Time Series Analysis

Topics in Data Mining
Fall 2015

Bruno Ribeiro

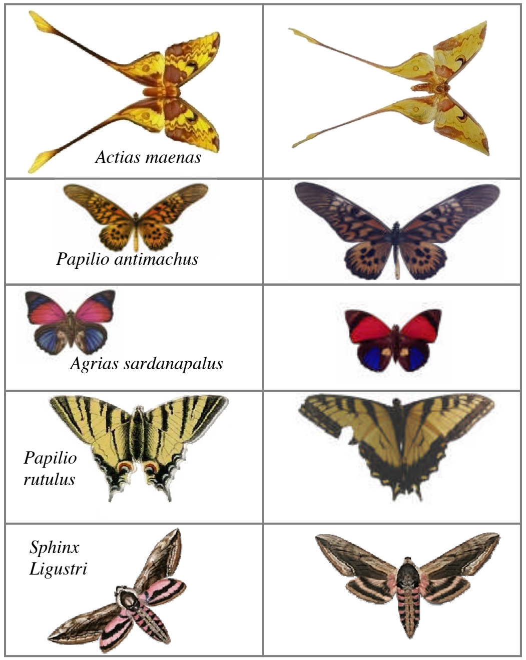
Motivation

What are time series?

- Sequence of data points

- Sequence must have meaning
  - Otherwise it is just a set of points

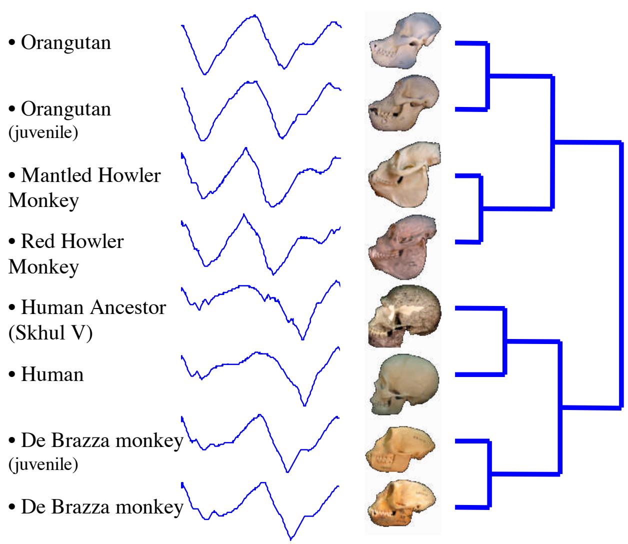
Google Finance



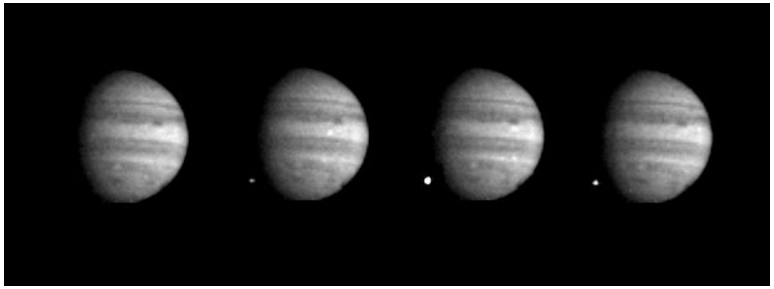
Keogh et al. VLDB 2006

1D Representation of Objects Very Common

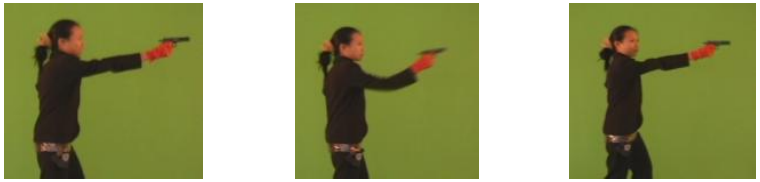
- Time series:
  - Sequence of data points
  - Sequence must have meaning (e.g., organization in space)
  - But 2D also very common



