Chapter 1: Getting Started

$$E[X] = \mu = \frac{1}{n}\sum_{i=0}^{n-1} x_i$$

$$<\vec{v},\vec{w}> = \frac{\vec{v}.\vec{w}}{|\vec{v}|.|\vec{w}|} \quad \alpha : \vec{v} \rightarrow <\vec{v},\vec{w}>$$

$$p(X,Y) = p(X \cap Y) = p(X).p(Y)$$

$$p(Y \mid X) = p(X \mid Y).\,p(Y)/p(X)$$

$$p(Y \mid X) = p(Y,X)/p(X)$$

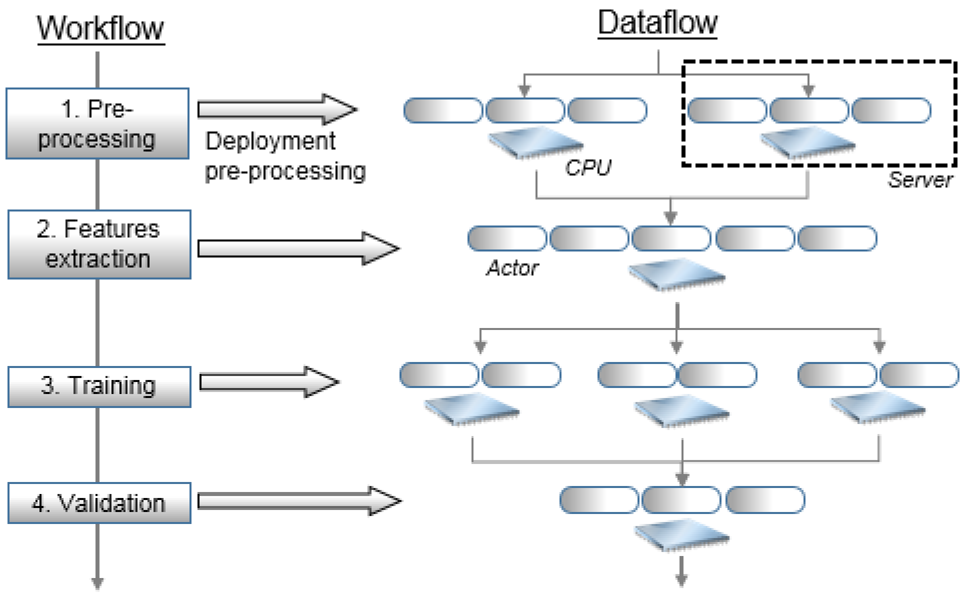
$$y_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

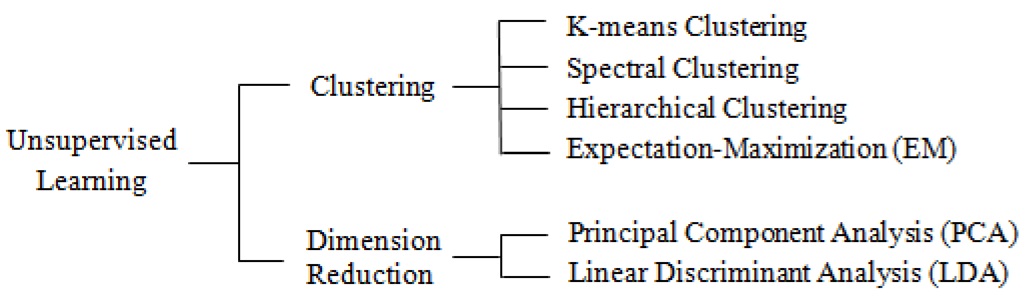
$$y_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}(h-l)+l$$

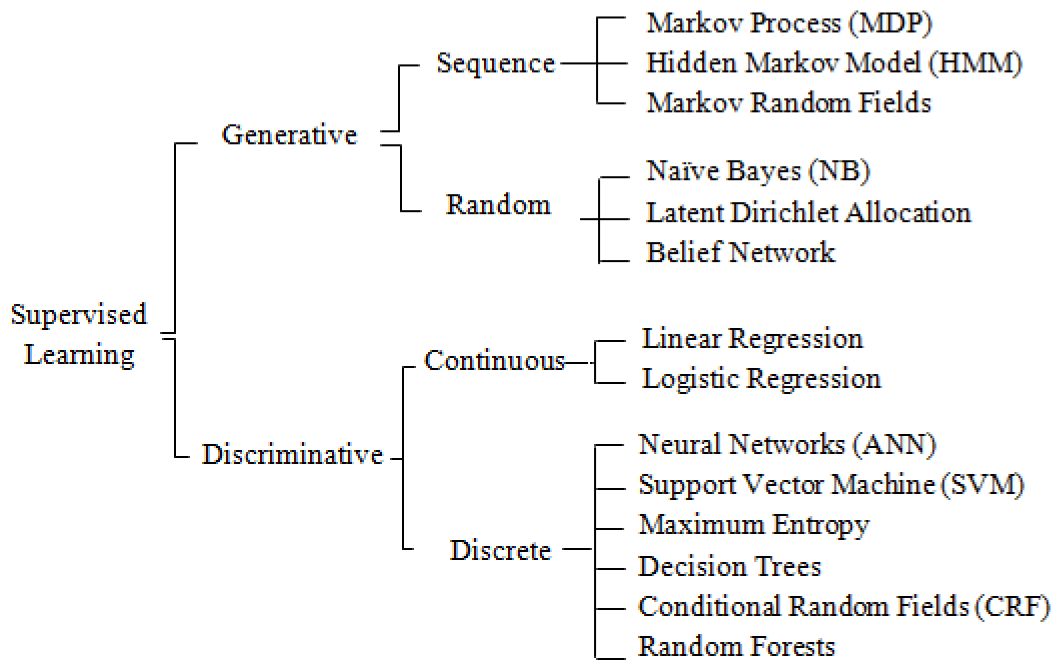
$$f(x \mid w) = w_0 + \sum_{i=1}^{N-1} x_i w_i \quad l(x \mid w) = \frac{1}{1+e^{-f(x|w)}}$$

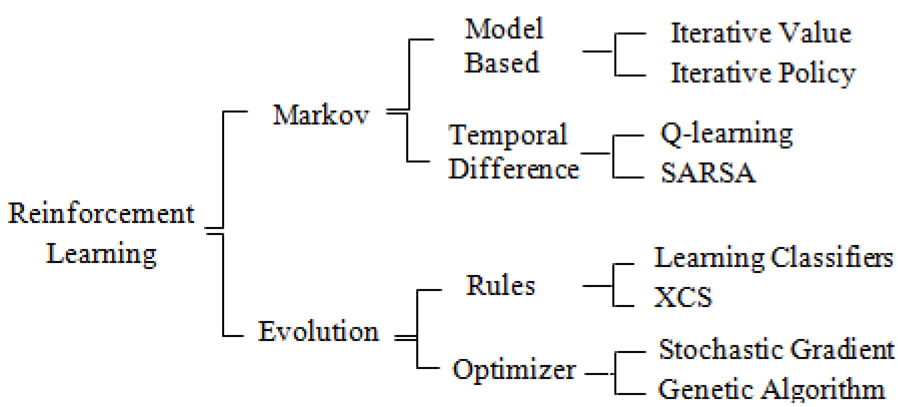
$$\mathcal{L}(x \mid w) = \frac{1}{N}\sum_{n=0}^{N-1} \log\left(1+e^{-y.W^T x}\right)$$

$$w_i^{(t+1)} = w_i^{(t)} + \eta \frac{x_i y}{1+e^{y w^T x}}$$

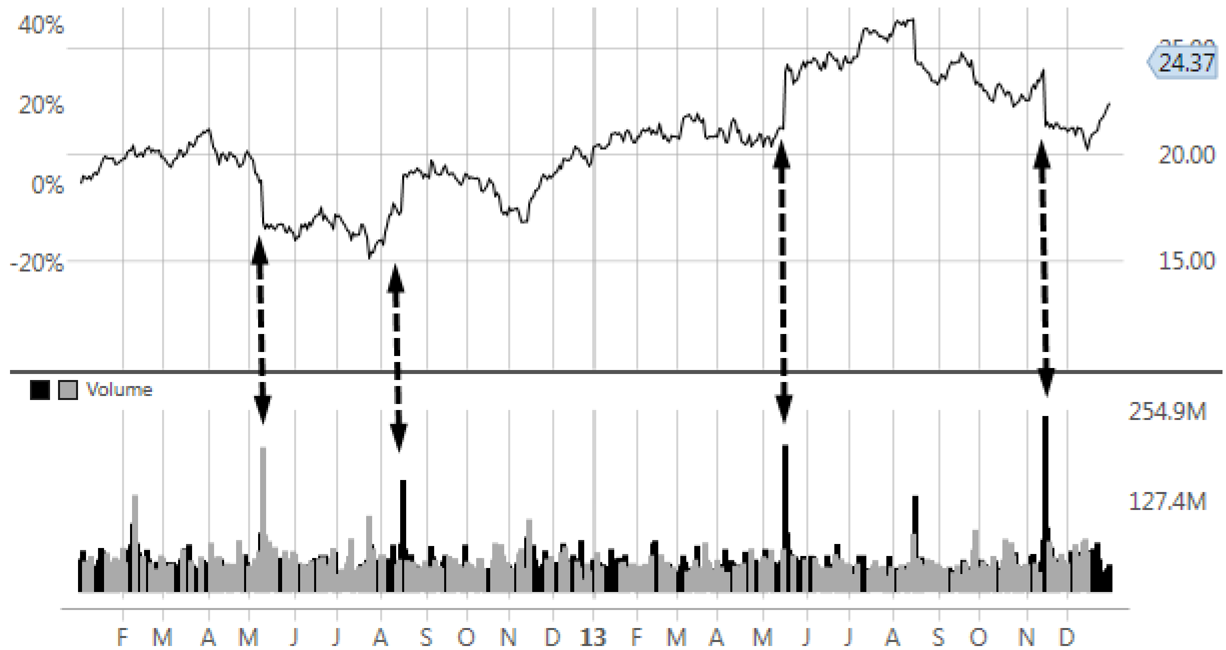


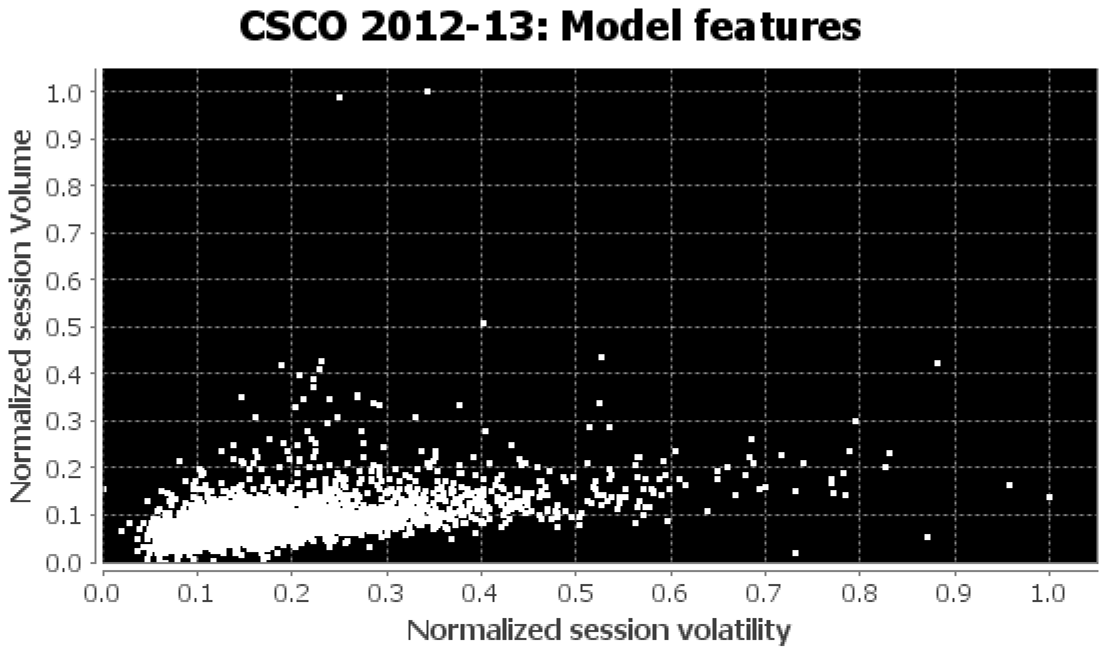


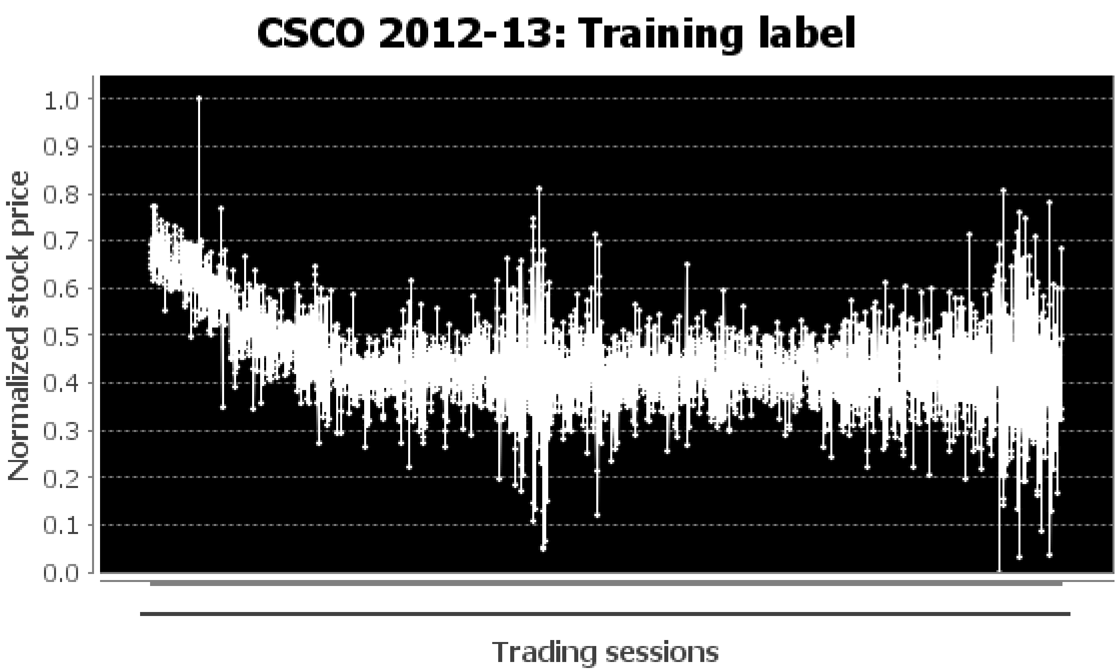

Supervised Learning
Generative
Sequence
Markov Process (MDP)
Hidden Markov Model (HMM)
Markov Random Fields
Random
Naïve Bayes (NB)
Latent Dirichlet Allocation
Belief Network
Discriminative
Continuous
Linear Regression
Logistic Regression
Discrete
Neural Networks (ANN)
Support Vector Machine (SVM)
Maximum Entropy
Decision Trees
Conditional Random Fields (CRF)
Random Forests

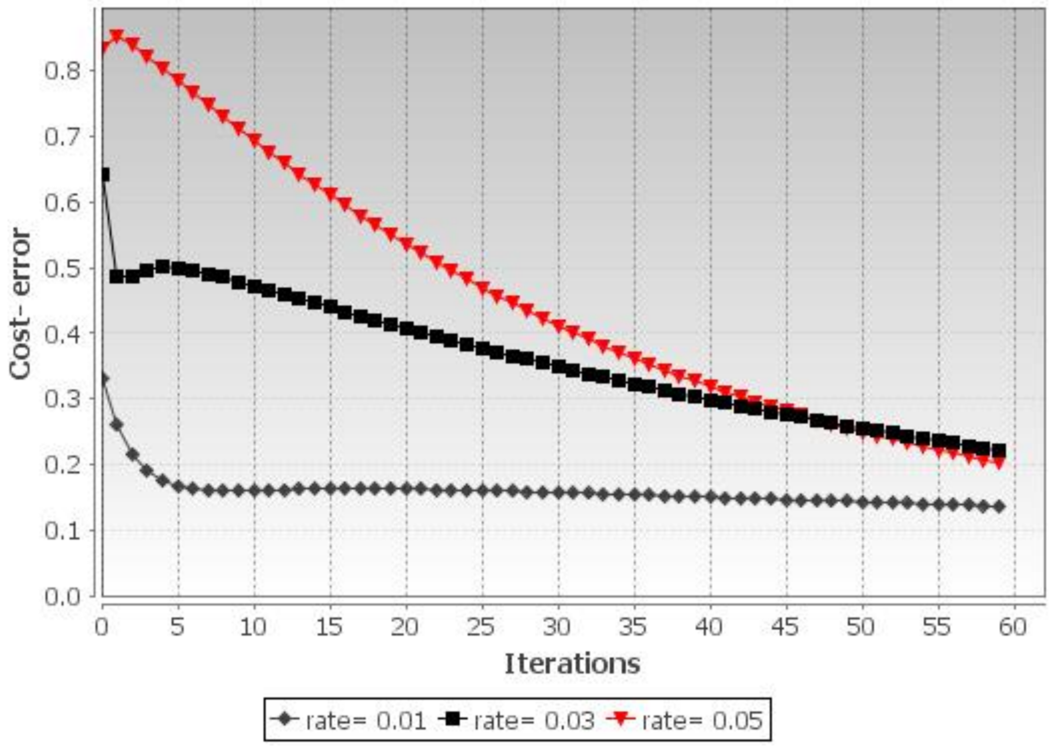

Reinforcement Learning
Markov
Model Based
Iterative Value
Iterative Policy
Temporal Difference
Q-learning
SARSA
Evolution
Rules
Learning Classifiers
XCS
Optimizer
Stochastic Gradient
Genetic Algorithm


40%
20%
0%
-20%
24.37
20.01
15.01
10.01
Volume
281.6M
140.8M
F M A M J J A S O N D 13 F M A M J J A S O N D

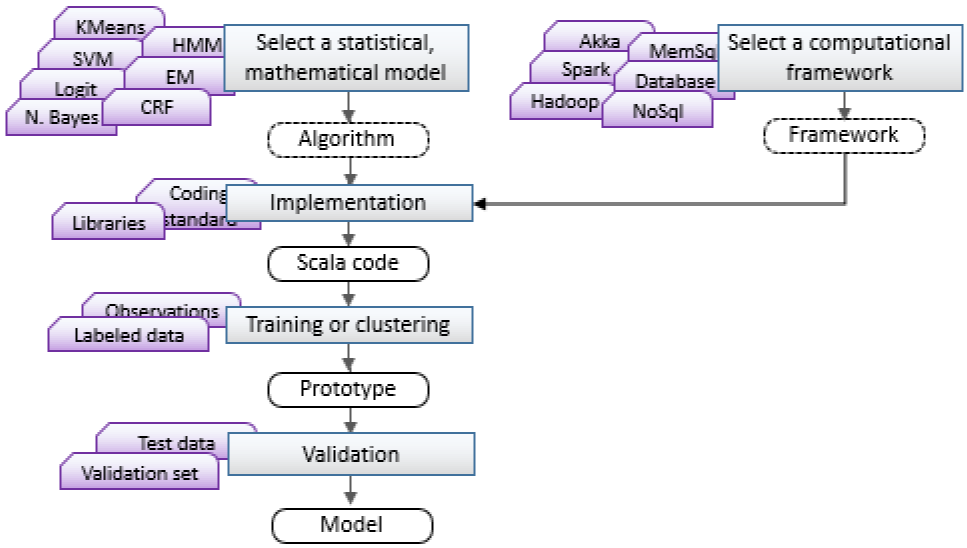

40%
20%
0%
-20%
24.37
20.00
15.00
Volume
254.9M
127.4M
F M A M J J A S O N D 13 F M A M J J A S O N D

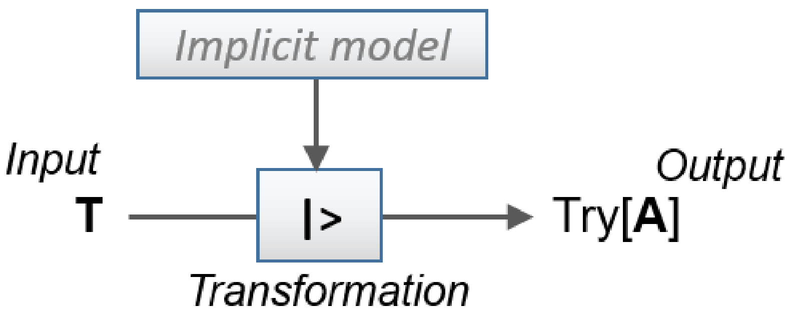

CSCO 2012-13: Model features
Normalized session Volume
Normalized session volatility
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

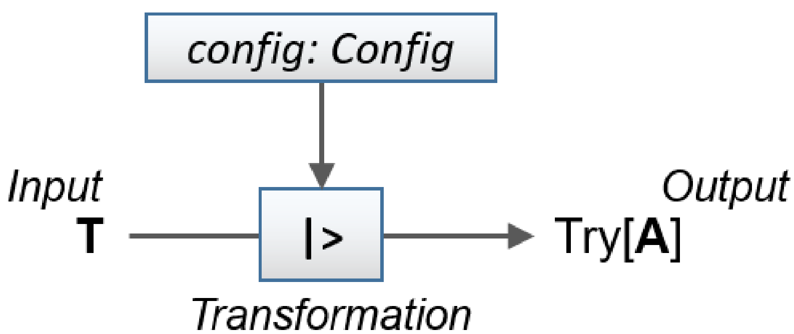

CSCO 2012-13: Training label
Normalized stock price
Trading sessions
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Cost- error
0
5
10
15
20
25
30
35
40
45
50
55
60
Iterations
rate= 0.01
rate= 0.03
rate= 0.05

Chapter 2: Data Pipelines

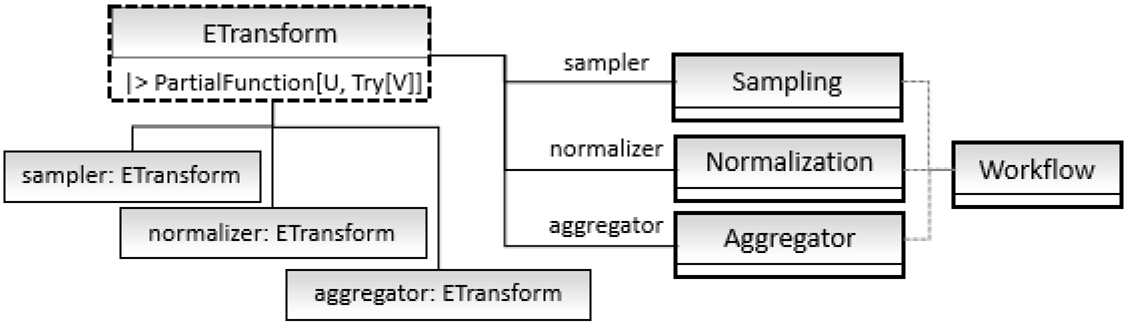


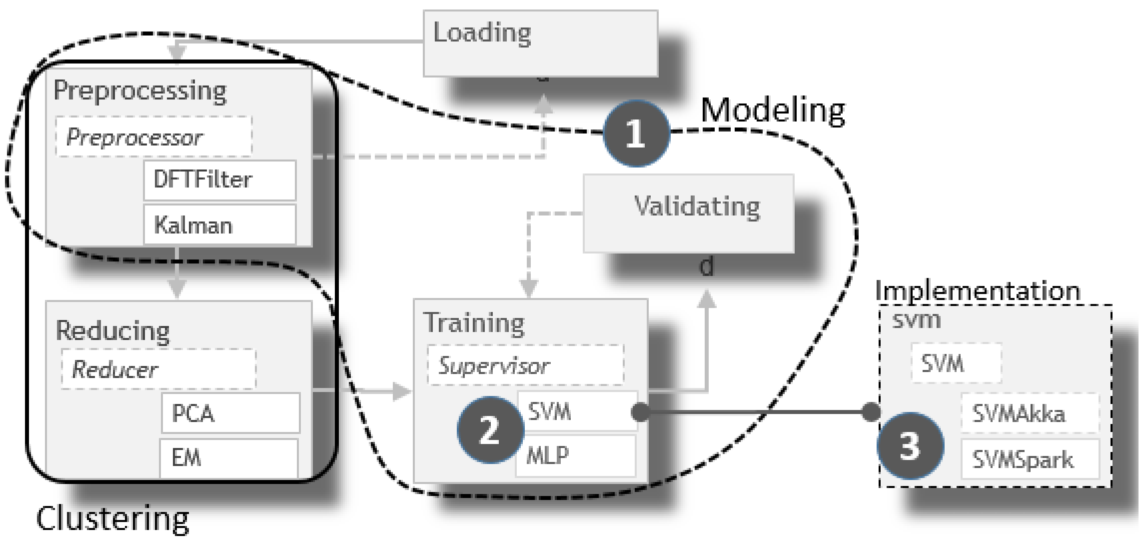




$$f : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad g : \mathbb{R}^n \rightarrow \mathbb{R}$$

$$f : x \to e^x \quad g : x \to \sum_{0}^{n-1} x_i$$





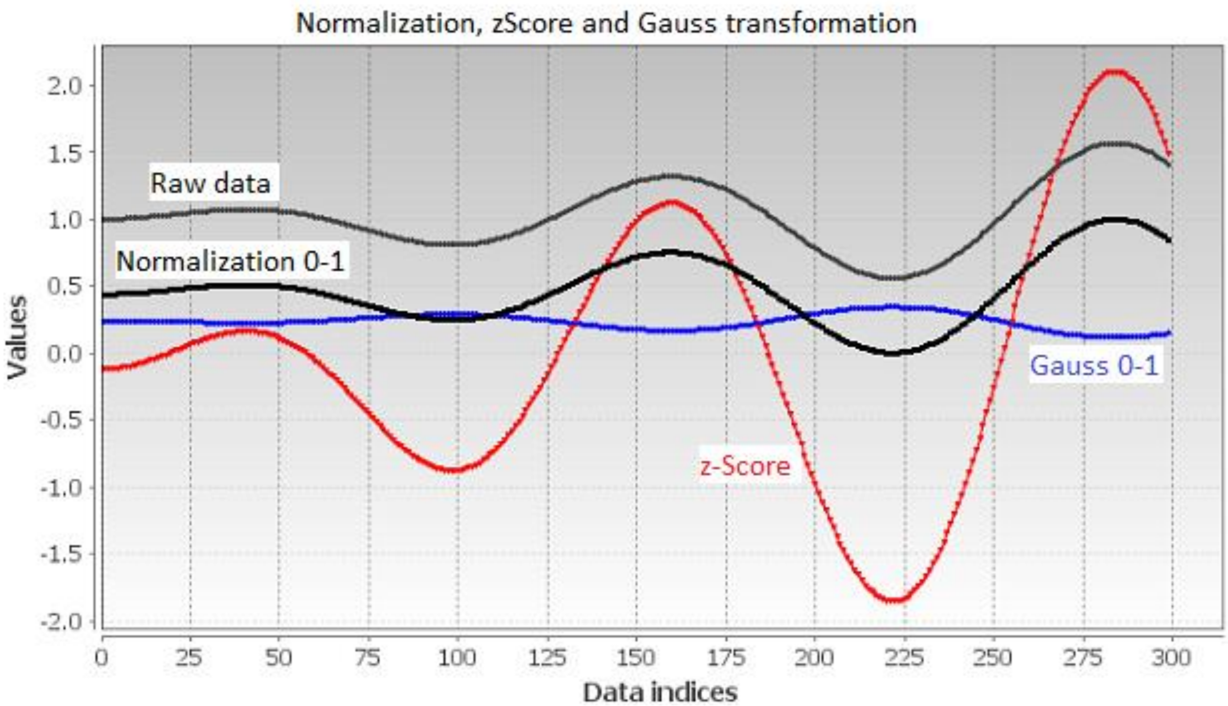
$$E[X] = \mu = \frac{1}{n}\sum_{i=0}^{n-1} x_i$$

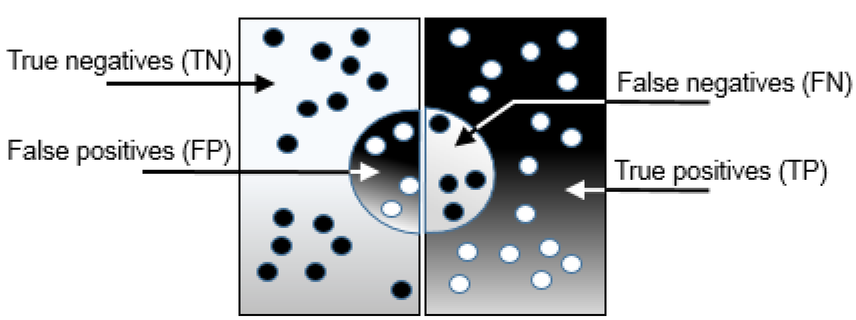
$$Var(X) = \frac{\sum \left(E(X) - x_j\right)^2}{n-1}$$

$$\overline{Var}(X) = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - E[X]\right)^2$$

$$y = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

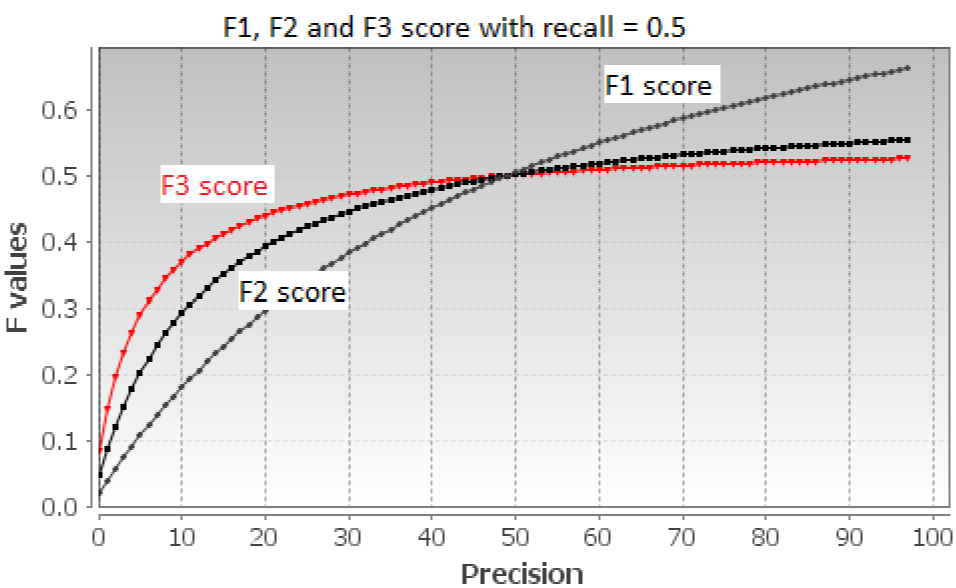
$$z_i = \frac{x_i - \mu}{\sigma}$$





$$ac = \frac{tp + tn}{tp + tn + fp + fn} \qquad p = \frac{tp}{tp + fp} \qquad r = \frac{tp}{tp + fn}$$

$$F_1 = \frac{2pr}{p + r} \qquad F_n = \frac{(1 + n^2)\,pr}{n^2 p + r} \qquad G = \sqrt{pr}$$



$$p^* = \frac{1}{c}\sum_{i=0}^{c-1}\frac{tp_i}{tp_i + fp_i} \qquad r^* = \frac{1}{c}\sum_{i=0}^{c-1}\frac{tp_i}{tp_i + fn_i}$$

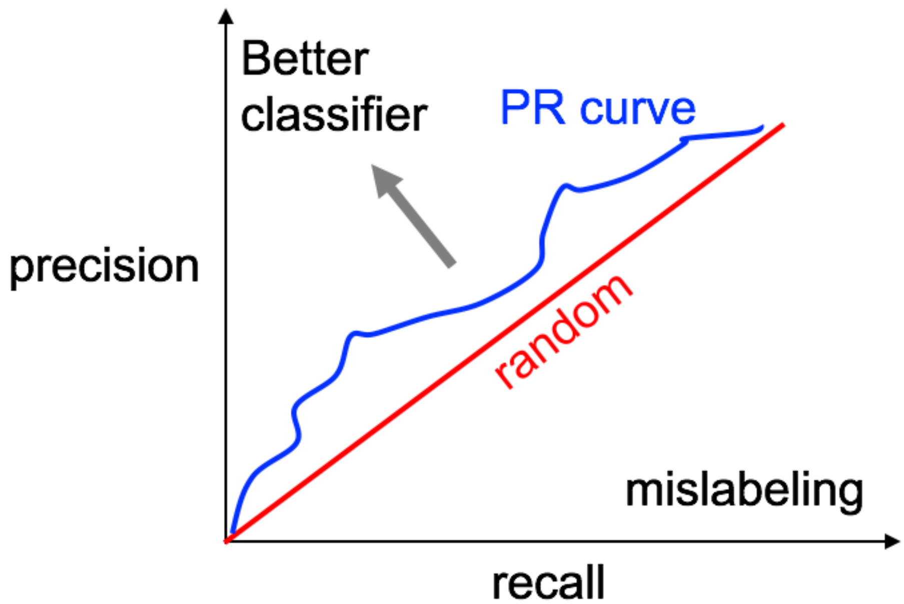
True positive

Actual

Predicted / classes	1	2	3	4	5	6
1	167	3	19	8	0	2
2	11	107	3	27	4	12
3	4	21	145	3	7	14
4	9	17	4	179	20	0
5	15	0	18	2	139	8
6	1	6	0	24	8	164

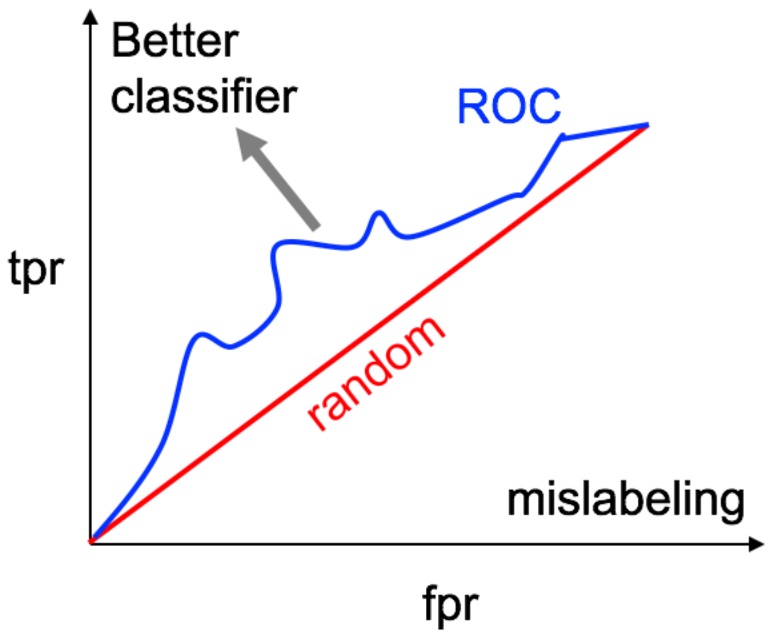
False positives

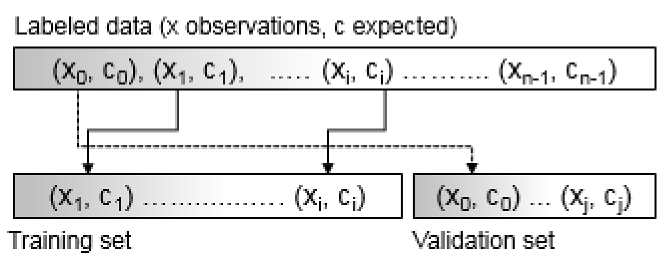
False negatives

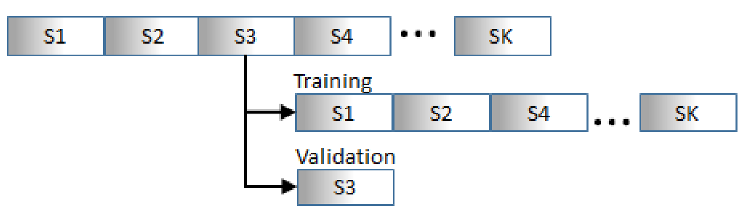


$$auPRC = 0.5 + \frac{1}{N}\sum_{i=0}^{N-1}\left(p_i - r_i\right)$$

$$trp = \frac{tp}{tp + fn} \quad fpr = \frac{fp}{fp + tn}$$





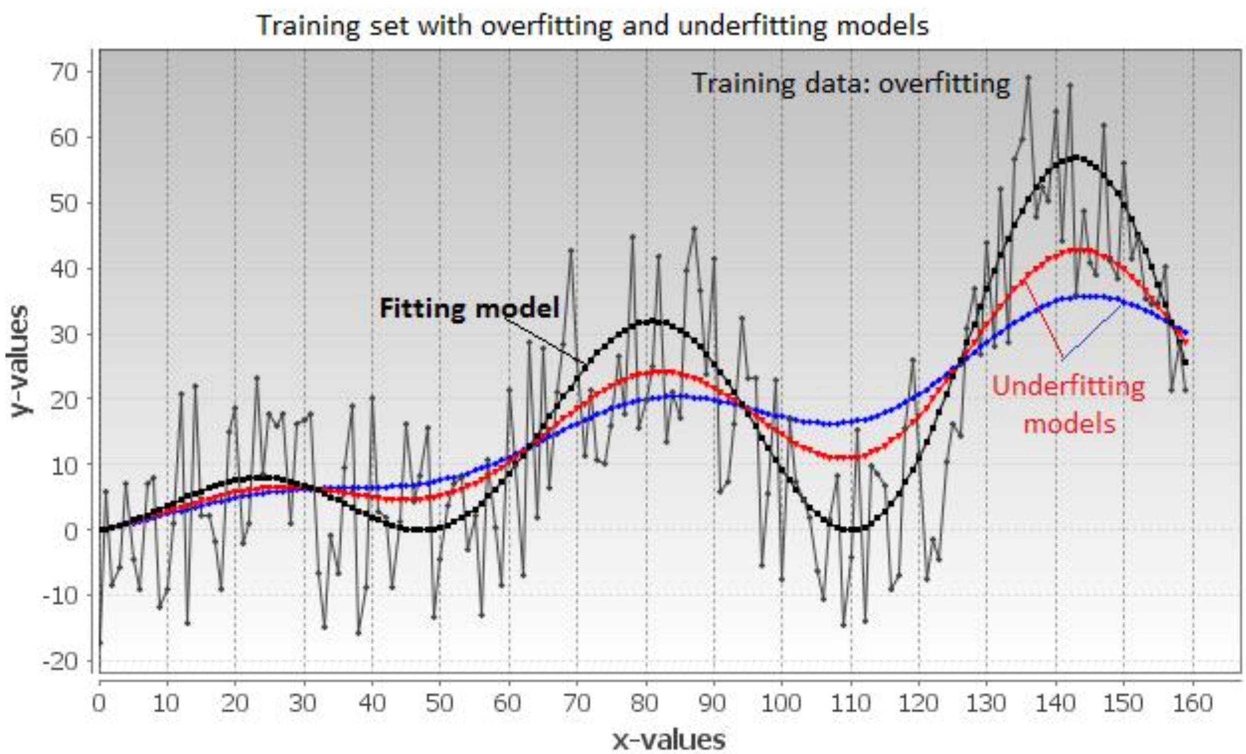


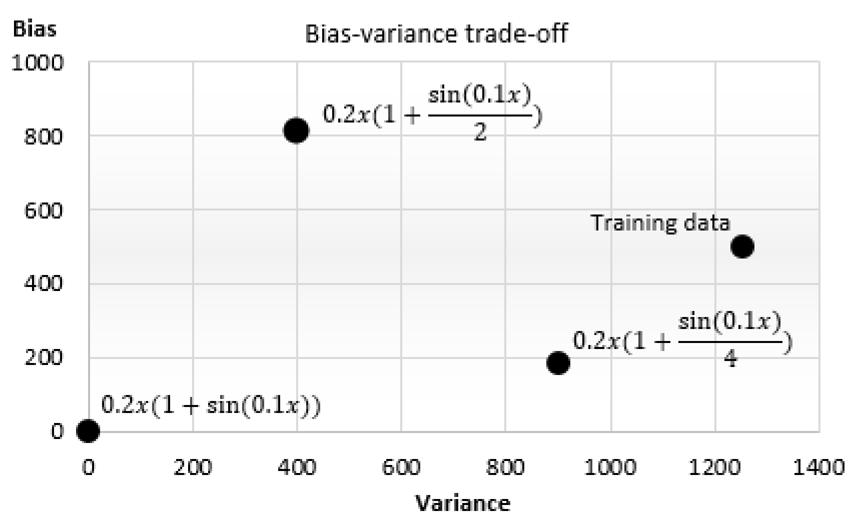
$$var\hat{\theta} = E\left[\left(\hat{\theta} - E\left[\hat{\theta}\right]\right)^2\right] \quad bias\hat{\theta} = \hat{\theta} - \theta$$

