インターネット計測とデータ解析 第 6 回

長 健二朗

2013 年 5 月 15 日

前回のおさらい

第 5 回 多様性と複雑さ (5/8)

- ロングテール
- Web アクセスとコンテンツ分布
- べき乗則と複雑系
- 演習: べき乗則解析

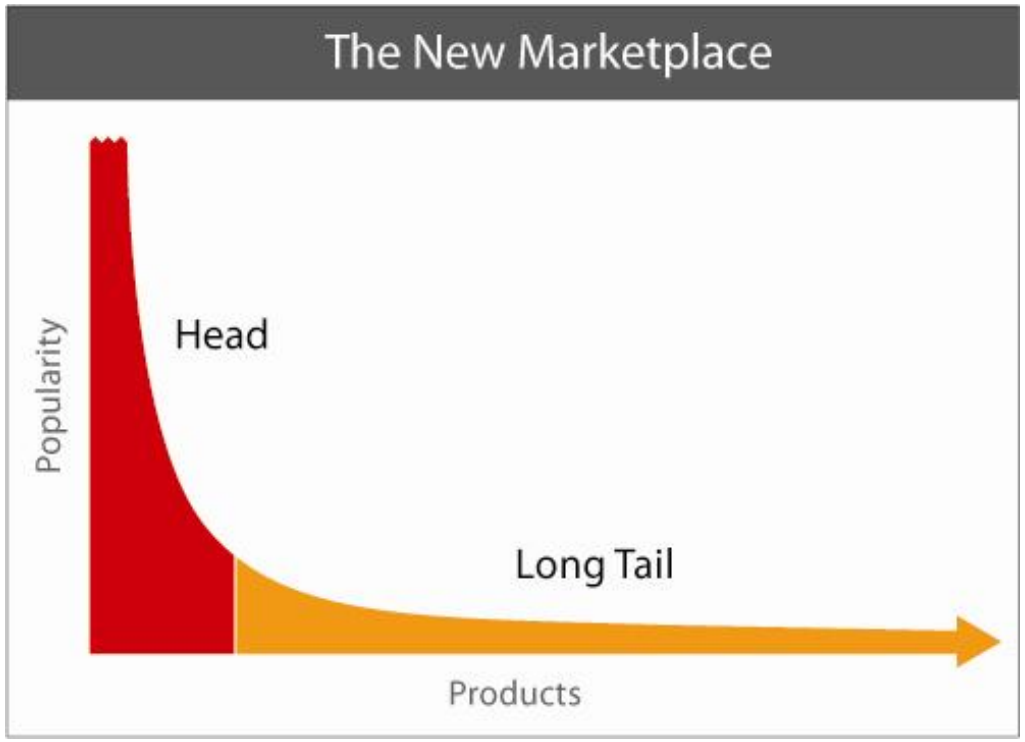
今日のテーマ

第 6 回 相関

- オンラインお勧めシステム
- 距離とエントロピー
- 相関係数
- 演習: 相関

オンラインお勧めシステム

- EC サイトにおけるロングテールのユーザの潜在ニーズ
  - ユーザの嗜好に合った商品を提示して購買に繋げる
- レコメンダーパッケージによる導入コスト低下で普及



source: http://longtail.com/

お勧めシステムの技術

- ユーザの行動を観察して有用な情報を予測して自動的に提示
- EC サイト: 購買履歴や閲覧履歴からお勧め商品を自動的に提示
- EC サイトだけでなく検索予測、かな漢字変換などへの応用も

データベースの作り方

- アイテムベース: アイテムごとに情報をまとめる
- ユーザベース: ユーザごとに情報をまとめる
- 実際にはこれらを組み合わせて使う

お勧めシステムの予測手法

- コンテンツベース:
  - ユーザが過去に利用したアイテムから類似アイテムを推薦
    - アイテムの属性分類
    - 機械学習クラスタリングによるグループ化
    - ノウハウのルール化
  - 比較的狭い範囲での推薦になりがち、意外性が低い
- 協調フィルタリング: amazon をはじめ広く利用されている
  - 購買履歴から顧客間の類似度を計算
  - 類似したユーザの実績から共通度の高いアイテムを推薦
  - 特徴: 個別のアイテムに関する情報は使わない
  - 思いがけない発見 (serendipity) の可能性
- 単純ベイズ分類器: スパム判定と同じ原理
  - アイテムやユーザに関する個別の多様な情報から確率計算、機械学習する

最近のターゲティング広告の進化

- ターゲティング広告
  - 特定ジャンルに興味を持つユーザに絞った広告
  - 広告効果や費用対効果の向上
- アドネットワーク
  - ネット広告枠と広告主を仲介するネット広告配信サービス
  - 例: 個人が運営するサイトにバナー広告を入れる
- Real Time Bidding
  - ネット広告枠をリアルタイムでオークションする仕組み
  - 広告枠を提供するオークション主催者
    - ユーザの属性情報、行動履歴情報など (cookie による追跡)
  - 広告枠を入札するオークション参加者
    - 提供された情報を元に入札価格を決める
    - リターゲティング: 過去に自社のサイトを訪問したユーザに対する広告
  - RTB 用のプラットフォーム: ミリ秒オーダーの落札を実現