Keogh et al. VLDB 2006



NASA



Formal Definition

- Given: one or more sequences

$(c_1 , c_2 , \ldots , c_t)$

$(q_1, q_2, \ldots , q_t)$

- Find
  - similar sequences; forecasts
  - patterns; clusters; outliers

Overview

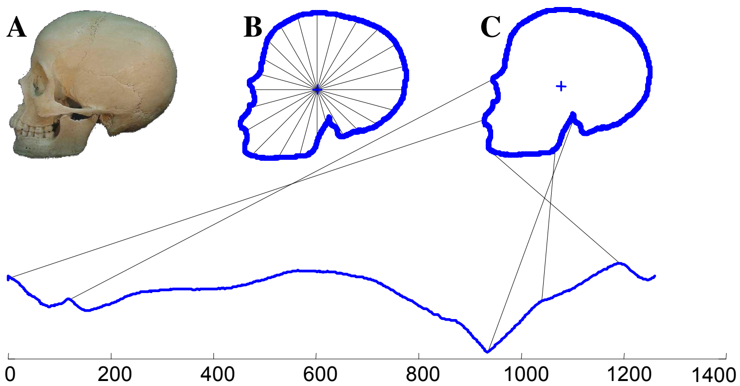
- Matching Time Series

- Models

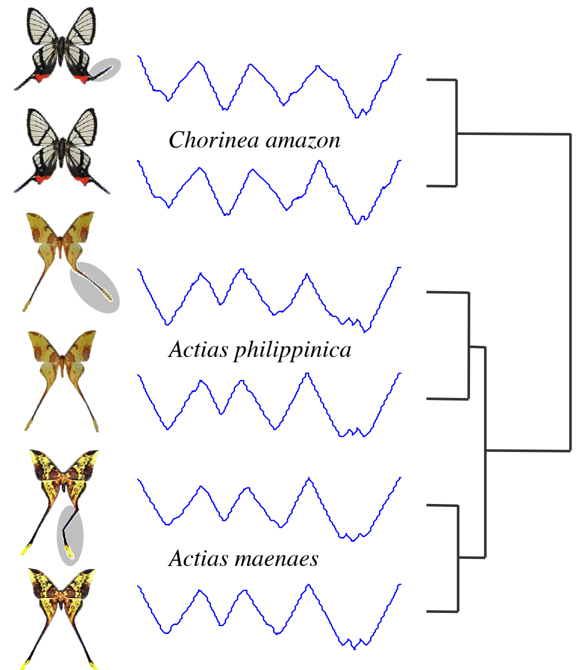
Overview

- Matching Time Series
- Models

Data Representation is Key



Keogh et al. VLDB 2006



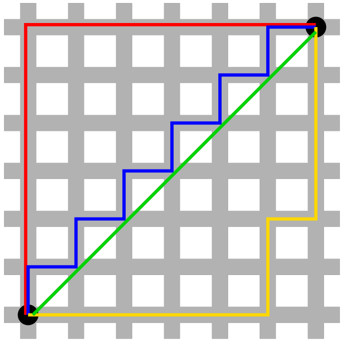
Keogh et al. VLDB 2006

$T_1 = 0$, **for** i = 1 **to** |DNAstring|
- **if** $DNAstring_i$ = A, **then** $T_{i+1} = T_i + 2$
- **if** $DNAstring_i$ = G, **then** $T_{i+1} = T_i + 1$
- **if** $DNAstring_i$ = C, **then** $T_{i+1} = T_i - 1$
- **if** $DNAstring_i$ = T, **then** $T_{i+1} = T_i - 2$

Rakthanmanon et al. KDD 2012

Similarity

$$L_p(\mathbf{c}, \mathbf{q}) = \sqrt[p]{\sum_{i=1}^{t} |c_i - q_i|^p}$$

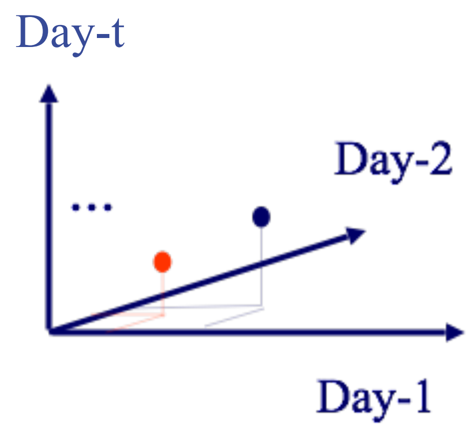


- L_1: Manhattan distance
- L_2 = Euclidean distance
- $L_\infty = \max(|c_1 - q_1|, \ldots, |c_t - q_t|)$