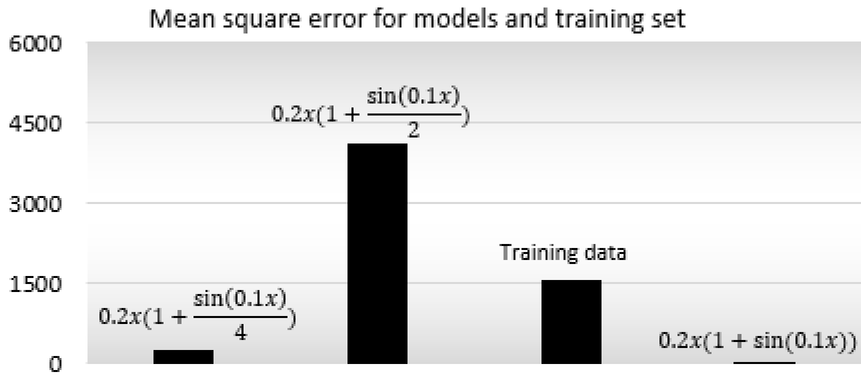
$$MSE = var\left(\tilde{\theta}\right) + bias\left(\tilde{\theta}\right)$$

$$y = \frac{x}{5}\left(1 + \frac{1}{n}\sin\left(\frac{x}{10} + r1\right)\right) + r2$$

$$y = \frac{x}{5}\left(1 + \frac{1}{n}\sin\left(\frac{x}{10}\right)\right)$$

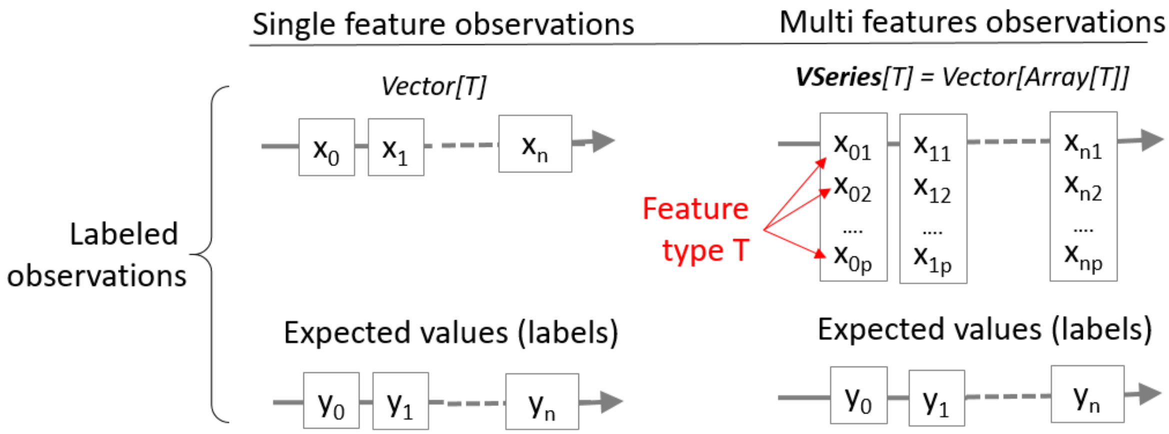






Mean square error for models and training set
6000
4500
3000
1500
0
$0.2x(1+\frac{\sin(0.1x)}{2})$
Training data
$0.2x(1+\frac{\sin(0.1x)}{4})$
$0.2x(1+\sin(0.1x))$

Chapter 3: Data Pre-processing

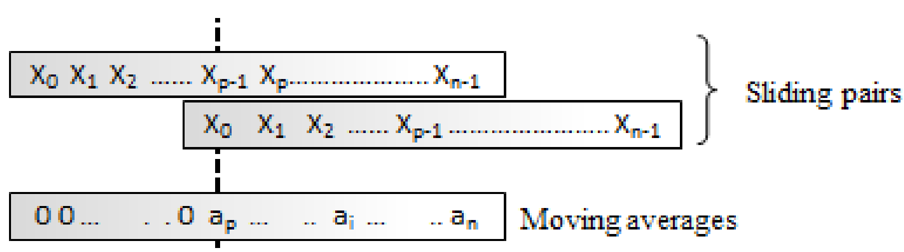


$$\tilde{x}_t = f\left(x_{t-p+1}, \ldots, x_t\right) \forall t \geq p$$

$$\tilde{x}_t = \frac{1}{p} \sum_{j=t-p+1}^{t} x_j \quad \forall t \geq p$$

$$0 \ \forall t < p$$

$$\tilde{x}_t = \tilde{x}_{t-1} + \frac{1}{p}\left(x_t - x_{t-p}\right) \quad \forall t \geq p$$

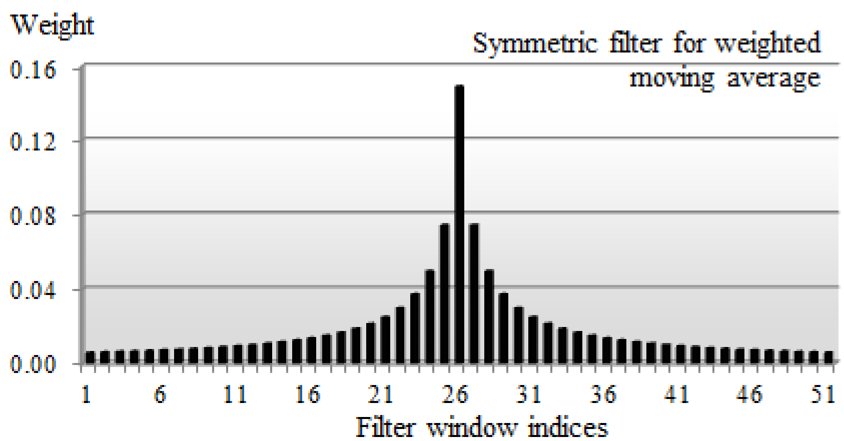


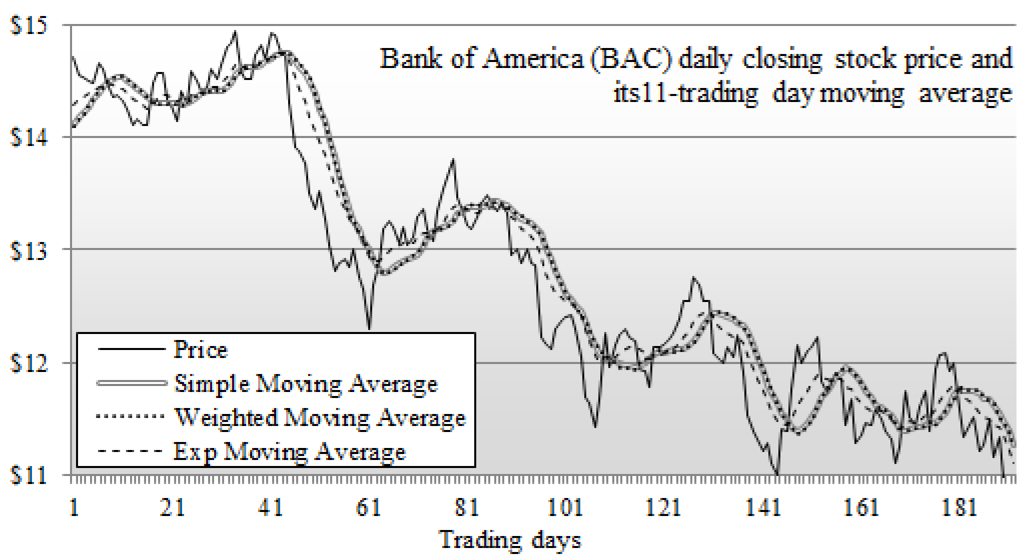
$$\tilde{x}_t = \sum_{j=t-p+1}^{t} \alpha_{j-t+p} x_j \quad \forall t \geq p \qquad subject\ to \sum_{i=0}^{p-1} \alpha_i = 1$$

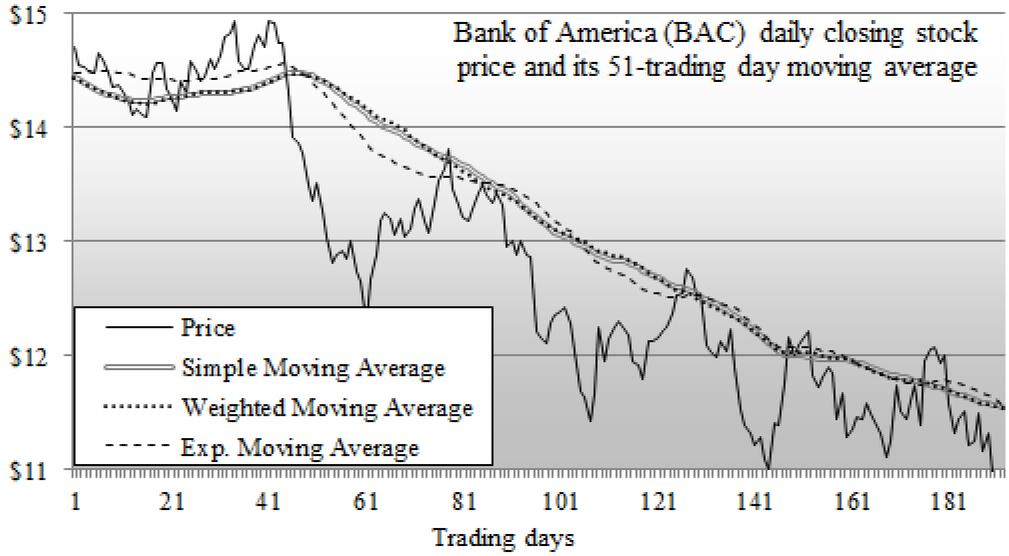
$$0 \ \forall t < p$$

$$\tilde{x}_t = (1-\alpha)\tilde{x}_{t-1} + \alpha x_t \ \ \forall t > 0 \quad 0 < \alpha < 1$$

$$\tilde{x}_0 = x_0$$







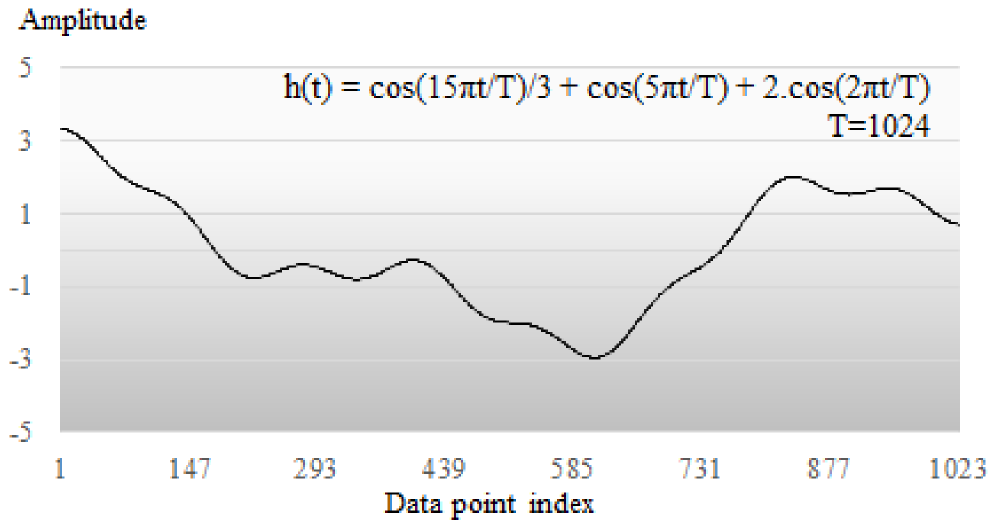
$$f\left(t\right)=\frac{a_0}{2}+\sum_{1}^{\infty}a_k\cos\left(nx\right)+\sum_{1}^{\infty}b_k\sin\left(nx\right)$$

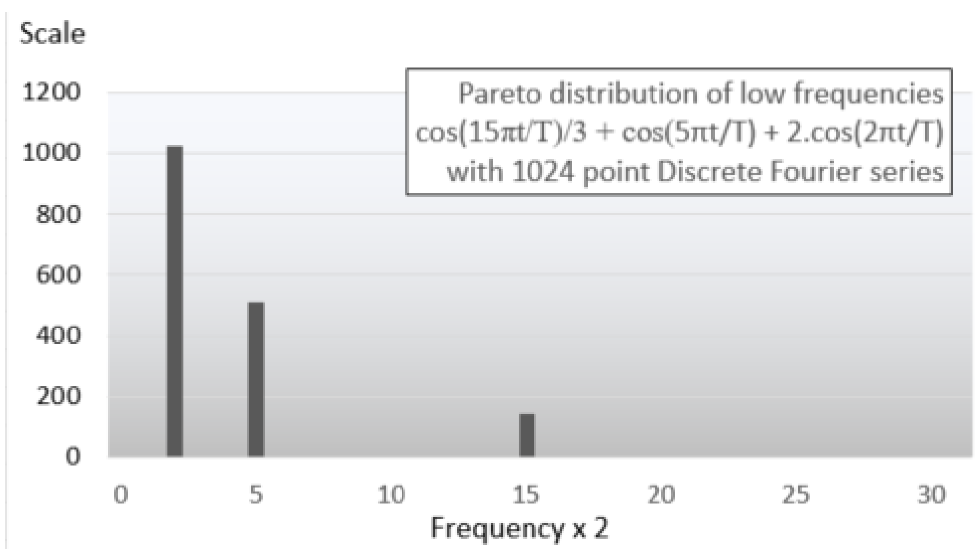
$$\mathcal{F}^c\left(f,k\right)=\int_{-\infty}^{\infty}\cos\left(2\pi k\pi\right)f\left(x\right)dx$$

$$f\left(x\right)=f\left(-x\right)=\frac{a_0}{2}+\sum_{k=1}^{2N-3}a_k cos\left(kx\right)\ \ where\, a_k=\frac{2}{\pi}\int_{0}^{\pi}f\left(t\right)\cos\left(kt\right).dt$$

$$\mathcal{F}^s\left(f,k\right)=\int_{-\infty}^{\infty}\sin\left(2\pi kx\right)f\left(x\right)dx$$

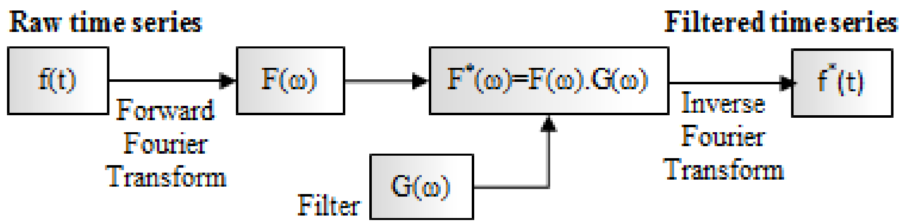
$$f(x) = -f(-x) = \sum_{k=1}^{2N-3} b_k \sin(kx) \ \ where\, b_k \frac{2}{\pi}\int_0^{\pi} f(t)\sin(kt).dt$$

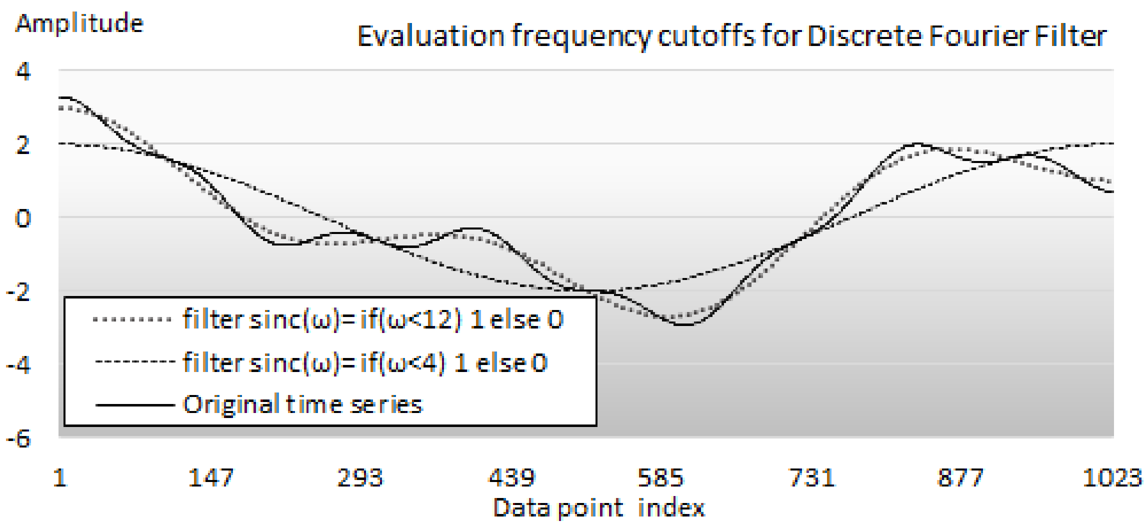


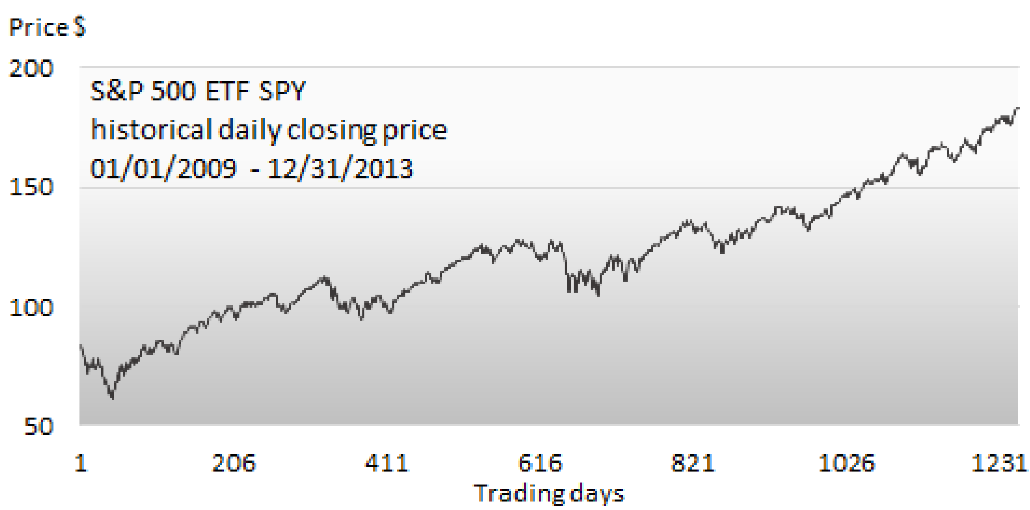


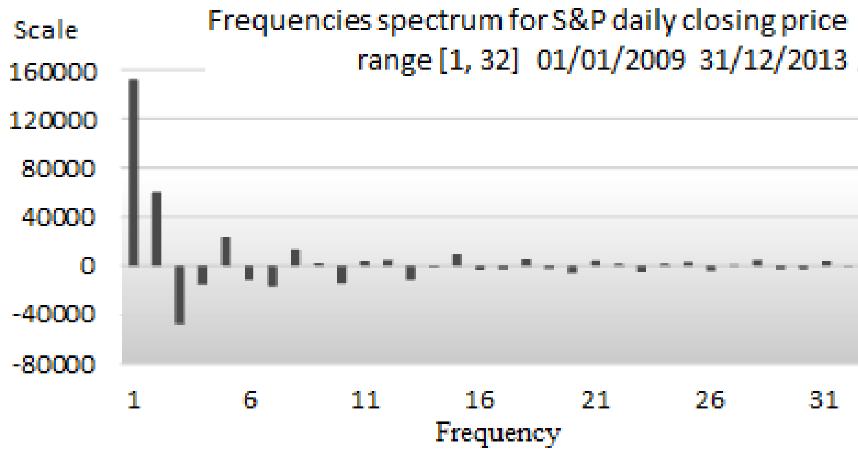
$$< f, g >= \int_{-\infty}^{\pi} f(t).g(x-t)\,dt$$

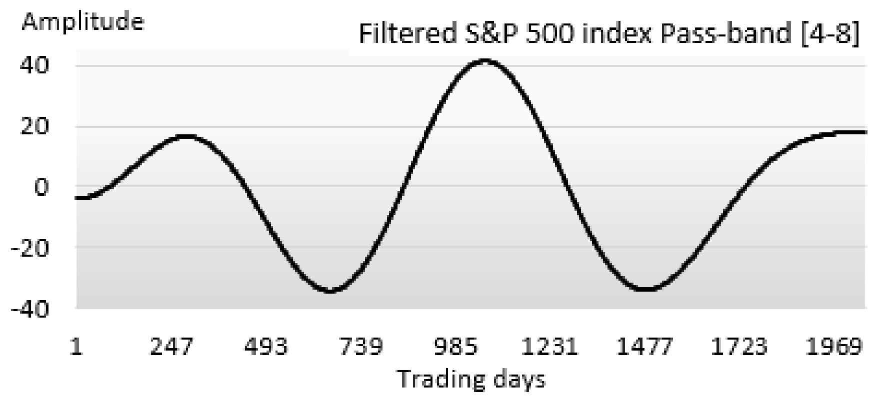
$$F\left(x*f\right)=F\left(x\right).F\left(g\right)=\sum_{j=0}^{N-1}\omega_j^x.\omega_{k-j}^f$$

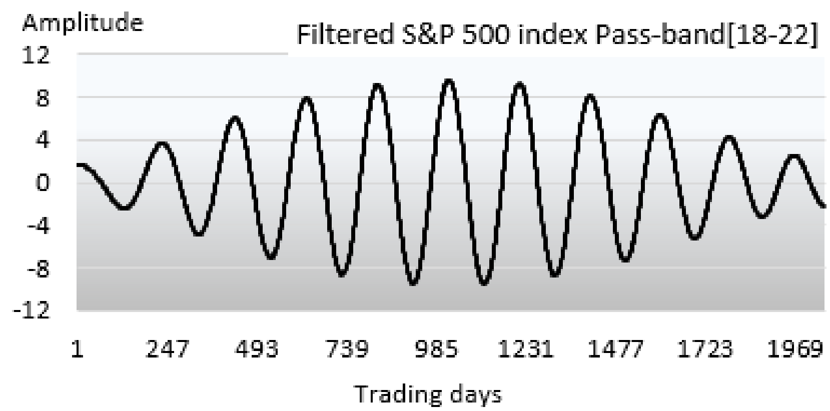


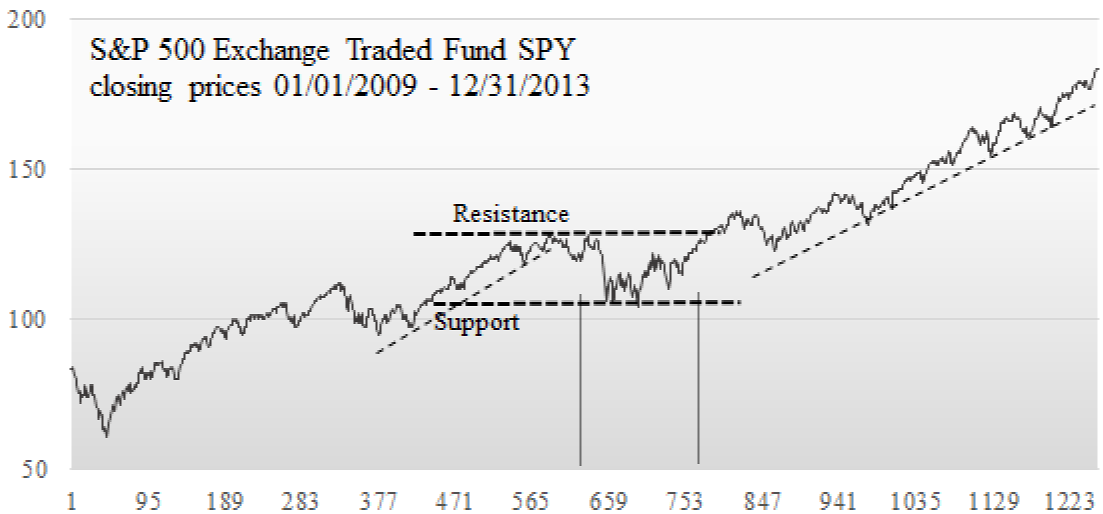





Scale
Frequencies spectrum for S&P daily closing price
range [1, 32] 01/01/2009 31/12/2013
160000
120000
80000
40000
0
-40000
-80000
1
6
11
16
21
26
31
Frequency

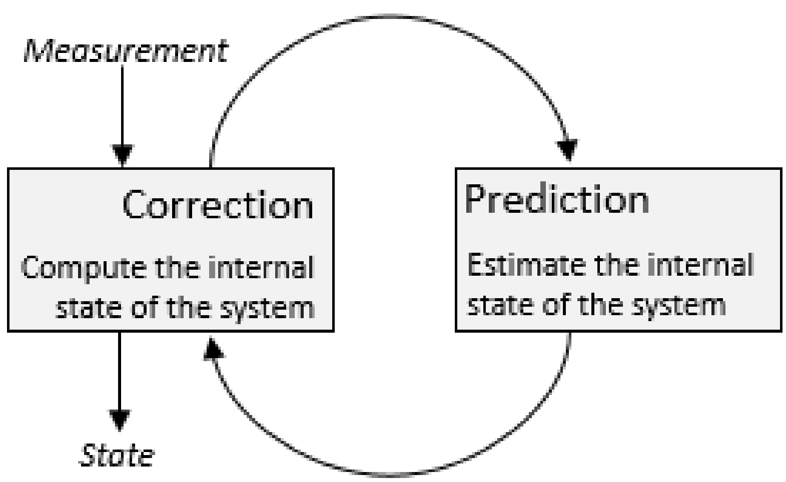

Amplitude
Filtered S&P 500 index Pass-band [4-8]
40
20
0
-20
-40
1
247
493
739
985
1231
1477
1723
1969
Trading days


Amplitude
Filtered S&P 500 index Pass-band[18-22]
12
8
4
0
-4
-8
-12
1
247
493
739
985
1231
1477
1723
1969
Trading days



$$x_t = A_t.x_{t-1} + B_t.u_t + w_t$$

$$z_t = H_t.x_t + v_t$$



$$\begin{bmatrix} \hat{x}_t \\ \hat{x}_{t-1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_t \\ u_{t-1} \end{bmatrix} + \begin{bmatrix} w_t \\ w_{t-1} \end{bmatrix}$$

$$\hat{x}_t' = A_t.\hat{x}_{t-1} + B_t.u_t$$

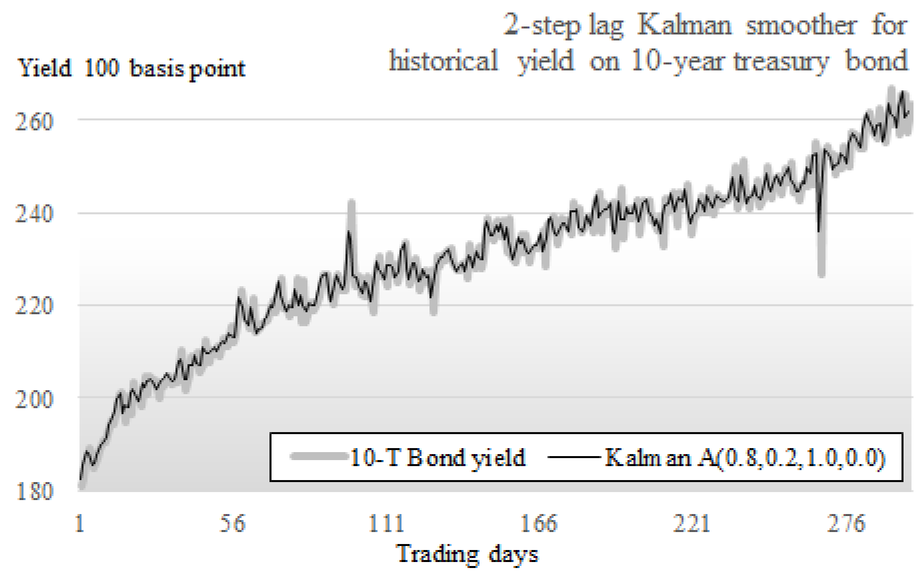
$$P_t' = A_t.P_{t-1} + A_t^T + Q_t$$

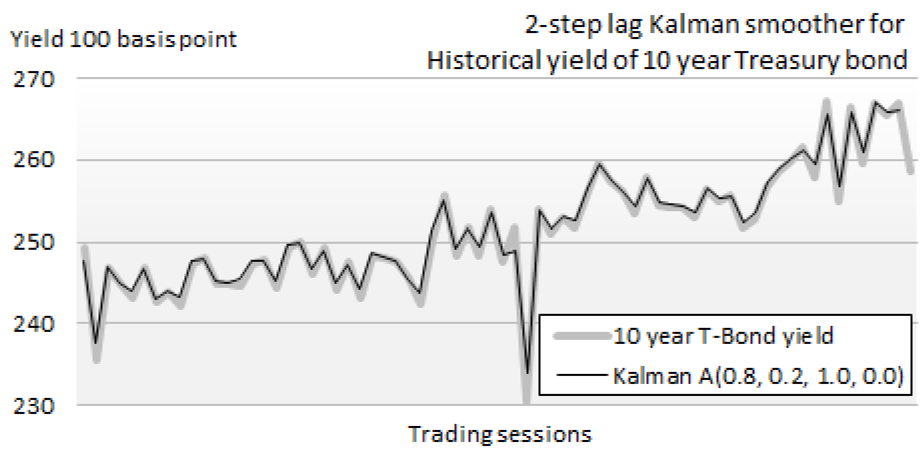
$$\begin{bmatrix} z_t \\ z_{t-1} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \cdot \begin{bmatrix} \hat{x}_t \\ \hat{x}_{t-1} \end{bmatrix} + \begin{bmatrix} v_t \\ v_{t-1} \end{bmatrix}$$

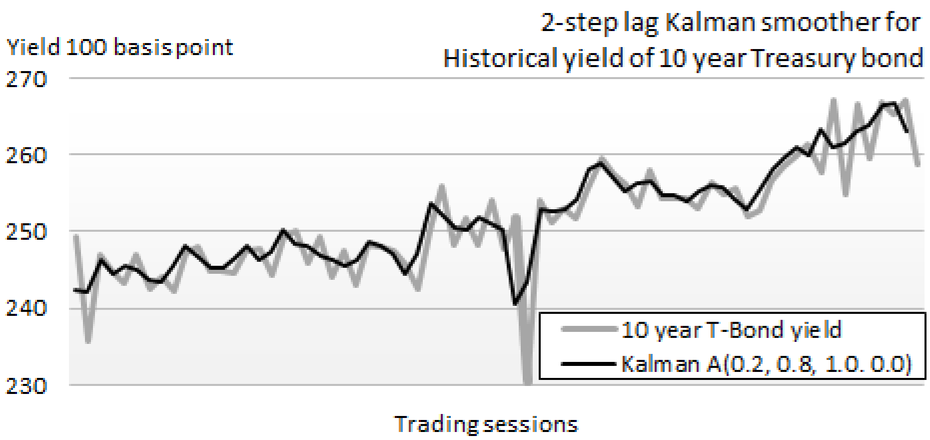
$$\hat{x}_t = \hat{x}_t' + K_t\left(z_t - H_t.\hat{x}_t'\right) \quad r_t = z_t - H_t.\hat{x}_t'$$

$$K_t = P_t'.H_t^T\left(H_t.P_t'.H_t^T + R_t\right)^{-1}$$

$$S_t = \left[x_{t+1}, x_t\right]^T \ with \begin{vmatrix} x_{t+1} \\ x_t \end{vmatrix} = \begin{vmatrix} \alpha & 1-\alpha \\ 1 & 0 \end{vmatrix} \cdot \begin{vmatrix} x_t \\ x_{t-1} \end{vmatrix}$$


2-step lag Kalman smoother for
historical yield on 10-year treasury bond
Yield 100 basis point
260
240
220
200
180
10-T Bond yield
Kalman A(0.8,0.2,1.0,0.0)
1
56
111
166
221
276
Trading days


Yield 100 basis point
2-step lag Kalman smoother for
Historical yield of 10 year Treasury bond
270
260
250
240
230
10 year T-Bond yield
Kalman A(0.8, 0.2, 1.0, 0.0)
Trading sessions