協調フィルタリング (collaborative filtering)

- 複数のアルゴリズムが存在
- シンプルなユーザ間相関分析
  - ユーザ間の相関をとり類似ユーザを抽出
  - 類似ユーザの類似度を重みに各アイテムの合計点数を計算

例: ユーザの購買履歴

	item						
user	a	b	c	d	e	f	···
A	1		1		1		···
B			1	1			···
C	1	1					···
D	1		1		1		···
···							···

A と相関高いユーザから A の持っていないアイテムのスコアを計算

	similarity	item						
user	σ	a	b	c	d	e	f	···
A	1	1		1		1		···
S	0.88		0.88		-		0.88	···
C	0.81		0.81		-		-	···
K	0.75		-		-		-	···
F	0.73		0.73		0.73		0.73	···
score			2.50		0.73		1.61	···

例: Netflix Prize

- 米国のオンライン DVD レンタルサービス Netflix のアルゴリズムコンテスト
- 同社のオンラインお薦めシステムの性能を 10%向上すれば 100 万ドルの賞金
- コンテスト用データセット:
  $< user\_id, movie\_id, date\_of\_grade, grade >$
  - トレーニング用データセット: 1 億件の評価スコア
  - 評価用データセット: 280 万件の評価スコア
    - 答え合わせ用データセット: 140 万件
    - 採点用データセット: 140 万件
  - 採点スコアは結果の誤差の平均二乗偏差 (10%改善目標)
- コンテストは 2006 年に始まり、2009 年に終了
  - プライバシー問題で批判
  - 匿名化されたユーザを他の映画評価サイトのユーザとマッチング可能

距離について

いろいろな距離

- ユークリッド距離 (Euclidean distance)
- 標準化ユークリッド距離 (standardized Euclidean distance)
- ミンコフスキー距離 (Minkowski distance)
- マハラノビス距離 (Mahalanobis distance)

類似度

- バイナリベクトルの類似度
- n 次元ベクトルの類似度

距離の性質

空間上の 2 点 (x, y) 間の距離 $d(x, y)$:
非負性 (positivity)

$$d(x, y) \geq 0$$

$$d(x, y) = 0 \Leftrightarrow x = y$$

対称性 (symmetry)

$$d(x, y) = d(y, x)$$

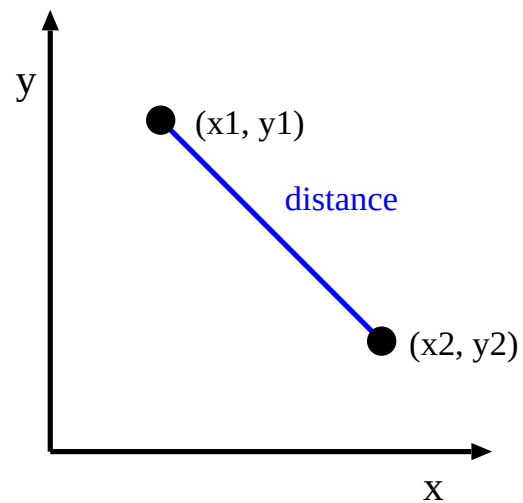
三角不等式 (triangle inequality)

$$d(x, z) \leq d(x, y) + d(y, z)$$

ユークリッド距離 (Euclidean distance)

普通に距離といえばユークリッド距離を指す
n 次元空間での 2 点 (x, y) の距離

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

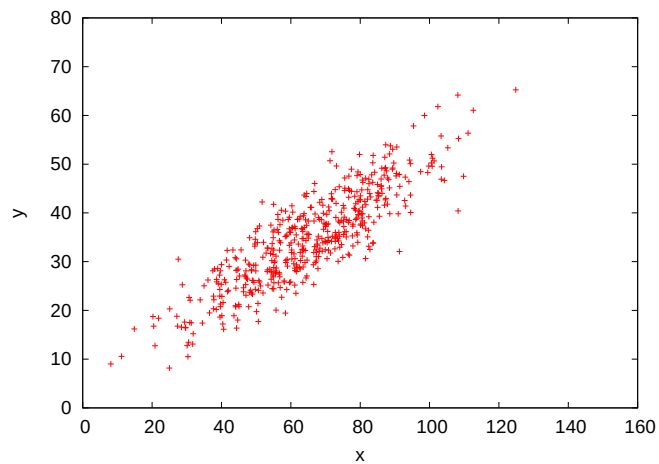


euclidean distance in 2-dimensional space

標準ユークリッド距離 (standardized Euclidean distance)

- 変数間でばらつきの大きさが異なると、距離が影響を受ける
- そこで、ユークリッド距離を各変数の分散で割って正規化

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(\frac{x_k}{s_k} - \frac{y_k}{s_k})^2} = \sqrt{\sum_{k=1}^{n}\frac{(x_k - y_k)^2}{s_k^2}}$$



