

データサイエンス

第7回

～時系列予測～

情報理工学系研究科
創造情報学専攻
中山 英樹

本日の内容

- 先週の補足（一般化線形モデル）
- レポート課題説明
- 時系列分析
 - 時系列予測

一般化線形モデル (generalized linear model)

- 目的変数が正規分布以外の指数分布族に従うモデル
 - 目的変数にある関数で非線形変換を加えると、説明変数の線形結合で表現できる

$$g(\underline{\mu_i}) = \mathbf{a}^T \mathbf{x}_i + b$$

$\mu_i = E[y_i]$: 目的変数の分布の平均 (期待値) $g()$ をリンク関数と呼ぶ

- Rのstatsパッケージの関数glm
> glm(formula, family, data)

回帰の種類	分布族 (family)	リンク関数	
線形回帰	正規分布 (gaussian)	μ	link="identity"
ポアソン回帰	ポアソン分布 (poisson)	$\log(\mu)$	link="log"
ロジスティック回帰	二項分布 (binomial)	$\log(\mu/(1-\mu))$	link="logit"

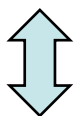
線形回帰（基本）

- 目的変数（の残差）は等分散の正規分布に従う

$$y_i = \mathbf{a}^T \mathbf{x}_i + b + \varepsilon_i \quad (\varepsilon_i \sim N(0, \sigma^2))$$

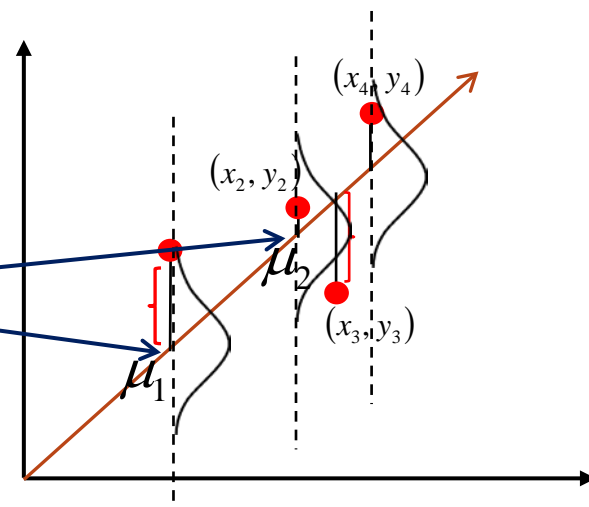
平均0、分散 σ^2 のガウシアン

$$\mu_i = E[y_i] = \mathbf{a}^T \mathbf{x}_i + b$$



$$g_{LR}(\mu_i)$$

リンク関数は恒等変換



ポアソン回帰

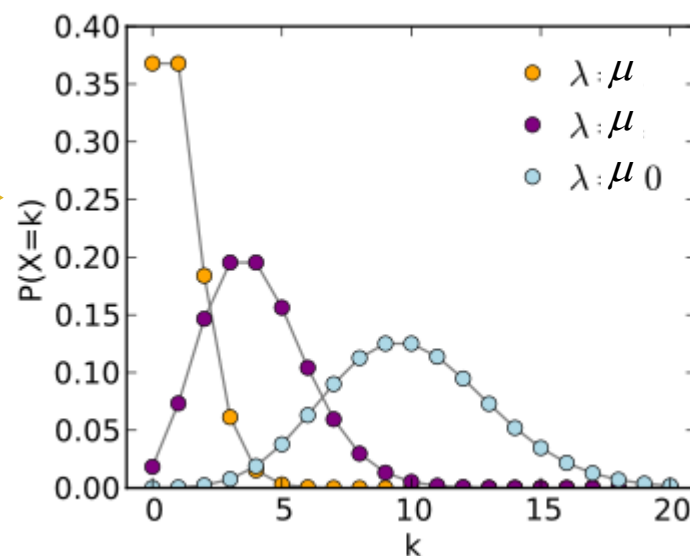
- 目的変数がポアソン分布に従う場合
 - ある一定の時間内に発生する離散的な事象を数える
特定の確率変数を持つ離散確率分布
- 計数（カウント）データに使う

$$P(y_i = k) = \frac{\mu_i^k e^{-\mu_i}}{k!}$$

$$E[y_i] = \mu_i = \exp(\mathbf{a}^T \mathbf{x}_i + b) \quad (\text{とおく})$$

$$g_{Po}(\mu_i) = \log(\mu_i) = \mathbf{a}^T \mathbf{x}_i + b$$

対数線形モデル



例) Large-scale behavioral targeting [Chen et al., KDD'09]