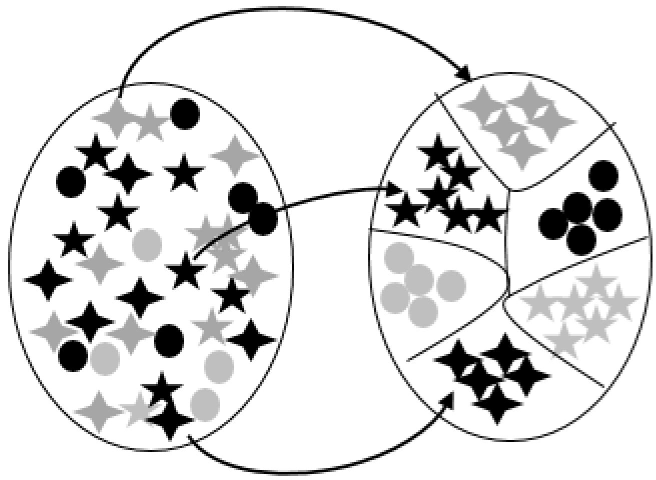
$$\left\{x_{t+1}-x_t\right\}$$

$$\left\{x_t\right\}$$

$$\tilde{x}_t$$

$$\tilde{x}$$

Chapter 4: Unsupervised Learning

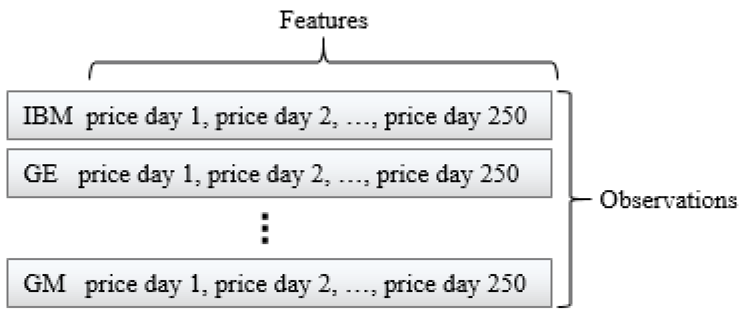


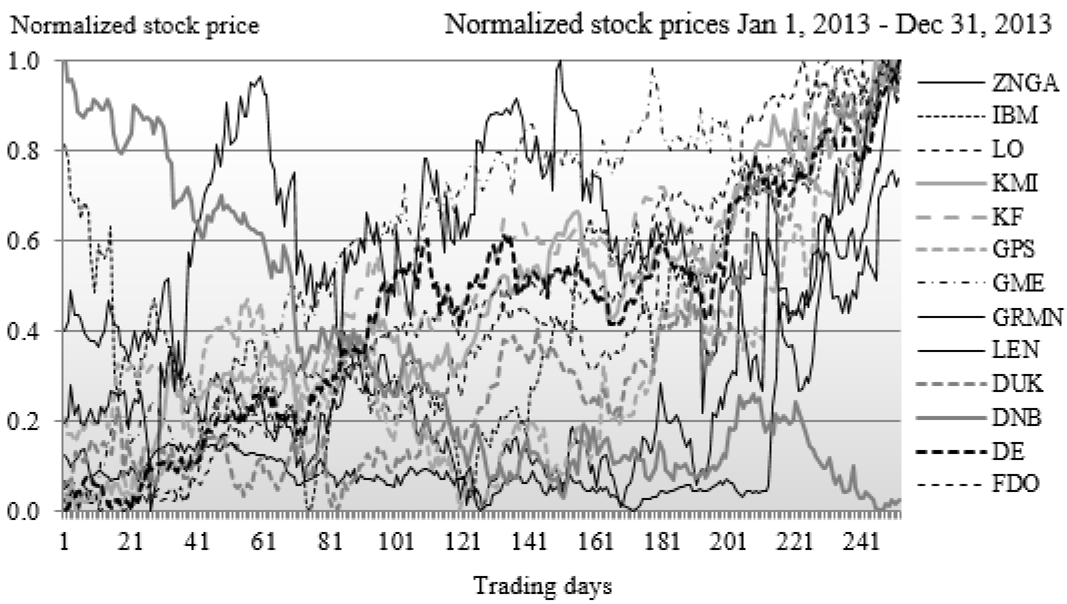
$$d(x,y) = \sum |x_i - y_i|$$

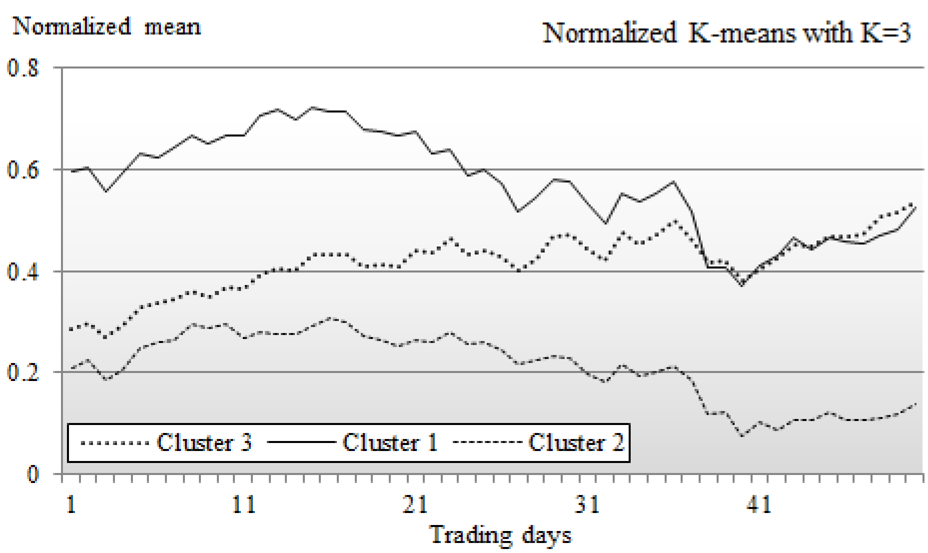
$$d(x,y) = \sum (x_i - y_i)^2$$

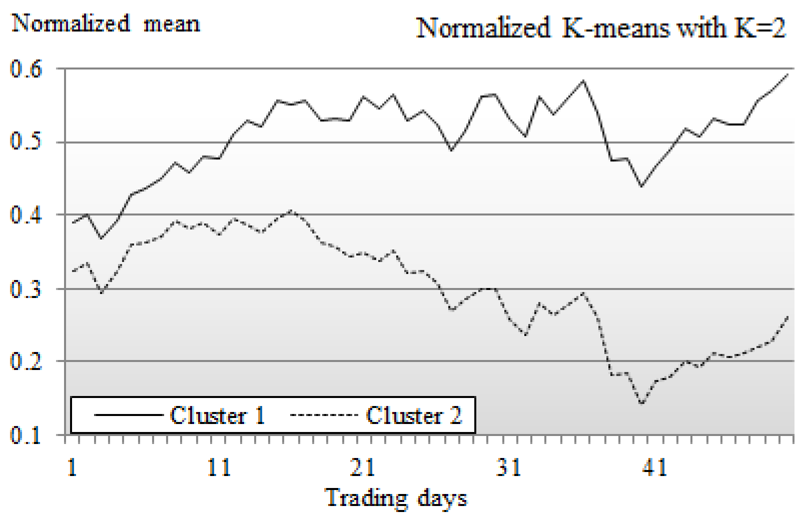
$$d(x,y) = \frac{\sum x_i y_i}{\left(\sum x_i^2 \sum y_i^2\right)^{1/2}}$$

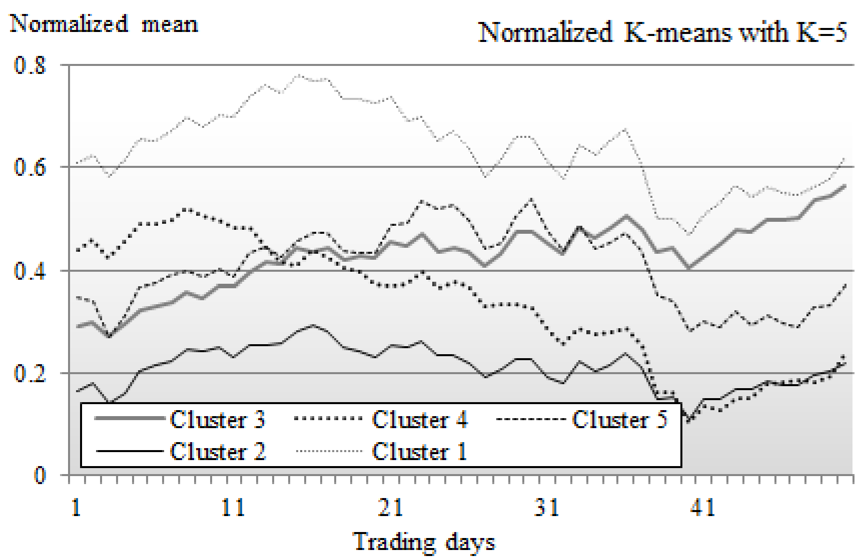
$$\min_{C_k} \sum_{1}^{K} \sum_{x_{i \in C_k}} d(x_i, m_k)$$

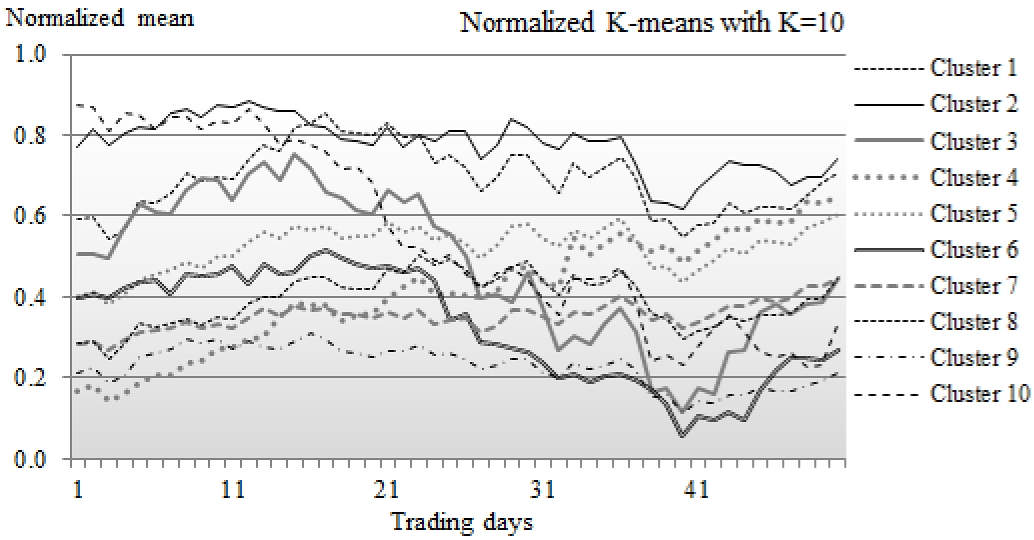

Features
IBM price day 1, price day 2, …, price day 250
GE price day 1, price day 2, …, price day 250
GM price day 1, price day 2, …, price day 250
Observations


Normalized stock price
Normalized stock prices Jan 1, 2013 - Dec 31, 2013
1.0
0.8
0.6
0.4
0.2
0.0
1
21
41
61
81
101
121
141
161
181
201
221
241
Trading days
ZNGA
IBM
LO
KMI
KF
GPS
GME
GRMN
LEN
DUK
DNB
DE
FDO

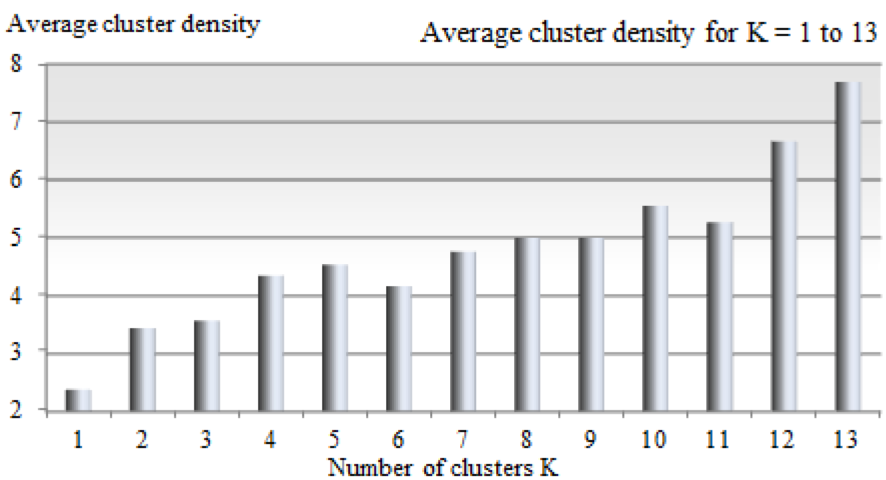

Normalized mean
Normalized K-means with K=3
0.8
0.6
0.4
0.2
0
1
11
21
31
41
Trading days
Cluster 3
Cluster 1
Cluster 2


Normalized mean
Normalized K-means with K=2
0.6
0.5
0.4
0.3
0.2
0.1
Cluster 1
Cluster 2
1
11
21
31
41
Trading days

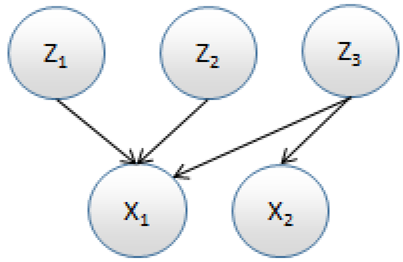

Normalized mean
Normalized K-means with K=5
0.8
0.6
0.4
0.2
0
Cluster 3
Cluster 4
Cluster 5
Cluster 2
Cluster 1
1
11
21
31
41
Trading days



$$d\left(C_j\right)=1\Big/\sum_{x\in C_j}\left(x-c_j\right)^2$$



$$H\left(p\right)=-\sum_{c}p_c.log_2 p_c$$



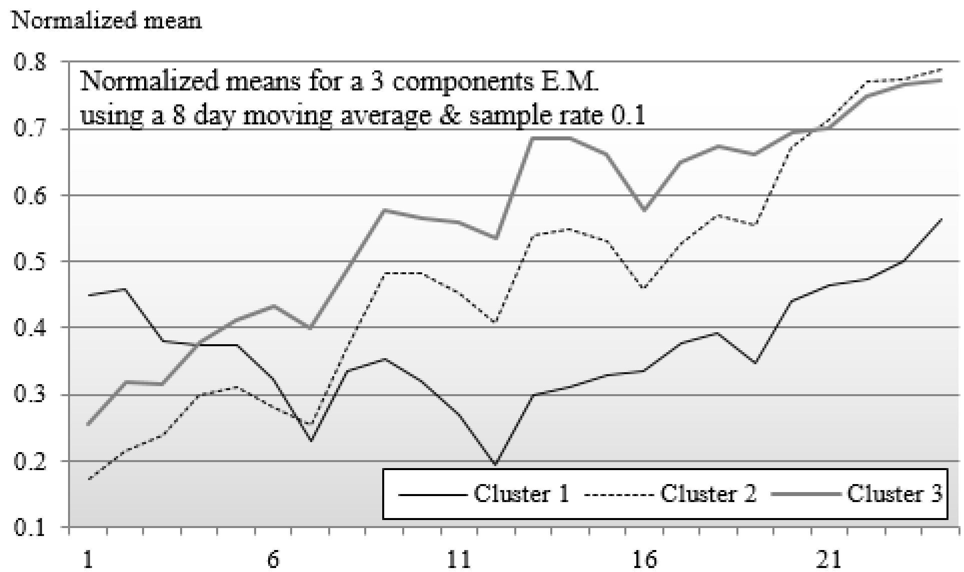
$$p(x_i \mid \theta) = -\sum_z p(x_i, z \mid \theta)$$

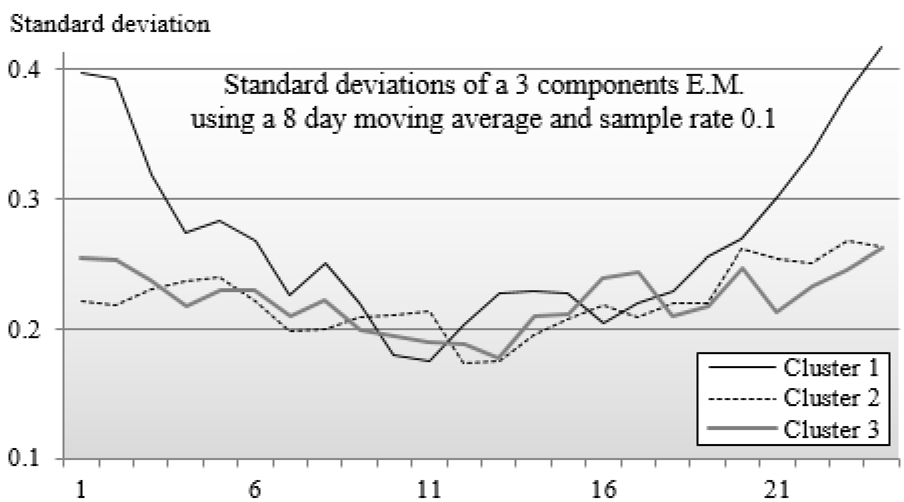
$$\mathcal{L}(\theta) = \sum_{i-0}^{N-1} log \left\{ \sum_z p(x_i, z \mid \theta) \right\} \quad \tilde{\theta} = argmax \, \mathcal{L}(\theta)$$

$$\mathbb{Q}(\theta, \theta^n) = \sum_z p(z \mid x_i, \theta^n) . \log p(x_i, z \mid \theta)$$

$$\mathbf{Q}$$

$$\theta^{n+1} = \arg\max \mathbb{Q}(\theta, \theta^n)_{\theta} \quad \left| \theta^{n+1} - \theta^n \right| < \varepsilon$$

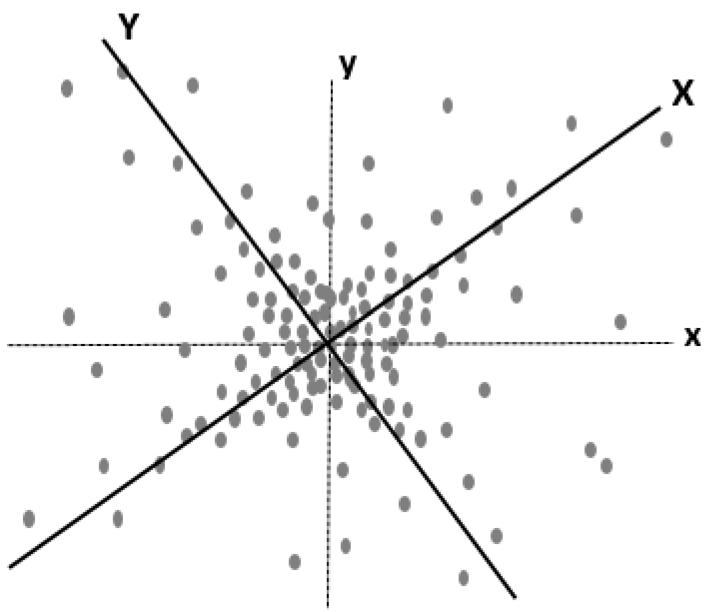

Normalized mean
Normalized means for a 3 components E.M.
using a 8 day moving average & sample rate 0.1
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1
6
11
16
21
Cluster 1
Cluster 2
Cluster 3


Standard deviation
Standard deviations of a 3 components E.M.
using a 8 day moving average and sample rate 0.1
0.4
0.3
0.2
0.1
1
6
11
16
21
Cluster 1
Cluster 2
Cluster 3

Chapter 5: Dimension Reduction

$$D_{KL}\left(P\|Q\right)=\sum_{i=0}^{n-1}p\left(x_i\right).\log\frac{p\left(x_i\right)}{q\left(x_i\right)}$$

$$I\left(X;Y\right)=D_{kl}\left[p\left(X,Y\right),p\left(X\right)p\left(Y\right)\right]=\sum_{i=0}^{n-1}p\left(x_i,y_i\right)\log\frac{p\left(x_i,y_i\right)}{p\left(x_i\right).p\left(y_i\right)}$$



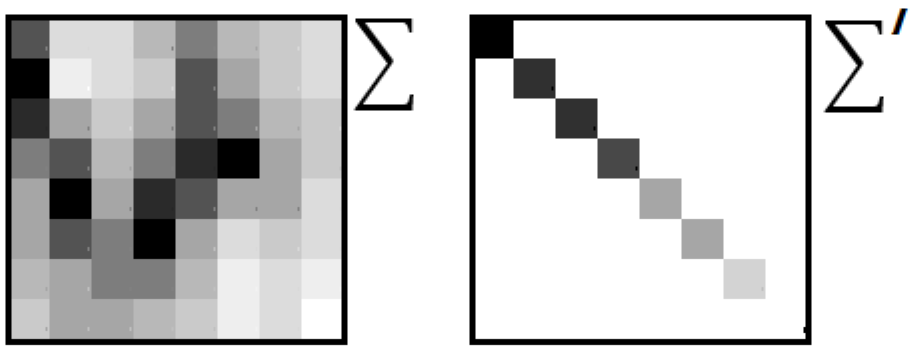
$$M=\left\{f_{i:1,m}\left|\sum_{i=1}^{m}\sigma^2\left(f_i\right)<\mu\right.\right\}$$

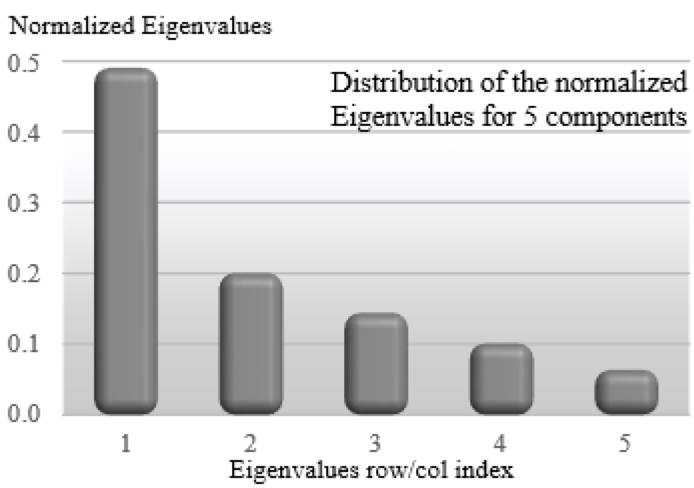
$$cov\left(X,Y\right)=\frac{1}{n-1}\sum_{i=0}^{n-1}\left(x_i-\overline{x}\right)\left(y_i-\overline{y}\right)$$

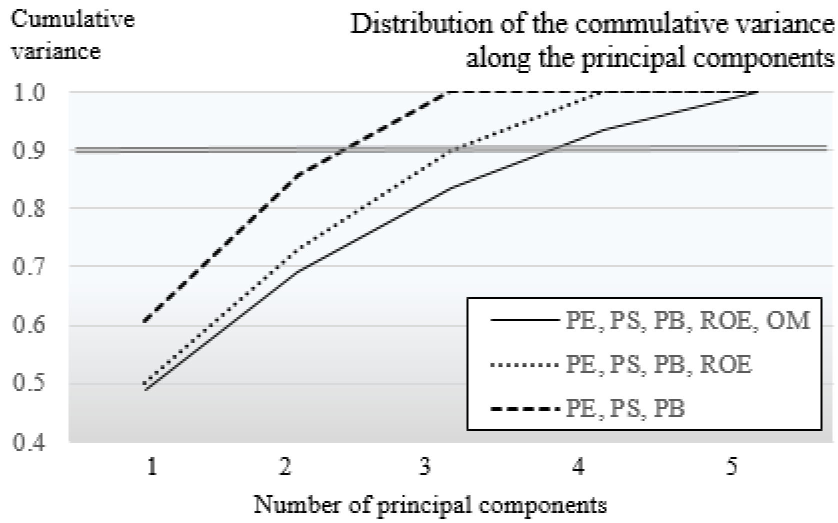
$$x_i\leftarrow\frac{x_i-\overline{x}}{\sigma}$$

$$\sum_{cov} = \begin{vmatrix} cov(x_0, x_0) & \dots & cov(x_0, x_{n-1}) \\ \dots & \mathrm{var}(x_i) & \dots \\ cov(x_{n-1}, x_0) & \dots & cov(x_{n-1}, x_{n-1}) \end{vmatrix}$$

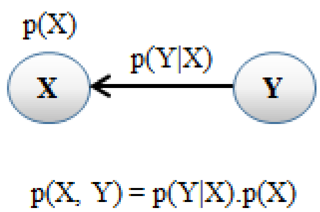
$$\sum{}' = W^T . \sum_{cov} . W$$

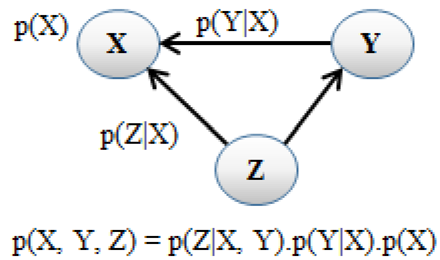


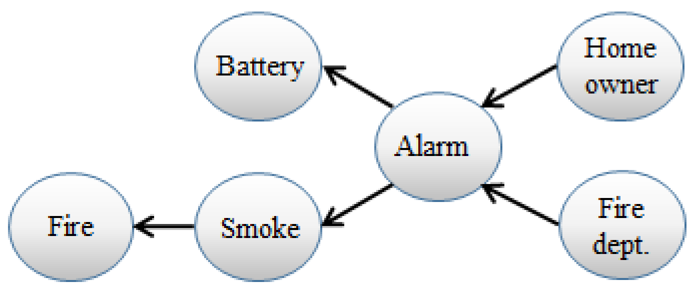


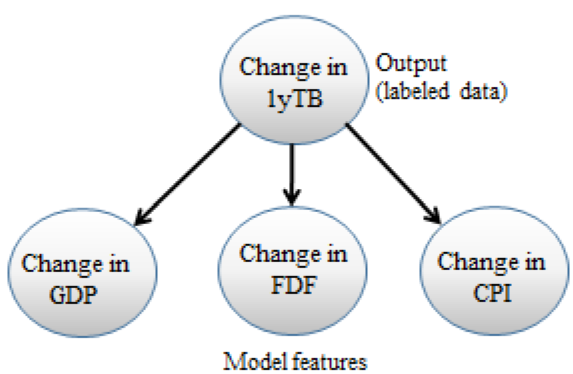

Cumulative variance
Distribution of the commulative variance along the principal components
1.0
0.9
0.8
0.7
0.6
0.5
0.4
PE, PS, PB, ROE, OM
PE, PS, PB, ROE
PE, PS, PB
1
2
3
4
5
Number of principal components

Chapter 6: Naïve Bayes Classifiers







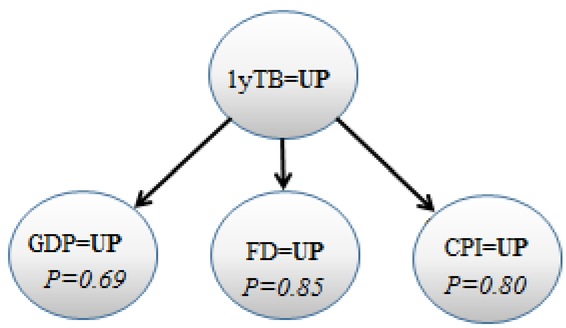


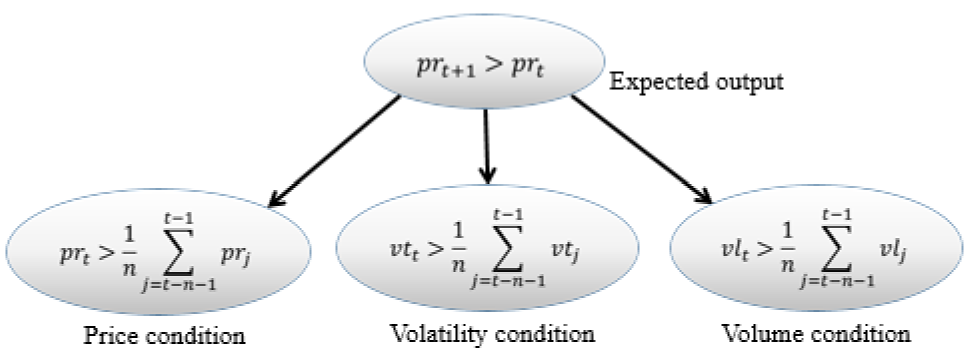
$$p\left(C_j \mid x\right) = \frac{p\left(x \mid C_j\right).p\left(C_j\right)}{p\left(x\right)}$$

$$p\left(C_j \mid x\right) = \prod_{i=0}^{n-1} p\left(x_i \mid C_j\right).p\left(C_j\right)$$

$$\mathcal{L}\left(C_j \mid x\right) = \sum_{i=0}^{n-1} \left\{\log p\left(x_i \mid C_j\right) + \log p\left(C_j\right)\right\}$$

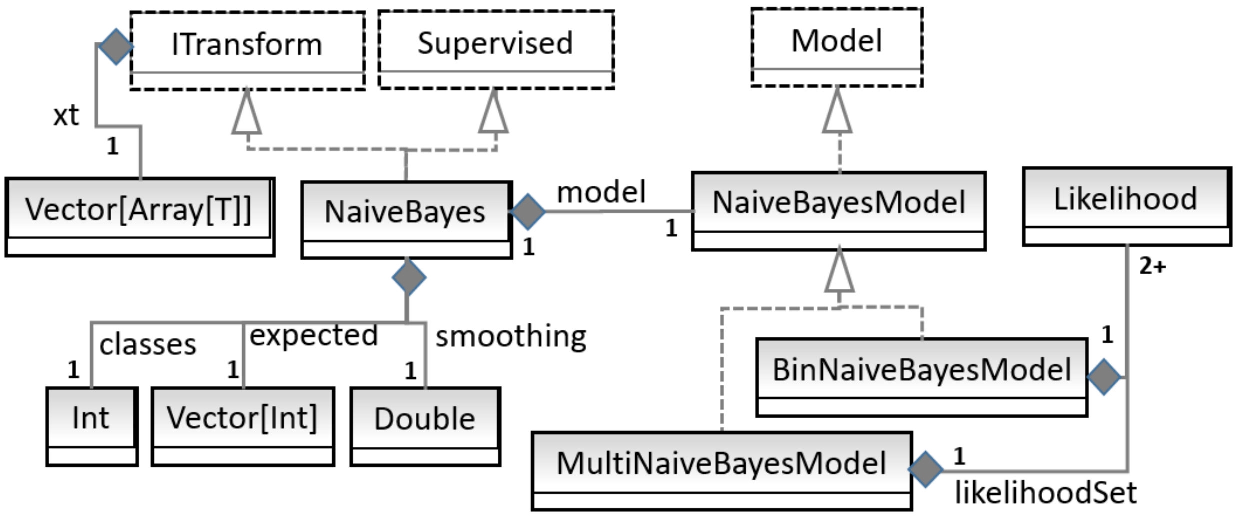
$$C_m = \arg\max_j \mathcal{L}\left(C_j \mid x\right)$$



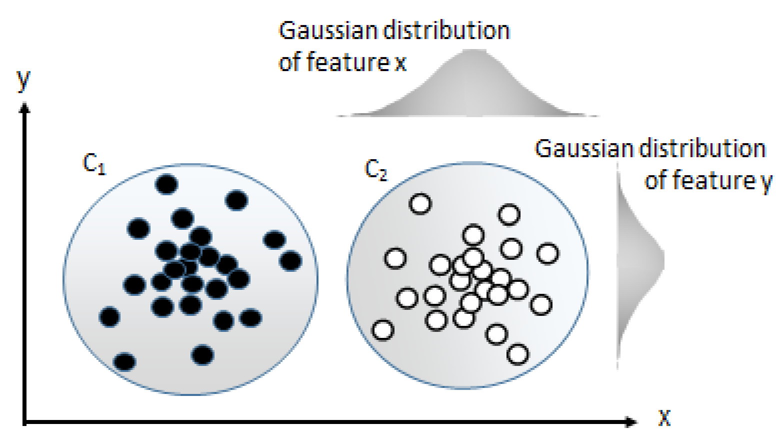


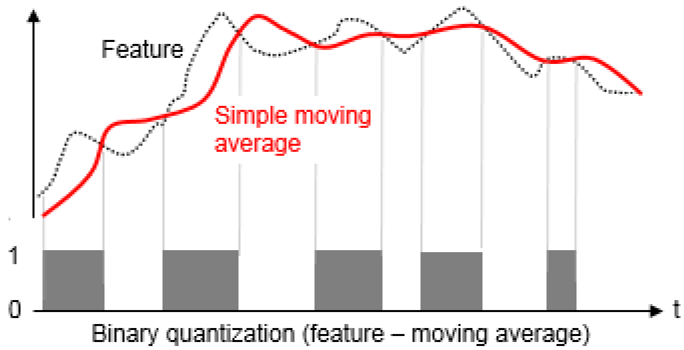
$$\mu' = \frac{k+1}{N+n}$$

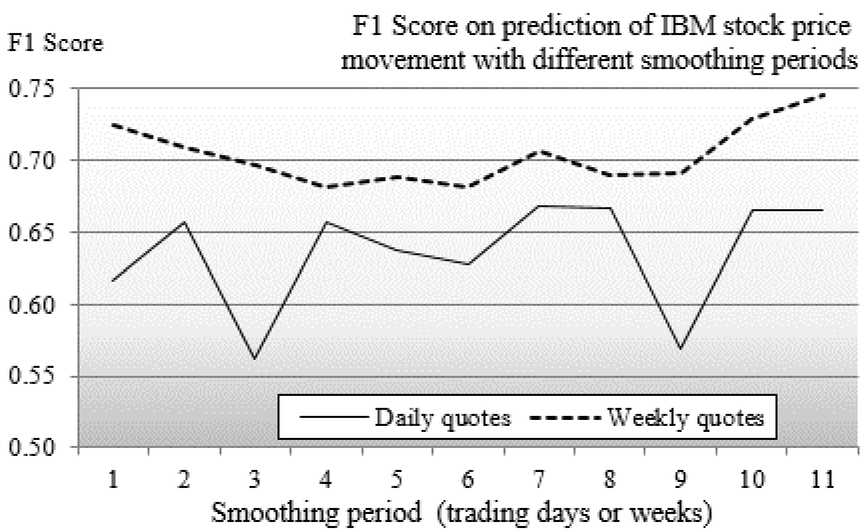
$$\mu' = \frac{k+a}{N+\alpha n}$$



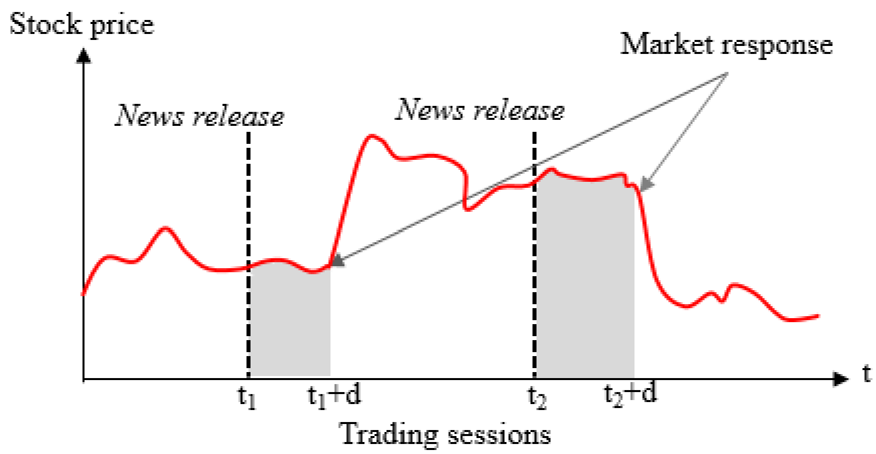
$$\mathcal{L}\left(C_j \mid x\right) = \sum_{i=0}^{n-1}\left[-\frac{1}{2}\log\left(2\pi\right) - log\sigma - \frac{\left(x_i - \mu'\right)^2}{2\sigma^2} + \log p\left(C_j\right)\right]$$

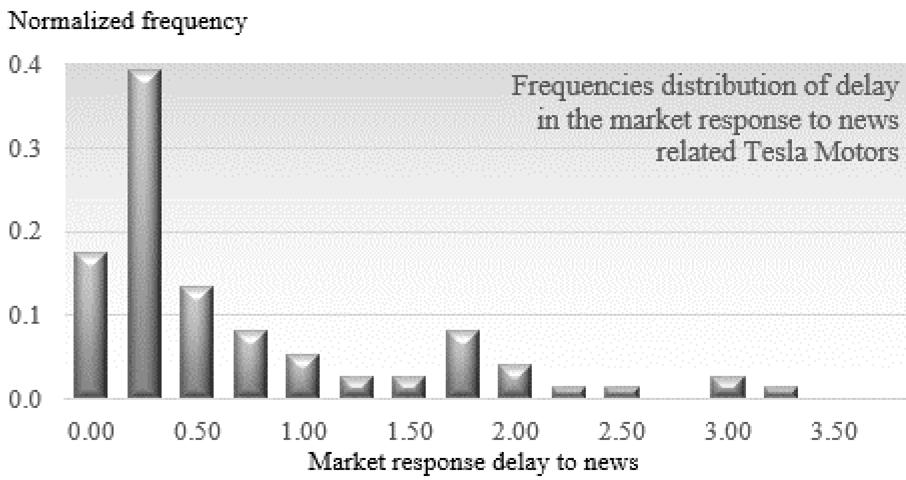


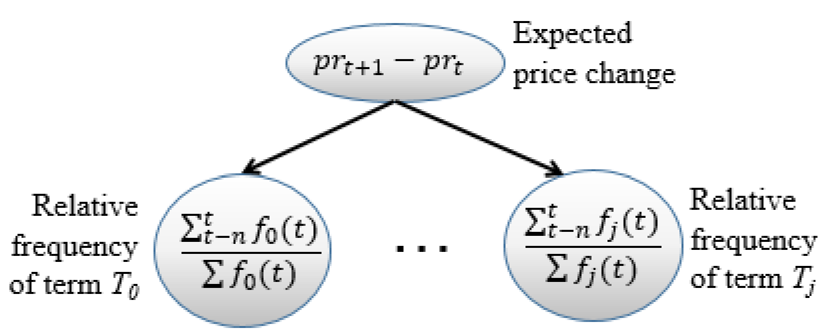




$$p\left(x \mid f, C_j\right) = \prod_{k=0}^{n-1} \left\{ f_k p\left(x_k \mid C_j\right) + \left(1 - f_k\right)\left(1 - p\left(x_k \mid C_j\right)\right) \right\}$$

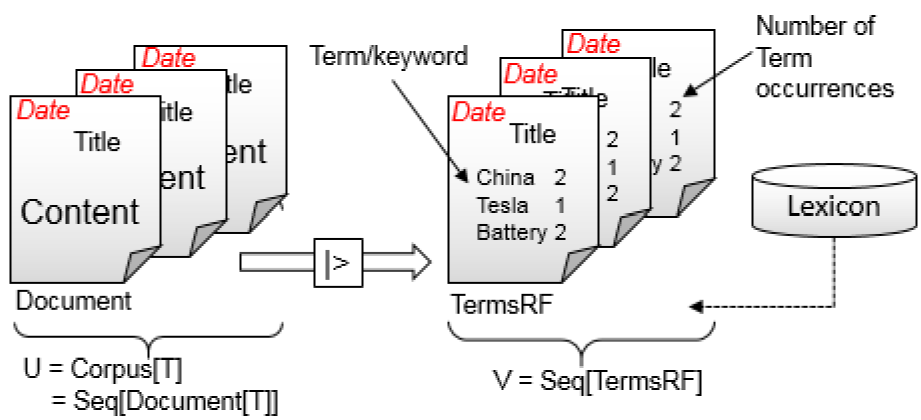

Stock price
Market response
News release
News release
t
t_1
t_1+d
t_2
t_2+d
Trading sessions

