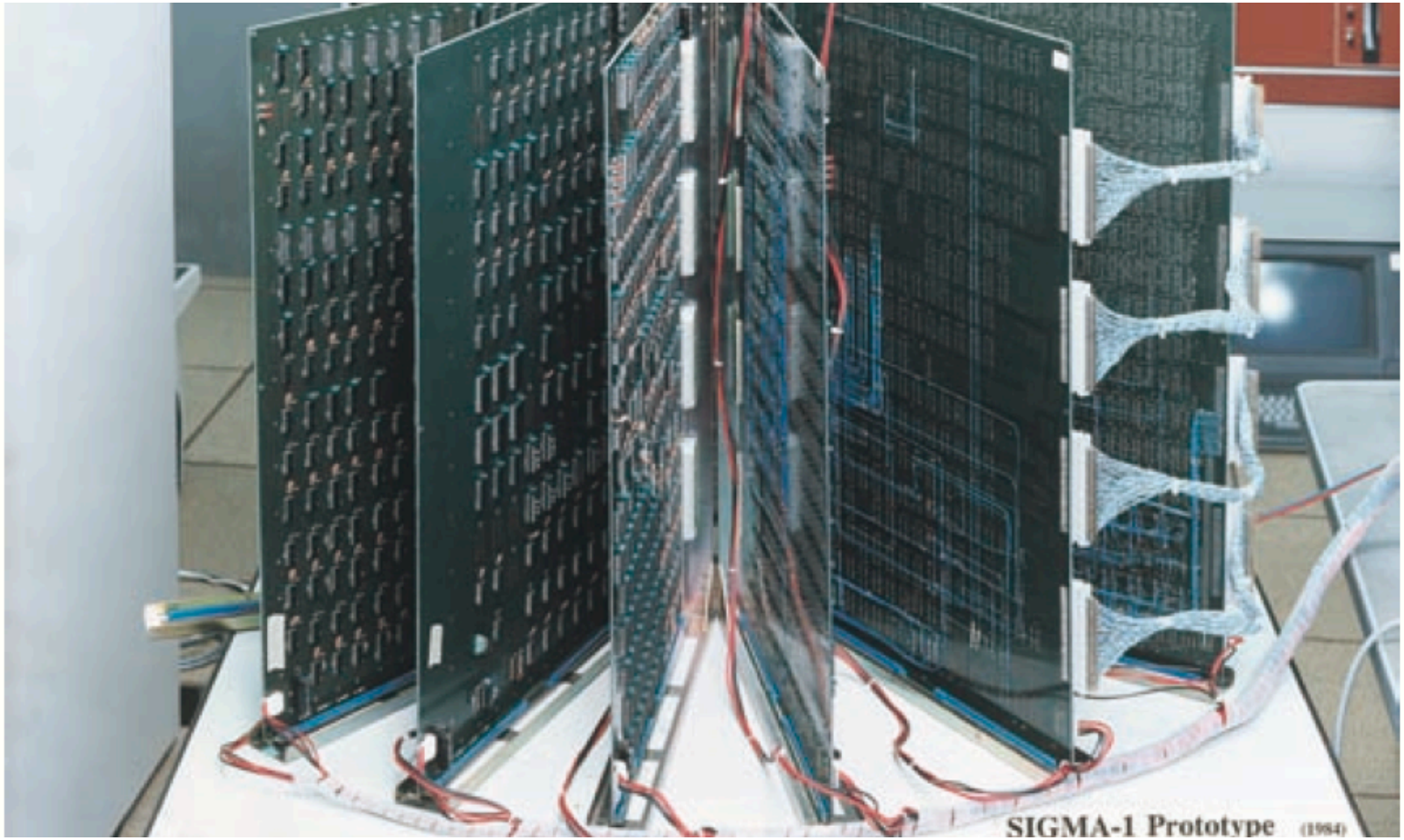


Figure 2. Control and data flow graph of m()

http://www.thomasalspaugh.org/pub/fnd/dataFlow.html

sigma-1 プロトタイプ　(1984)