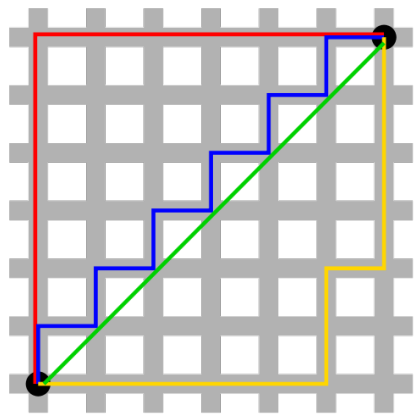
ミンコフスキー距離 (Minkowski distance)

ユークリッド距離を一般化

- パラメータ r が大きいほど、次元軸にとらわれない移動 (斜め方向のショートカット) を重視する距離

$$d(x, y) = (\sum_{k=1}^{n} |x_k - y_k|^r)^{\frac{1}{r}}$$

- $r = 1$: マンハッタン距離
  - ハミング距離: 2 つの文字列間の同じ位置の文字の不一致数
  - 例えば、111111 と 101010 のハミング距離は 3
- $r = 2$: ユークリッド距離



Manhattan distance vs. Euclidean distance

マハラノビス距離 (Mahalanobis distance)

変数間に相関がある場合に、相関を考慮した距離

$$mahalanobis(x, y) = (x - y)\Sigma^{-1}(x - y)^T$$

ここで Σ^{-1} は共分散行列の逆行列

類似度

類似度

- ふたつのデータの似ている度合の数値表現

類似度の性質

非負性 (positivity)

$$0 \leq s(x, y) \leq 1$$

$$s(x, y) = 1 \Leftrightarrow x = y$$

対称性 (symmetry)

$$s(x, y) = s(y, x)$$

三角不等式 (triangle inequality) は一般に類似度には当てはまらない

バイナリベクトルの類似度

Jaccard 係数

- 1 の出現が少ないバイナリベクトル同士の類似度に使われる
- 文書中に出現する単語から文書の類似度を示す場合など
- 多くの単語は両方ともに出現しない $\Rightarrow$ これらは考慮しない
- 2 つのベクトルの各要素の対応関係を表のように集計

		vector y	
		1	0
vector x	1	n_{11}	n_{10}
	0	n_{01}	n_{00}

Jaccard 係数は以下で表される

$$J = \frac{n_{11}}{n_{11} + n_{10} + n_{01}}$$

n 次元ベクトルの類似度

一般のベクトルの類似度

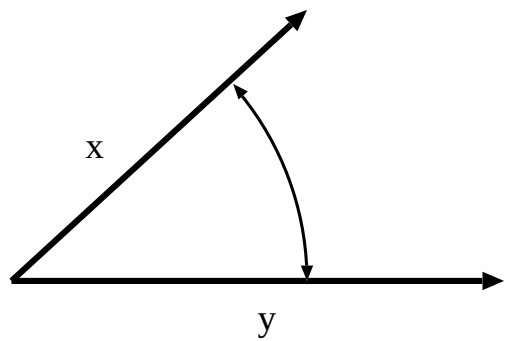
- 文書の類似度で出現頻度も考慮する場合など

コサイン類似度

- ベクトルの x, y の cosine を取る、向きが一致:1、直交:0、向きが逆:-1
- ベクトルの長さで正規化 $\Rightarrow$ 大きさは考慮しない

$$cos(x, y) = \frac{x \cdot y}{\|x\|\|y\|}$$

$x \cdot y = \sum_{k=1}^{n} x_k y_k$: ベクトルの積
$\|x\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{x \cdot x}$: ベクトルの長さ



コサイン類似度の例題

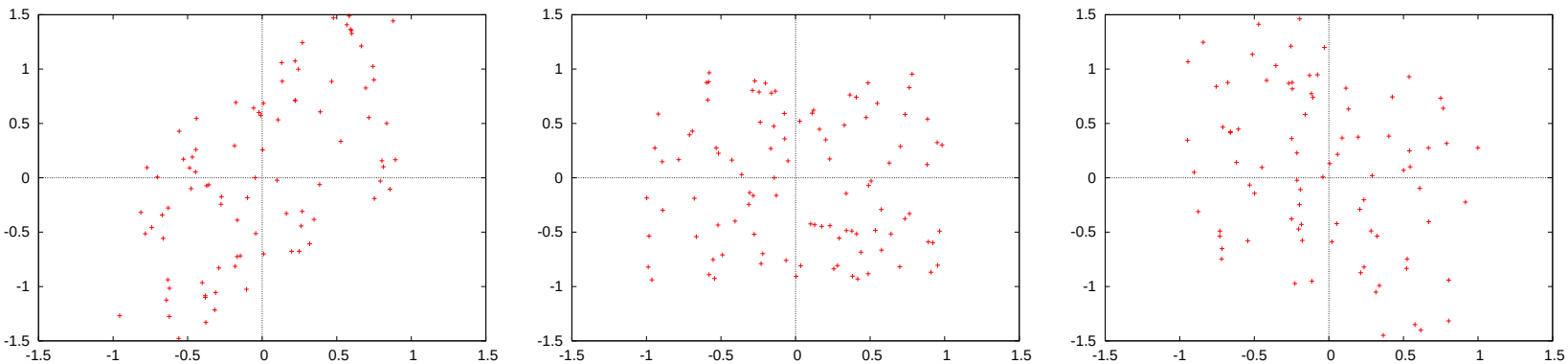
$$x = 3 \quad 2 \quad 0 \quad 5 \quad 0 \quad 0 \quad 0 \quad 2 \quad 0 \quad 0$$
$$y = 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 2$$