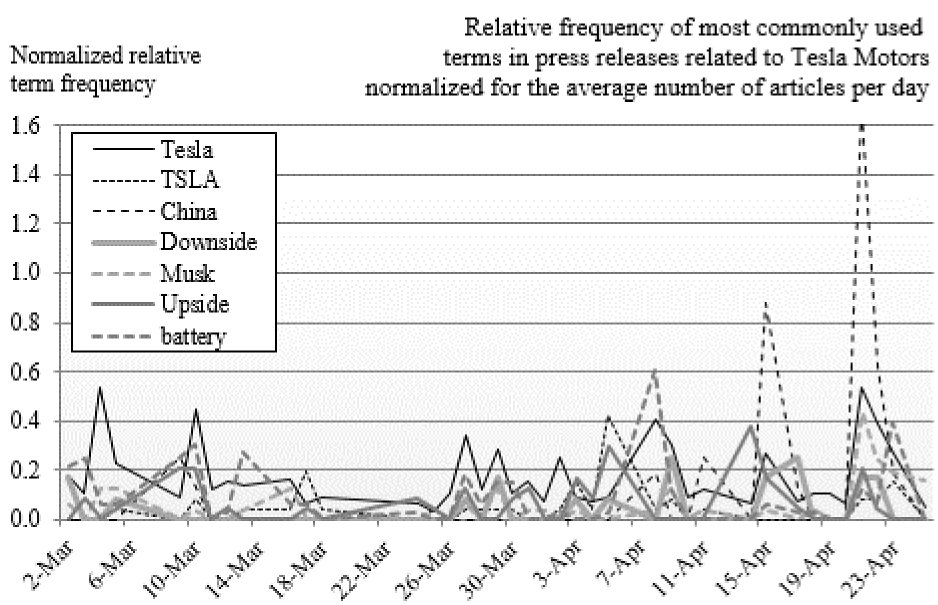

Normalized frequency
0.4
0.3
0.2
0.1
0.0
Frequencies distribution of delay
in the market response to news
related Tesla Motors
0.00
0.50
1.00
1.50
2.00
2.50
3.00
3.50
Market response delay to news

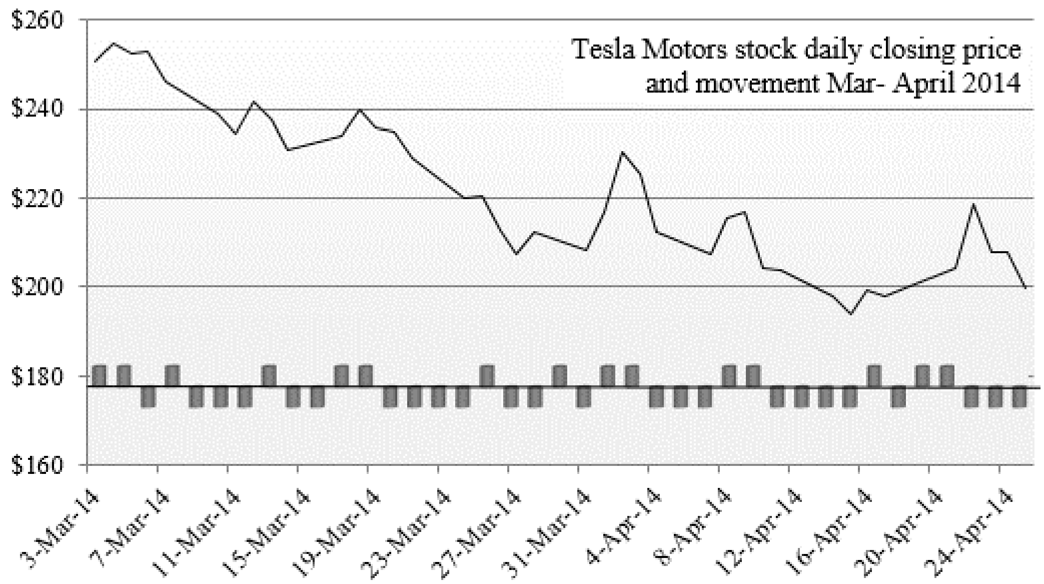

$pr_{t+1} - pr_t$
Expected
price change
Relative
frequency
of term T_0
$\frac{\sum_{t-n}^{t} f_0(t)}{\sum f_0(t)}$
. . .
$\frac{\sum_{t-n}^{t} f_j(t)}{\sum f_j(t)}$
Relative
frequency
of term T_j

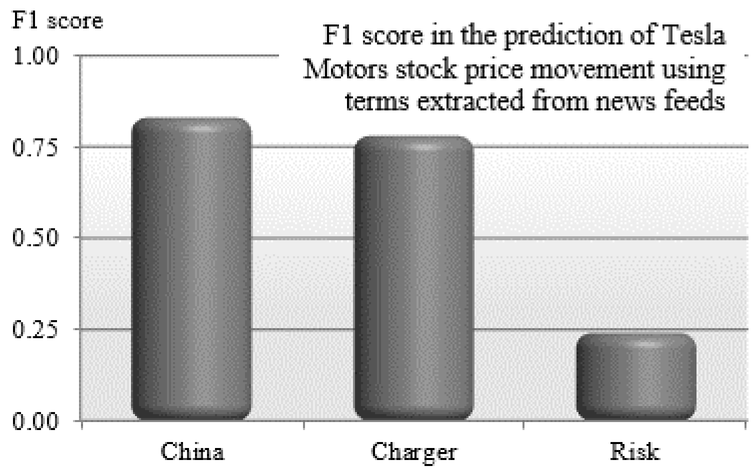
$$rtf\{t_i\} = \frac{\sum_{a \in D_t} n_i^a}{\sum_{a \in Corpus} n_i^a}$$

$$nrt\{t_i\} = \frac{rtf\{t_i\} N_d}{N_a}$$



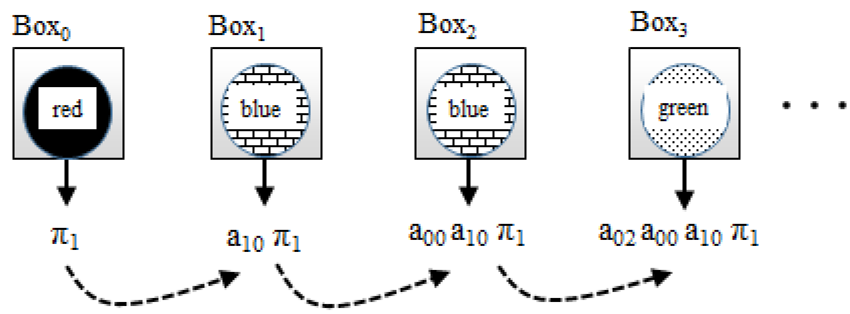


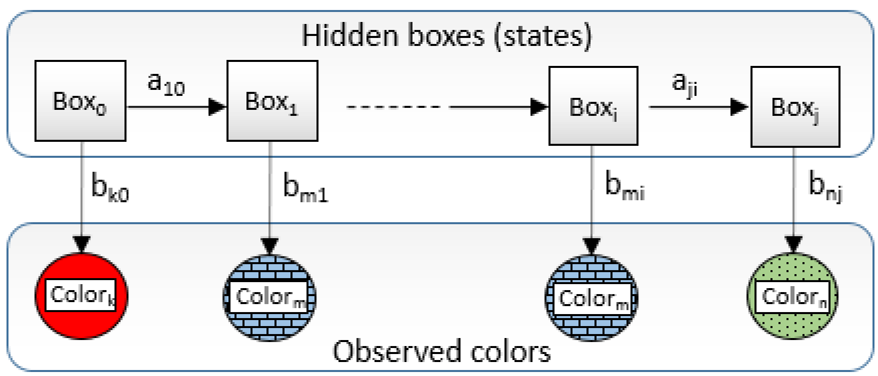


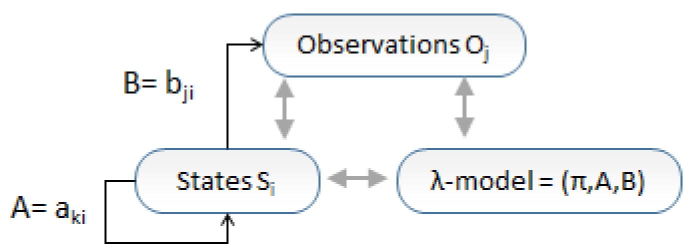


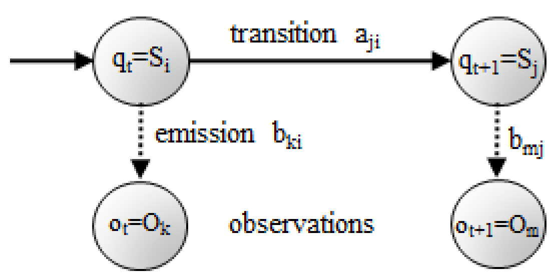
	Keyword 1	Keyword 2	Keyword 3	...	Keyword N
Date 1	0.42	*0.00*	0.07		0.23
Date 2	*0.00*	0.11	0.18		0.04
...					
Date J	0.13	0.29	*0.00*		*0.00*

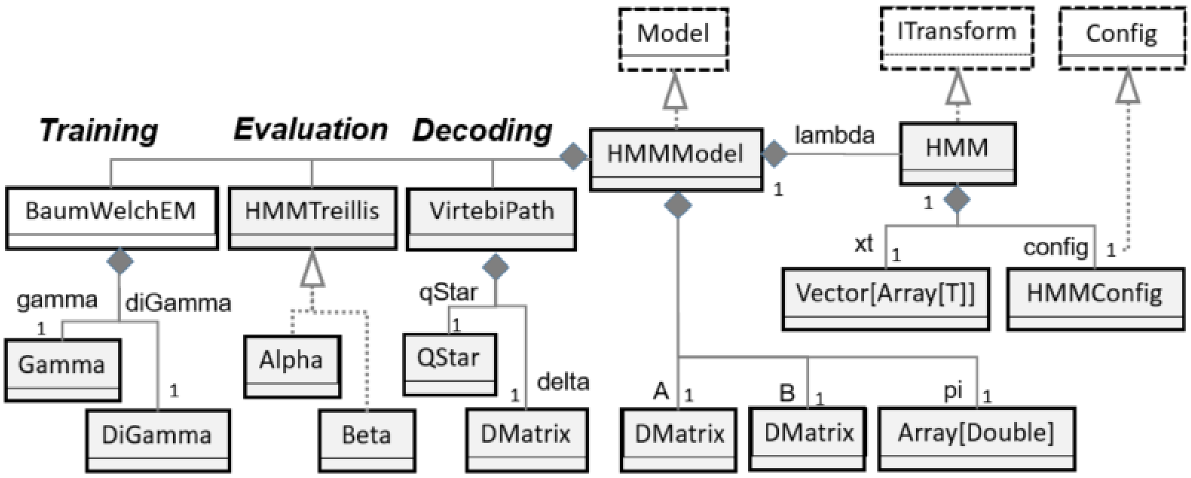
Chapter 7: Sequential Data Models











$$p\left(O_{0:T-1} \mid \lambda\right) \alpha\, p\left(O_{0:t} \mid \lambda\right).p\left(O_{t+1:T-1} \mid \lambda\right)$$

$$p\left(O_{0:T-1} \mid \lambda\right) = \sum_{i=0}^{N-1} p\left(O_{0:T-1}, q_t = S_i \mid \lambda\right)$$

$$p\left(O_{0:T-1} \mid \lambda\right) = \sum_{i} \alpha_t\left(i\right).\beta_t\left(i\right)$$

$$\alpha_0\left(i\right) = \pi_i.b_i\left(O_0\right)$$

$$\hat{a}_0\left(i\right) = \alpha_0\left(i\right) \Big/ \sum_{j=0}^{N-1} \alpha_0\left(i\right)$$

$$\alpha_t\left(i\right) = \sum_{j=0}^{N-1} \alpha_{t-1}\left(\mathrm{j}\right) a_{ji} b_i\left(\mathrm{O_t}\right) \quad c_t = 1 \Big/ \sum_{i=0}^{N-1} \alpha_t\left(i\right) \quad \hat{a}_t\left(i\right) = \alpha_t\left(i\right).c_t$$

$$\log p(O \mid \lambda) = -\sum_{j=0}^{T-1} \log\left(\frac{1}{\sum_{i=0}^{N-1} \hat{\alpha}_t(i)}\right)$$

$$\hat{\beta}_{T-1}(i) = \beta_{T-1}(i) \Big/ \sum_{j=0}^{N-1} \beta_{T-1}(j)$$

$$\beta_t(i) = \sum_{j=0}^{N-1} \beta_{t-1}(j).\mathrm{a}_{ij}.b_j(O_{t+1}) \quad c_t = 1 \Big/ \sum_{j=0}^{N-1} \beta_t(j) \quad \widehat{\beta}_t(i) = \beta_t(i).c_t$$

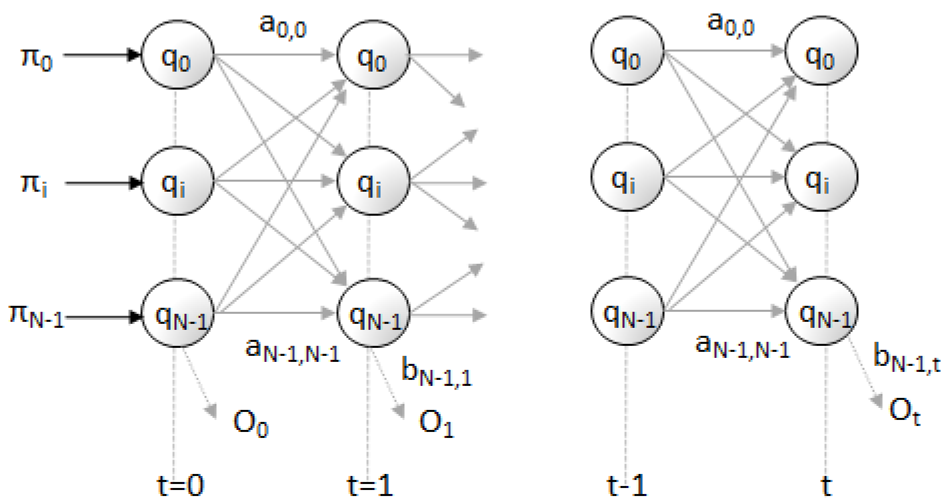
$$\gamma_t(i,j) = p\left(q_t = S_i, q_{t+1} = S_j \mid O, \lambda\right)$$

$$\gamma_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{j=0}^{N-1} \alpha_t(i) \beta_t(i)}$$

$$\hat{\pi}_i = \gamma_0(i) \quad \gamma_t(i) = \sum_{j=0}^{N-1} \gamma_t(i,j)$$

$$\hat{a}_{ij} = \frac{\sum_{t=0}^{T-1}\left[\left(\gamma_t(i,j)\right)\right.}{\sum_{t=0}^{T-1} \gamma_t(i)}$$

$$\hat{b}_{ij} = \frac{\sum_{t=0}^{O_j} \gamma_t(i)}{\sum_{t=0}^{T-1} \gamma_t(i)}$$

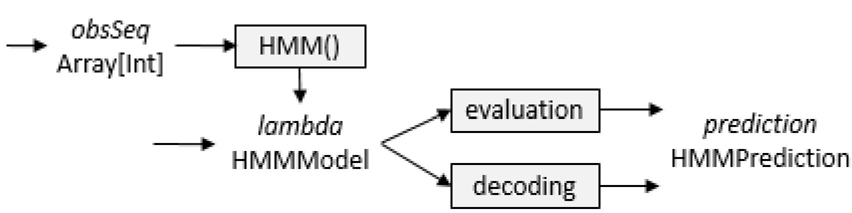


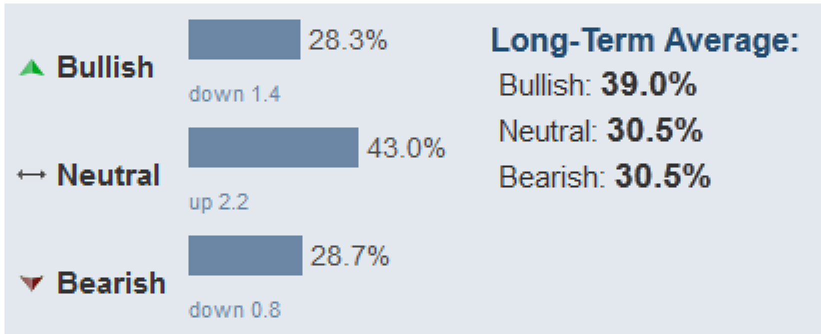
$$\delta_t(i) = \max_{qj:\{o,T-1\}} p\left(q_{0:T-1} = S_i, O_{o:T-1} \mid \lambda\right)$$

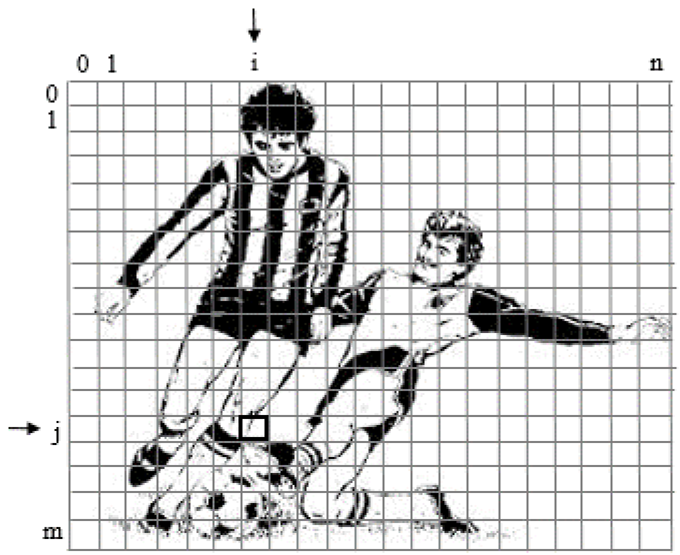
$$\delta_0(i) = \pi_i b_i(O_0) \psi_0(i) = 0 \forall i$$

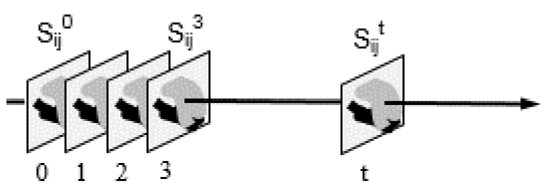
$$\delta_t(j) = \max_i\left(\delta_{t-1}(i).a_{ij}.b_j(O_t)\right) \quad \psi_t(j) = \arg\max_i\left(\delta_{t-1}(i).a_j\right)$$

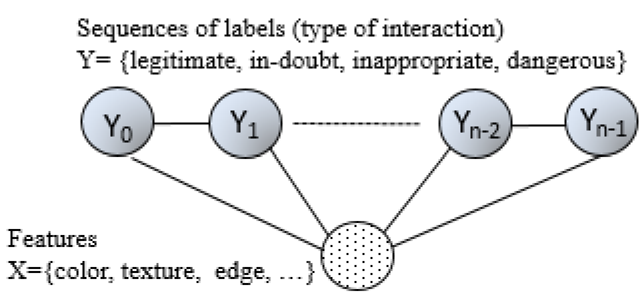
$$q^*_t = \psi_{t+1}\left(q^*_{t+1}\right) \; q^*_T = \arg\max_i \delta_T(i)$$

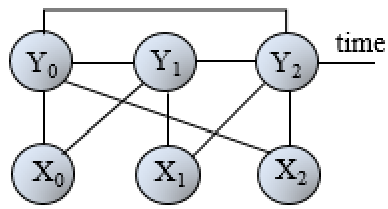



Bullish
28.3%
down 1.4
Neutral
43.0%
up 2.2
Bearish
28.7%
down 0.8
Long-Term Average:
Bullish: 39.0%
Neutral: 30.5%
Bearish: 30.5%

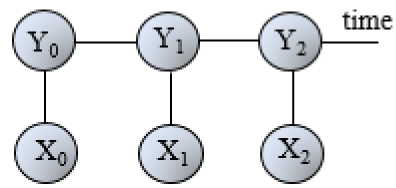

0 1
i
n
0
1
j
m


S_{ij}^0
S_{ij}^3
S_{ij}^t
0 1 2 3
t

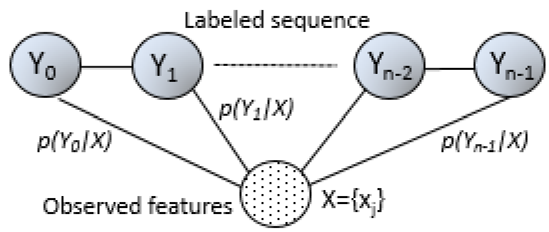

Sequences of labels (type of interaction)
Y= {legitimate, in-doubt, inappropriate, dangerous}
Y_0
Y_1
Y_{n-2}
Y_{n-1}
Features
X={color, texture, edge, …}



Example non-linear CRF



Linear chain CRF

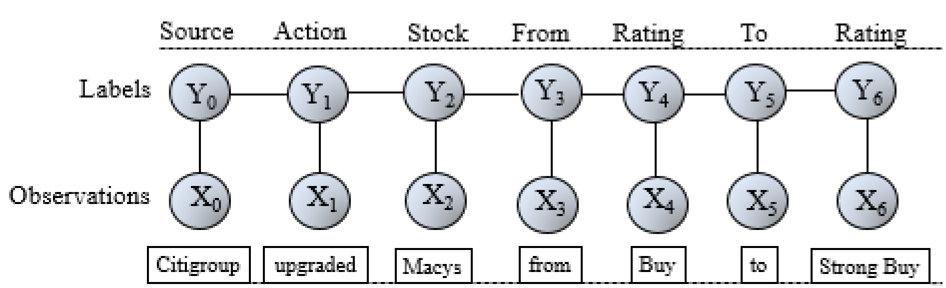


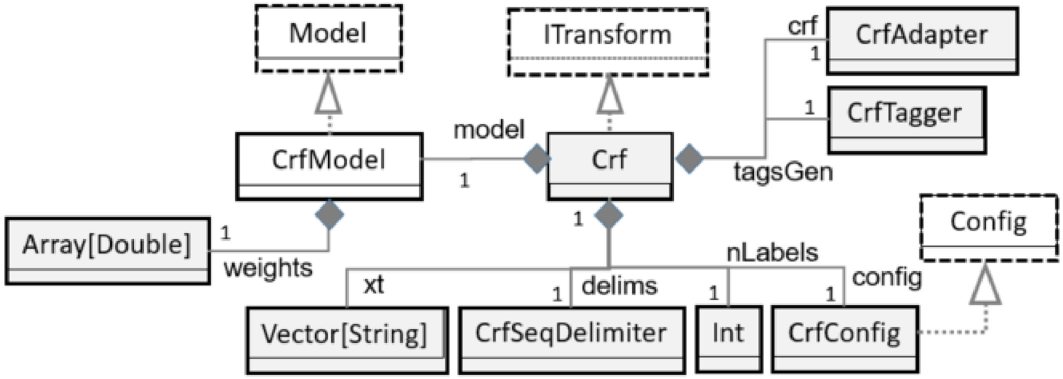
$$\log f_i\left(y_{i-1}, y_i \mathrm{x}, \mathrm{i}\right) = w_c + \sum_{i=0}^{K-1} w_i t_i\left(y_{i-1}, y_i, \mathrm{x}, \mathrm{i}\right) + \sum_{j=0}^{K-1} \mu_j s_j\left(y_i, \mathrm{x}, \mathrm{i}\right)$$

$$t_i\left(y_{i-1}, y_i, \mathrm{x}, \mathrm{i}\right) = \mathrm{I}\left(y_{i-1} = l_1\right).\mathrm{I}\left(y_i = l_2\right).\mathrm{I}\left(x = 0\right)$$

$$F_i\left(y, x\right) = \sum_{j=0}^{K-1} f_i\left(y_{j-1} y_j, x, i\right) \quad \log p\left(x, \lambda\right) \propto \sum_{j=0}^{K-1} w_j F_j\left(x, y\right)$$

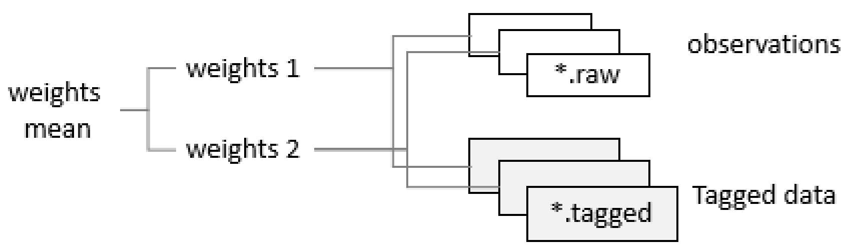
$$p\left(y \mid x, w\right) = \frac{1}{Z\left(x\right)} e^{\sum_{j=0}^{K-1} w_j F_j\left(x, y\right)} \quad z\left(x\right) = \sum_{i=0}^{N-1} \sum_{j=0}^{K-1} w_j F_j\left(x, y\right)$$

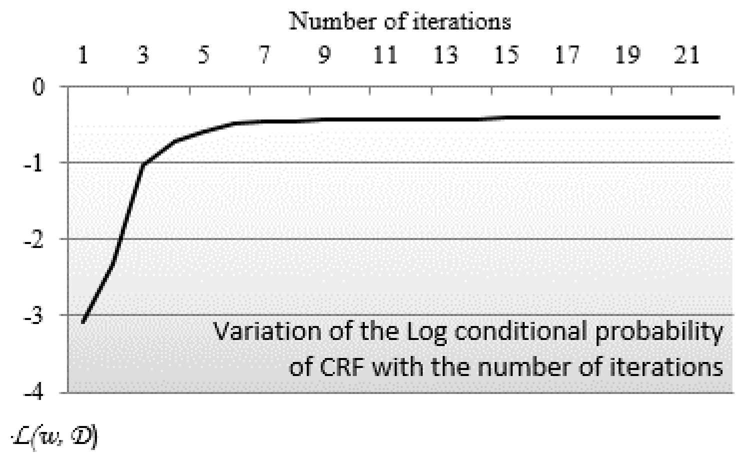




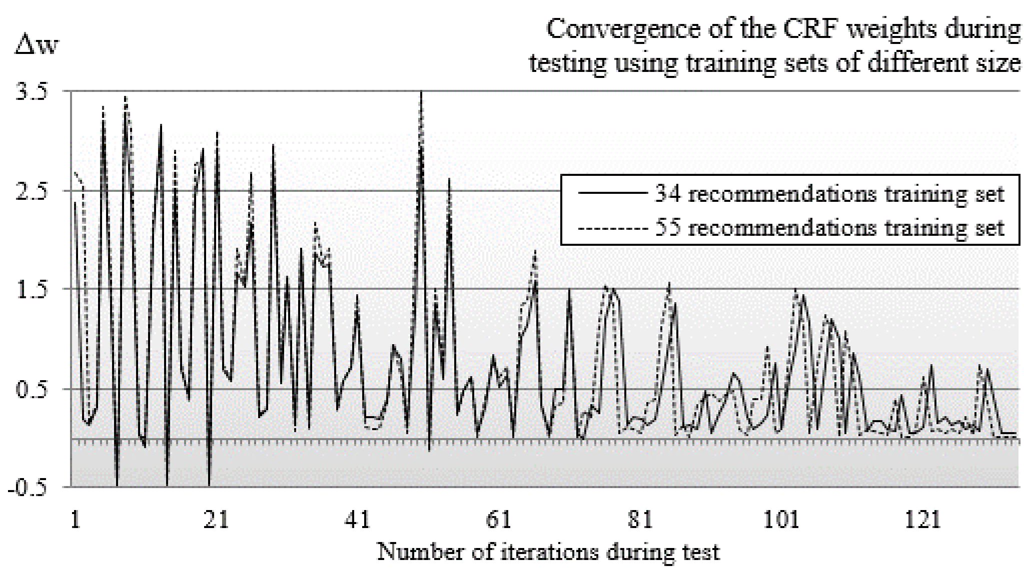
$$\mathcal{L}(w,D) = -\sum_{i=0}^{n-1} \log p(y_i \mid x_i, w)$$

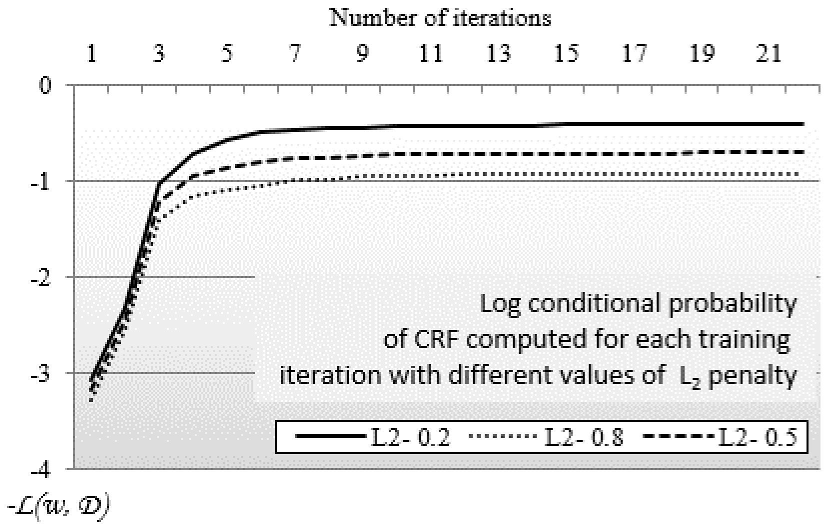
$$w^* = \underset{\lambda}{\arg\max}\left[\mathcal{L}(w,D) + \lambda \|w\|^2\right] \quad \lambda = \frac{1}{2\sigma^2}$$





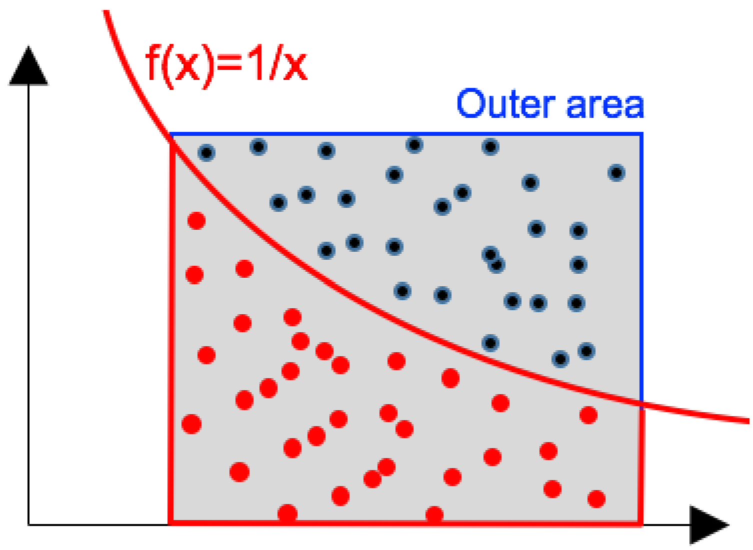
$$\Delta w = \sum_{i=0}^{D-1} \left(w_i^{t+1} - w_i^{t} \right)$$

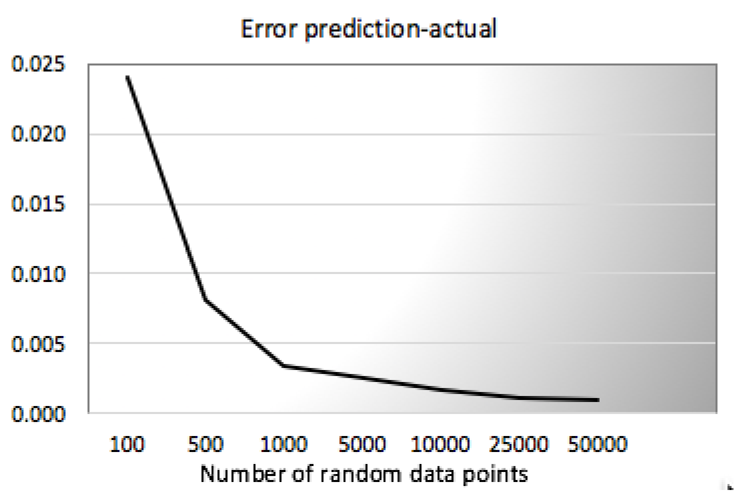



Number of iterations
1
3
5
7
9
11
13
15
17
19
21
0
-1
-2
-3
-4
-L(w, D)
Log conditional probability
of CRF computed for each training
iteration with different values of L_2 penalty
L2- 0.2
L2- 0.8
L2- 0.5

Chapter 8: Monte Carlo Inference

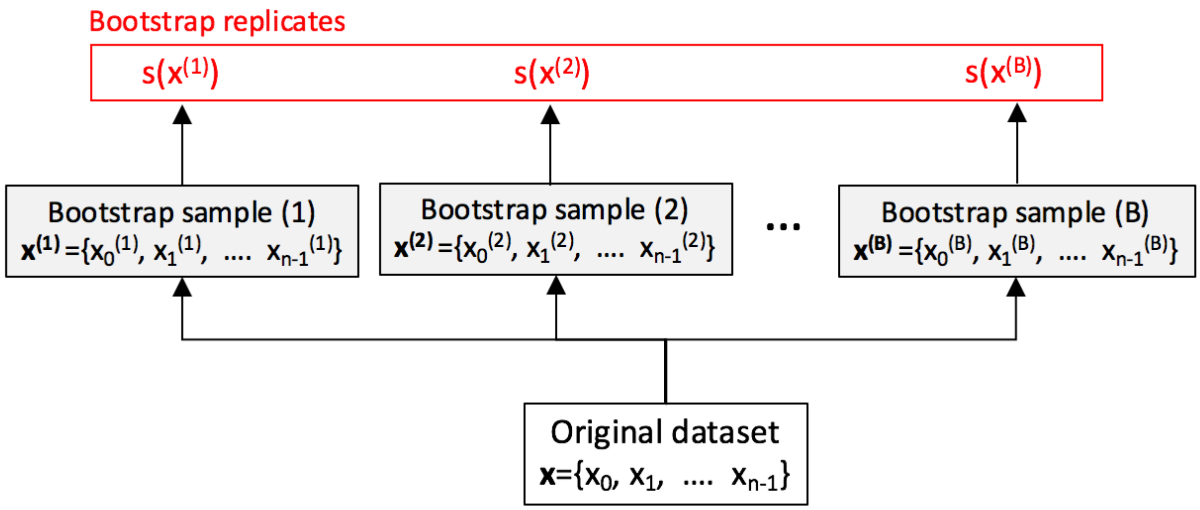
$$\sqrt{-2\log(u_1)}.\sin(2\pi u_2)\ \sqrt{-2\log(u_1)}.\cos(2\pi u_2)$$

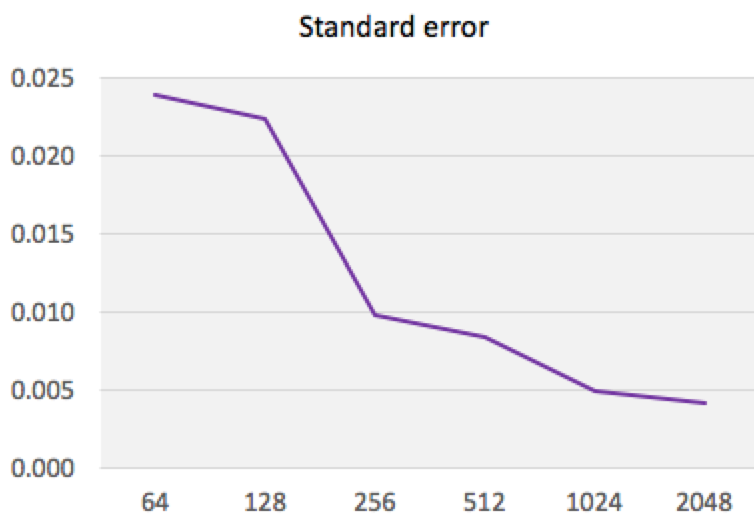




$$x^{(i)} = \left\{x_j^{(i)}\right\}\ \hat{\theta}^{(i)} = s\left(x^{(i)}\right)$$

$$\hat{s} = \sqrt{\frac{1}{B-1}\sum_{j=0}^{B-1}\left(\hat{\theta}_j^{(i)} - \bar{\theta}\right)^2} \quad \hat{\theta} = \frac{1}{B}\sum_{j=0}^{B-1}\hat{\theta}_j^{(i)}$$





$$\alpha = \frac{\hat{p}\left(s'\right)q\left(s^t \mid s'\right)}{\hat{p}\left(s^t\right)q\left(s' \mid s^t\right)}$$

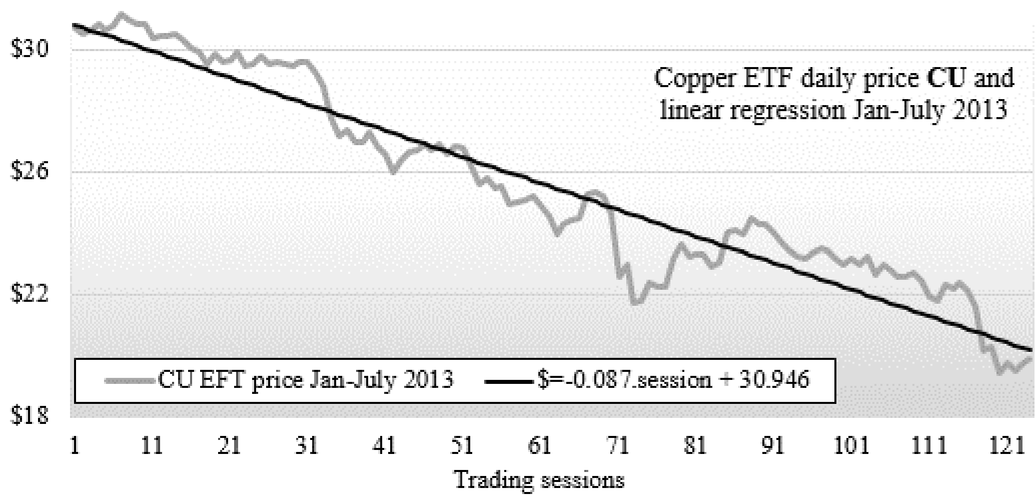
$$s^{t+1} = \begin{cases} s' & if\ u < \min(1, \alpha) \\ s^t & if\ u \geq \min(1, \alpha) \end{cases}$$

$$\log \alpha = \log\left(\hat{p}(s')\right) - \log\left(\hat{p}(s^t)\right) + \log\left(q(s^t \mid s')\right) - \log\left(q(s' \mid s^t)\right)$$