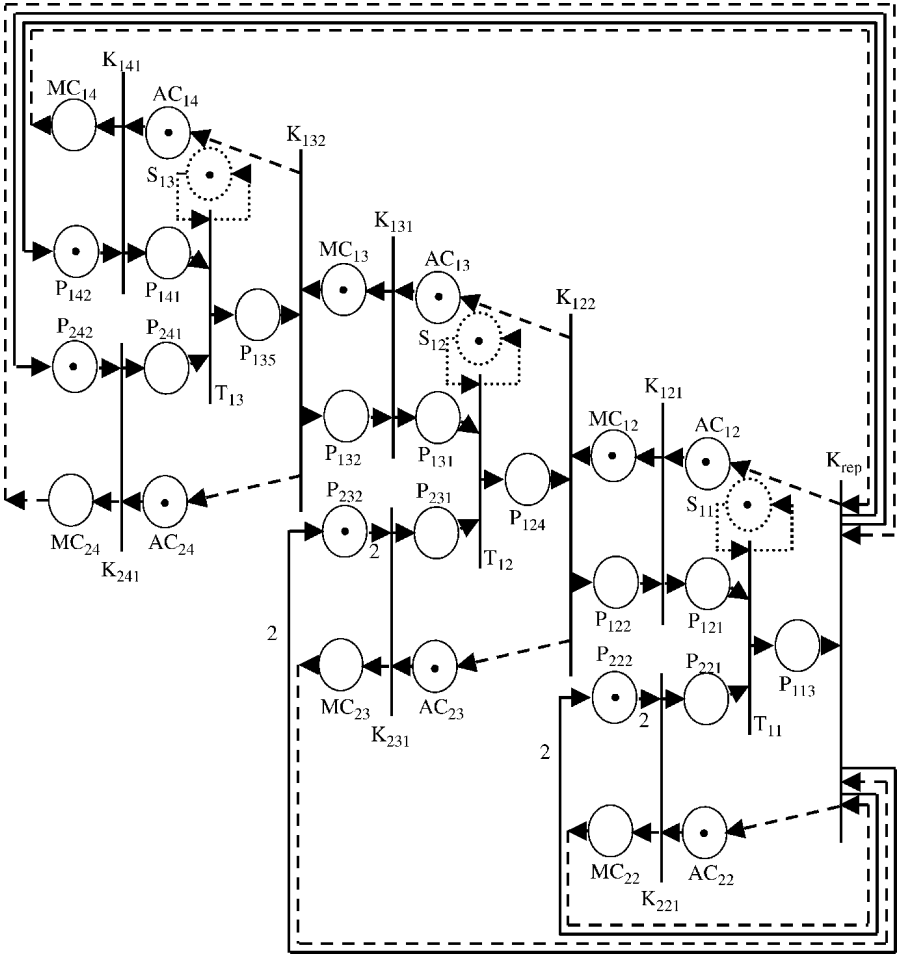


sigma-1 (1988)

（おまけのおまけ）ペトリネット
分散モデル

- データフローグラフよりも、抽象度が高い
- place と transition という 二種類の node で 二部グラフ。
- 黒丸がトークン

（おまけのおまけ）ペトリネット



http://www.emeraldinsight.com/journals.htm?articleid=1747512

推薦システムの歴史

1986　Information Lens (MIT, Malone)

　メールやネットニュースのメッセージヘッダに

　内容に関する場所・時間・トピック等、情報付与

1991　適合性フィードバック(Foltz)

1994　GroupLens (Resnick)

1990 年代後半　情報フィルタリング

2001 ITEM-BASED フィルタリング

情報推薦の方式

- Contents-based filtering

  推薦する内容に基づき、情報の取捨選択。
  いくつか候補をだし、一つを選ぶと、それに
  近いものを、推薦していく。
  →適合性フィードバック（ユーザプロファイル）
- Collaborative filtering (協調フィルタリング）

  ネットワーク上で、同じ好みを持つ人達を
  みつけ、共通して好むような情報を選択

Collaborative filtering (協調フィルタリング）

- メモリベース
  - ユーザーベース （あるユーザ a とする）
    アイテム集合が似ているユーザの集合 neighbor(a) を求める。
    neighbor(a) が好評価で、a が未評価な物を推薦する
  - アイテムベース
    あらかじめ、item 間の類似度を求めておく。
- モデルベース
  あらかじめ、ユーザーやアイテムを特徴を使ってモデル化（クラスタリング）しておく

協調フィルタリングの何が難しいか

- データを集めること(特に最初)
- データが疎 （買ってない物のほうが多い）
  → 類似なユーザを見つけにくい

対策

- 次元削減 （特異値分解）
- ハイブリッド化
- モデルの構築

情報資本主義？

- 持てる者が、ますます富んでいくシステム

  e.x. Google, Amazon

→　資本家の暴走を防げるか？

外れ値検出と不正検出

- ネットワークの不正侵入検出
- 携帯電話のなりすまし利用検出
- クレジットカードの不正利用の検出
- 医療や保険業界における不正請求検出

難しいところ　→　on-line 検出をしたい！