$$x \cdot y = 3 * 1 + 2 * 1 = 5$$
$$\|x\| = \sqrt{3 * 3 + 2 * 2 + 5 * 5 + 2 * 2} = \sqrt{42} = 6.481$$
$$\|y\| = \sqrt{1 * 1 + 1 * 1 + 2 * 2} = \sqrt{6} = 2.449$$

$$cos(x, y) = \frac{5}{6.481 * 2.449} = 0.315$$

散布図と相関係数

- 散布図は 2 つの変数の関係を見るのに有効
  - X 軸: 変数 X
  - Y 軸: それに対応する変数 Y の値
- 散布図で分かる事
  - X と Y に関連があるか
    - 無相関、正の相関、負の相関
  - 外れ値の存在があるか
- 相関係数: 相関の方向 (正負) と強さを表す量



例: (左) 正の相関 0.7 (中) 無相関 0.0 (右) 負の相関 -0.5

相関 (correlation)

- 共分散 (covariance):

$$\sigma_{xy}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

- 相関係数 (correlation coefficient):

$$\rho_{xy} = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- 相関係数は共分散を正規化したもの。 -1 から 1 の値を取る。
- 相関係数は外れ値の影響を大きく受ける。 散布図と併用し、外れ値を確認する必要。
- 相関関係と因果関係
  - 相関関係が因果関係を示すとは限らない。
    - 未知の第 3 の共通の要因が存在する場合
    - 単なる偶然

相関係数の計算 (1)

偏差平方和 (sum of squares)

$$
\begin{aligned}
\sum_{i=1}^{n}(x_i - \bar{x})^2 &= \sum_{i=1}^{n}(x_i^2 - 2x_i\bar{x} + \bar{x}^2) \\
&= \sum_{i=1}^{n} x_i^2 - 2\bar{x}\sum_{i=1}^{n} x_i + n\,\bar{x}^2 \\
&= \sum_{i=1}^{n} x_i^2 - 2\bar{x}\cdot n\,\bar{x} + n\,\bar{x}^2 \\
&= \sum_{i=1}^{n} x_i^2 - n\,\bar{x}^2 = \sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}
\end{aligned}
$$

偏差積和 (sum of products)

$$
\begin{aligned}
\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) &= \sum_{i=1}^{n}(x_i y_i - x_i\bar{y} - \bar{x}y_i + \bar{x}\bar{y}) \\
&= \sum_{i=1}^{n} x_i y_i - \bar{x}\sum_{i=1}^{n} y_i - \bar{y}\sum_{i=1}^{n} x_i + n\,\bar{x}\bar{y} \\
&= \sum_{i=1}^{n} x_i y_i - \bar{x}\cdot n\,\bar{y} - \bar{y}\cdot n\,\bar{x} + n\,\bar{x}\bar{y} \\
&= \sum_{i=1}^{n} x_i y_i - n\,\bar{x}\bar{y} = \sum_{i=1}^{n} x_i y_i - \frac{(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n}
\end{aligned}
$$

相関係数の計算 (2)

相関係数 (correlation coefficient)

$$
\begin{aligned}
\rho_{xy} &= \frac{\sigma_{xy}^2}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \\
&= \frac{\sum_{i=1}^n x_i y_i - n\,\bar{x}\bar{y}}{\sqrt{(\sum_{i=1}^n x_i^2 - n\,\bar{x}^2)(\sum_{i=1}^n y_i^2 - n\,\bar{y}^2)}} \\
&= \frac{\sum_{i=1}^n x_i y_i - \frac{(\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n}}{\sqrt{(\sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n})(\sum_{i=1}^n y_i^2 - \frac{(\sum_{i=1}^n y_i)^2}{n})}}
\end{aligned}
$$

他の相関係数

- ピアソンの積率相関係数 (Pearson's product-moment correlation coefficient)
  - 単に相関係数といえばこれを指す (この授業でも同様)
- 順位相関係数 (rank correlation coefficient):
  - 同じ項目を持つデータセットに対する順位付けの違いとその関係
  - スピアマンの順位相関係数
  - ケンドールの順位相関係数
- その他の相関係数

課題 1: ホノルルマラソン完走時間のプロット

- ねらい: 実データから分布を調べる
- データ: 2012 年のホノルルマラソンの記録
  - http://results.sportstats.ca/res2012/honolulumarathon_m.htm
  - 完走者 24,070 人
- 提出項目
  1. 全完走者、男性完走者、女性完走者それぞれの、完走時間の平均、標準偏差、中間値
  2. それぞれの完走時間のヒストグラム
     - 3 つのヒストグラムを別々の図に書く
     - ビン幅は 10 分にする
     - 3 つのプロットは比較できるように目盛を合わせること
  3. それぞれの CDF プロット
     - ひとつの図に 3 つのプロットを書く
  4. オプション
     - 年代別や国別の CDF プロットなど自由
  5. 考察
     - データから読みとれることを記述
- 提出形式: レポートをひとつの PDF ファイルにして SFC-SFS から提出
- 提出〆切: 2013 年 5 月 16 日