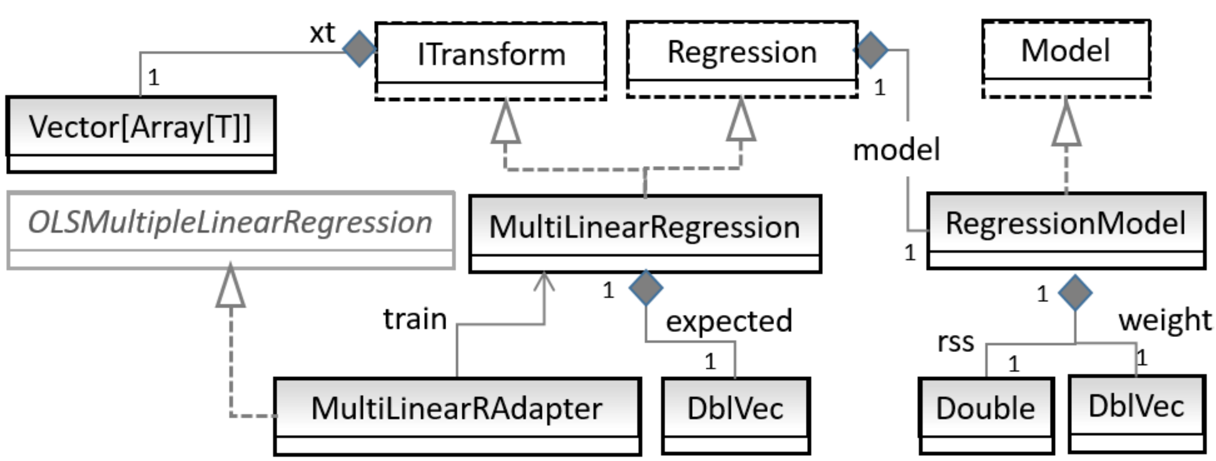
$$s^{t+1} = \begin{cases} s' & if\ u < e^{\alpha} \\ s^t & if\ u \geq e^{\alpha} \end{cases}$$

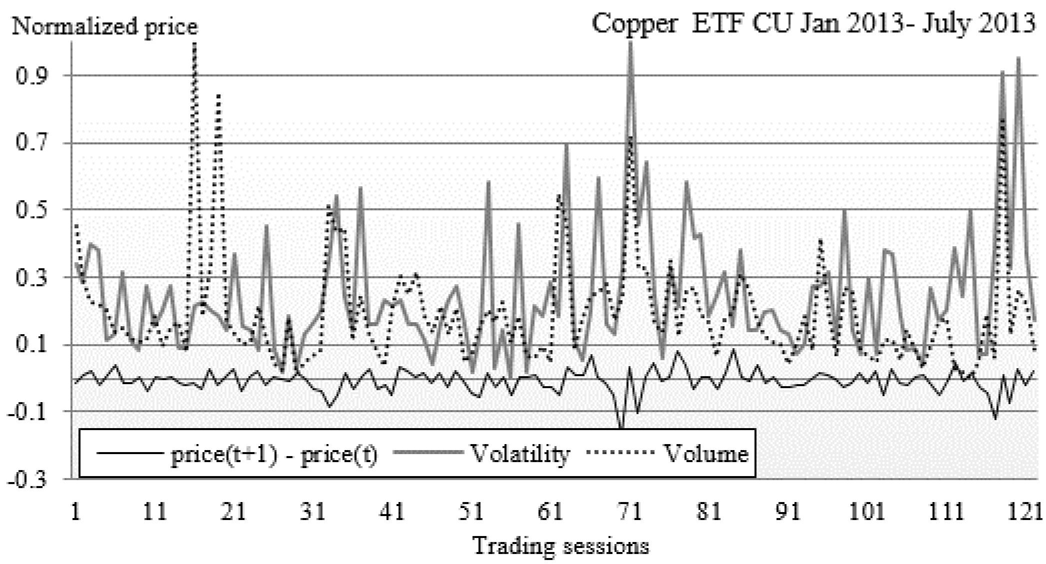
Chapter 9: Regression and Regularization

$$\tilde{w} = arg\,\min_{w,r}\left\{\sum_{j=0}^{N-1}\left(y_j - f\left(x_j|w\right)\right)^2\right\} \quad f\left(x|w\right) = w_0 + w_1 x$$

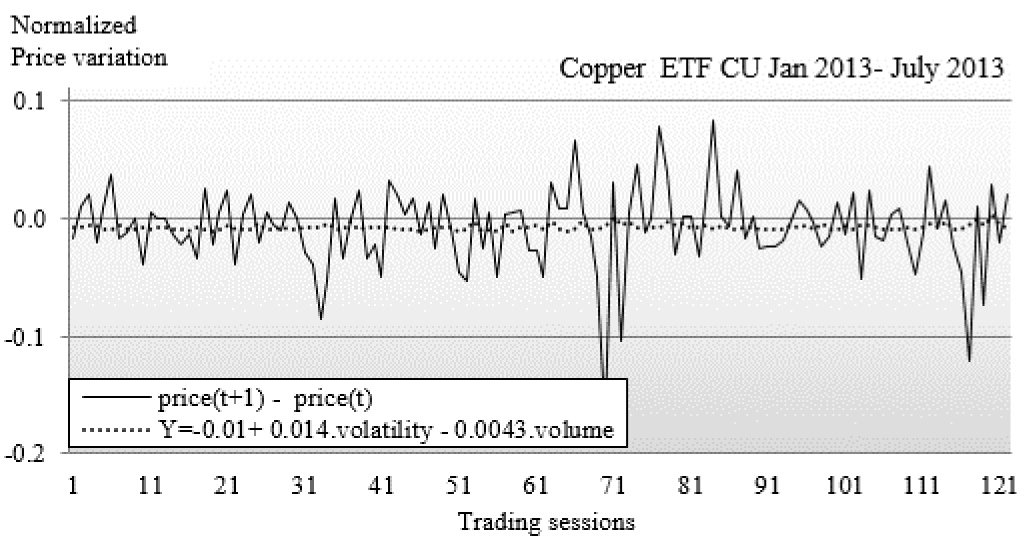


$$\tilde{w} = arg\,\min_{w,r}\left\{\sum_{j=0}^{N-1}\left(y_j - f\left(x_j|w\right)\right)^2\right\} \quad f\left(x|w\right) = w_0 + \sum_{1}^{D-1} w_d x_d$$

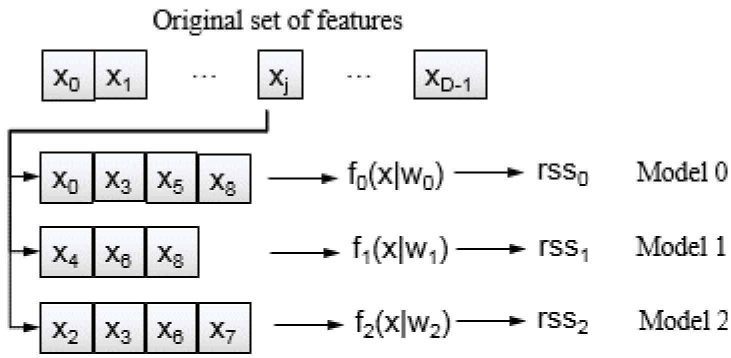




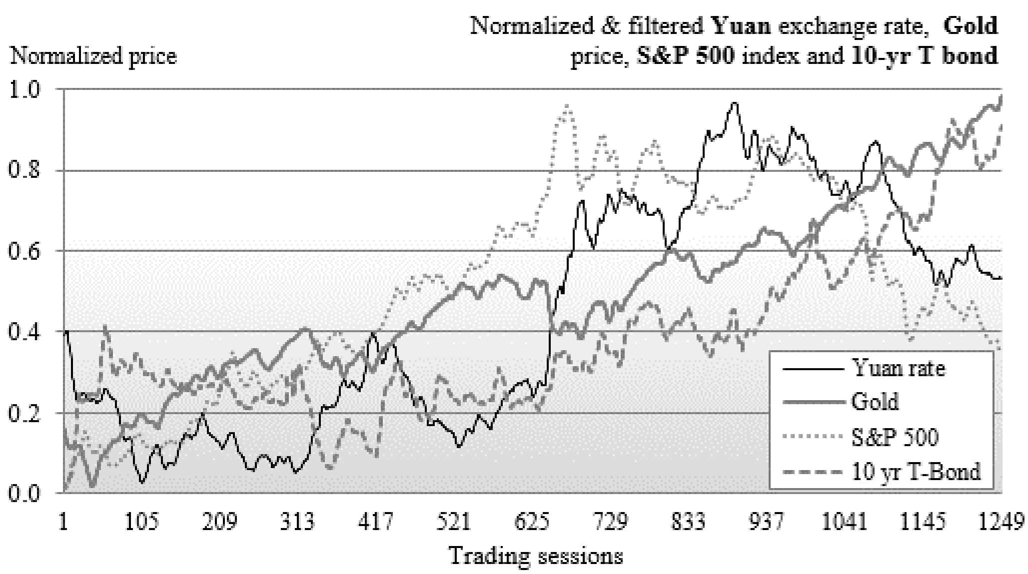
Normalized price
Copper ETF CU Jan 2013- July 2013
0.9
0.7
0.5
0.3
0.1
-0.1
-0.3
price(t+1) - price(t)
Volatility
Volume
1
11
21
31
41
51
61
71
81
91
101
111
121
Trading sessions



Normalized
Price variation
Copper ETF CU Jan 2013- July 2013
0.1
0.0
-0.1
-0.2
price(t+1) - price(t)
Y=-0.01+ 0.014.volatility - 0.0043.volume
1
11
21
31
41
51
61
71
81
91
101
111
121
Trading sessions



$$\breve{f} = arg\,min_{f_j}\left\{\sum_{i=0}^{n-1}\left(y_j - f_j\left(x|w\right)\right)^2\right\} \quad f_j\left(\boldsymbol{x}|\boldsymbol{w}\right) = w_{j0} + \sum_{d=1}^{D_j-1} w_{jd}x_d$$



$$r^2 = 1 - \frac{RSS}{TSS} \quad TSS = \sum_{i=0}^{n-1}\left(y_i - \bar{f}\left(x|w\right)\right)^2 \quad \bar{f} = \sum_f f_j$$

CNY = f (SPY, GLD, TLT)
0.16089780923264457 + -0.21189413823325406.x1 + 0.26299169969099795.x2 + 0.3556562652009136.x3
RSS: 3.681353535940423

CNY = f (SPY, TLT)
0.2039015515038045 + -0.03796334296279046.x1 + 0.26219728078589966.x2
RSS: 3.8589613138639227

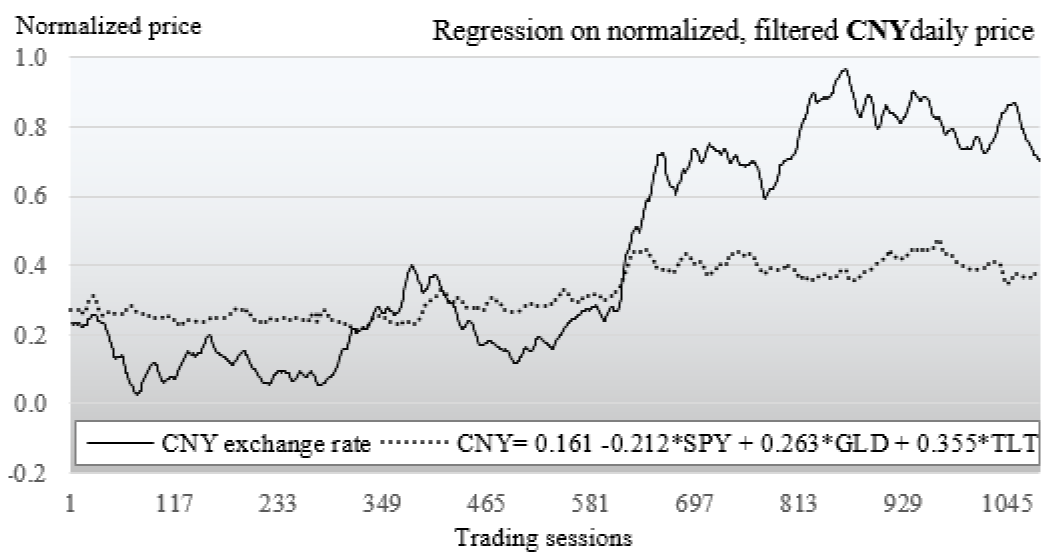
CNY = f (GLD, TLT)
0.19290917330324198 + 0.015507174710195552.x1 + 0.2204800144601237.x2
RSS: 3.8849688539396317

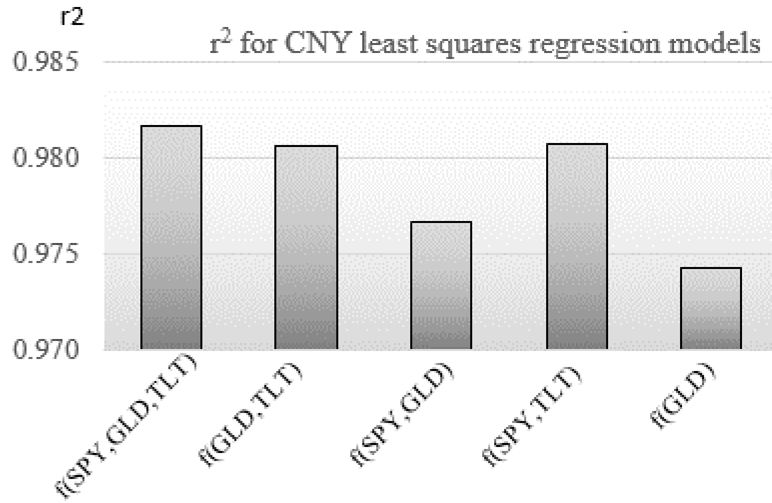
CNY = f (SPY, GLD)
0.22242202699107552 + 0.17842973203100937.x1 + -0.12099602178260839.x2
RSS: 4.6681933948464645

CNY = f (SPY)
0.20238901847764892 + 0.1251591898720694.x1
RSS: 4.7291908591838405

CNY = f (TLT)
0.19724352413716711 + 0.22501420632545652.x1
RSS: 3.8876824376753705

CNY = f (GLD)
0.198195293931846 + 0.16413676262473123.x1
RSS: 5.149661975952835





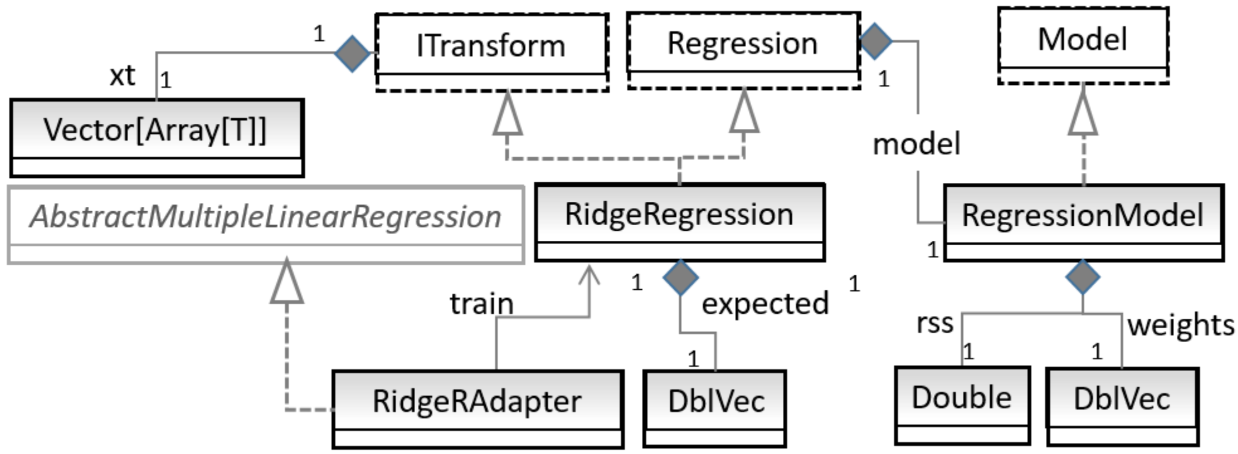
$$\hat{\boldsymbol{w}} = \arg\min_{w_d} \left\{ \sum_{i=0}^{n-1} \left(y_i - f\left(\boldsymbol{x}_i \middle| \boldsymbol{w} \right) \right)^2 + \lambda J\left(\boldsymbol{w} \right) \right\}$$

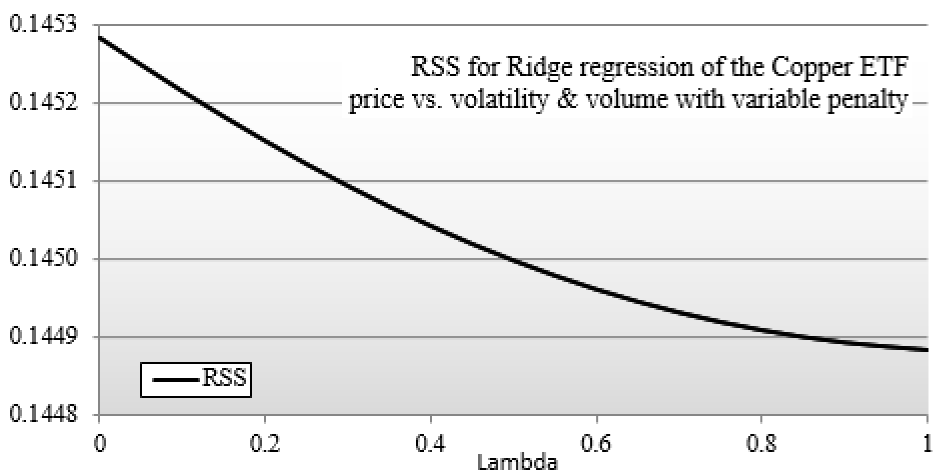
$$J_{pq}\left(w \right) = \left\| w \right\|_p^q = \left[\sum_{d=1}^{D-1} \left| w_d \right|^p \right]^{q/p}$$

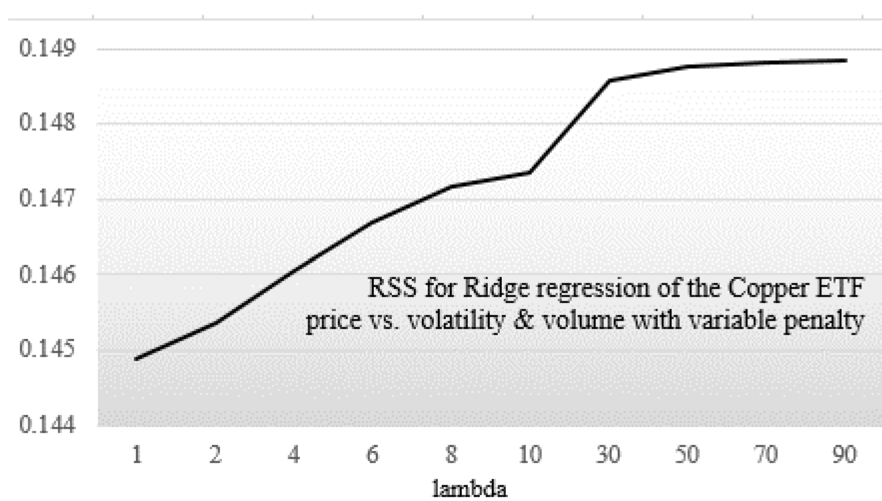
$$\tilde{w}_{ridge} = arg\,\min_{w} \left\{ \sum_{j=0}^{N-1} \left(y - w_0 - w^T x \right)^2 + \lambda \left| w \right|_2^2 \right\} \quad \left| w \right|_2^2 = \sum_{1}^{D-1} w_d^2$$

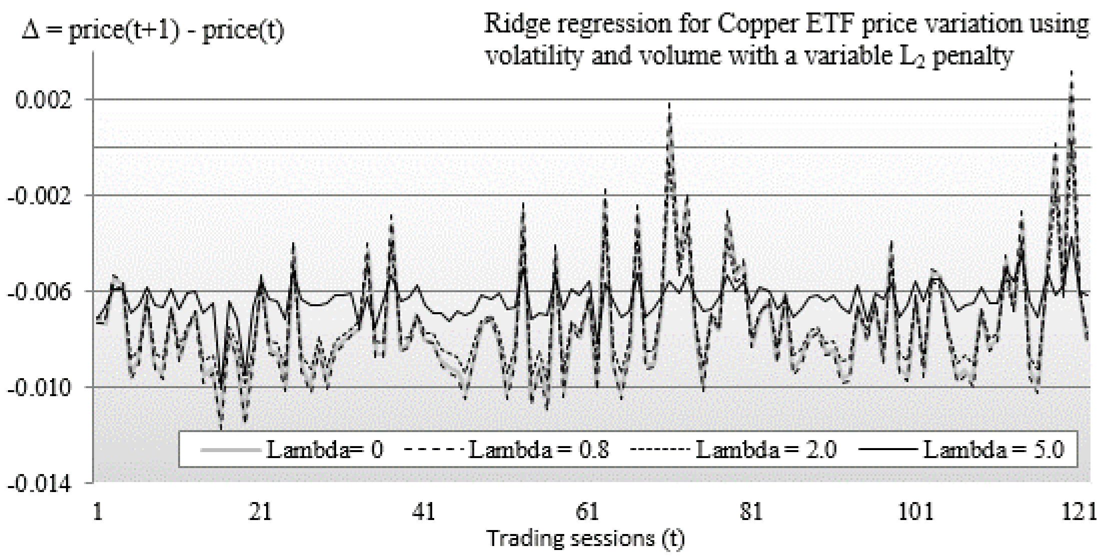
$$\left\{ X^T X - \lambda.I \right\}.\hat{w}_{Ridge} = X^T y$$

$$\left\{ X^T X - \lambda.I \right\} = Q \begin{vmatrix} R \\ 0 \end{vmatrix} \quad w_{Ridge} = Q^T y \begin{bmatrix} R \\ 0 \end{bmatrix}^{-1}$$


ITransform
Regression
Model
xt
Vector[Array[T]]
model
AbstractMultipleLinearRegression
RidgeRegression
RegressionModel
train
expected
rss
weights
RidgeRAdapter
DblVec
Double
DblVec


RSS for Ridge regression of the Copper ETF
price vs. volatility & volume with variable penalty
RSS
Lambda
0.1453
0.1452
0.1451
0.1450
0.1449
0.1448
0
0.2
0.4
0.6
0.8
1


RSS for Ridge regression of the Copper ETF
price vs. volatility & volume with variable penalty
lambda
0.149
0.148
0.147
0.146
0.145
0.144
1
2
4
6
8
10
30
50
70
90

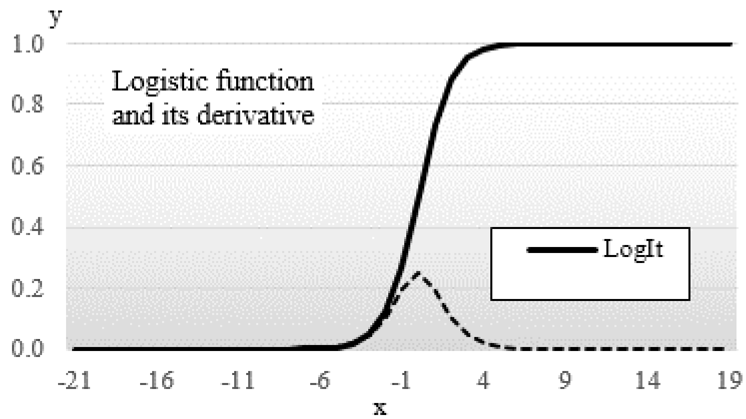


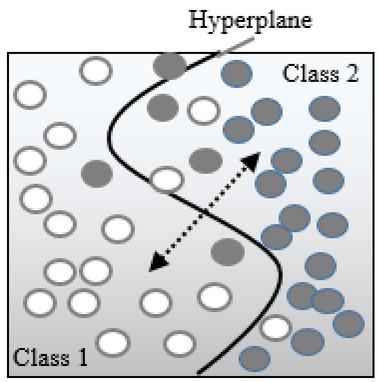
$$L\left(w\right)=\sum_{i=0}^{n-1} r_i\left(w\right)^{2} \quad r_i\left(w\right)=y_i-f\left(x_i\middle|w\right)$$

$$\sum_{i=0}^{n-1} r_i\left(w\right)J_{id}\left(w\right)=0 \quad J_{id}\left(w\right)=-\frac{\partial r_i\left(w\right)}{\partial w_d}$$

$$f\left(x_i\middle|w\right)-f\left(x_i\middle|w^{(k)}\right)\sim\sum_{jd=0}^{D-1}\frac{\partial f\left(x_i\middle|w^{(k)}\right)}{\partial w_d}\left(w-w^{(k)}\right)$$

$$f\left(x\right)=\frac{1}{\left(1-e^{-x}\right)} \qquad \frac{df}{dx}=f\left(x\right)\left(1-f\left(x\right)\right)$$



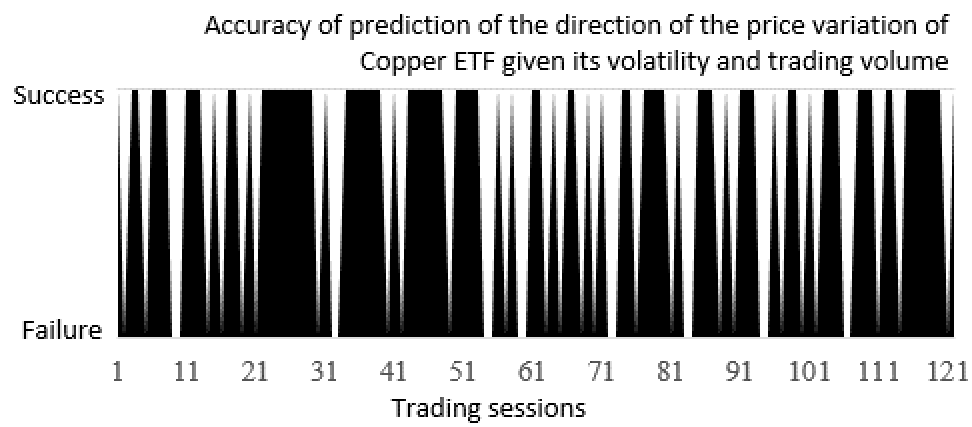


$$L\ (w) = \sum_{i=0}^{N-1} \log p\left(x_i \middle| w\right)$$

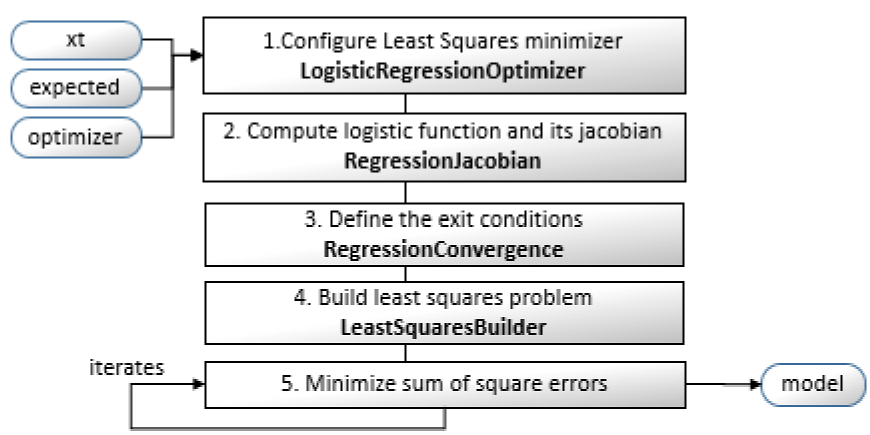
$$x_i = \{1, x_{i0}, \dots, x_{id-1}\} \quad p\left(x_i \middle| w\right) = \frac{1}{1+e^{-w^T x_i}} \quad w^T x_i = \sum_{j=0}^{d} w_j x_{ij}$$

$$sse(w) = \frac{1}{2}\sum_{i=0}^{N-1}\left\{y_i - \log\left(1+e^{-w^T x_i}\right)\right\}^2 \quad y \in \{0,1\}$$

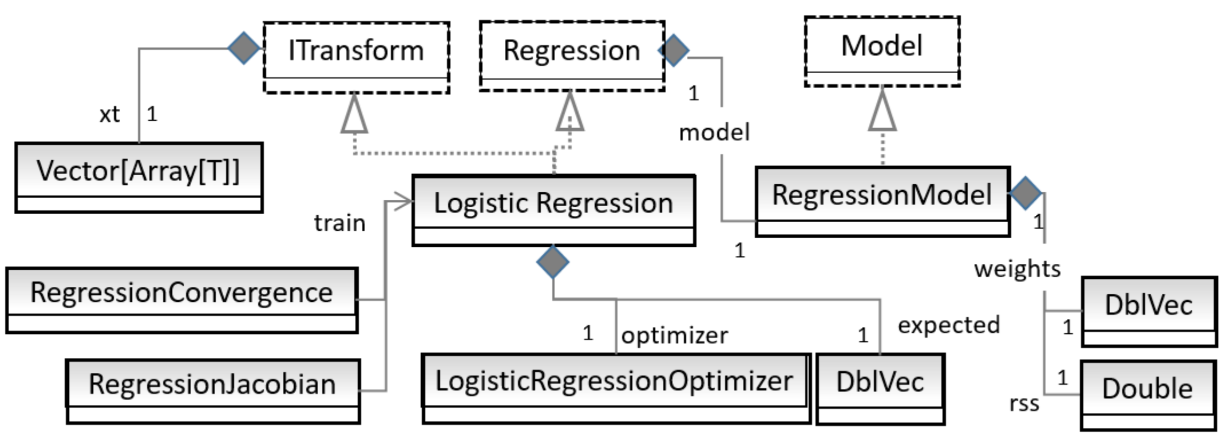
$$\frac{\partial L\ (\tilde{w})}{\partial w_j} = \sum_{i=0}^{N-1} x_{ij}\left(y_i - \frac{1}{1+e^{-\tilde{w}^T x_i}}\right) = 0$$



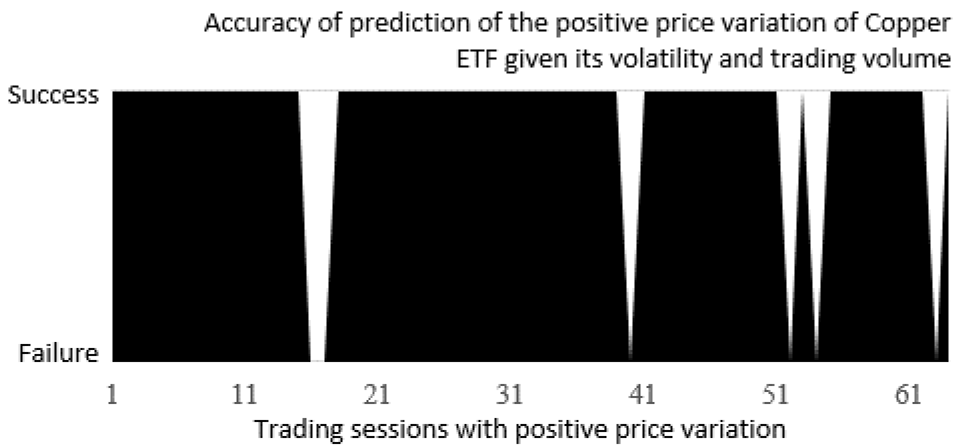
Accuracy of prediction of the direction of the price variation of Copper ETF given its volatility and trading volume
Success
Failure
1
11
21
31
41
51
61
71
81
91
101
111
121
Trading sessions



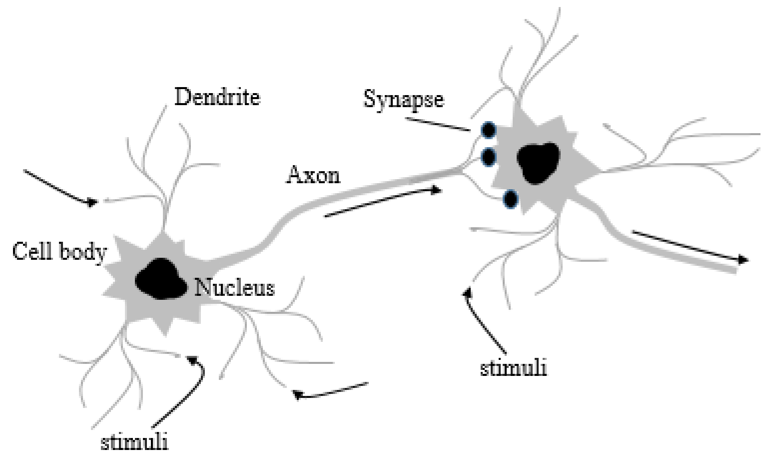
xt
expected
optimizer
1.Configure Least Squares minimizer
LogisticRegressionOptimizer
2. Compute logistic function and its jacobian
RegressionJacobian
3. Define the exit conditions
RegressionConvergence
4. Build least squares problem
LeastSquaresBuilder
iterates
5. Minimize sum of square errors
model

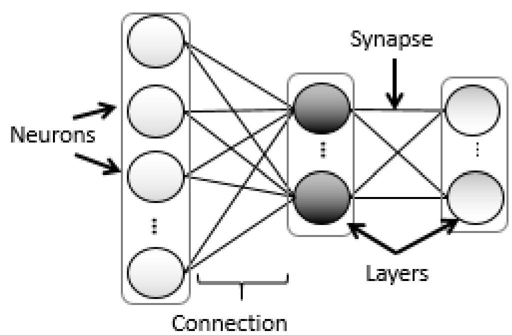


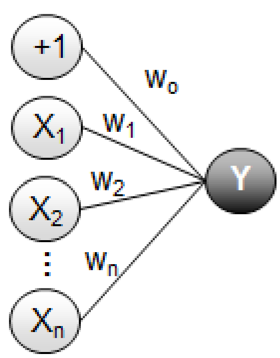
ITransform
Regression
Model
xt
1
Vector[Array[T]]
1
model
Logistic Regression
RegressionModel
train
1
1
weights
RegressionConvergence
1
optimizer
1
expected
DblVec
1
RegressionJacobian
LogisticRegressionOptimizer
DblVec
1
Double
rss


Accuracy of prediction of the positive price variation of Copper ETF given its volatility and trading volume
Success
Failure
1
11
21
31
41
51
61
Trading sessions with positive price variation

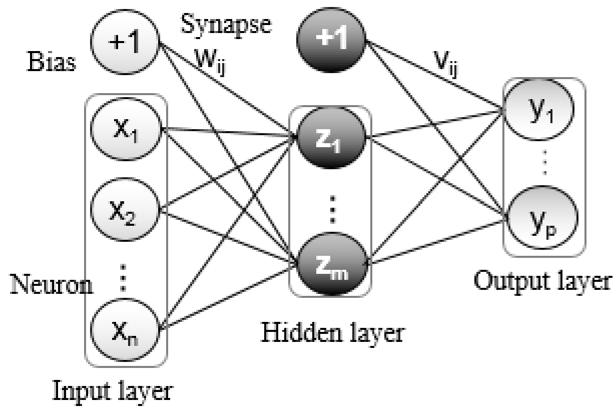
Chapter 10: Multilayer Perceptron



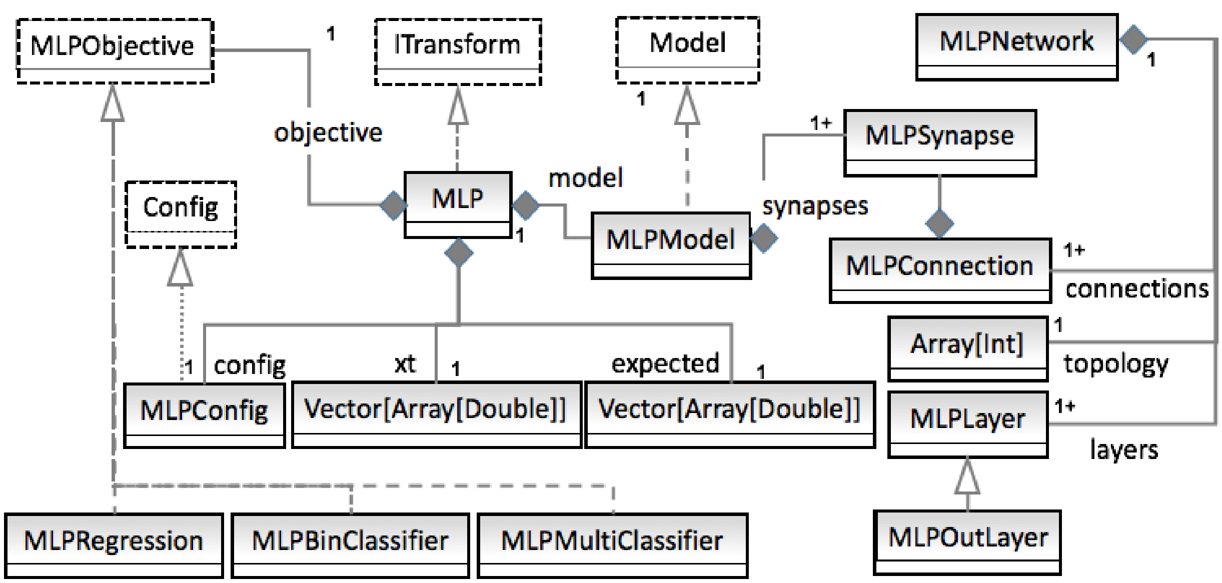


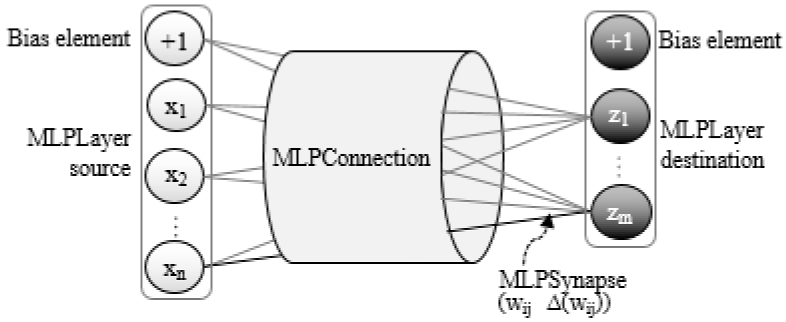


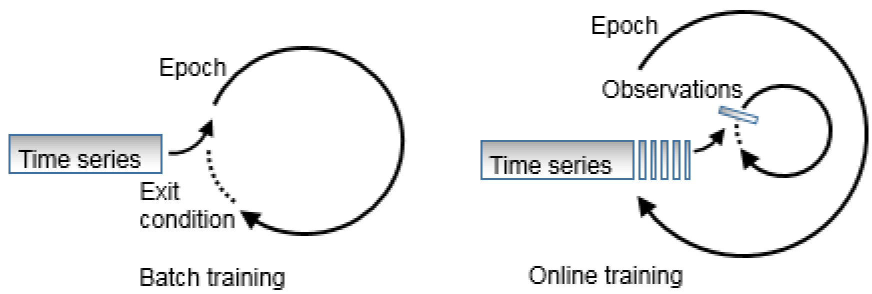
$$y = \sigma\left(w_0 + w^T x\right) = \frac{1}{1 + e^{-\left(w_0 + w^T x\right)}}$$