In Time Series Space

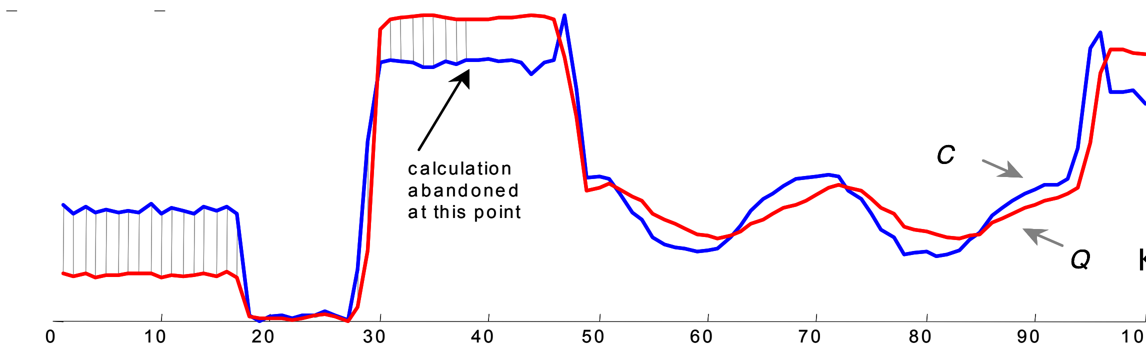
- Euclidean distance is inner product in t-dimensional space

- Related to
  - cosine similarity
  - cross-correlation



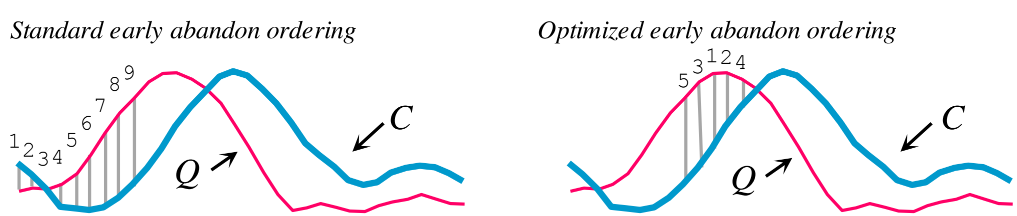
Early Termination

- Fast matching of time series



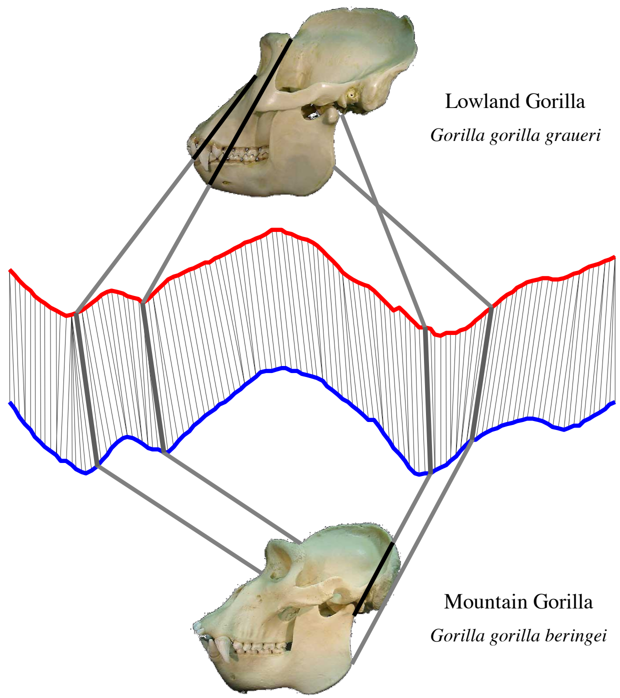
Keorgh et al. 2006

- Even faster matching of time series



Rakthanmanon et al. 2012

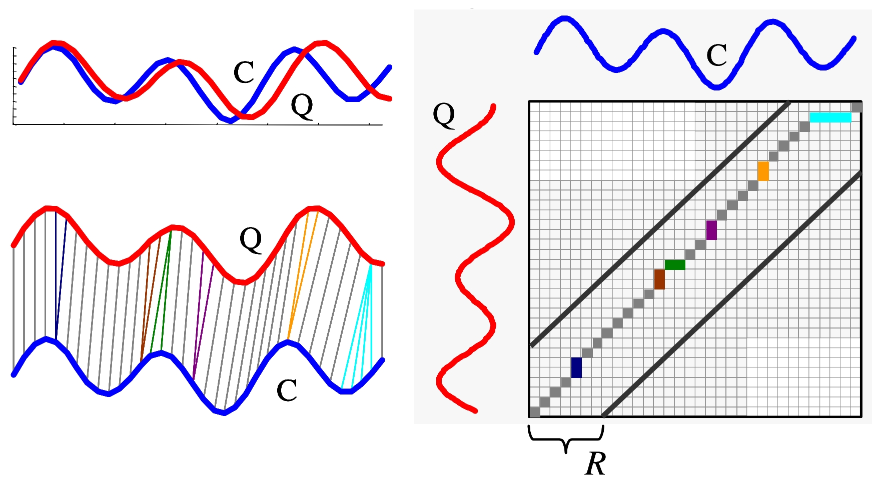
But Alignment not Always Perfect



Rakthanmanon et al. KDD 2012

Dynamic Time Warping (DTW)