ホノルルマラソンデータ

データフォーマット

Place	Chip Time	Pace /mi	#	Name	City	ST	CNT	Gender Plce/Tot	Category Plc/Tot	Category	@10km Split1	@21.1 Split2	@3
1	02:12:31	5:04	6	Kipsang, Wilson	Iten		KEN	1/12690	1/16	MElite	31:40	1:07:07	1:3
2	02:13:08	5:05	7	Geneti, Markos	Addis Ababa		ETH	2/12690	2/16	MElite	31:39	1:07:02	1:3
3	02:14:15	5:08	11	Kimutai, Kiplimo	Eldoret		KEN	3/12690	3/16	MElite	31:40	1:07:02	1:3
4	02:14:55	5:09	2	Ivuti, Patrick	Kangundo		KEN	4/12690	4/16	MElite	31:40	1:07:02	1:3
5	02:15:17	5:10	12	Arile, Julius	Kepenguria		KEN	5/12690	5/16	MElite	31:39	1:07:02	1:3
6	02:15:53	5:11	9	Bouramdane, Abderr	Champs De Cou		MAR	6/12690	6/16	MElite	31:40	1:07:01	1:3
7	02:18:27	5:17	8	Manza, Nicholas	Ngong Hills		KEN	7/12690	7/16	MElite	31:39	1:07:01	1:3
8	02:19:46	5:20	1	Chelimo, Nicholas	Ngong Hills		KEN	8/12690	8/16	MElite	31:40	1:07:02	1:3
9	02:25:23	5:33	20850	Harada, Taku	Nagoya-Shi	AI	JPN	9/12690	1/1238	M25-29	31:54	1:09:52	1:4
10	02:27:12	5:37	25474	Hagawa, Eiichi	Matsumoto	NA	JPN	10/12690	1/1501	M30-34	32:46	1:12:21	1:4

...

- Chip Time: 完走時間
- Category: MElite, WElite, M15-19, M20-24, ..., W15-29, W20-24, ...
  - "No Age"となっている人がいるので注意
- Country: 3-letter country code: e.g., JPN, USA
  - "UK"が交じっているので注意
- 完走者を抽出したら、総数が合っているかチェックすること

前回の演習: CCDF のプロット

twitter ユーザの following/follower の数の分布を CCDF にプロット

- データ: Kwak らによる 2009 年 7 月の twitter data
  - http://an.kaist.ac.kr/traces/WWW2010.html
  - twitter API を使って全ユーザ (4000 万以上) を crawl
    - API 変更、ユーザ増加によって、現在では出来ない
  - ここから、各ユーザの following/follower 数を抽出し、1 万人をサンプル

```
% head -10 twitter_degrees-10000.txt
# id followings followers
2058    1       1
11097   5       4
12329   375     1132
596043  63      97
638173  407     428
643423  2       18
659943  1958    1294
698823  503     344
730013  23      13
% ./make_ccdf.rb twitter_degrees-10000.txt > followings-ccdf.txt
% ./make_ccdf.rb -c 3 twitter_degrees-10000.txt > followers-ccdf.txt
```

前回の演習: CCDF のプロット (続き)

- CCDF 用データ形式: follow 数 人数 累積人数 CDF CCDF

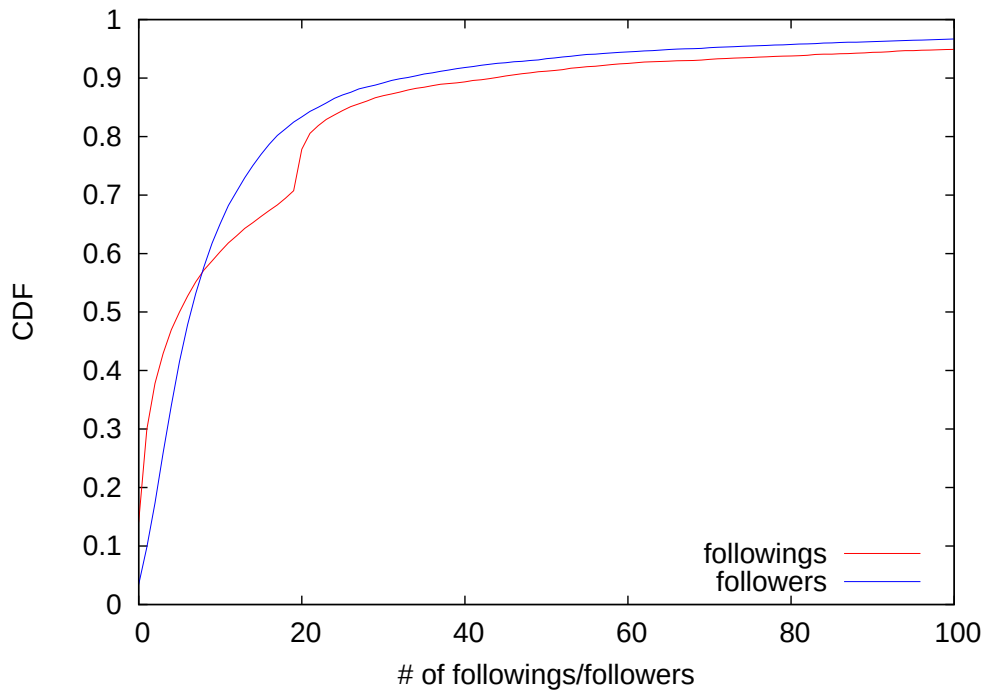
```
% cat followings-ccdf.txt
0       1407    1407    0.1407  1.0000
1       1586    2993    0.2993  0.8593
2       787     3780    0.3780  0.7007
3       513     4293    0.4293  0.6220
4       403     4696    0.4696  0.5707
5       302     4998    0.4998  0.5304
6       275     5273    0.5273  0.5002
7       242     5515    0.5515  0.4727
8       202     5717    0.5717  0.4485
9       158     5875    0.5875  0.4283
...
2134    1       9991    0.9991  0.0010
2167    1       9992    0.9992  0.0009
2510    1       9993    0.9993  0.0008
2512    1       9994    0.9994  0.0007
2657    1       9995    0.9995  0.0006
3410    1       9996    0.9996  0.0005
5042    1       9997    0.9997  0.0004
6605    1       9998    0.9998  0.0003
11350   1       9999    0.9999  0.0002
12335   1       10000   1.0000  0.0001
```