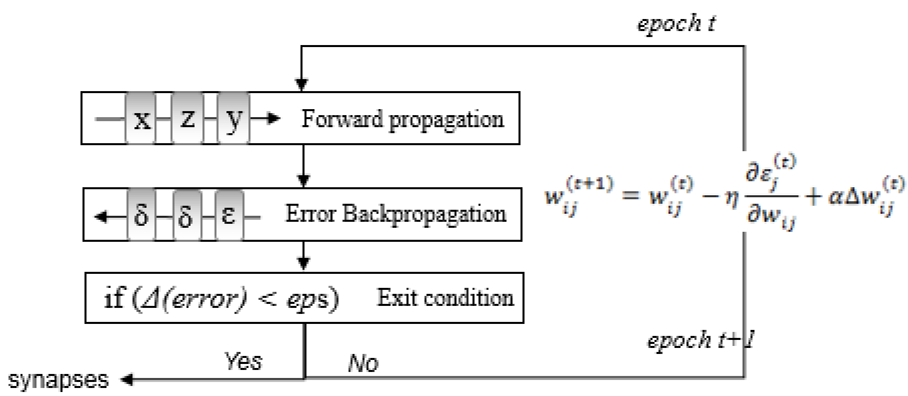


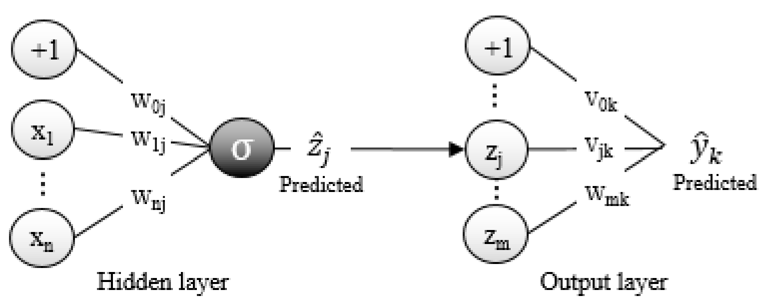
$$\hat{y} = h\left(w_0 + \sum_{i=1}^{n} w_i x_i \right) = h\left(w_0 + w^T x \right)$$




Bias element
+1
x_1
MLPLayer source
x_2
x_n
MLPConnection
+1
Bias element
z_1
MLPLayer destination
z_m
MLPSynapse
(w_{ij} $\Delta(w_{ij})$)

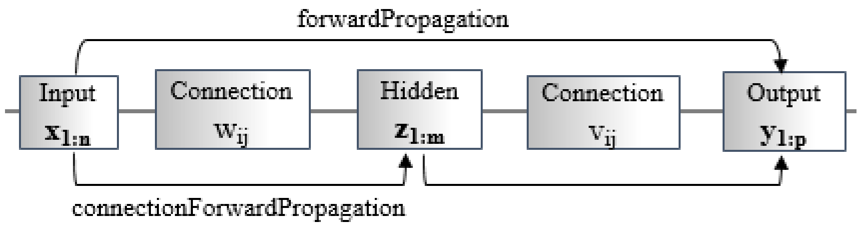

Epoch
Time series
Exit condition
Batch training
Epoch
Observations
Time series
Online training


epoch t
x z y Forward propagation
δ δ ε Error Backpropagation
$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial \varepsilon_j^{(t)}}{\partial w_{ij}} + \alpha \Delta w_{ij}^{(t)}$
if (Δ(error) < eps) Exit condition
epoch t+1
Yes
No
synapses


+1
x_1
x_n
w_{0j}
w_{1j}
w_{nj}
σ
$\hat{z}_j$
Predicted
Hidden layer
+1
z_j
z_m
v_{0k}
v_{jk}
w_{mk}
$\hat{y}_k$
Predicted
Output layer

$$\tilde{y}_k = v_{0j} + \sum_{j=1}^{m} v_{kj} z_j$$

$$\tilde{z}_j = \sigma\left(w_{0j} + \sum_{i=1}^{m} w_{ij} x_i \right) = \frac{1}{1 + e^{-w_{0j} - \sum_{i=1}^{m} w_{ij} x_i}}$$



$$\varepsilon = \frac{1}{2} \sum_{i=0}^{n-1} \sum_{j=0}^{k-1} \left(y_{ij} - \tilde{y}_{ij} \right)^2 \quad \overline{\varepsilon} = \frac{\varepsilon}{n}$$

$$ce = -\sum_{i=0}^{n-1} \left\{ y_i \log\left(\tilde{y}_i \right) + \left(1 - y_i \right) . \log\left(\tilde{y}_i \right) \right\}$$

$$ce = -\sum_{i=0}^{n-1} \sum_{j=0}^{k-1} \tilde{y}_{ij} \log\left(y_{ij} \right)$$

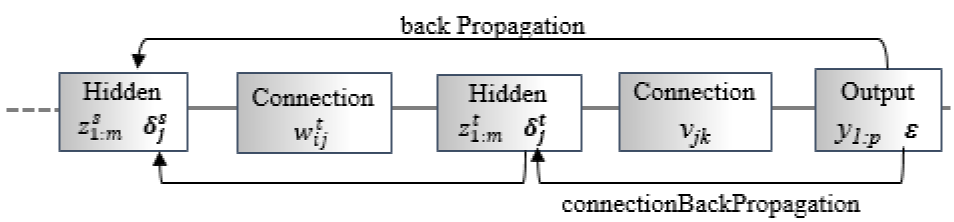
$$\hat{\tilde{y}} = \frac{e^{-\hat{y}_k}}{\sum_i e^{-\hat{y}_i}}$$

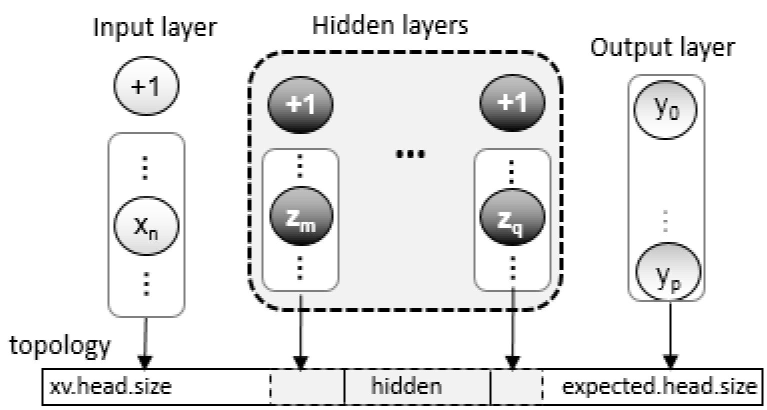
$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial \varepsilon_j^{(t)}}{\partial w_{ij}}$$

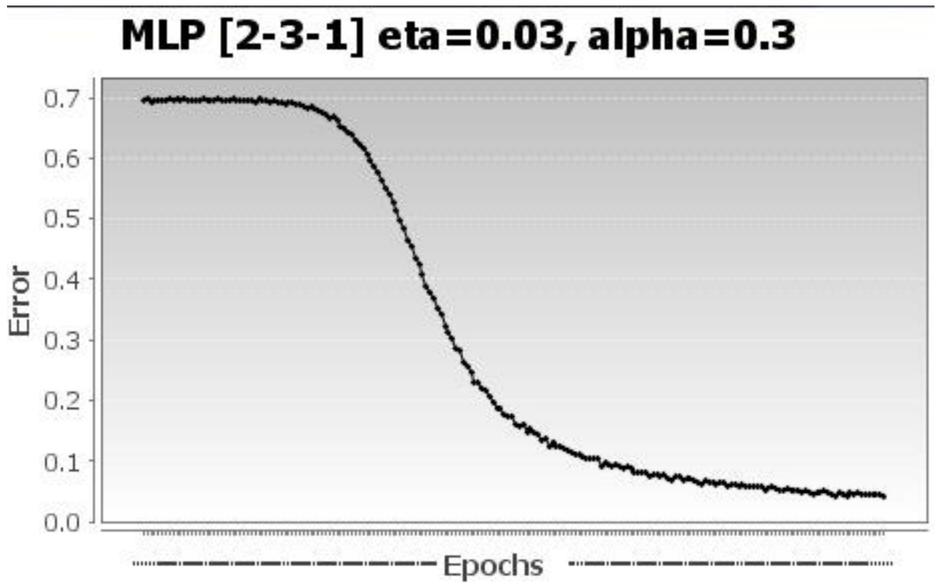
$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial \varepsilon_j^{(t)}}{\partial w_{ij}} + \alpha \Delta w_{ij}^{(t)}$$

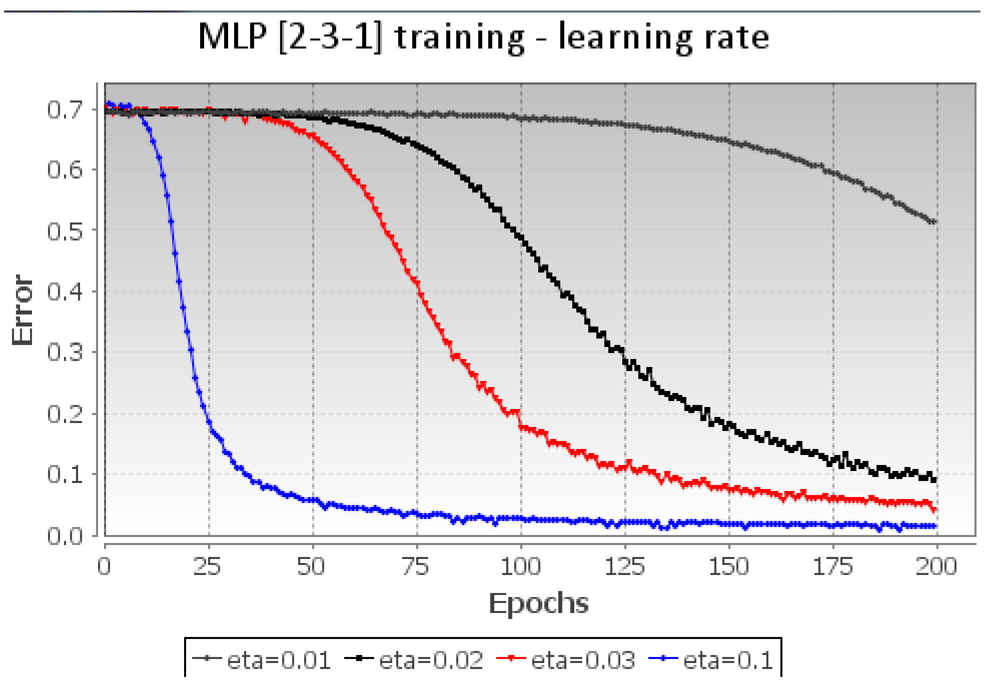
$$\delta_{ih} = \left(\tilde{y}_i - y_i\right).z_h \quad \Delta v_{ih} = -\delta_{ih}$$

$$\delta_{hi} = \sum_{j=0}^{k-1} \left\{\left(\tilde{y}_j - y_j\right).v_{jh}\right\}.z_h\left(1 - z_h\right).x_i \quad \Delta w_{hi} = -\eta \delta_{hi}$$

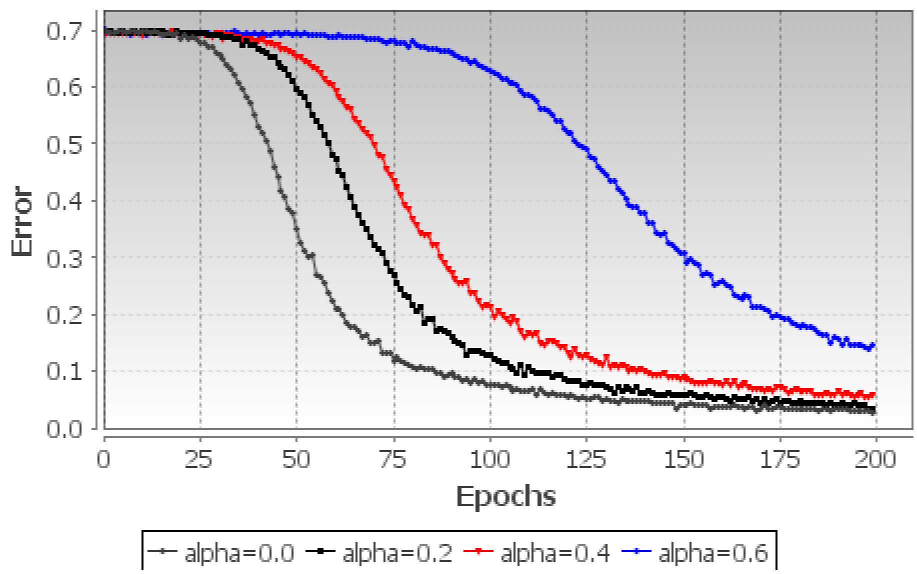




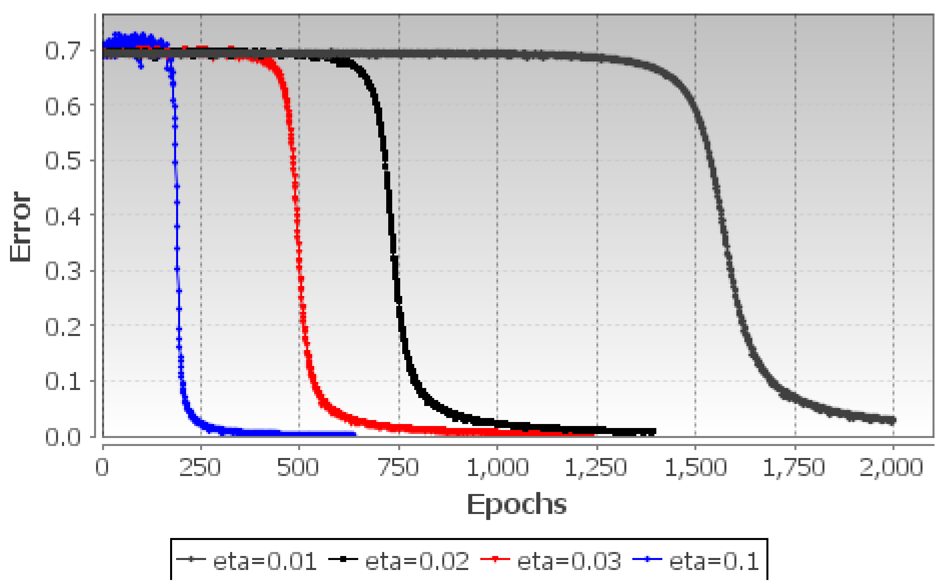

MLP [2-3-1] eta=0.03, alpha=0.3
Error
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Epochs

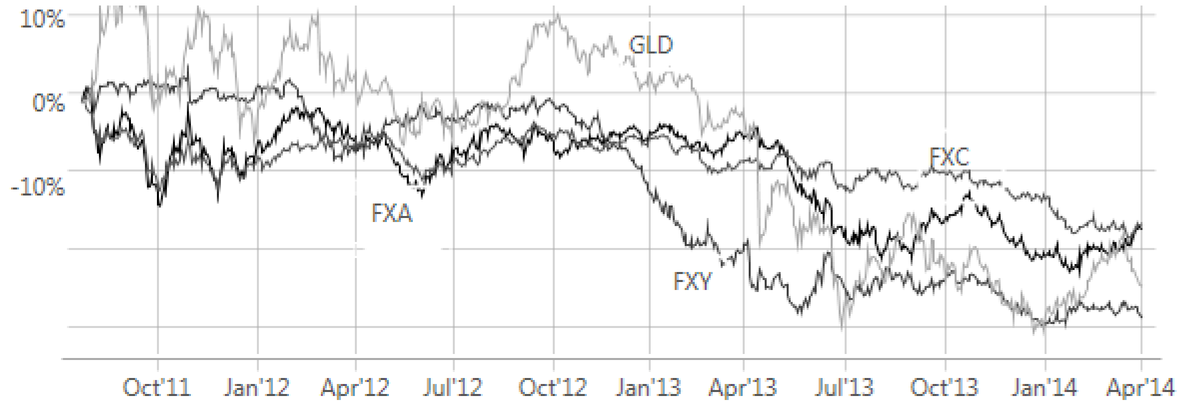

MLP [2-3-1] training - learning rate
Error
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
25
50
75
100
125
150
175
200
Epochs
eta=0.01
eta=0.02
eta=0.03
eta=0.1

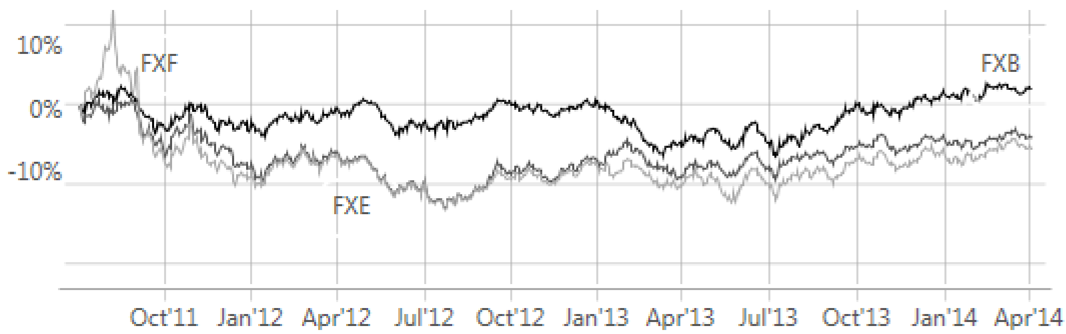
MLP [2-3-1] training - momentum

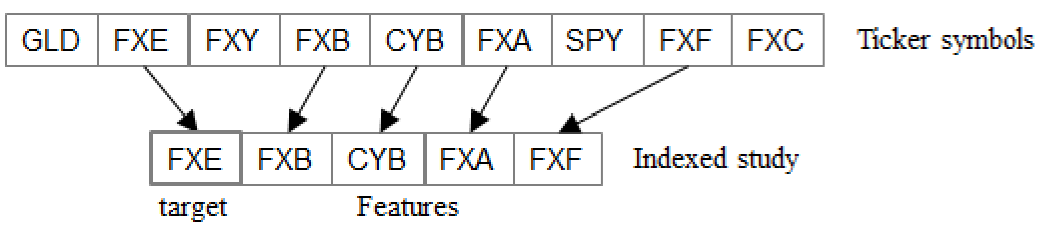


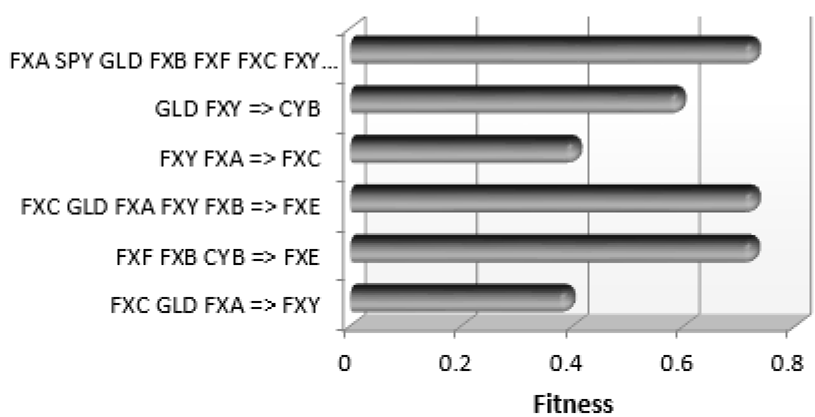
MLP [2-7-3-1] training - learning rate

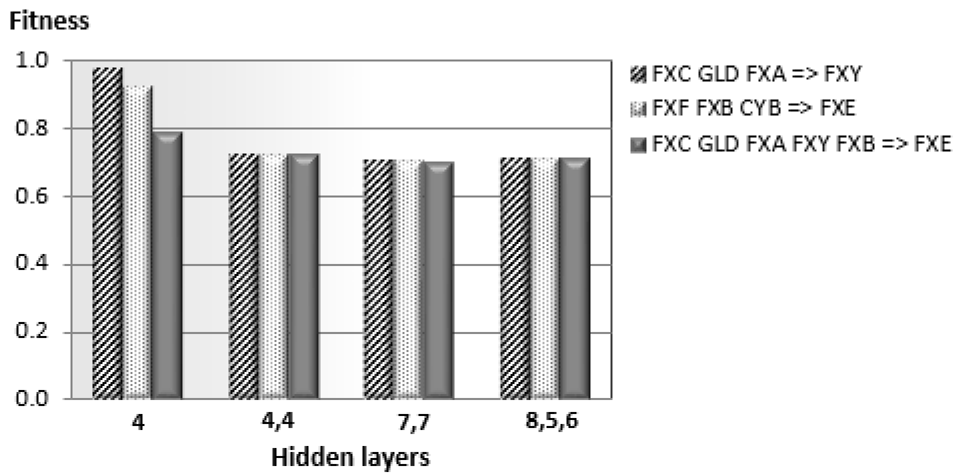


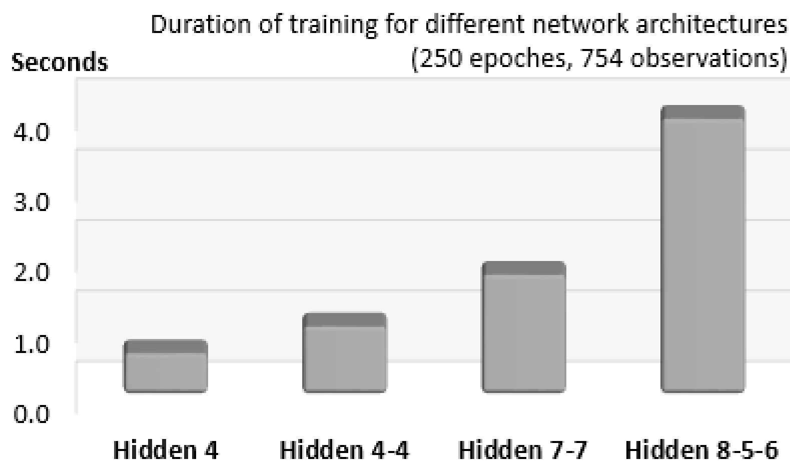

10%
0%
-10%
GLD
FXC
FXA
FXY
Oct'11 Jan'12 Apr'12 Jul'12 Oct'12 Jan'13 Apr'13 Jul'13 Oct'13 Jan'14 Apr'14

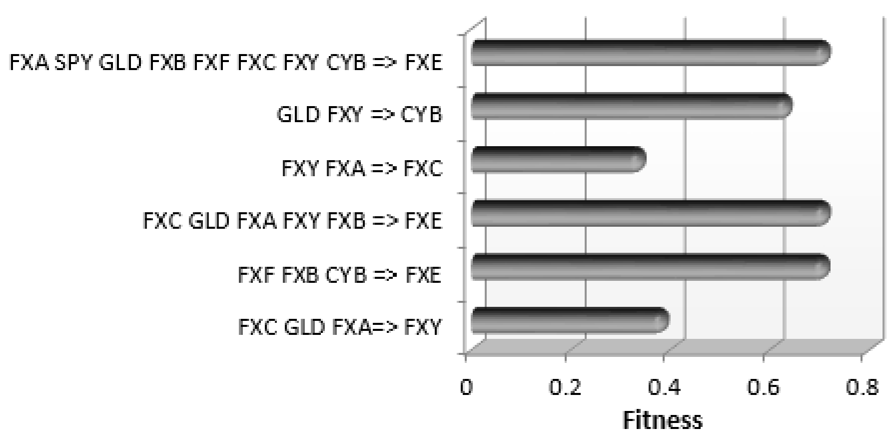

10%
0%
-10%
FXF
FXB
FXE
Oct'11 Jan'12 Apr'12 Jul'12 Oct'12 Jan'13 Apr'13 Jul'13 Oct'13 Jan'14 Apr'14

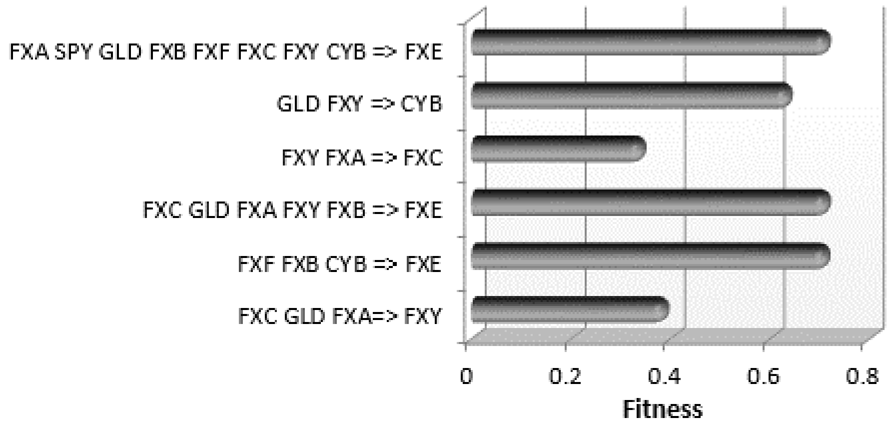

GLD FXE FXY FXB CYB FXA SPY FXF FXC
Ticker symbols
FXE FXB CYB FXA FXF
Indexed study
target
Features


FXA SPY GLD FXB FXF FXC FXY...
GLD FXY => CYB
FXY FXA => FXC
FXC GLD FXA FXY FXB => FXE
FXF FXB CYB => FXE
FXC GLD FXA => FXY
0
0.2
0.4
0.6
0.8
Fitness

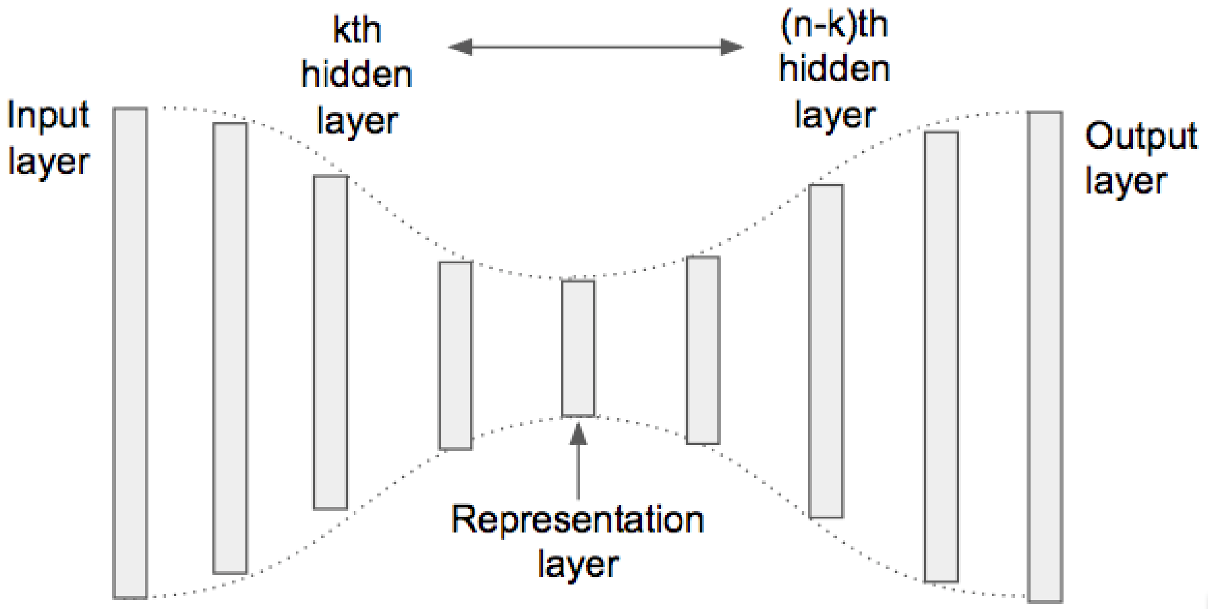

Fitness
1.0
0.8
0.6
0.4
0.2
0.0
FXC GLD FXA => FXY
FXF FXB CYB => FXE
FXC GLD FXA FXY FXB => FXE
4
4,4
7,7
8,5,6
Hidden layers


Duration of training for different network architectures
(250 epoches, 754 observations)
Seconds
4.0
3.0
2.0
1.0
0.0
Hidden 4
Hidden 4-4
Hidden 7-7
Hidden 8-5-6


FXA SPY GLD FXB FXF FXC FXY CYB => FXE
GLD FXY => CYB
FXY FXA => FXC
FXC GLD FXA FXY FXB => FXE
FXF FXB CYB => FXE
FXC GLD FXA=> FXY
0
0.2
0.4
0.6
0.8
Fitness

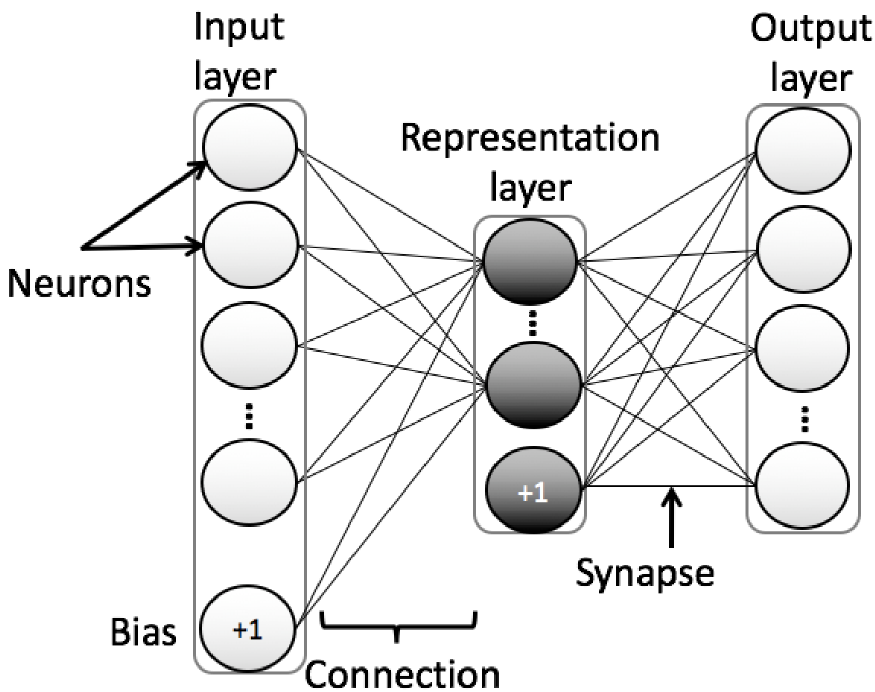

FXA SPY GLD FXB FXF FXC FXY CYB => FXE
GLD FXY => CYB
FXY FXA => FXC
FXC GLD FXA FXY FXB => FXE
FXF FXB CYB => FXE
FXC GLD FXA=> FXY
0
0.2
0.4
0.6
0.8
Fitness

Chapter 11: Deep Learning

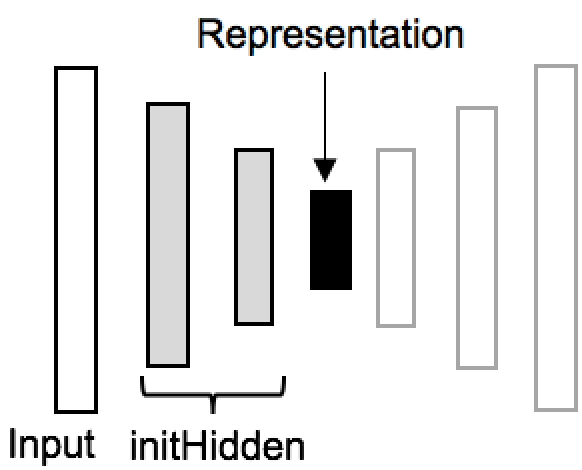


$$x' = \varnothing(\psi(x)) loss = \mathcal{L}(x, \varnothing(\psi(x)))$$

$$h = \varnothing(x) = \sigma(w^t.x + b)$$
$$x' = \psi(h) = \sigma'(w''.h + b)$$
$$loss = \|x - x'\|^2$$



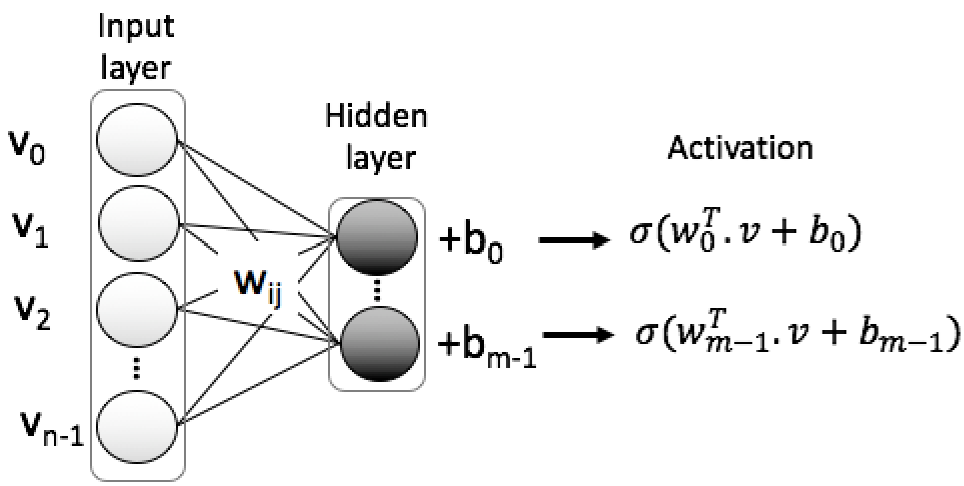
$$\rho_i' = \lambda . f\left(w_i^T . x + b_i'\right) + \left(1-\lambda\right) . \rho$$
$$b_i' = b_i - \alpha\beta\left(\rho_i - \rho_i'\right)$$



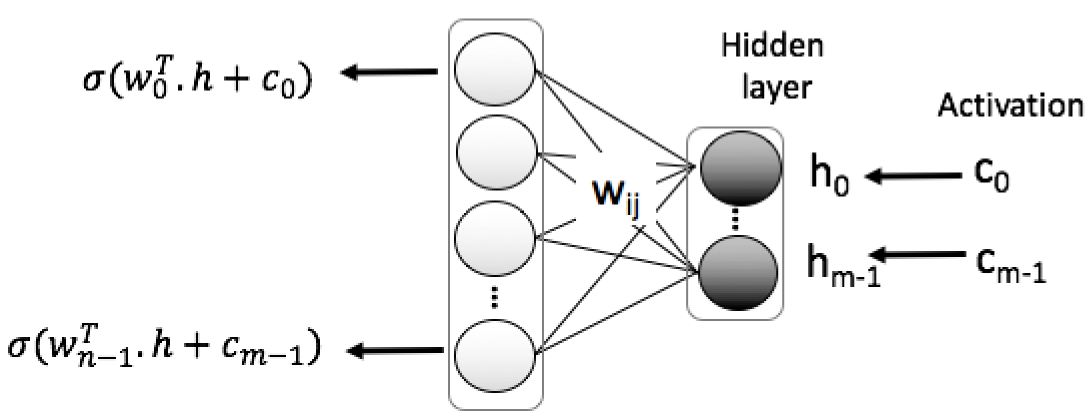
$$E\left(x\right) = -\left(x^T W x + b^T x\right) = -\sum_{i<j}^{n-1} w_{ij} x_i x_j - \sum_{i=0}^{n-1} b_i x_i$$

$$p(x=1) = \frac{1}{1+e^{-\frac{E(x=1)-E(x=0)}{T}}}$$

$$p(v|h) = \prod_{j=0}^{n-1} p(v_j|x) \quad p(v_j=1|h) = \sigma\left(\sum_{i=0}^{n-1} w_{ij}h_i + b_j\right)$$



$$p(h|v) = \prod_{i=0}^{m-1} p(h_i|v) \quad p(h_i=1|v) = \sigma\left(\sum_{j=0}^{m-1} w_{ij}v_j + c_i\right)$$



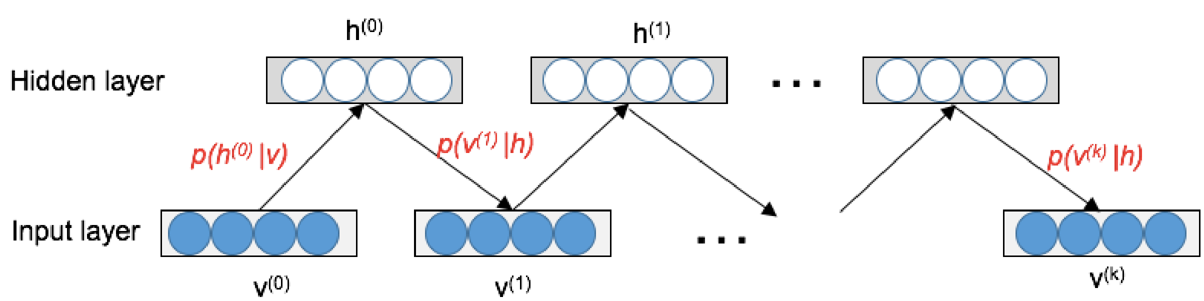
$$p(v \mid w) = \frac{1}{Z(w)} \tilde{p}(v \mid w)$$