- Accelerations & decelerations possible
- Compute Euclidean with accel. and decc. in mind
- Closely related to string-editing distance



Keogh et al. VLDB 2006

Similar: Approximate matching - string editing distance:
d(‘holes’, ‘mole’) = 2
= min # of insertions, deletions, substitutions to transform the first string into the second

Dynamic Time Warping Computation

- Dynamic programming

  $D(\ i, j\ )$ = matching cost up to times i and j

of sequences c and q

$$D(i,j) = \|\mathbf{c}_i - \mathbf{q}_j\| + \min \begin{cases} D(i-1, j-1) & \#\ \text{“substitution”} \\ D(i-1, j) & \#\ \text{deletion} \\ D(i, j-1) & \#\ \text{insertion} \end{cases}$$

Substitution cost

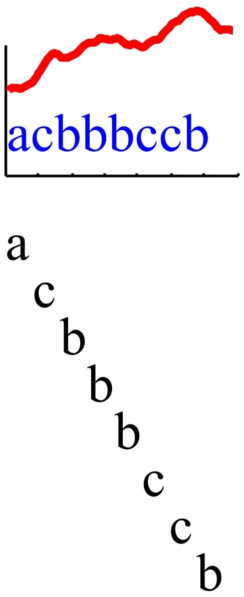
Complexity: $O(t_1\ t_2)$ - quadratic on the length of the sequences

Quantization

- Modify time series into a sequence of symbols

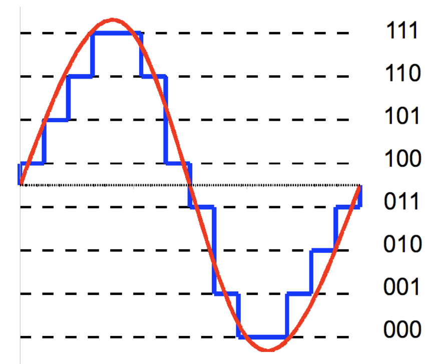
Allows:

- Hashing
- Suffix Trees
- Simpler Markov Models
- Use more text processing ideas

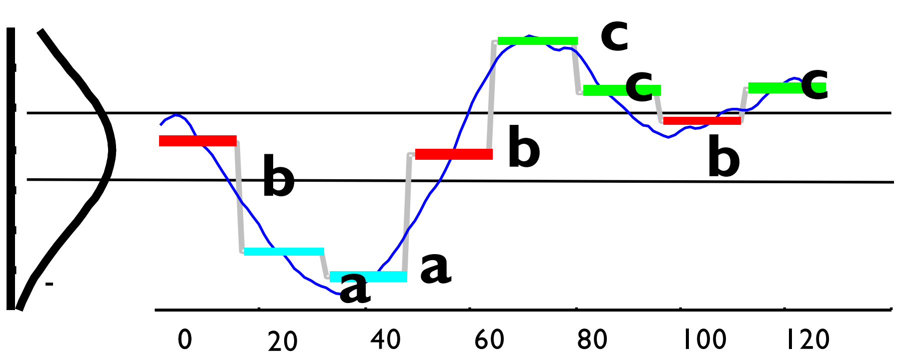


What is Quantization?

- Quantization: Comes from the statistical term "quantiles"
  - Use quantiles of signal density function to replace time series by symbols
- Widely used in signal processing
- Examples in time series mining
  - SAX – Lin et al. 2007