- 広告CTRをユーザの行動データから予測

← $\mu_i = \mathbf{a}^T \mathbf{x}_i$ で解いている

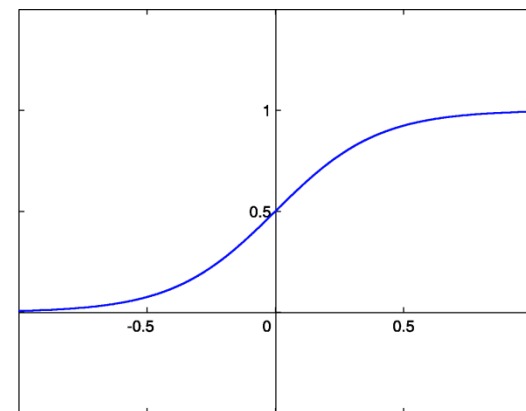
ロジスティック回帰

- 目的変数がベルヌーイ分布に従う場合
- 二値データ（質的データ）の回帰
 - 実用上はクラス識別の手法として解釈される場合が多い

y_i は0か1の二値（ベルヌーイ分布）

$$P(y_i = 1 | \mathbf{x}_i) = p_i, \quad P(y_i = 0 | \mathbf{x}_i) = 1 - p_i$$

$$p_i = \frac{\exp(\mathbf{a}^T \mathbf{x}_i + b)}{1 + \exp(\mathbf{a}^T \mathbf{x}_i + b)} \quad \text{とおく（ロジスティック関数）}$$



(y の分布ではないので注意)

$$E[y_i] = \mu_i = 1 \times p_i + 0 \times (1 - p_i) = \frac{\exp(\mathbf{a}^T \mathbf{x}_i + b)}{1 + \exp(\mathbf{a}^T \mathbf{x}_i + b)}$$

$$g(\mu_i) = \log\left(\frac{\mu_i}{1 - \mu_i}\right) = \mathbf{a}^T \mathbf{x}_i + b \quad \text{リンク関数はロジット関数}$$

解き方

- 最尤推定

訓練データ集合 $\{\mathbf{x}_i, t_i\}, t_i \in \{0, 1\}$

尤度関数は $L = \prod_{i=1}^N p_i^{t_i} \{1 - p_i\}^{1-t_i} \quad p_i = P(y_i = 1 | \mathbf{x}_i) = \frac{\exp(\mathbf{a}^T \mathbf{x}_i + b)}{1 + \exp(\mathbf{a}^T \mathbf{x}_i + b)}$

負の対数尤度は $E(\mathbf{a}) = -\ln L = \sum_{i=1}^N \{t_i \ln p_i + (1 - t_i) \ln(1 - p_i)\}$



$$\frac{\partial E(\mathbf{a})}{\partial \mathbf{a}} = \sum_{i=1}^N \underline{(p_i - t_i) \mathbf{x}_i}$$

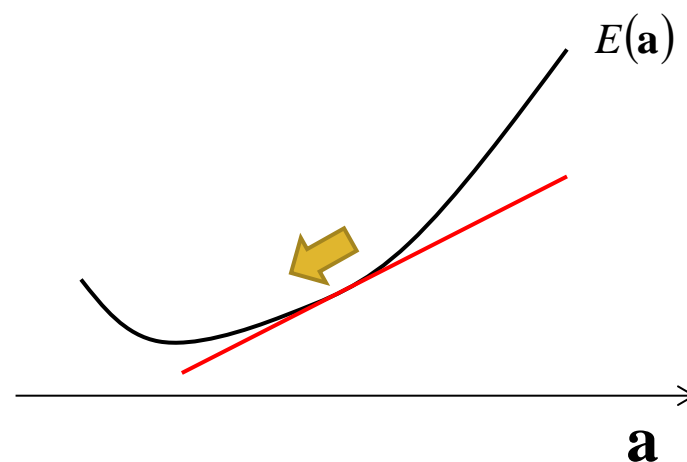
エラーに説明変数をかけたもの

最急降下法

- 収束するまで以下のように更新
 - α は微小な正のパラメータ

$$\mathbf{a} \leftarrow \mathbf{a} - \alpha \sum_{i=1}^N (p_i - t_i) \mathbf{x}_i$$

$\frac{E(\mathbf{a})}{\partial \mathbf{a}}$



第一回レポート課題（回帰）

- Wine quality prediction dataset

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

- 物理化学的特徴からワインの質（10段階）を予測（回帰）

- 説明変数

- 0 - red or white
- 1 - fixed acidity
- 2 - volatile acidity
- 3 - citric acid
- 4 - residual sugar
- 5 - chlorides
- 6 - free sulfur dioxide
- 7 - total sulfur dioxide
- 8 - density
- 9 - pH
- 10 - sulphates
- 11 - alcohol

- 目的変数

- 12 - quality (score between 0 and 10)



コンペティション形式

- 課題用に一部のデータを利用（ランダムに抽出）

- 学習データ：赤200、白200
- テストデータ：赤500、白500
- 講義ページからダウンロード

- 評価指標

- Mean absolute error
$$\text{MAE} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} |y_i - \hat{y}_i|$$