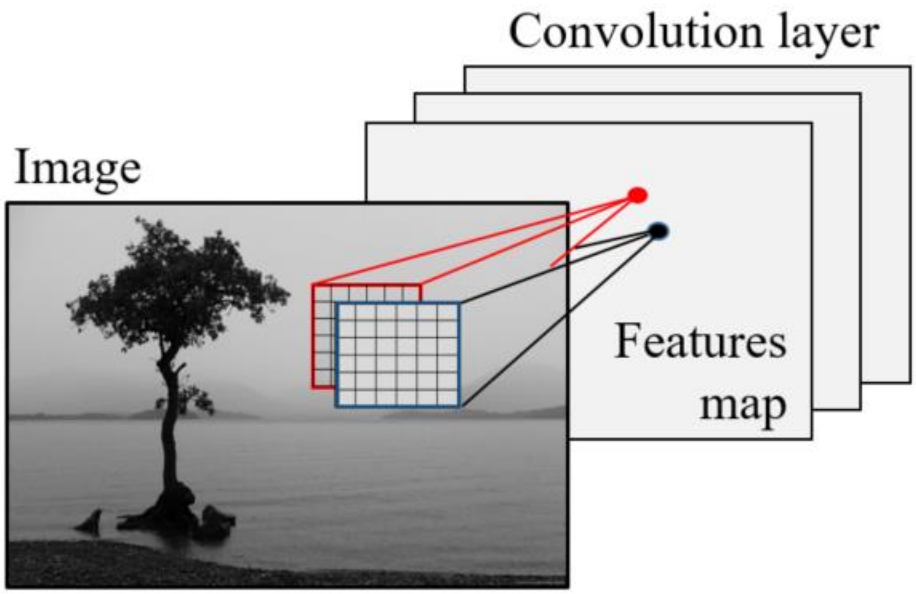
$$\frac{\partial \ln p(v \mid w)}{\partial w} = \frac{\partial \ln \tilde{p}(v \mid w)}{\partial w} - \frac{\partial Z(w)}{\partial w}$$

$$\frac{\partial \ln\left(\mathcal{L}(w \mid v)\right)}{\partial w_{ij}} \triangleq \Delta_e = \frac{1}{n}(e_+ - e_-)$$

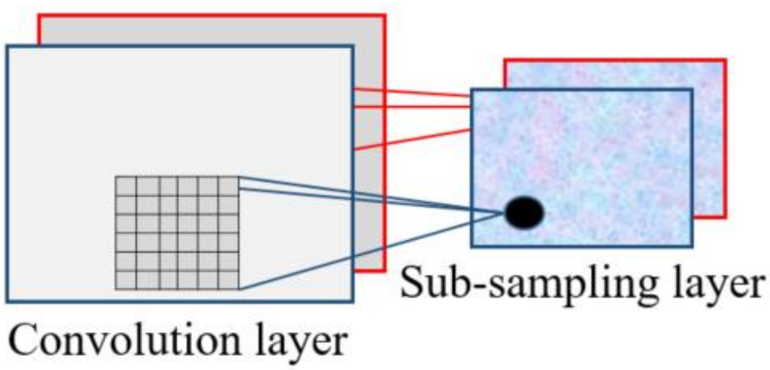
$$e_+ = v^{(0)}.p\left(v^{(0)} \mid h\right)$$

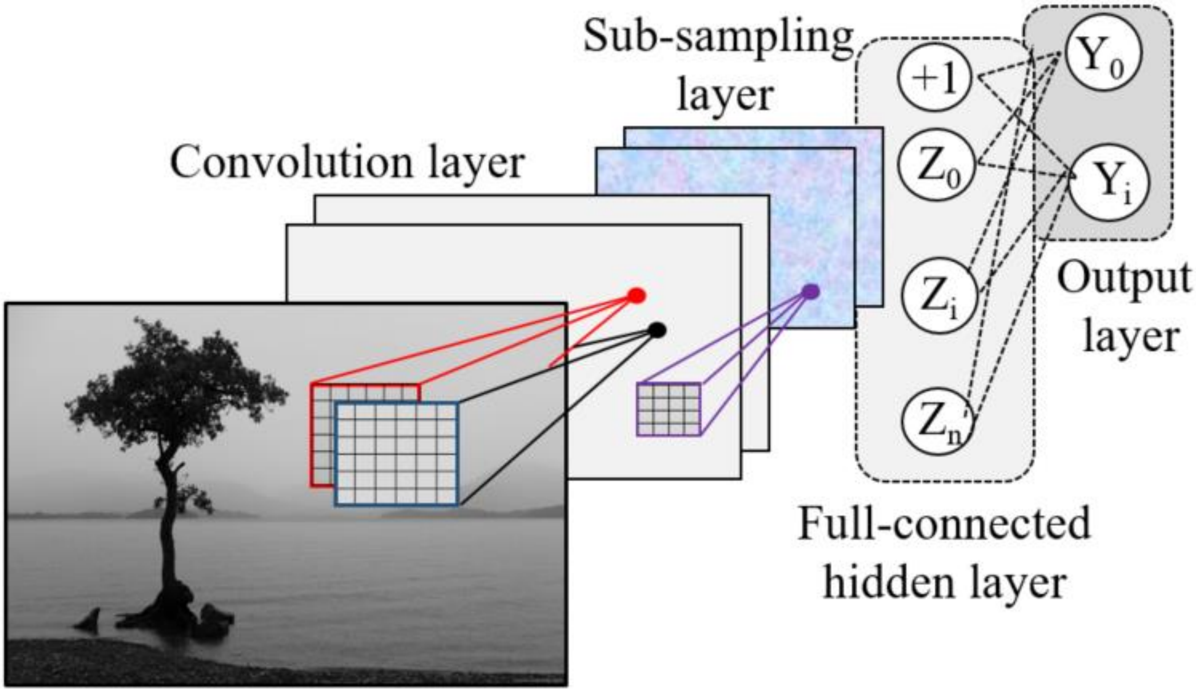
$$e_- = v^{(1)}.p\left(v^{(1)} \mid h\right) \; v^{(1)} = p\left(h^{(0)} \mid v\right) \;\; h^{(0)} = p\left(v^{(0)} \mid h\right)$$



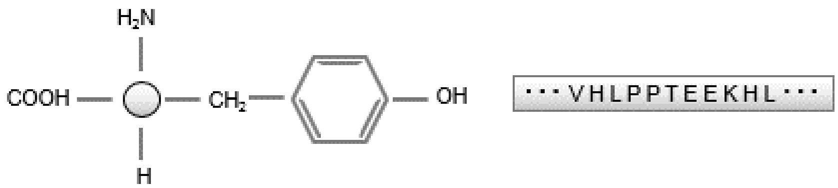


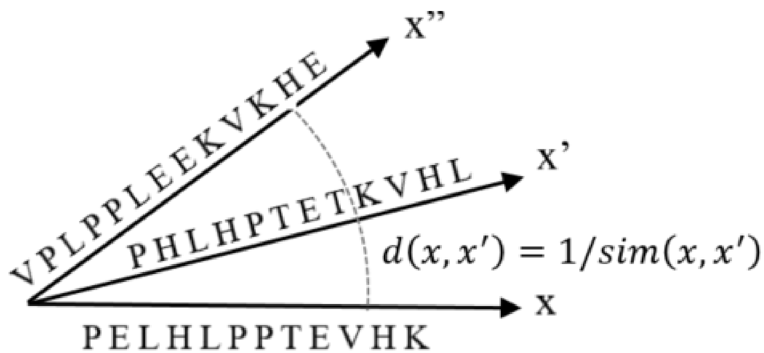
$$\tilde{z}_j = \sigma\left(w_0 + \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} w_{u,v} x_{j+u,i+v} \right)$$




Sub-sampling
layer
Convolution layer
+1
Z_0
Z_i
Z_n
Y_0
Y_i
Output
layer
Full-connected
hidden layer

Chapter 12: Kernel Models and SVM





$$sim\left(x_{cp}, x'_{c'p'}\right) = \frac{1}{mx}\sum_{i=1}^{mx}\left(c = c'\right) \cap \left(p = p'\right) \; mx = \max\left(n, n'\right)$$

$$f\left(x \mid w\right) = w_0 + \sum_{d=1}^{D} w_d \varnothing_d\left(x\right) \;\; \varnothing_d : \mathbb{R} \rightarrow \mathbb{R}$$

$$K\left(x, x'\right) = \varnothing\left(x\right).\varnothing\left(x'\right) = \sum_{d=1}^{D} \varnothing_d\left(x\right) \varnothing_d\left(x'\right)$$

$$K\left(x, x'\right) = x^T x' = \sum_{d=1}^{D} x_i . x'_i$$

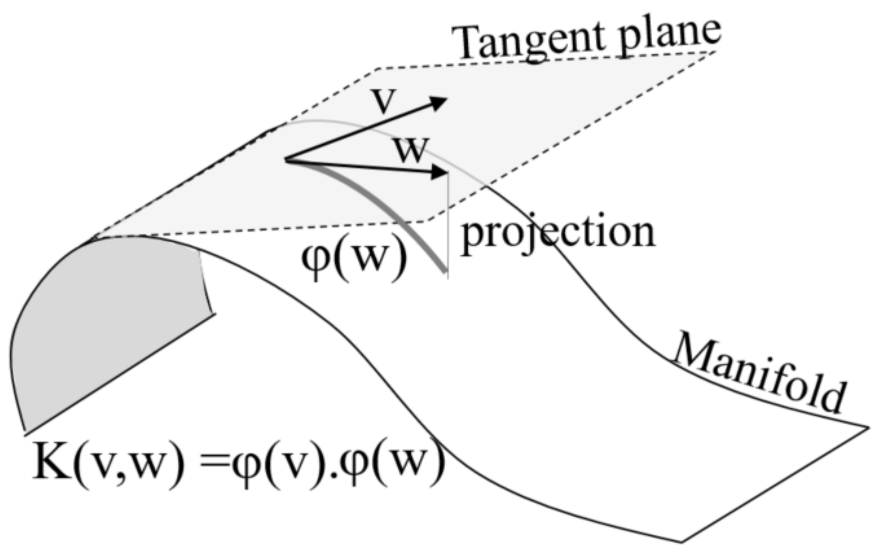
$$K(x, x') = (\gamma x^T x' + c)^n \;\; \gamma > 0, c \geq 0$$

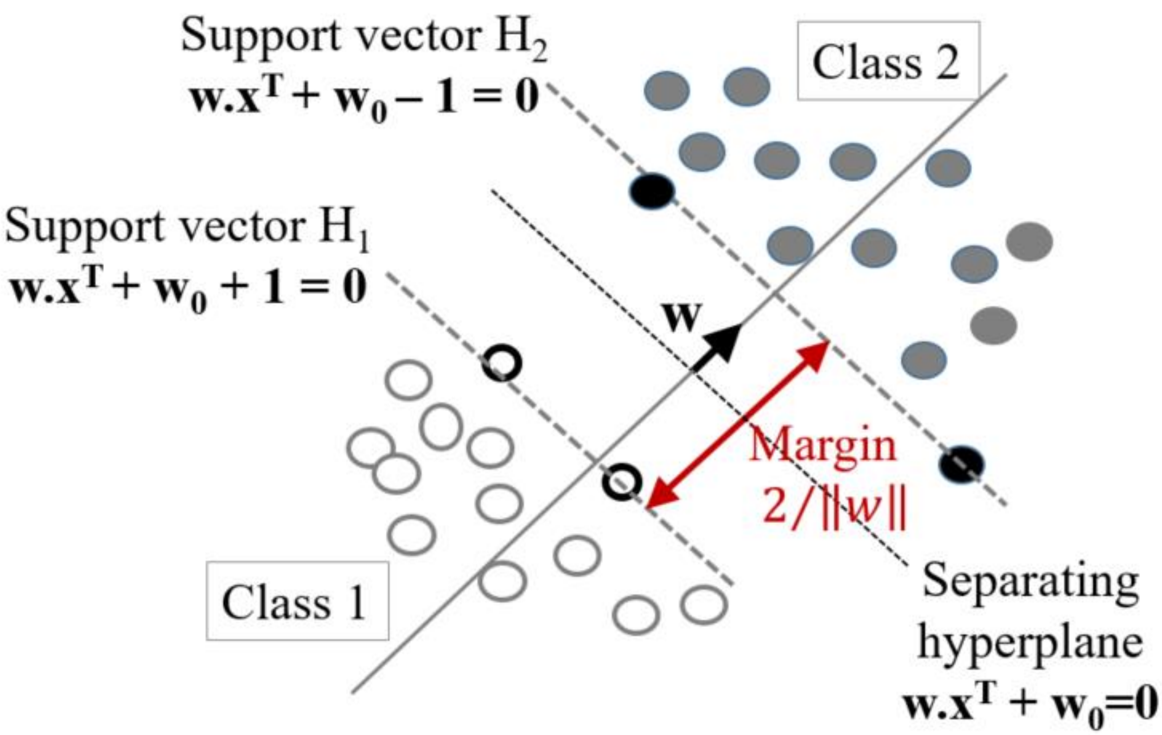
$$K(x, x') = \tanh(\gamma x^T x' + c) \;\; \gamma > 0, c \geq 0$$

$$K(x, x') = e^{-\gamma \|x - x'\|^2} \gamma > 0$$

$$K(x, x') = e^{-\gamma \|x - x'\|} \gamma > 0$$

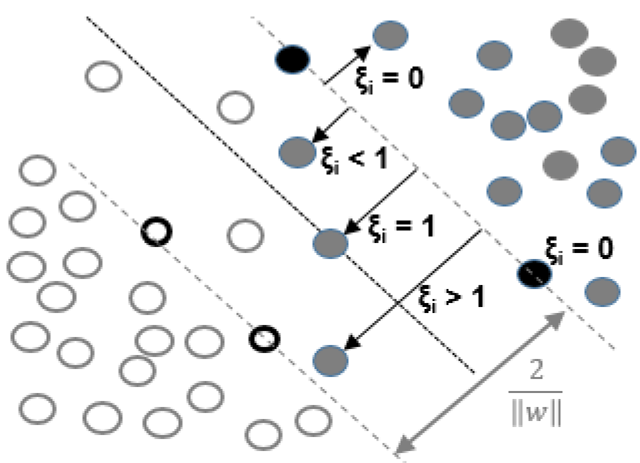
$$K(x, x') = -\log(1 + \|x - x'\|^n)$$





$$y_i\left(w^T x + w_0\right) \geq 1 \ \forall i$$

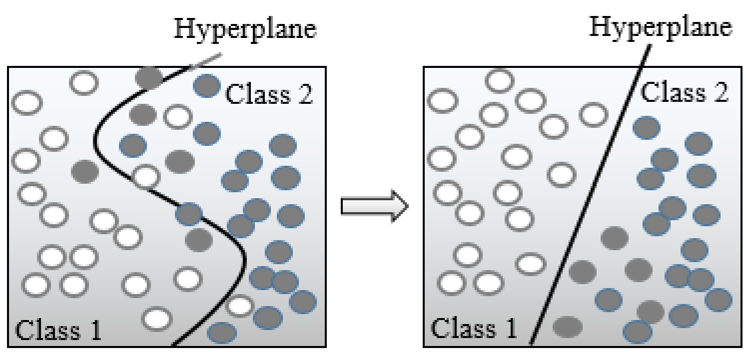
$$\min_{w,w_0}\left\{\frac{w^T w}{2}\right\} \textit{subject to } y_i\left(w^T x + w_0\right) \geq 1 \ \forall i$$



$$\min_{w,,\xi}\left\{\frac{w^T w}{2}+C\sum_{i=0}^{n-1}\xi_i\right\}$$
$$\xi_i \geq 0,\ \ y_i\left(w^T x+w_0\right)\geq 1-\xi_i \ \ \forall i$$

$$\min_{w,w_0}\left\{\frac{w^T w}{2}+C\sum_{i=0}^{n-1}\mathcal{L}_i\right\}\mathcal{L}_i=\left|1-y_i\left(w^T x+w_0\right)\right|$$

$$\min_{w,\rho,\xi}\left\{\frac{w^T w}{2}-\rho+\frac{1}{vn}\sum_{i-0}^{n-1}\xi_i\right\}$$
$$\xi_i \geq 0,\ \ y_i\left(w^T x+w_0\right)\geq \rho-\xi_i \ \ \forall i$$

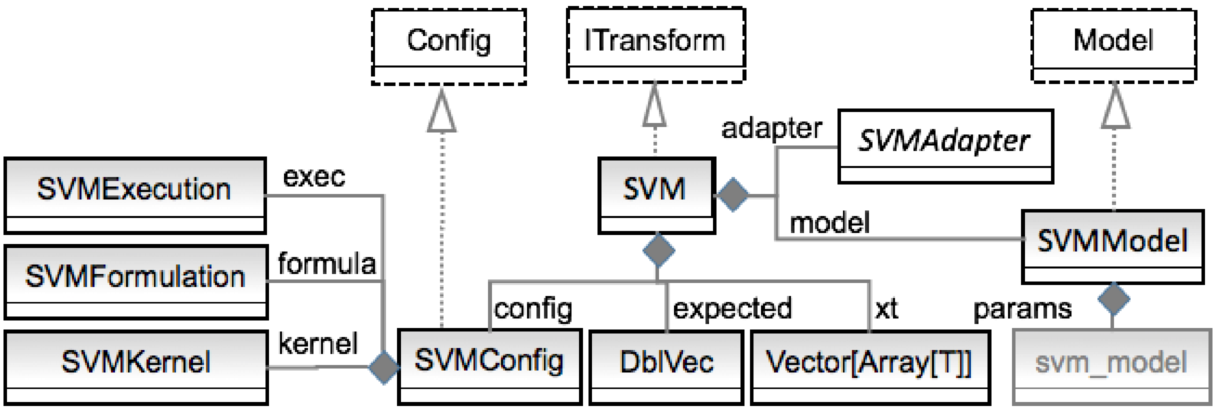


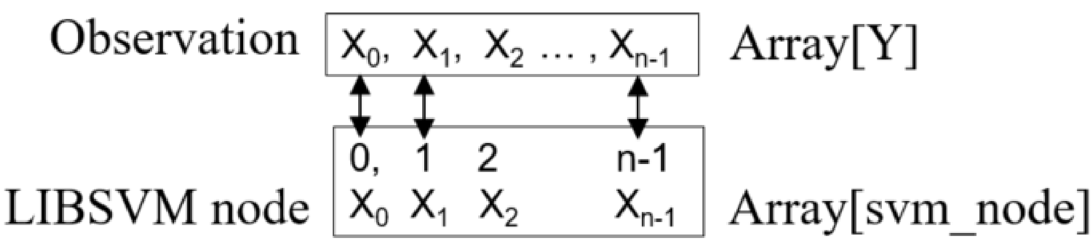
$$y_i\left(w^T\varnothing\left(x\right)+w_0\right)\geq 1-\xi_i \ \ \xi_i\geq 0 \ \forall i$$

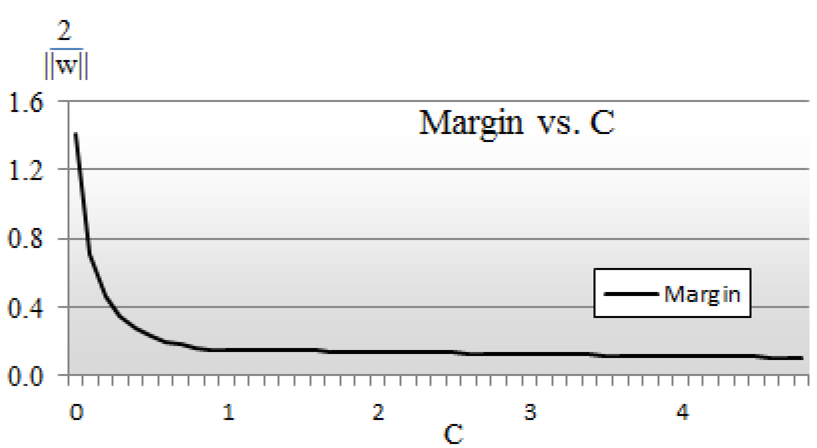
$$w^*=\sum_{i=0}^{n-1}\alpha_i y_i \varnothing\left(x_i\right)$$

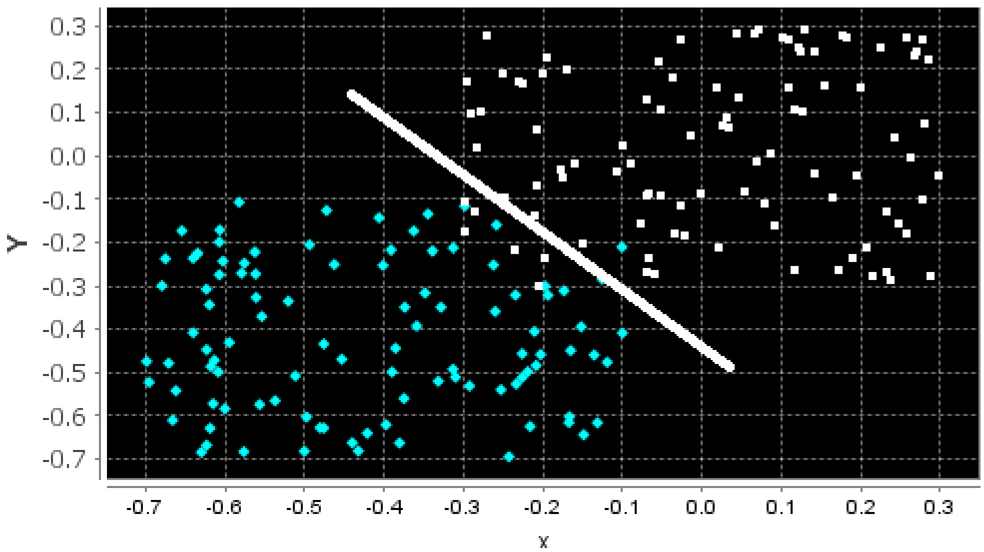
$$y_i\left(w^T.\varnothing\left(x\right)+w_0\right)=y_i\left(\sum_{i=0}^{n-1}\alpha_i y_i K\left(x_i,x\right)+w_0\right)\geq 1$$
$$K\left(x_i,x\right)=\varnothing\left(x_i\right)\varnothing\left(x\right)\forall_i$$

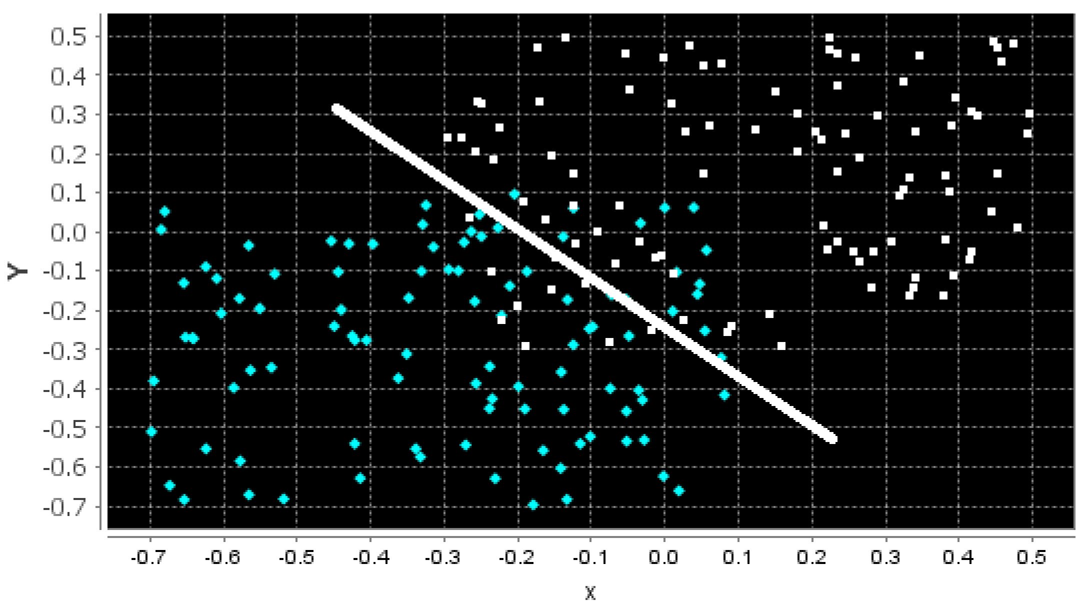
$$K(x_i, x) = (1 + x^T x')^2$$
$$= 1 + 2x_1x_1' + 2x_2x_2' + 2x_1x_1'x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2$$
$$= \varnothing_1(x).\varnothing_1(x') + \varnothing_2(x).\varnothing_2(x') + \varnothing_3(x).\varnothing_3(x') + \cdots$$
$$\varnothing_2(x) = 1, \varnothing_2(x) = \sqrt{2}x_1, \varnothing_3(x) = \sqrt{2}x_2, \varnothing_4(x) = x_1^2 \cdots$$

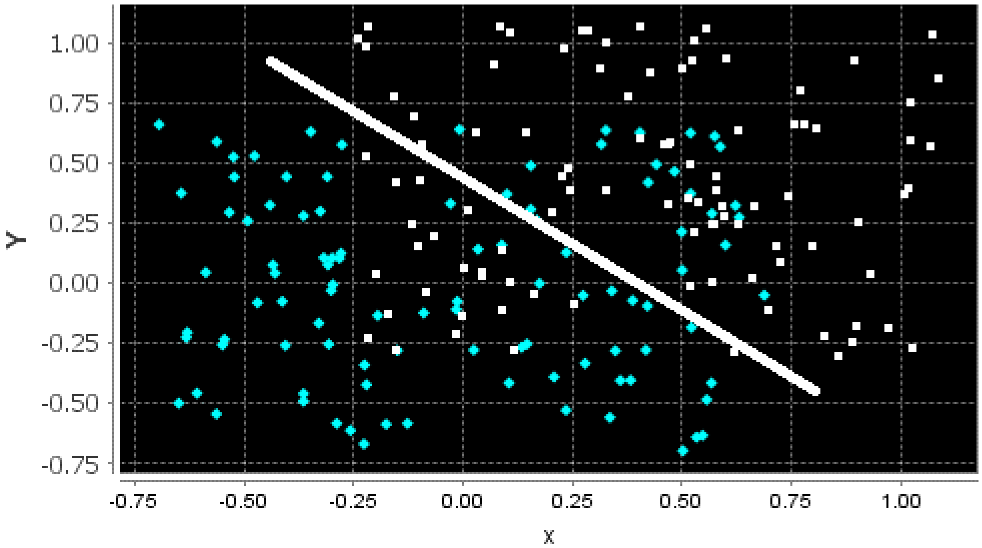


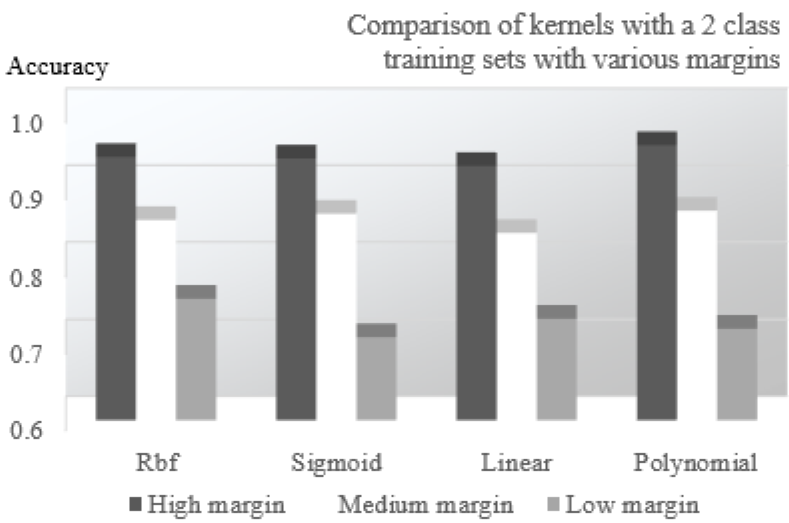


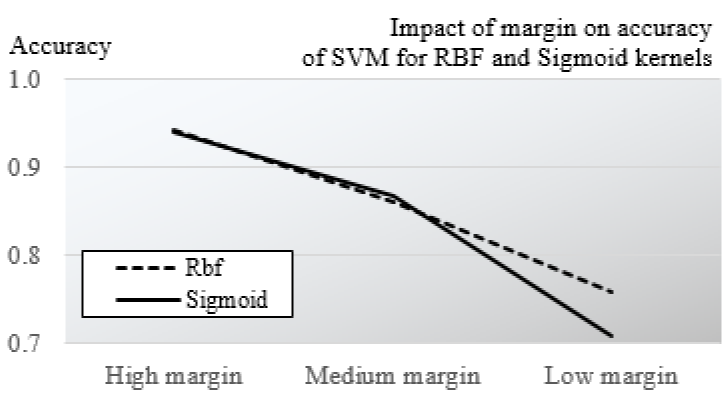


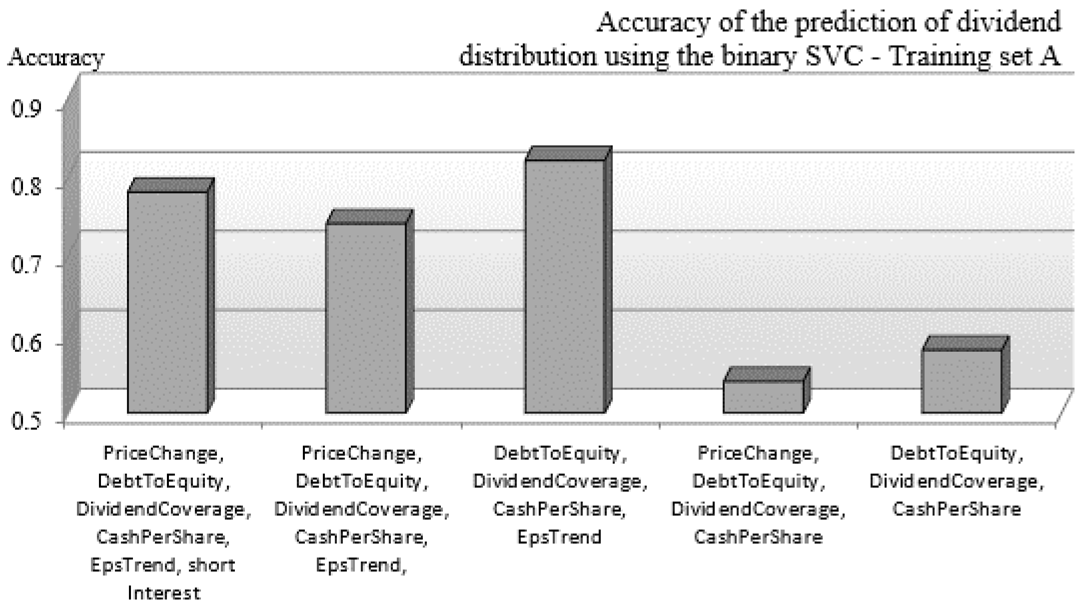

0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
Y
-0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3
x


0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
Y
-0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5
x

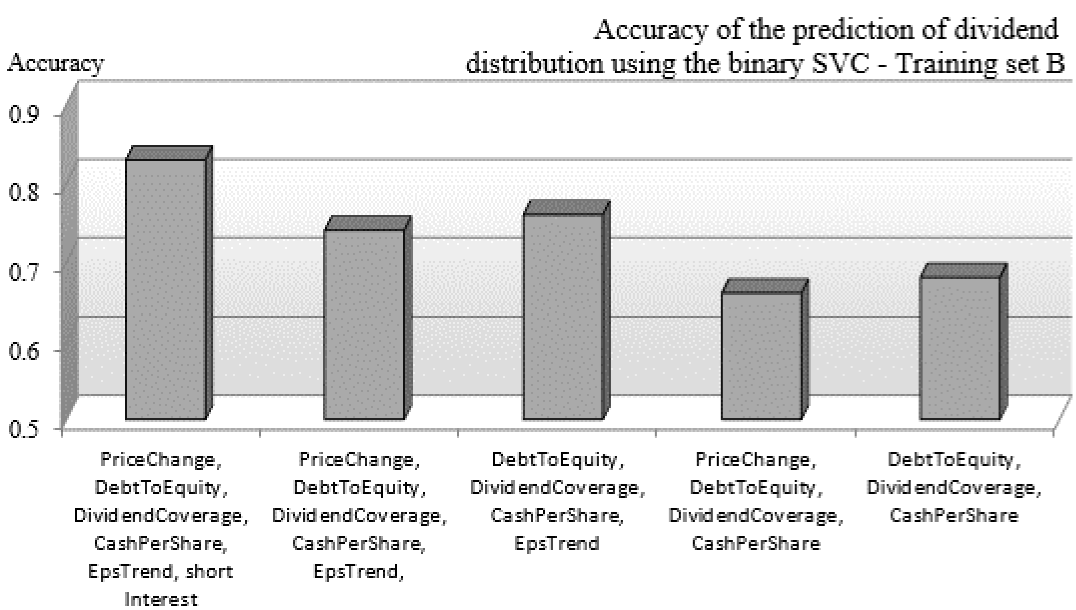

1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
Y
-0.75
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
X


Comparison of kernels with a 2 class
training sets with various margins
Accuracy
1.0
0.9
0.8
0.7
0.6
Rbf
Sigmoid
Linear
Polynomial
High margin
Medium margin
Low margin

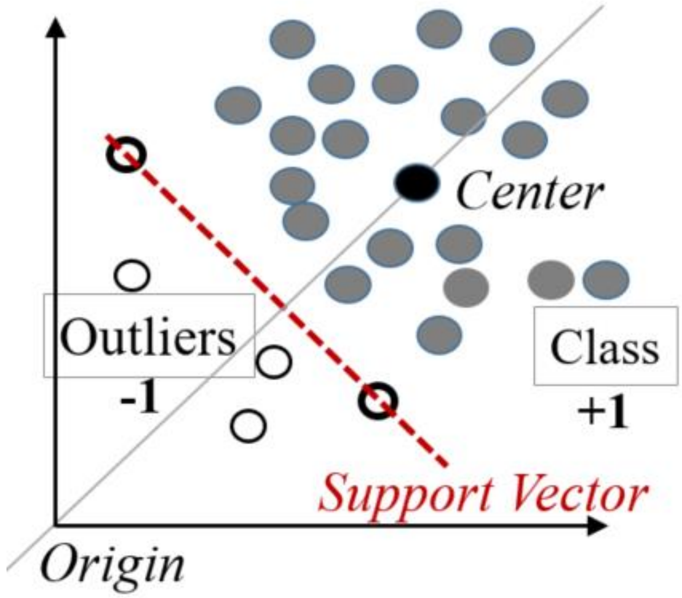

Impact of margin on accuracy
of SVM for RBF and Sigmoid kernels
Accuracy
1.0
0.9
0.8
0.7
Rbf
Sigmoid
High margin
Medium margin
Low margin

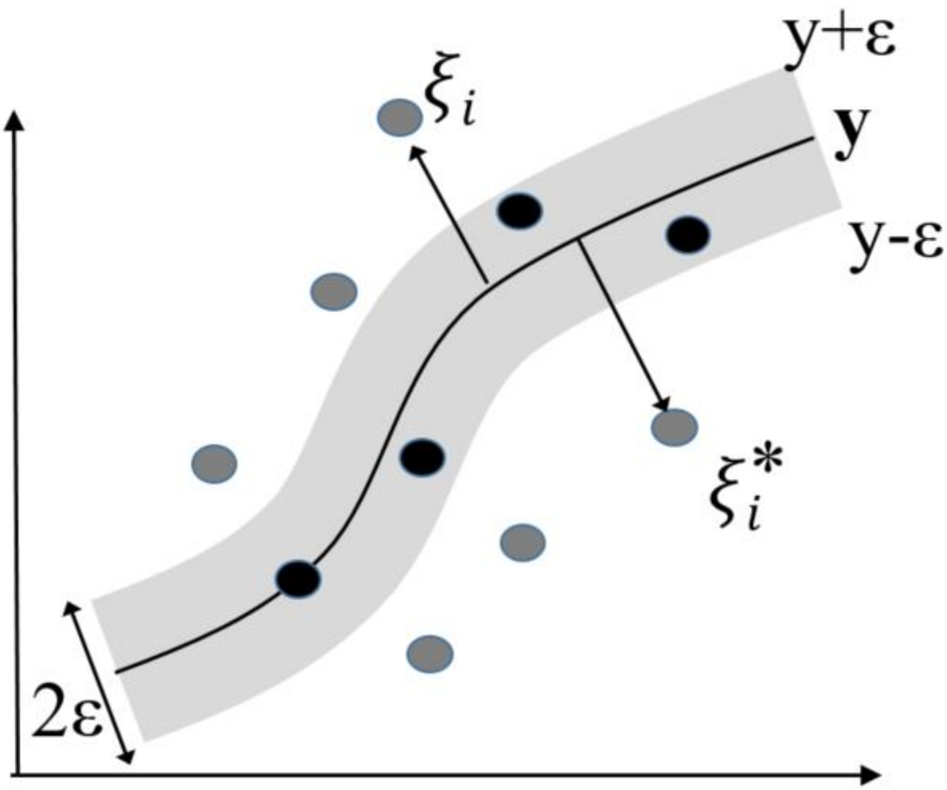


Accuracy of the prediction of dividend distribution using the binary SVC - Training set A
Accuracy
0.9
0.8
0.7
0.6
0.5
PriceChange, DebtToEquity, DividendCoverage, CashPerShare, EpsTrend, short Interest
PriceChange, DebtToEquity, DividendCoverage, CashPerShare, EpsTrend,
DebtToEquity, DividendCoverage, CashPerShare, EpsTrend
PriceChange, DebtToEquity, DividendCoverage, CashPerShare
DebtToEquity, DividendCoverage, CashPerShare