Wikipedia



Keorgh 2007

baabccbc

Euclidean Distance Lower Bound

$\hat{C}$ = **bbabcbac**

$\hat{Q}$ = **bbaccbac**

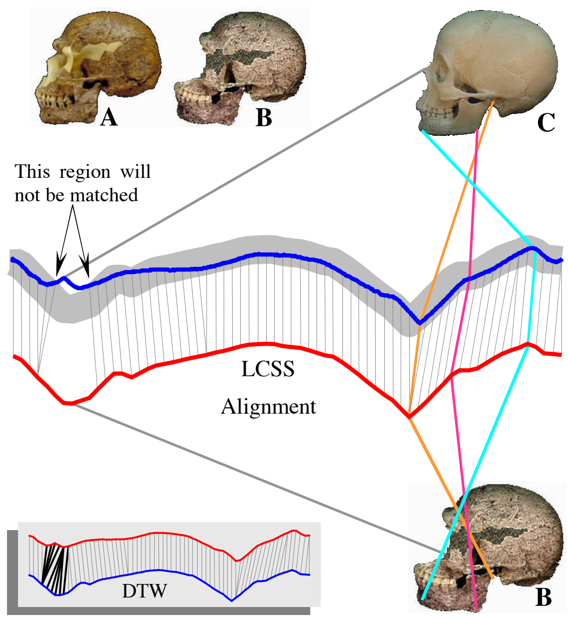
$$MINDIST(\hat{Q},\hat{C}) \equiv \sqrt{\tfrac{n}{w}}\sqrt{\sum_{i=1}^{w}\left(dist(\hat{q}_i,\hat{c}_i)\right)^2}$$

dist() can be implemented using a table lookup.

Camerra et al. ICDM 2010

Matching with Errors

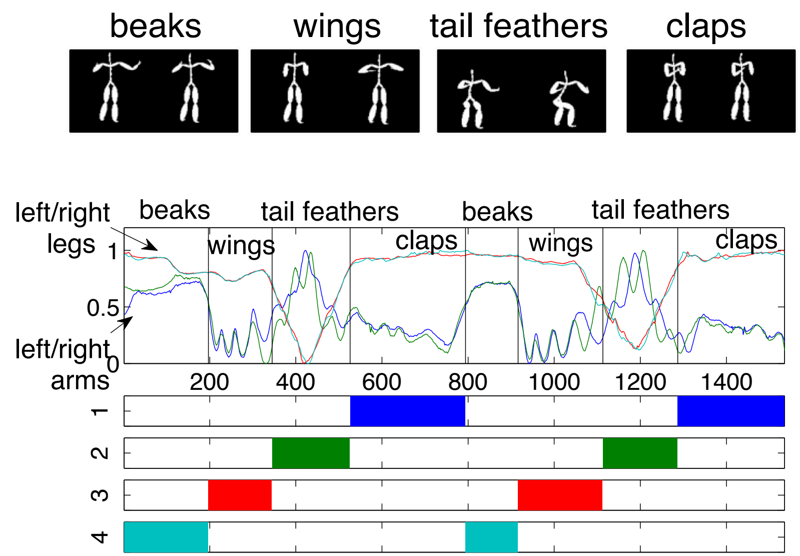
- Longest Common Sub-String



Keorgh et al. 2006

Time Series Motifs

- Helps identify commands / words
- Can be used in prediction
- Obtain more “meaningful” symbolic representation



Matsubara et al. SIGMOD 2014

Overview

- Matching Time Series
- Models

Linear Forecasting

Linear Forecasting

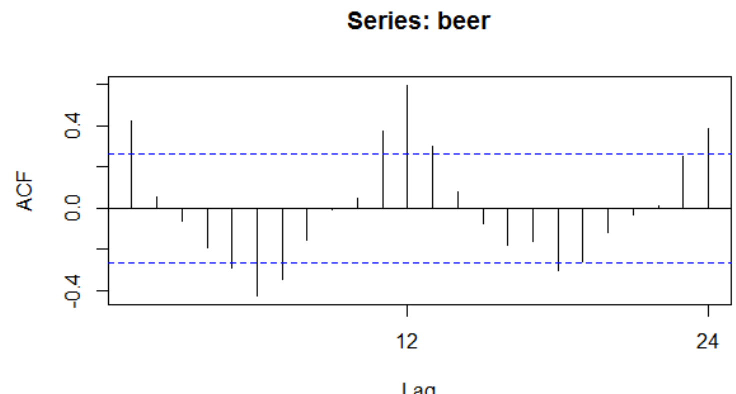
- Linear Forecasting
  - Auto-regression: Least Squares; Recursive Least Squares