- ランキング

- スコアリングサーバ: <http://www.nlab.ci.i.u-tokyo.ac.jp/~nakayama/ds13/report1/index.php>
 - テストサンプルの推定結果を一行ずつ記載したテキストファイルを提出
 - ユーザ名を忘れずに（区別できればなんでもよい。ただし同じ名前をずっと使用すること）
 - 提出した結果は上書きされる。最後のものだけ保存されるので注意。
- 現在は、提出されたテストデータ1000の推定結果のうち、所定の200サンプルでスコアリングしている
- 最終的なスコアは、締め切り後に残りの800サンプルで算出

サンプルプログラム (線形回帰、sample.py)

```
import numpy as np
import pandas as pd
import regression as reg
```

```
nsample = 300
```

```
train_data = pd.read_csv("train.csv")
test_data = pd.read_csv("test.csv")
```

```
train_data = train_data.drop(['red or white'], axis=1) #remove 'red or white' information.
test_data = test_data.drop(['red or white'], axis=1)
```

```
X = np.matrix(train_data.drop(['quality'], axis=1))
y = np.matrix(train_data['quality'])
```

```
XTrain = X[:nsample,:] #use the first 300 samples for training
yTrain = y[:, :nsample]
XVal = X[nsample:,:] #use the rests for validation
yVal = y[:, nsample:]
```

```
w = reg.standRegres(XTrain,yTrain) #linear regression
```

```
yHatTrain = np.dot(XTrain,w)
yHatVal = np.dot(XVal,w)
```

```
print "Training error ", np.mean(np.abs(yTrain - yHatTrain.T))
print "Validation error ", np.mean(np.abs(yVal - yHatVal.T))
```

```
yHatTest = np.dot(np.matrix(test_data),w)
np.savetxt('result.txt', yHatTest)
```

(train.csv)

```
"red or white", "fixed acidity", "volatile acidity", "citric
acid", "residual sugar", "chlorides", "free sulfur dioxide", "total
sulfur dioxide", "density", "pH", "sulphates", "alcohol", "quality"
R, 7.7, 0.705, 0.1, 2.6, 0.084, 9, 26, 0.9976, 3.39, 0.49, 9.7, 5
R, 10.3, 0.27, 0.24, 2.1, 0.072, 15, 33, 0.9956, 3.22, 0.66, 12.8, 6
R, 8.5, 0.37, 0.32, 1.8, 0.066, 26, 51, 0.99456, 3.38, 0.72, 11.8, 6
R, 6.3, 0.3, 0.48, 1.8, 0.069, 18, 61, 0.9959, 3.44, 0.78, 10.3, 6
```

```
5.791953101912569402e+00
4.706598774763293136e+00
5.376977275096278319e+00
6.060673895165684222e+00
6.068585089649218389e+00
5.963326573799613506e+00
5.388982109344969018e+00
5.561331325216619881e+00
5.597390900380342593e+00
5.813484727415301201e+00
5.245247663082410305e+00
5.488444266101371483e+00
5.488444266101371483e+00
5.594112325354758219e+00
```

test.csv と同じ順番で、
各テストサンプルの予
測値が行ずつ入っ
ている。

(result.txt)

課題詳細

- レポート内容

- Wine quality predictionの問題に取り組み、以下の点を中心にA4用紙1~2枚程度にまとめよ。講義で扱っていない技術を用いても構わない。

1. 予測性能を向上させるための自分なりの工夫点と結果、考察（必須）

- コードを載せる必要はない（もちろん、実装がポイントの場合は載せてかまわない）

2. その他、自由にデータを分析した結果（+α）

- 学習データ数と性能、正則化パラメータの関係
- 各特徴の寄与の分析
- 赤ワインと白ワインを判別分析してみる …など

3. ここまでの講義の感想、要望など（必須）

- フィードバックをください！

課題詳細

- どうしても実装が難しい場合… (プログラミング未経験者の方など)
 - 1・2の代わりに、元論文を読み内容をまとめることでOKとします。
 - できれば挑戦して欲しいですが…
- 情報理工の人はだめですよ