CCDF を集計するスクリプト: make_ccdf.rb

```
#!/usr/bin/env ruby
# create ccdf: usage: make_ccdf.rb [-c n] file
#
require 'optparse'

re = /^(\d+)\s+(\d+)\s+(\d+)/
col_no = 2 # column no (1 origin)

OptionParser.new {|opt|
  opt.on('-c VAL', Integer) {|v| col_no = v}
  opt.parse!(ARGV)
}

counts = Hash.new(0)
n = 0
ARGF.each_line do |line|
  if re.match(line)
    val = $~[col_no].to_i
    counts[val] += 1
    n += 1
  end
end

cum = 0
cdf = 0.0
counts.sort.each do |key, value|
  comp = 1.0 - cdf
  cum += value
  cdf = Float(cum) / n
  printf "%d\t%d\t%d\t%.4f\t%.4f\n", key, value, cum, cdf, comp
end
```

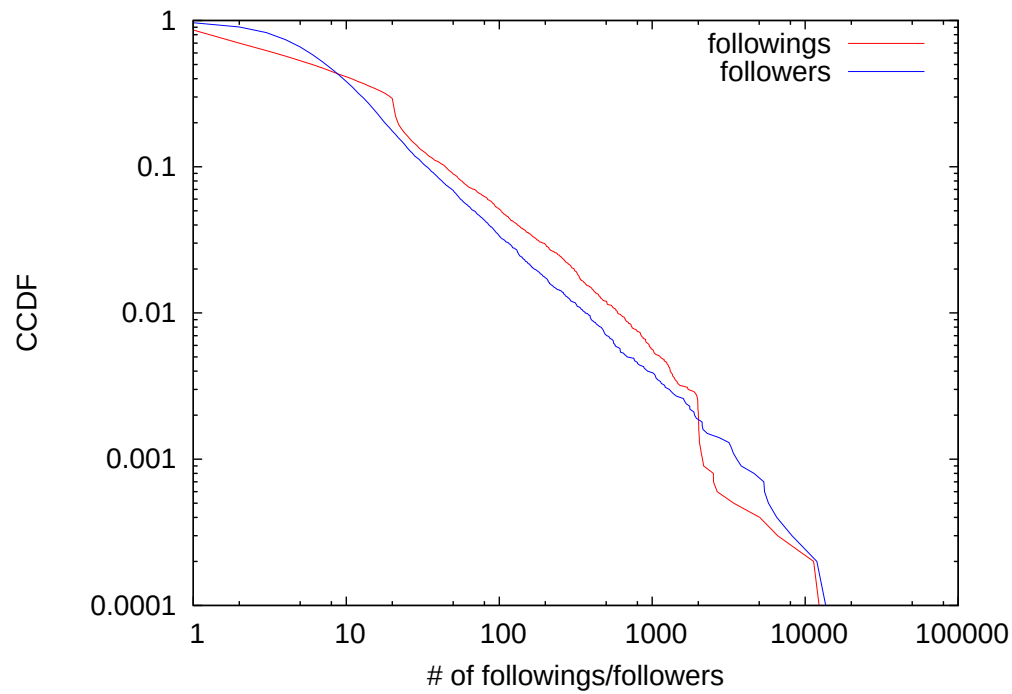
CDF の表示: gnuplot スクリプト

```
set xlabel "# of followings/followers"
set ylabel "CDF"
set key bottom
set xrange [0:100]

plot "followings-ccdf.txt" using 1:4 title 'followings' with lines, \
"followers-ccdf.txt" using 1:4 title 'followers' with lines lt 3
```