Google Finance

Pre-processing Time Series

- Remove trends (drift)
- Remove periodicity

Google Finance

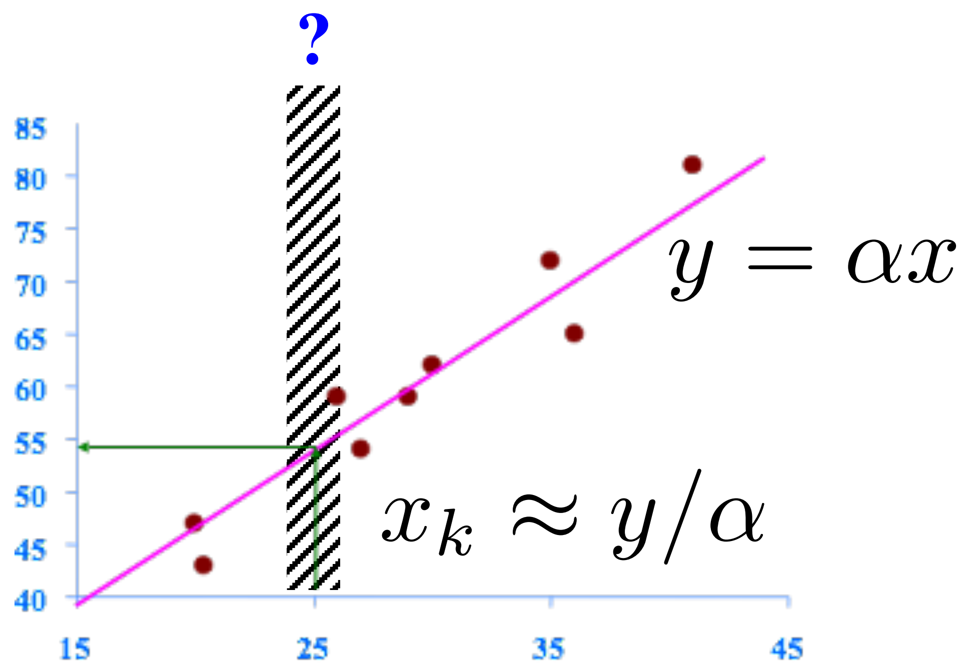


Australian beer consumption shows periodicity (seasonality)
Autocorrelation function

Linear Regression

- Interpolation (something is missing)

- $(x_1, \ldots, x_t)$
- $(y_1, \ldots, y_t)$



Faloutsos 2014

Auto-regression: Predicting Point in w Steps

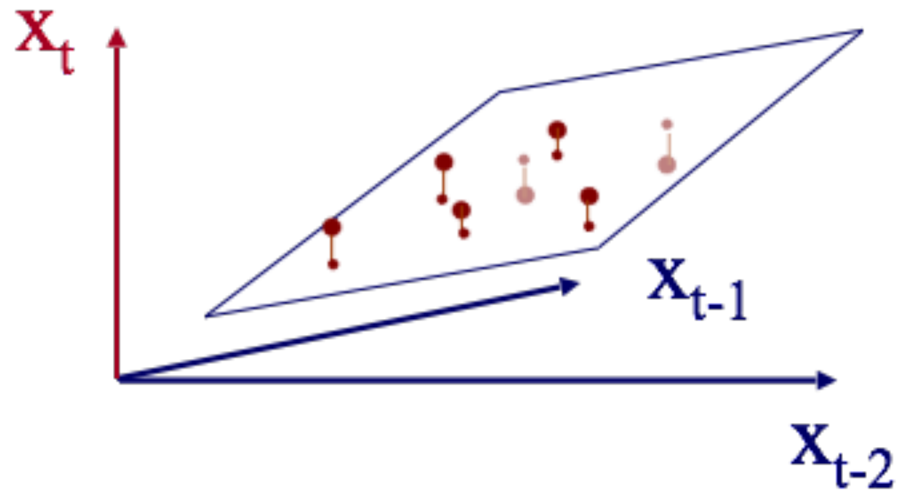
Google Finance

$$x_{w+1} = \sum_{i=1}^{w} a_i x_i + \epsilon_{\text{noise}}$$

Similar problem to linear regression:
express unknowns as a linear function of knowns

Predictions from Historical Data

- $\mathbf{X}_{[\mathbf{t} \cdot \mathrm{w}]} \cdot \mathbf{a}_{[\mathbf{w} \cdot 1]} = \mathbf{y}_{[\mathbf{t} \cdot 1]}$



Faloutosos 2014

- Over-constrained problem
  - $\mathbf{a}$ is the vector of the regression coefficients
  - $\mathbf{X}$ has the t values of the w indep. variables
  - $\mathbf{y}$ has the t values of the dependent variable

Looking Into Multiplication

- $\mathbf{X}_{[\mathbf{t}\ \cdot \mathrm{w}]} \cdot \mathbf{a}_{[\mathbf{w}\ \cdot 1]} = \mathbf{y}_{[\mathbf{t}\ \cdot 1]}$

time

$$\begin{bmatrix} X_{11}, X_{12}, \ldots, X_{1w} \\ X_{21}, X_{22}, \ldots, X_{2w} \\ \vdots \\ \vdots \\ \vdots \\ X_{t1}, X_{t2}, \ldots, X_{tw} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{bmatrix}$$

How to Estimate a?