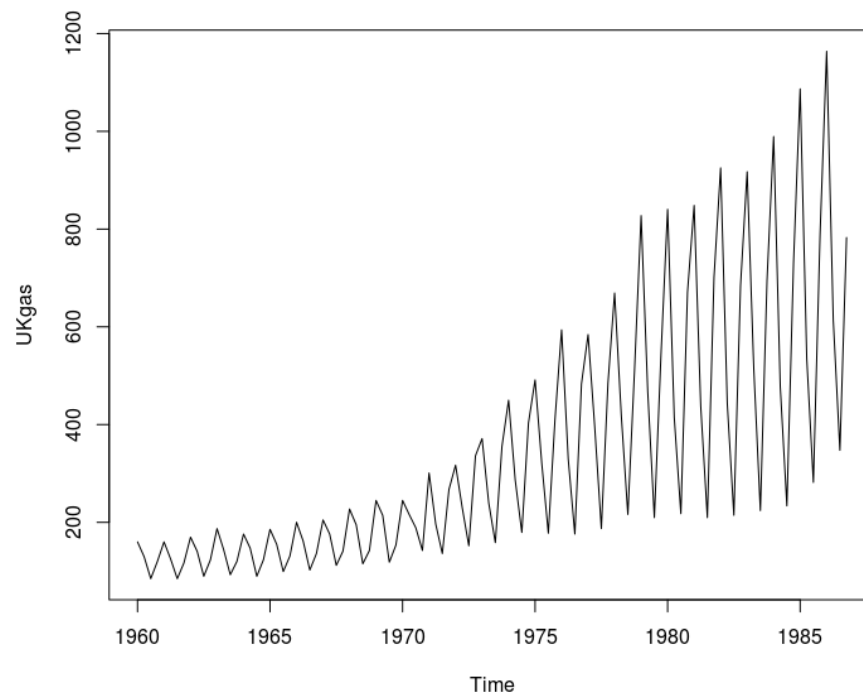
課題詳細

- 評価の方針
 - アイデアや試行錯誤の過程を重視
 - スコアが悪くても、しっかり考察してくれればOK
(もちろん良くなればプラスに評価しますが)
 - 面白い分析を期待します
- 提出先
 - 以下のアドレスへメールで提出すること（質問もこちらへ）
 - ds2013@nlab.ci.i.u-tokyo.ac.jp
 - 件名は「データサイエンスレポート課題1」
 - 氏名、学籍番号、所属を忘れずに
- 締め切り
 - 12月16日（月）
 - ただし、スコアリングサーバは13日に締め切り、最終結果発表

工夫できそうなところ

- 回帰のオフセット
- カテゴリ変数の扱い（赤 or 白）
 - ダミー変数？
 - 分けてモデルを作る？
- 手法・パラメータチューニング
 - 正則化項の入れ方（リッジ回帰）
 - クロスバリデーション
- 誤差の評価
 - L2? L1?
- 原論文を読んでみる

時系列予測

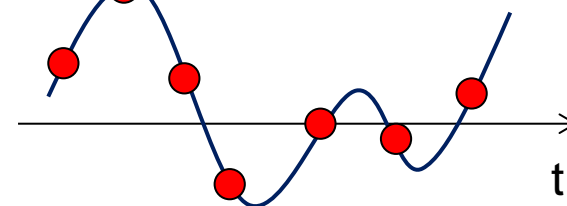


時系列分析

- 時系列データ：時間とともに変動する現象に対して、時間の順序で測定・観測した結果を記録したデータ
 - 気象データ
 - 金融・経済データ
 - 医療データ
- 時系列データは、確率過程からの標本
 - ある時点 n において、 y_n がどのような確率で出現するか

確率過程 $(\cdots, y_{-1}, y_0, \boxed{y_1, \cdots, y_n}, y_{n+1}, \cdots)$

標本（観測データ）



- 現象の本質的な構造を捉え、将来の変動を予測・制御したい

定常確率過程

- 弱定常（広義定常）：以下の二つの性質を有する確率過程
 - (1) $E(y_t) = \mu$ （平均が時間に依存せずに一定）
 - (2) $E(y_t^2) = \mu$ （分散が時間に依存せずに一定）
 - (3) $\text{Cov}(y_t, y_{t+h}) = \gamma(|h|)$ （異時点間の共分散が時間差のみに依存）
- ホワイト・ノイズ：上記(1)(2)に加え、共分散も時刻に依存せず常にゼロ
- 強定常（狭義定常）
 - (y_{t1}, \dots, y_{tn}) の同時分布と、時間 h だけシフトした $(y_{t1+h}, \dots, y_{tn+h})$ の同時分布が、すべての自然数 n とすべての整数 h に対して互いに同じになる
 - モーメント（平均、分散など）の存在を仮定しない
- 単に定常という場合、通常は弱定常のことを指す
 - 物理・工学の諸問題で頻出
 - 音声波形（短時間であれば定常とみなせる）
 - 種々の時系列分析手法において重要な前提となっている

線形定常過程（あるいは単に線形過程）

- 定常過程を生み出す確率モデルの一群
- ある時点の確率変数 y_t が、過去のホワイトノイズによる系列の線形和となる
- 一般的な標記

$$y_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}, \quad \{\varepsilon_t\} \sim \text{i.i.d. } N(0, \delta^2), \quad \sum_{j=0}^{\infty} |\psi_j| < \infty$$