CCDF の表示: gnuplot スクリプト

```
set logscale
set xlabel "# of followings/followers"
set ylabel "CCDF"

plot "followings-ccdf.txt" using 1:5 title 'followings' with lines, \
"followers-ccdf.txt" using 1:5 title 'followers' with lines lt 3
```



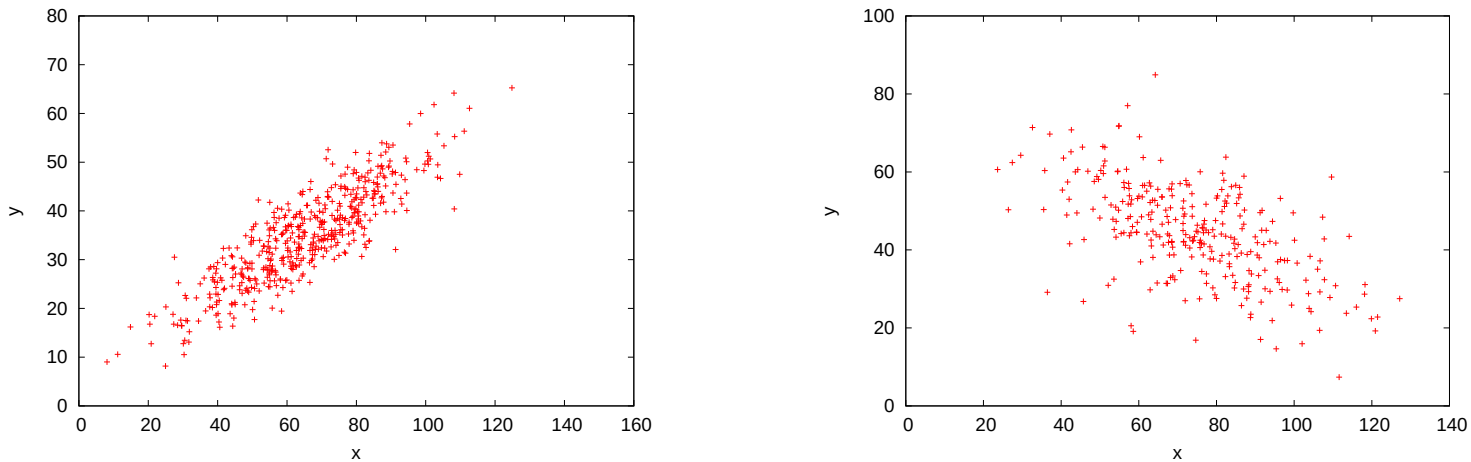
演習の考察

CDF/CCDF から分かる事

- フォロー数、フォロワー数ともにべき分布に従う
  - 極端に少ない/多い部分はその限りでない
- フォロー数は、20 と 2000 あたりにギャップ
  - 20: 初期設定でフォローする 20 人を薦められる
  - 2000: 以前は最大 2000 人までしかフォローできなかった

今回の演習: 相関係数の計算

- データの相関係数を計算する
  - correlation-data-1.txt, correlation-data-2.txt



data-1:r=0.87 (left), data-2:r=-0.60 (right)