- $\mathbf{a} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot (\mathbf{X}^T \cdot \mathbf{y})$

$\mathbf{X}^+ = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T$ is the Moore–Penrose pseudoinverse

Or: $\mathbf{a} = \mathbf{X}^+ \mathrm{y}$

a is the vector that minimizes the RMSE of $(\mathbf{y} - \mathrm{X} \cdot \mathbf{a}')$

Problems:

Matrix X grows over time needs matrix inversion

- $O(t \cdot w^2)$ computation
- $O(t \cdot w)$ storage

Recursive Least Squares

At time t we know $\mathbf{X}_t = (x_1, \ldots, x_t), \quad \mathbf{y}_t = (y_1, \ldots, y_{t-w})$

Least squares is solving

$$\underset{\mathbf{a}^\star}{\operatorname{argmax}} \|\mathbf{a}^T \mathbf{X}_t - \mathbf{y}_t\|^2$$

which gives

$$\mathbf{a}^\star = \mathbf{X}^+ \mathbf{y}$$

Let

$$\mathbf{\Phi}_t = \mathbf{X}_t^T \mathbf{X}_t \qquad \theta_t = \mathbf{X}_t^T y_t$$

Then $\mathbf{\Phi}_{t+1}^{-1}$ is

$$\mathbf{\Phi}_{t+1}^{-1} = (\mathbf{\Phi}_t + \mathbf{x}^T(t+1)\mathbf{x}^T(t+1))^{-1} = \mathbf{\Phi}_t^{-1} - \frac{\mathbf{\Phi}_t^{-1}\mathbf{x}^T(t+1)\mathbf{x}^T(t+1)\mathbf{\Phi}_t^{-1}}{1 + \mathbf{x}^T(t+1)\mathbf{\Phi}_t^{-1}\mathbf{x}(t+1)}$$

Recursive Least Squares Algorithm

$$\mathbf{\Phi}_{t+1}^{-1} = \mathbf{\Phi}_t^{-1} - \frac{\mathbf{\Phi}_t^{-1}\mathbf{x}^T(t+1)\mathbf{x}^T(t+1)\mathbf{\Phi}_t^{-1}}{1+\mathbf{x}^T(t+1)\mathbf{\Phi}_t^{-1}\mathbf{x}(t+1)}$$

$$\theta_{t+1} = \theta_t + \mathbf{x}^T(t+1)\mathbf{y}_{t+1}$$

$$\mathbf{a}_{t+1} = \mathbf{\Phi}_{t+1}^{-1}\theta_{t+1}$$

Exponentially Weighted Recursive Least Squares Algorithm

for $\lambda > 1$

$$\mathbf{\Phi}_{t+1}^{-1} = \frac{1}{\lambda}\mathbf{\Phi}_t^{-1} - \frac{1}{\lambda^2}\frac{\mathbf{\Phi}_t^{-1}\mathbf{x}^T(t+1)\mathbf{x}^T(t+1)\mathbf{\Phi}_t^{-1}}{1+\mathbf{x}^T(t+1)\mathbf{\Phi}_t^{-1}\mathbf{x}(t+1)}$$

$$\theta_{t+1} = \lambda\theta_t + \mathbf{x}^T(t+1)\mathbf{y}_{t+1}$$

$$\mathbf{a}_{t+1} = \mathbf{\Phi}_{t+1}^{-1}\theta_{t+1}$$

Comparison

Original Least Squares

- Needs huge matrix (growing in size) $O(t \times w)$
- Costly matrix operation $O(t \times w^2)$

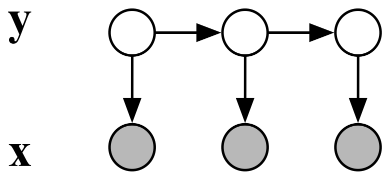
Recursive LS

- Need much smaller, fixed size matrix $O(w \times w)$
- Fast, incremental computation $O(1 \times w^2)$
- no matrix inversion