- 現実的なパラメータ数で推定を行いたい

分解定理 [Brockwell-Davis 1996, Fuller 1996]など

- 任意の定常過程は、線形過程（ただし、IIDノイズによるモデル）と、決定論的な定常過程の和として表現できる.
- したがって、後者の成分がなければ、線形過程は定常過程の一般的な表現とみなすことができる
 - 通常、決定論的な部分分かっている場合は、長期トレンドなどと同様に別枠で考える
- 要するに、実用上は線形過程で十分であろうという期待

ARモデル (auto regression)

- p次のARモデル AR(p)

$$y_t = m + \sum_{j=1}^p a_j y_{t-j} + \varepsilon_t, \quad \{\varepsilon_t\} \sim \text{i.i.d. } N(0, \delta^2)$$

- 最も基本的な時系列予測手法
- 過去の自分の値の重み付き和で現在の値を回帰

ARモデルの推定

- Rの関数 `ar(x, aic = TRUE, method="", order.max=NULL, ...)`
 - デフォルトでは、AICを用いて最適な次数(p)を選択する

```
> (lh.ar <- ar(lh)) #ARモデルのフィッティング
```

Call:

```
ar(x = lh)       $\hat{y}_t = 0.653y_{t-1} - 0.064y_{t-2} - 0.227y_{t-3}$ 
```

Coefficients:

1	2	3
0.6534	-0.0636	-0.2269

↑

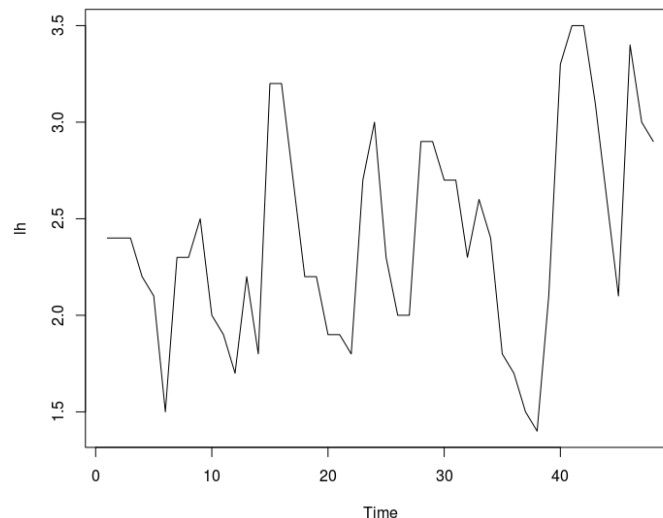
Order selected 3 sigma^2 estimated as 0.1959

```
> lh.ar$order
```

```
[1] 3
```

```
> lh.ar$aic
```

(省略)



lh データセット

(血中の黄体ホルモン量を
10分置きに計測したもの)

情報量規準（再掲）

- 赤池情報量規準 (AIC) (×基準)
 - 以下を最小とするモデルを選択する

$$-2\ln(L) + 2M$$

対数尤度

パラメータ数

- BIC (Bayesian information criterion)

$$-2\ln(L) + M \ln(N)$$

- MDL (minimal description length)

$$-\ln(L) + \frac{M \ln(N)}{2}$$

- 訓練データだけから、「そこそこいいモデル」が得られる
 - 実際には、過度に単純なものが得られる場合が多いらしい
 - 変数選択などに使える

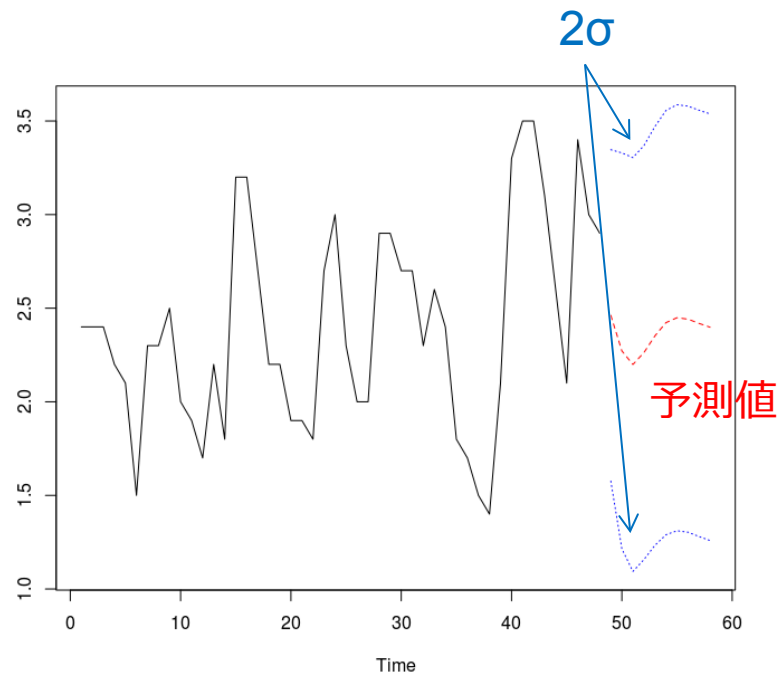
予測してみる

```
> (lh.pr<-predict(lh.ar,n.ahead=10)) #10点先まで予測
$pred
Time Series:
Start = 49
End = 58
Frequency = 1
[1] 2.461588 2.272267 2.199151 2.262914 2.352194 2.423066
2.449223 2.441544
[9] 2.418779 2.398456
```