演習: 相関係数の計算スクリプト

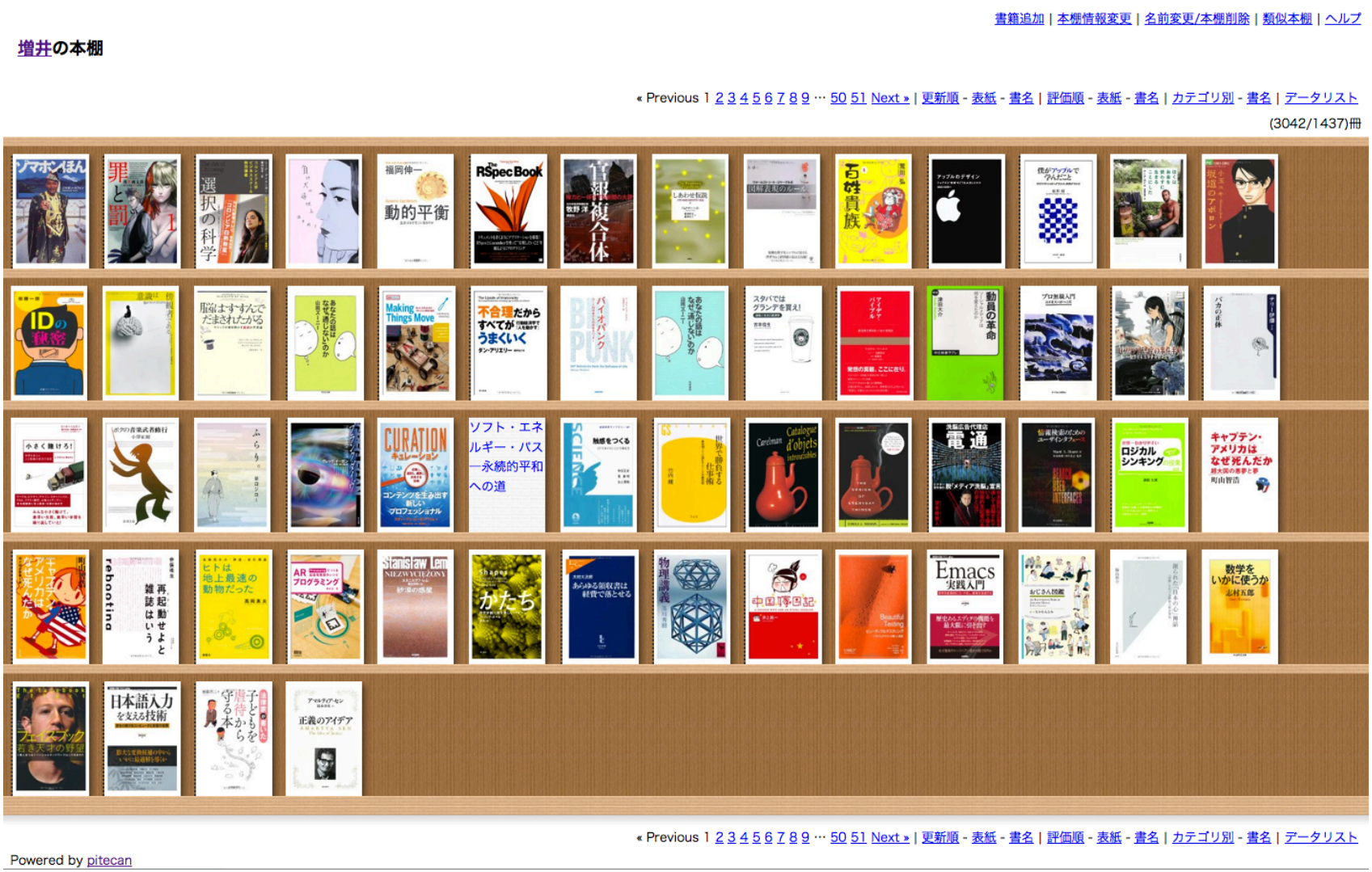
```
#!/usr/bin/env ruby

# regular expression for matching 2 floating numbers
re = /([-+]?\d+(?:\.\d+)?)\s+([-+]?\d+(?:\.\d+)?)/

sum_x = 0.0 # sum of x
sum_y = 0.0 # sum of y
sum_xx = 0.0 # sum of x^2
sum_yy = 0.0 # sum of y^2
sum_xy = 0.0 # sum of xy
n = 0 # the number of data

ARGF.each_line do |line|
    if re.match(line)
      x = $1.to_f
      y = $2.to_f
      sum_x += x
      sum_y += y
      sum_xx += x**2
      sum_yy += y**2
      sum_xy += x * y
      n += 1
    end
end

r = (sum_xy - sum_x * sum_y / n) /
  Math.sqrt((sum_xx - sum_x**2 / n) * (sum_yy - sum_y**2 / n))

printf "n:%d r:%.3f\n", n, r
```

トピック: 本棚.org

増井俊之先生の本棚.org

- 自分の持つ本のリストを共有するサイト

  http://www.hondana.org/



トピック: 本棚演算

本棚演算: 本棚.org のデータに演算を適用し、協調フィルタリング的に本の情報を集める

- 本棚演算のページ
  `http://www.pitecan.com/Enzan/`
  - 2007 年のデータと ruby で書かれたソースコードも
  - 日本語は EUC コード
- 本棚演算を解説した UnixMagazine の原稿
  `http://www.pitecan.com/UnixMagazine/PDF/if0512.pdf`
- MySQL のデータも公開されている
  - 最新データがダウンロード可能 (約 80MB)
  - 文字コードは UTF-8
- 現在は Rails2 で動作、github でソースや API を公開

まとめ

第 6 回 相関

- オンラインお勧めシステム
- 距離とエントロピー
- 相関係数
- 演習: 相関

次回予定

5/22 休講

第 7 回 多変量解析 (5/29)

- データセンシング
- 線形回帰
- 主成分分析
- 演習: 線形回帰

補講予定

- 6/19 (水) 6 限 (18:10-19:40) $\lambda 13$
- 7/17 (水) 4 限 (14:45-16:15) $\epsilon 12$

参考文献

[1] Ruby official site. `http://www.ruby-lang.org/`

[2] gnuplot official site. `http://gnuplot.info/`

[3] Mark Crovella and Balachander Krishnamurthy. *Internet measurement: infrastructure, traffic, and applications*. Wiley, 2006.

[4] Pang-Ning Tan, Michael Steinbach and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

[5] Raj Jain. *The art of computer systems performance analysis*. Wiley, 1991.

[6] Toby Segaran. (當山仁健 鴨澤眞夫 訳). 集合知プログラミング. オライリージャパン. 2008.

[7] Chris Sanders. (高橋基信 宮本久仁男 監訳 岡真由美 訳). 実践パケット解析 第 *2* 版 — *Wireshark* を使ったトラブルシューティング. オライリージャパン. 2012.

[8] あきみち、空閑洋平. インターネットのカタチ. オーム社. 2011.

[9] 井上洋, 野澤昌弘. 例題で学ぶ統計的方法. 創成社, 2010.

[10] 平岡和幸, 掘玄. プログラミングのための確率統計. オーム社, 2009.