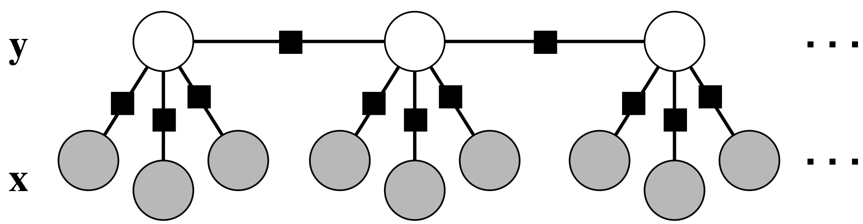
More Advanced Models

Graphical Models

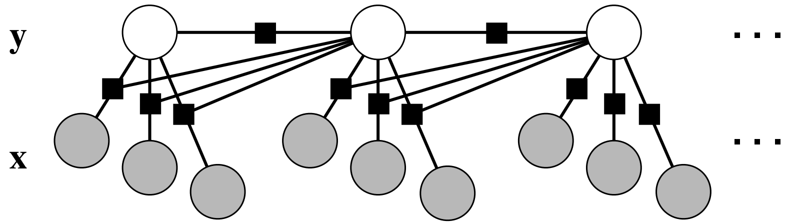
- HMMs



- Conditional Random Fields

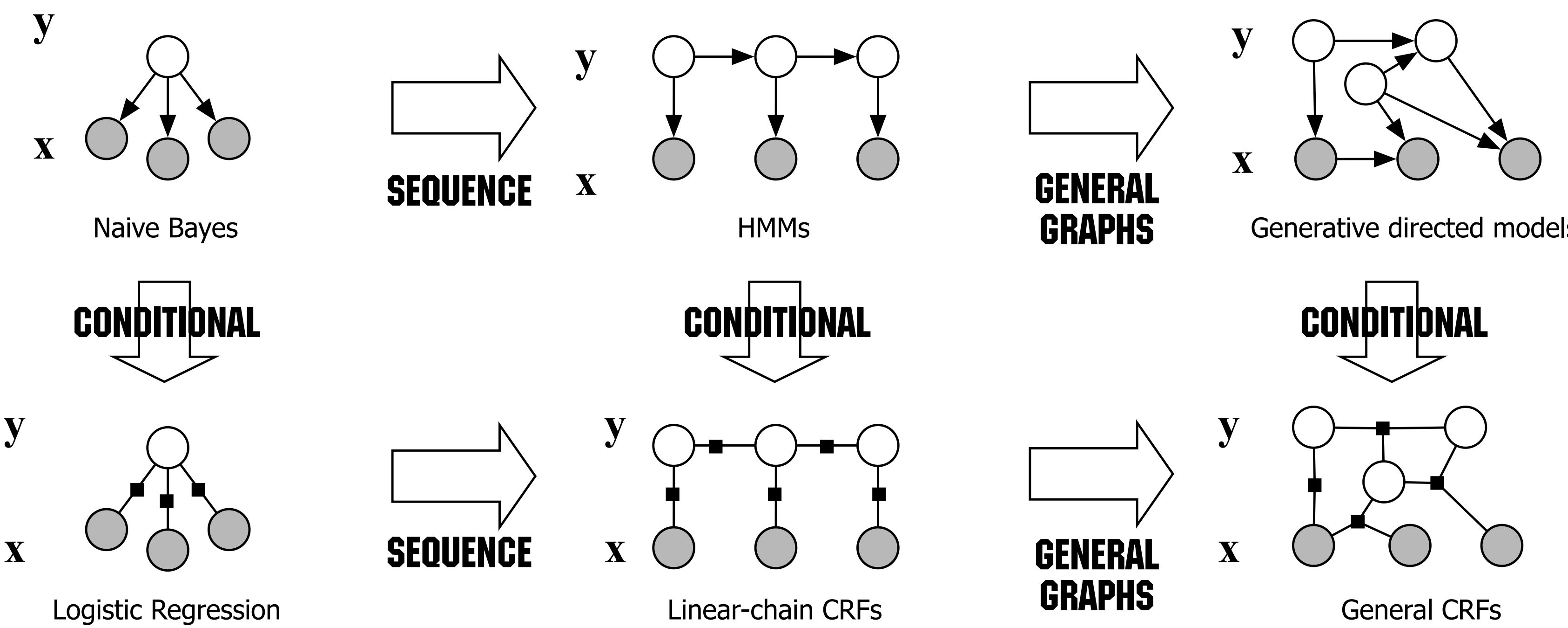


- Linear-chain CRF



Figures: Sutton & McCallum 2002

More Generally



Recommended Reading: C. Sutton & A. McCallum 2002

Figures: Sutton & McCallum 2002