\$se #残差（正規分布の標準偏差）

```
Time Series:
Start = 49
End = 58
Frequency = 1
[1] 0.4425687 0.5286675 0.5525786 0.5527502 0.5592254
0.5665903 0.5688786
[8] 0.5689385 0.5692396 0.5697534
```

```
> SE1<-lh.pr$pred+2*lh.pr$se #2σの線
> SE2<-lh.pr$pred-2*lh.pr$se
> ts.plot(lh,lh.pr$pred,SE1,SE2,gpars=list(lt=c(1,2,3,3),col=c(1,2,4,4)))
```



MAモデル (移動平均、moving average)

- q次のMAモデル MA(q)

$$y_t = m + \varepsilon_t + \sum_{j=1}^q b_j \varepsilon_{t-j}, \quad \{\varepsilon_t\} \sim \text{i.i.d. } N(0, \delta^2)$$

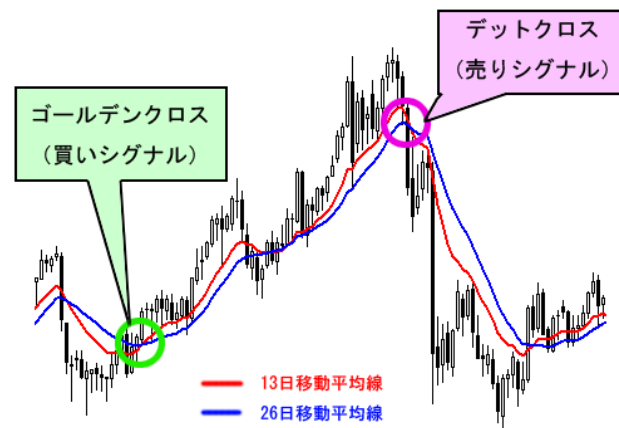
- 過去のホワイトノイズの重み付和で現在の値を表現
- y_t, y_{t-1} でオーバーラップができる = 自己相関のモデル化

注意：移動平均法（移動平均線）と混同しないこと

- 移動平均法

- 時系列データの平滑化の方法
 - ノイズ除去、トレンド解析、予測
- 過去の一定期間の観測値の平均値をとって代表値とする方法
- 金融分野などで非常によく用いられる

$$\bar{y}_t = \frac{1}{n} \sum_{i=0}^{n-1} y_{t-i}$$



<http://finalrich.com/fx/fx-technical-moving-average.html>

- 確率変数間の相関をモデリングするために、オーバーラップさせた変数を介在させるところだけが似ている

寄り道：移動平均いろいろ

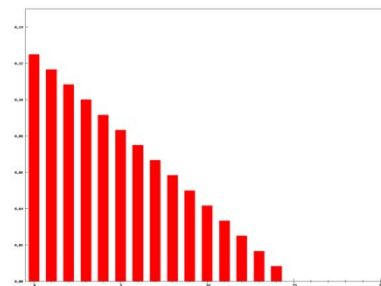
- 単純移動平均

- 単に移動平均という場合はこれ $\bar{y}_t^{SMA} = \frac{1}{n} \sum_{i=0}^{n-1} y_{t-i}$

- 加重移動平均

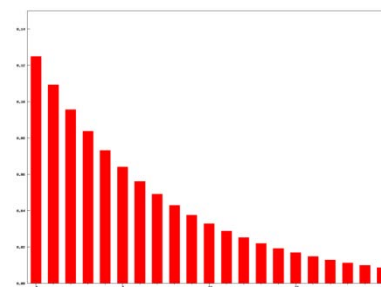
- 直近の観測値に重みづけ

$$\bar{y}_t^{WMA} = \frac{2}{n(n+1)} \sum_{i=0}^{n-1} (n-i) y_{t-i}$$



- 指数移動平均（指数平滑法）

$$\bar{y}_t^{EMA} = (1 - \beta) \sum_{i=0}^{n-1} \beta^i y_{t-i} \quad (0 < \beta < 1)$$



ARMAモデル (AR+MA)

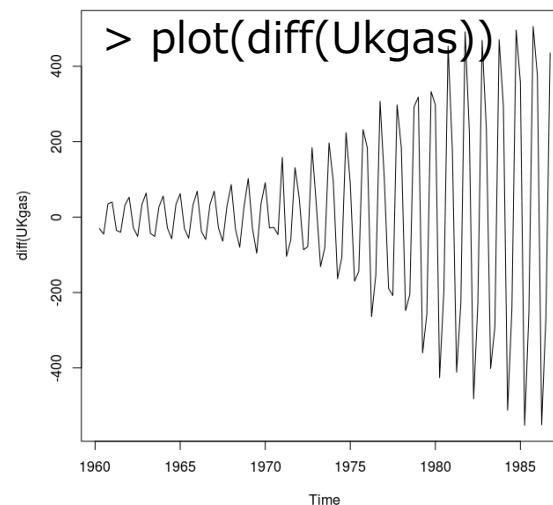
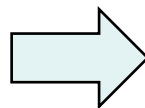
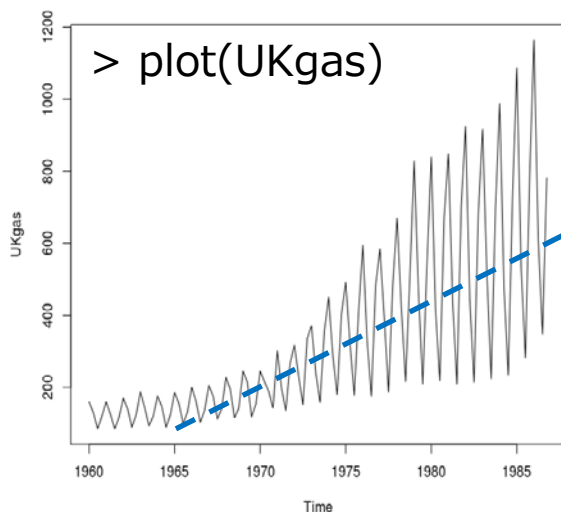
- ARMA(p,q)

$$y_t = m + \varepsilon_t \sum_{j=1}^p a_j y_{t-j} + \sum_{j=1}^q b_j \varepsilon_{t-j}, \quad \{\varepsilon_t\} \sim \text{i.i.d. } N(0, \delta^2)$$

- 少ないパラメータで複雑な変動パターンを表現できる
 - 一般に、AR・MAモデルでもラグを十分とれば同等の表現能力をも持ちうるが、パラメータが多くなる（過学習しやすい）
- p, qは実用上1~3程度で問題がない [Box-Jenkins]
- 定常過程の時系列データに対する最も標準的な方法
 - ただし、実際は次で説明するARIMAの枠組の中で扱われる

ARIMAモデル

- $ARIMA(p, d, q)$: ARMAモデルを非定常データへ拡張
- 時系列データの階差をとり、トレンドを除去した後、ARMAモデルをフィッティング



$$y_t = (\alpha + \beta t) + \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}$$

$$\Delta y_t = y_t - y_{t-1} = \beta + \psi_0 \varepsilon_t + \sum_{j=1}^{\infty} (\psi_j - \psi_{j-1}) \varepsilon_{t-j}$$

d 次多項式によるトレンドはd回階差をとれば消える

ARIMAモデルのフィッティング

- Rの関数 `arima(x, order = c(p,d,q)...`)
 - p,d,qはそれぞれARの次数、差分の回数、MAの次数
- `arima(2,0,1)`の例 #d=0なのでARMA
> (lh.ari<-arima(lh,order=c(2,0,1)))

Call:

```
arima(x = lh, order = c(2, 0, 1))
```

Coefficients:

	ar1	ar2	ma1	intercept
	1.1765	-0.5044	-0.5080	2.3946
s.e.	0.3990	0.2190	0.4517	0.0944

sigma^2 estimated as 0.1827: log likelihood = -27.6, aic = 65.2

パラメータの推定

- 例えば、AICを基準に選ぶ

```
data<-lh; cnt<-0;
for(p in 1:4)
  for(d in 0:1)
    for(q in 0:4){
      fit<-arima(data,order=c(p,d,q));
      cnt<-cnt+1;
      if(cnt==1){
        minaic<-fit$aic;
        orderP<-p; orderD<-d; orderQ<-q;
      }else{
        if (fit$aic<minaic){
          minaic<-fit$aic;
          orderP<-p; orderD<-d; orderQ<-q;
        }
      }
    }
  }
}
```

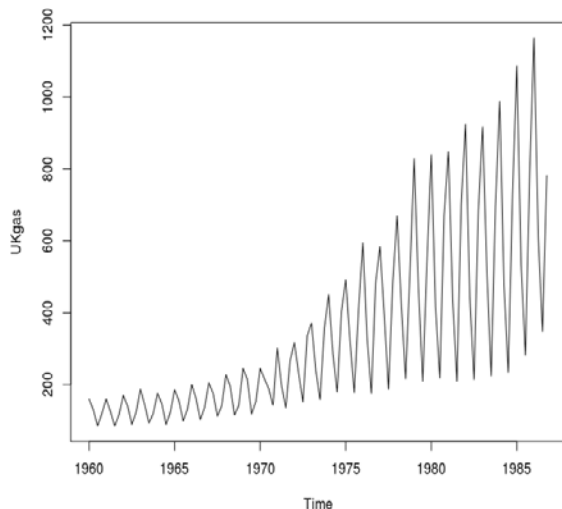
```
> cat("Results: p=",orderP, "d=",orderD, "q=",orderQ, "AIC=",minaic,"¥n");
Results: p= 3 d= 0 q= 0 AIC= 64.18482
```



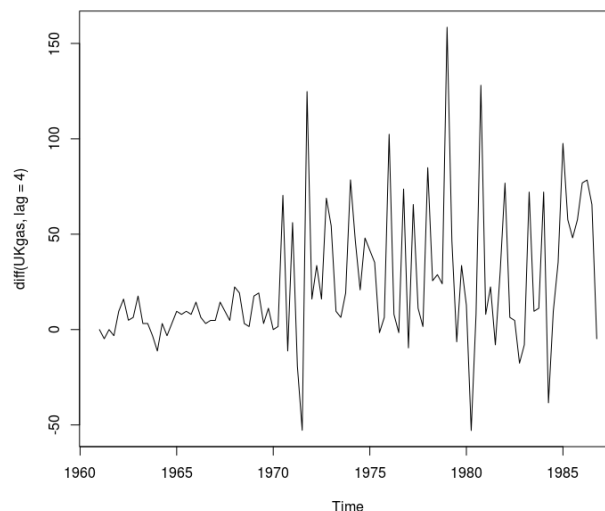
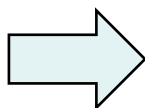
この場合、結局ARモデルと等価

SARIMAモデル (Seasonal ARIMA)

- 季節性の変動を考慮したARIMA
- 季節階差（前年同期）との差をとり、季節性の影響をキャンセルしたあとARIMAをフィッティングする



> plot(UKgas)

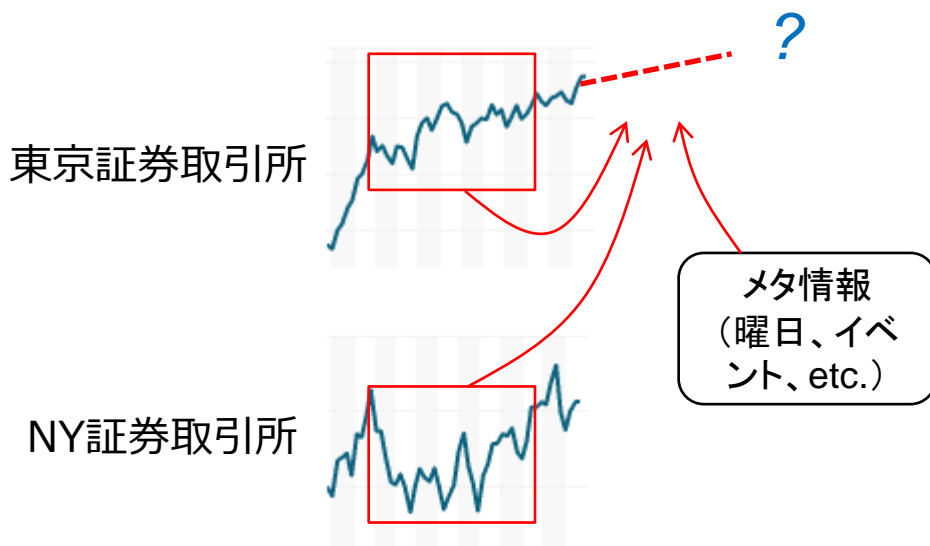


> plot(diff(UKgas, lag=4))

- Rでは、arima関数の"seasonal"パラメータで指定可能

補足：外部情報の利用

- 他の時系列、ダミー変数等を加えた回帰
 - 実際のデータマイニングの場面ではこちらの方が多い？



```
fit <- arima(data, c(2, 0, 3), xreg = my_x )  
predict(fit, n.ahead = 20, newxreg =  
my_newx)
```

のように外部の説明変数をarimaに入れることが可能

- ニューラルネットワーク等もよく使われる
- 時間差があるだけで、一般的な回帰の問題に

まとめ

- 時系列は確率過程
- 定常性、線形過程の定義と各手法の位置付けを理解
 - ARMA, ARIMAなどが代表的な方法
 - MAと移動平均法を混同しないように
- 外部情報を用いた時系列予測もよく行われる
 - 回帰分析で使われる方法論は広く利用可能
- 非定常（トレンドがある場合）は注意が必要
 - 階差をとってグラフを描いてみる。定常っぽくなるか？
 - 低次多項式でトレンドをモデル化するか、ARIMAを使うか…