Accuracy of the prediction of dividend distribution using the binary SVC - Training set B
Accuracy
0.9
0.8
0.7
0.6
0.5
PriceChange, DebtToEquity, DividendCoverage, CashPerShare, EpsTrend, short Interest
PriceChange, DebtToEquity, DividendCoverage, CashPerShare, EpsTrend,
DebtToEquity, DividendCoverage, CashPerShare, EpsTrend
PriceChange, DebtToEquity, DividendCoverage, CashPerShare
DebtToEquity, DividendCoverage, CashPerShare

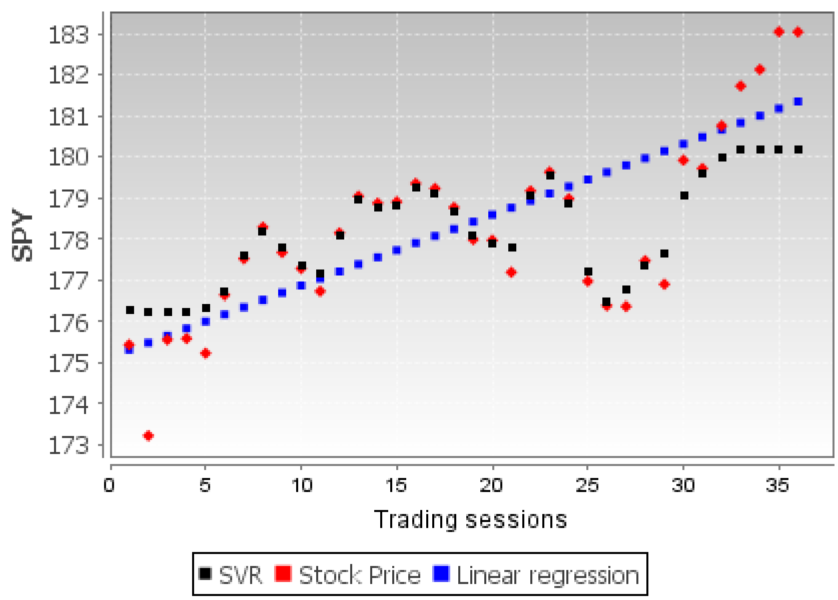

Center
Outliers
-1
Class
+1
Support Vector
Origin

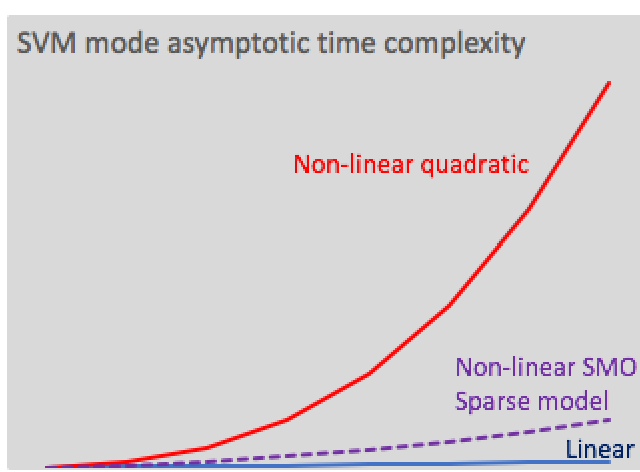

y+ε
ξi
y
y-ε
ξi*
2ε

$$\min_{w,\xi,\xi^*} \left\{ \frac{w^T w}{2} + C \sum_{i=0}^{n-1} \left(\xi_i + \xi_i^* \right) \right\}$$

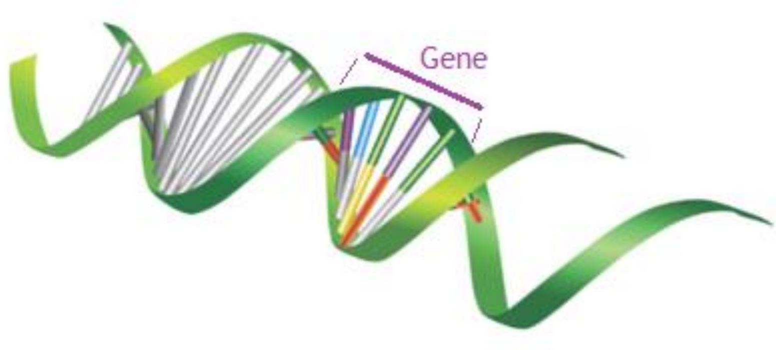
$$-\in -\xi_i^* \leq w^T \varnothing\left(x_i\right) + w_0 - y_i \leq \varepsilon + \xi_i \quad \forall i$$

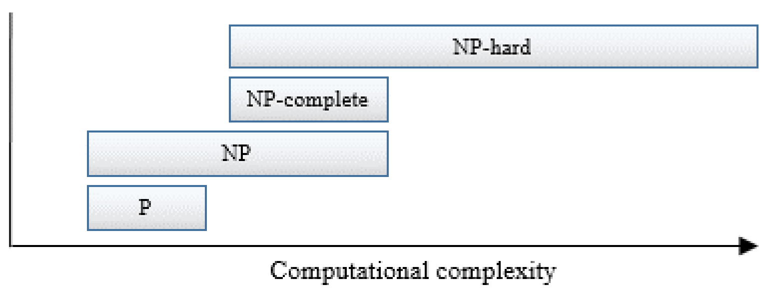
$$\hat{y}\left(x\right) = \sum_{i=0}^{n-1} \alpha_i K\left(x_i, x\right) + \hat{w}_0$$

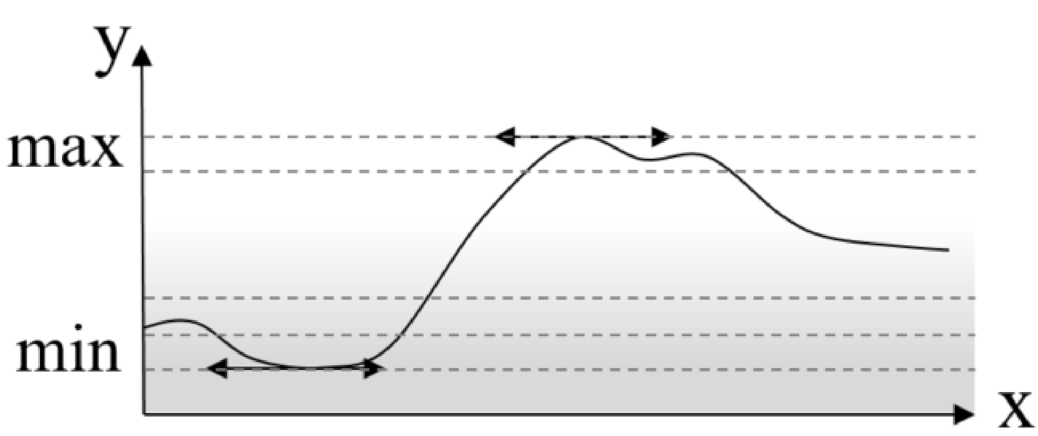




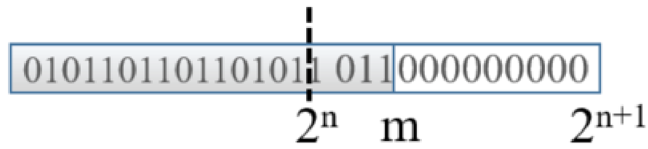
Chapter 13: Evolutionary Computing

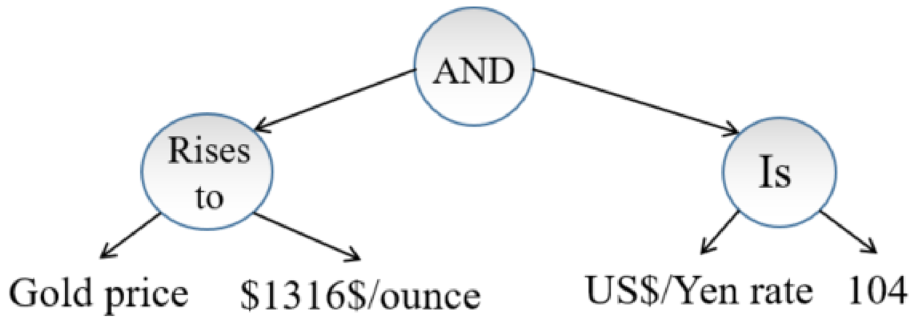


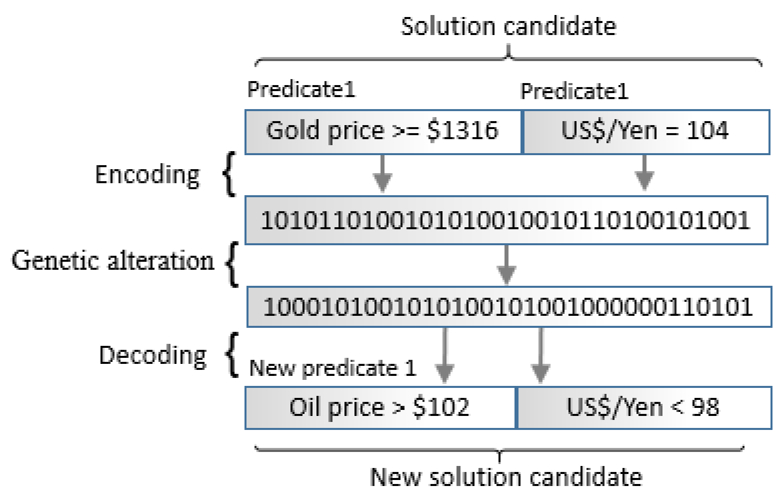


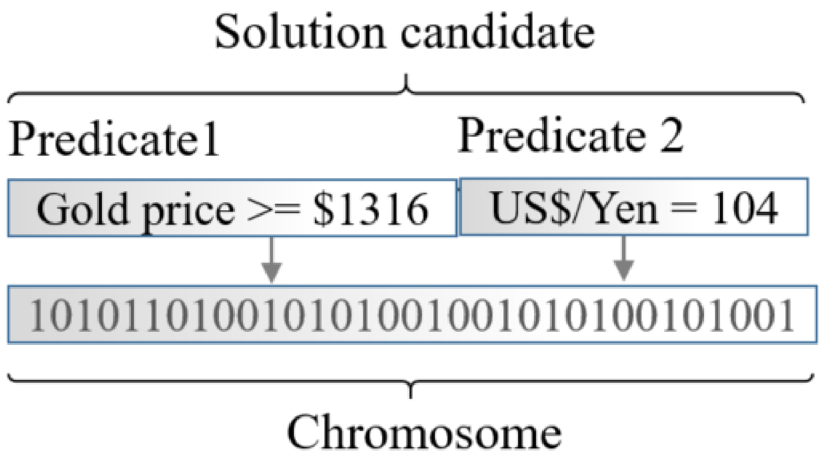


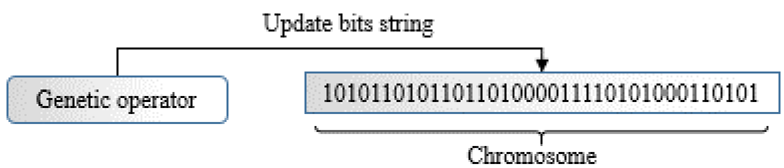
$$step = \frac{max - min}{2^n}$$

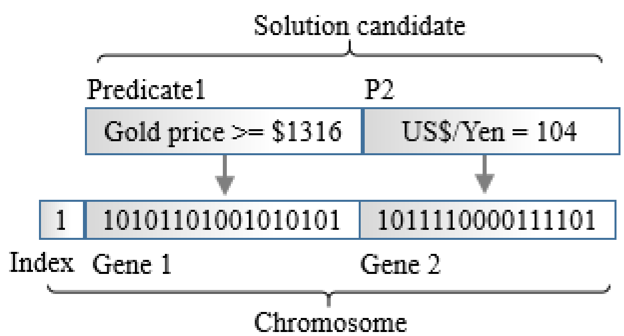

0101101101101011 011000000000
2n
m
2n+1

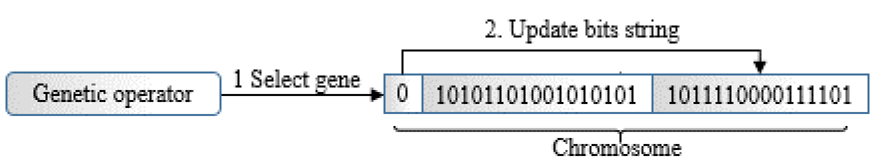

AND
Rises to
Is
Gold price
1316/ounce
US$/Yen rate
104

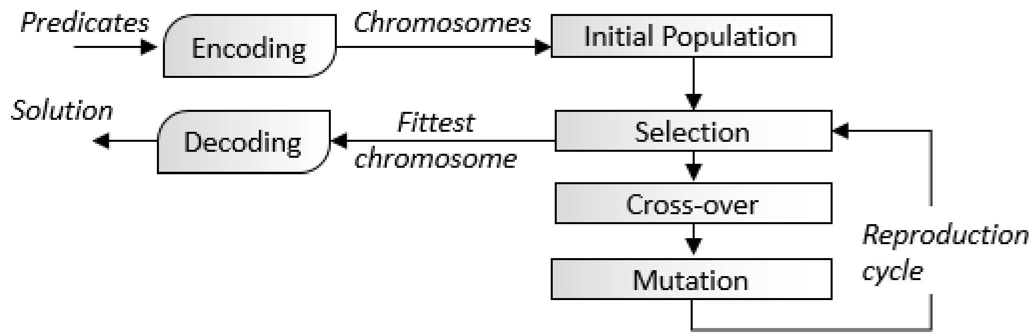

Solution candidate
Predicate1
Predicate1
Gold price >= $1316
US$/Yen = 104
Encoding
10101101001010100100101101001010001
Genetic alteration
1000101001010100101001000000110101
Decoding
New predicate 1
Oil price > $102
US$/Yen < 98
New solution candidate

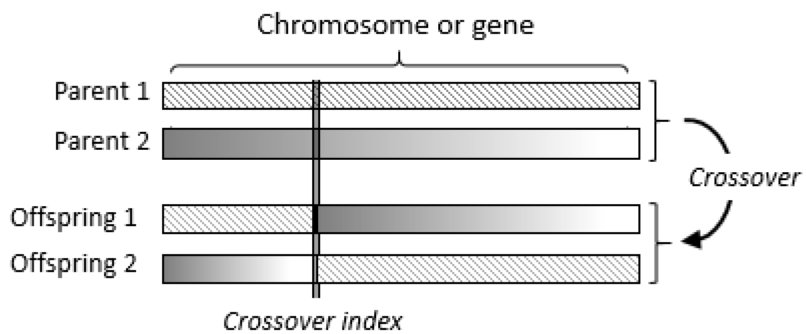

Solution candidate
Predicate1
Predicate 2
Gold price >= $1316
US$/Yen = 104
10101101001010100100101010010100 1
Chromosome

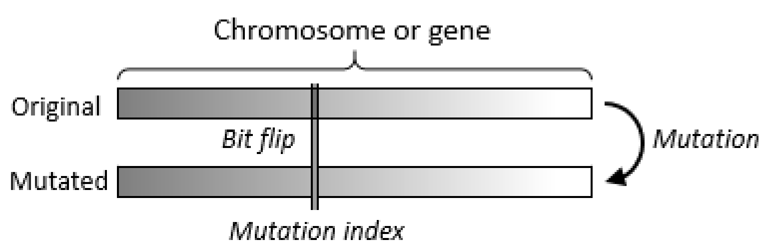

Update bits string
Genetic operator
1010110101101101000011110101000110101
Chromosome

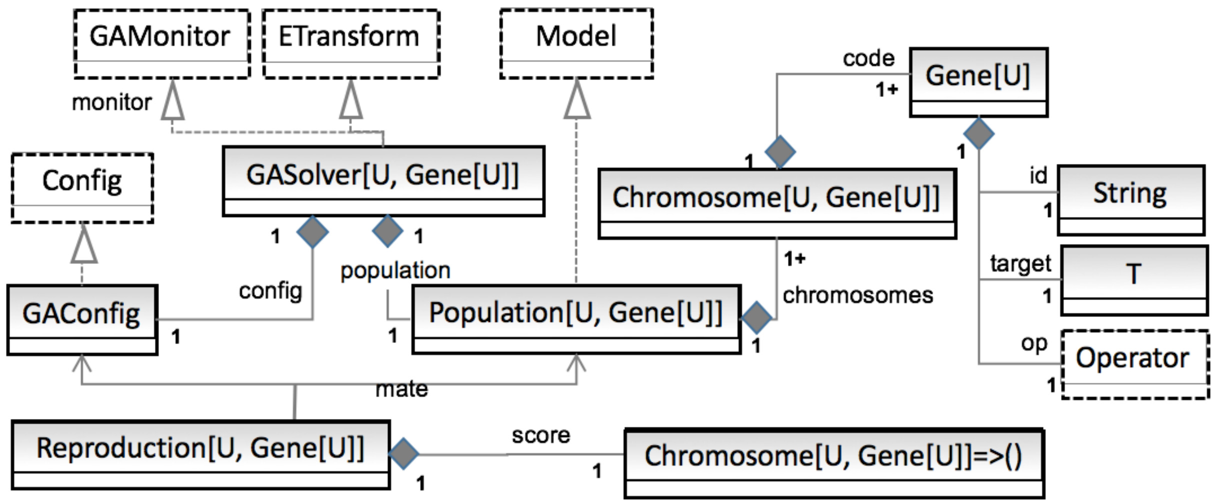

Solution candidate
Predicate1
P2
Gold price >= $1316
US$/Yen = 104
1
101011010010101 01
1011110000111101
Index
Gene 1
Gene 2
Chromosome


2. Update bits string
Genetic operator
1 Select gene
0
10101101001010101
1011110000111101
Chromosome

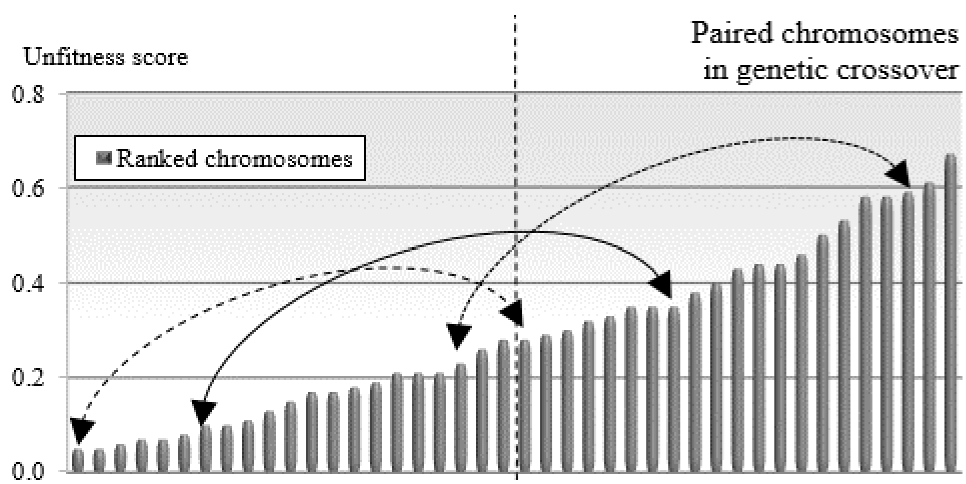

Predicates
Encoding
Chromosomes
Initial Population
Solution
Decoding
Fittest chromosome
Selection
Cross-over
Mutation
Reproduction cycle


Chromosome or gene
Parent 1
Parent 2
Crossover
Offspring 1
Offspring 2
Crossover index

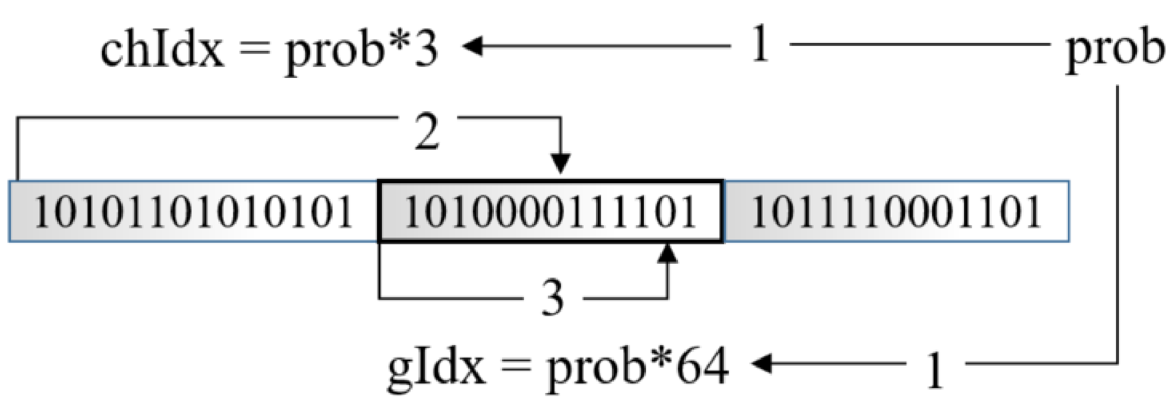

Chromosome or gene
Original
Bit flip
Mutation
Mutated
Mutation index



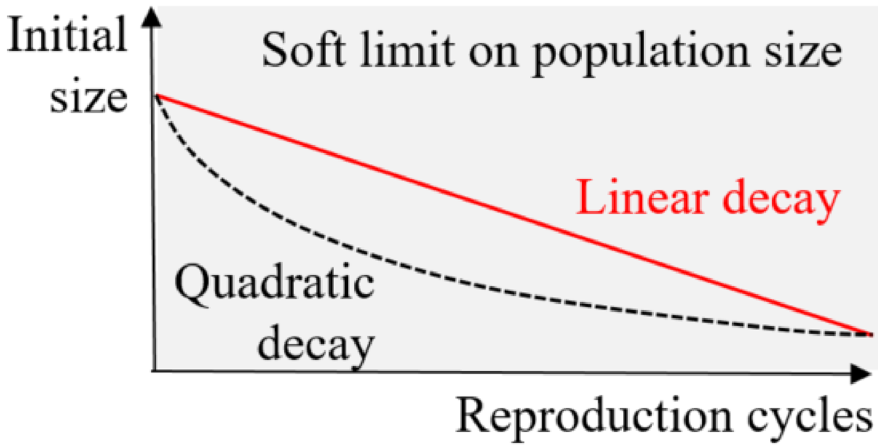
GAMonitor
ETransform
Model
code
Gene[U]
1+
monitor
Config
GASolver[U, Gene[U]]
1
Chromosome[U, Gene[U]]
1
id
String
1
1
1
population
1+
target
T
config
chromosomes
1
GAConfig
Population[U, Gene[U]]
1
1
1
op
Operator
1
mate
Reproduction[U, Gene[U]]
score
Chromosome[U, Gene[U]]=>()
1
1

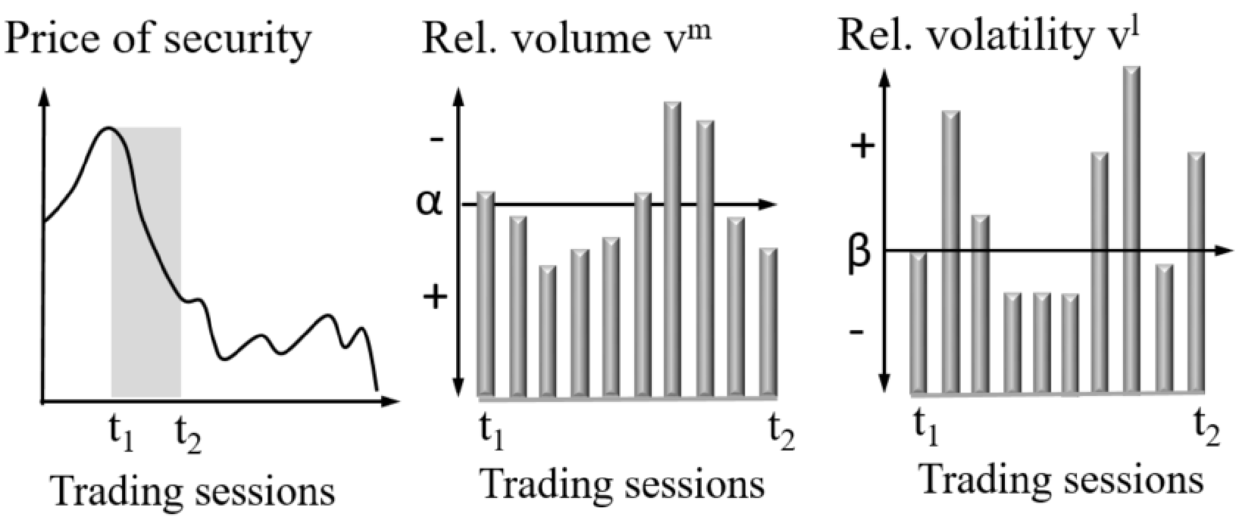


Paired chromosomes
in genetic crossover
Unfitness score
0.8
Ranked chromosomes
0.6
0.4
0.2
0.0



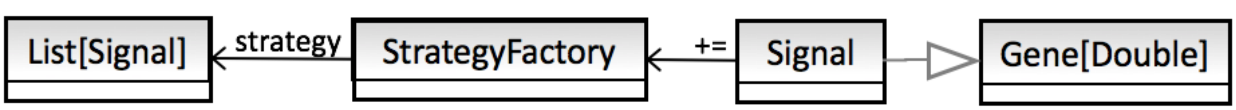
chIdx = prob*3
1
prob
2
101011101010101
1010000111101
1011110001101
3
gIdx = prob*64
1

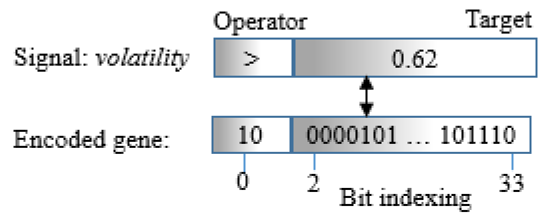


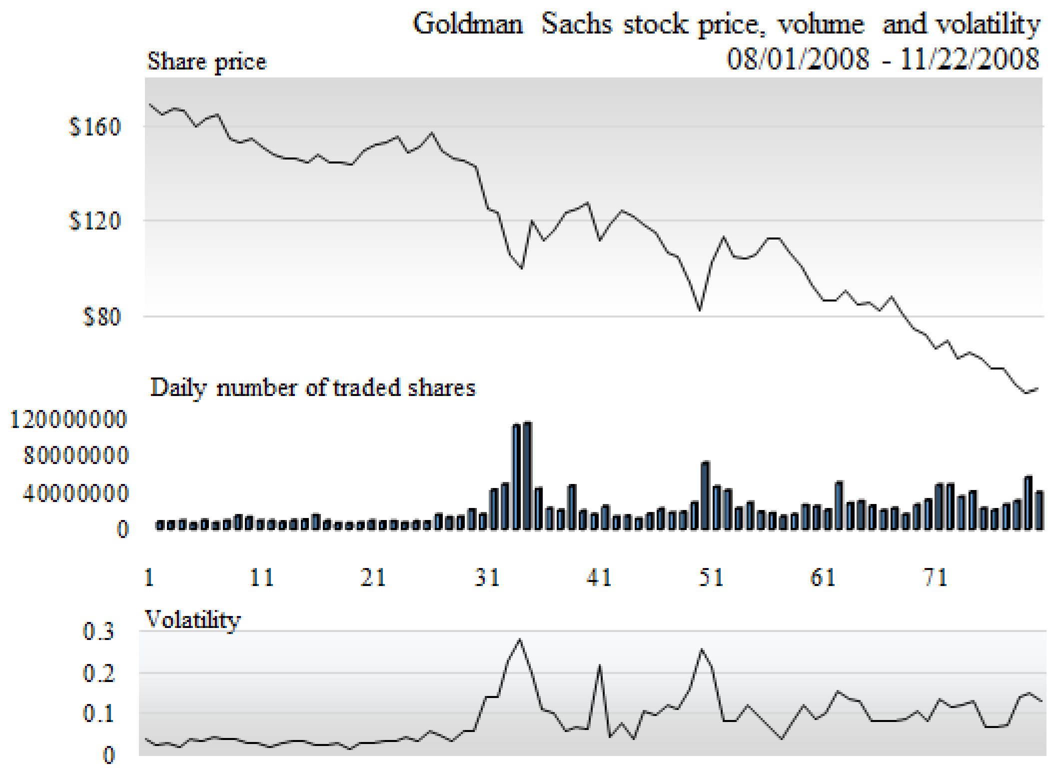


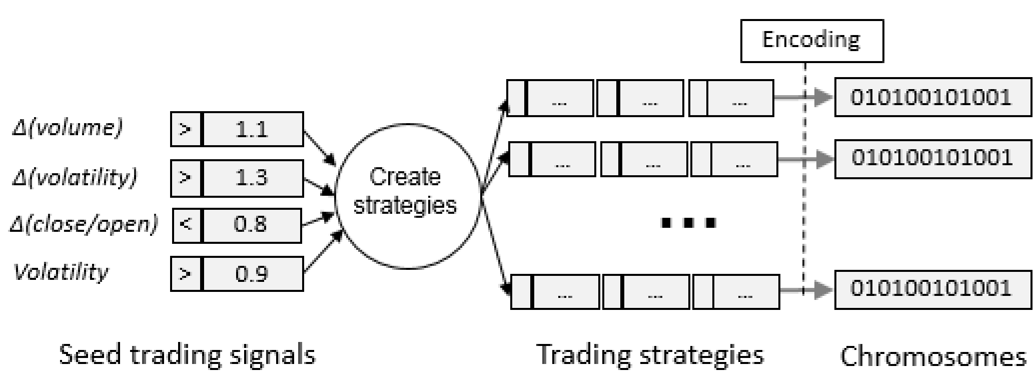
$$w_t = -\Delta p_t$$

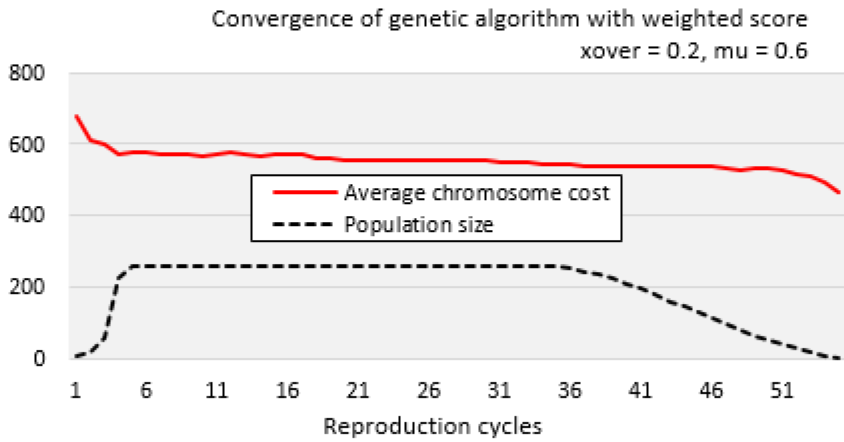
$$C\left(p, v^m, v^l \mid \alpha, \beta\right) = \sum_{t=0}^{n-1} \left(\alpha - v_t^m\right) w_t + \left(v_t^l - \beta\right) w_t$$

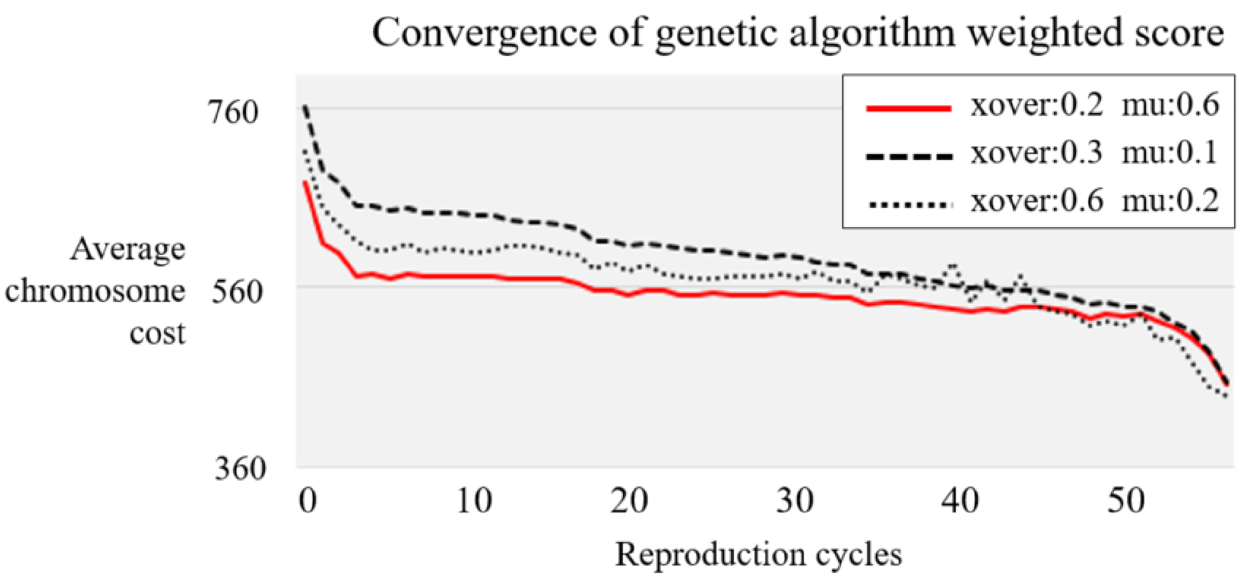


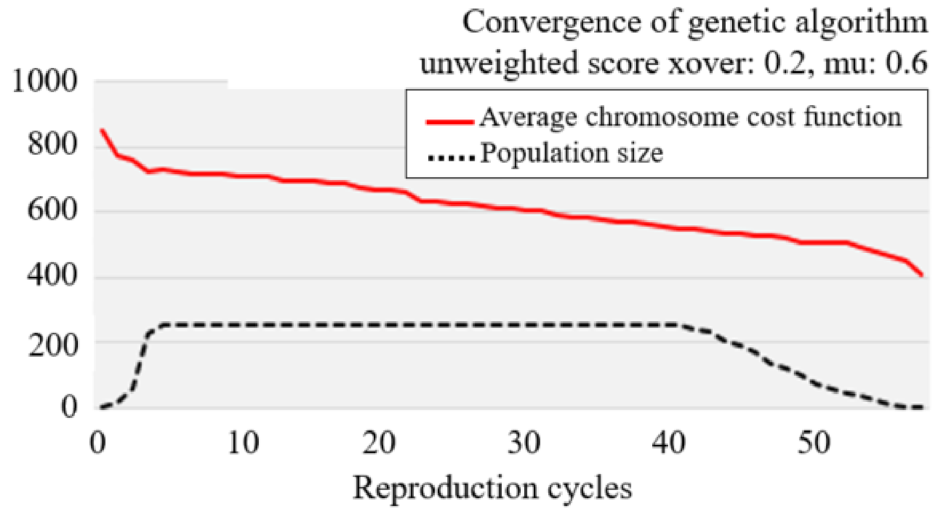

Operator
Target
Signal: *volatility*
>
0.62
Encoded gene:
10
0000101 … 101110
0
2
Bit indexing
33


Goldman Sachs stock price, volume and volatility
08/01/2008 - 11/22/2008
Share price
$160
$120
$80
Daily number of traded shares
120000000
80000000
40000000
0
1
11
21
31
41
51
61
71
Volatility
0.3
0.2
0.1
0


Encoding
Δ(volume)
>
1.1
Δ(volatility)
>
1.3
Δ(close/open)
<
0.8
Volatility
>
0.9
Create strategies
010100101001
010100101001
010100101001
Seed trading signals
Trading strategies
Chromosomes

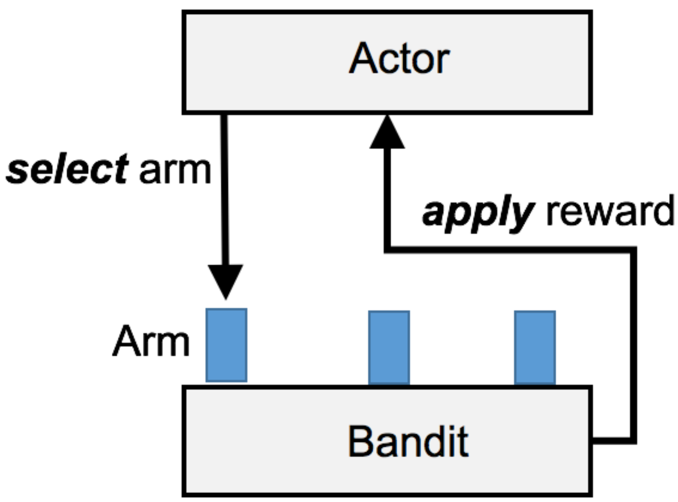

Convergence of genetic algorithm with weighted score
xover = 0.2, mu = 0.6
800
600
400
200
0
Average chromosome cost
Population size
1
6
11
16
21
26
31
36
41
46
51
Reproduction cycles


Convergence of genetic algorithm weighted score
760
560
360
Average chromosome cost
xover:0.2 mu:0.6
xover:0.3 mu:0.1
xover:0.6 mu:0.2
0
10
20
30
40
50
Reproduction cycles

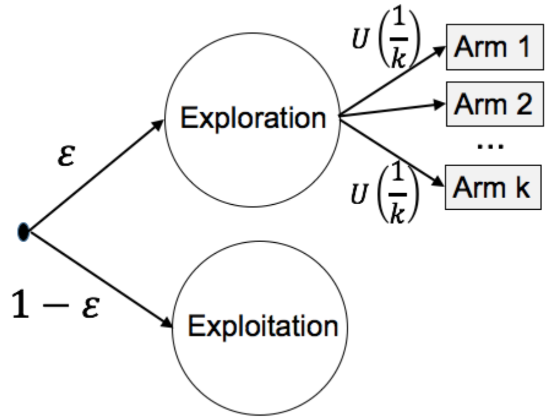


Convergence of genetic algorithm
unweighted score xover: 0.2, mu: 0.6
Average chromosome cost function
Population size
1000
800
600
400
200
0
0
10
20
30
40
50
Reproduction cycles

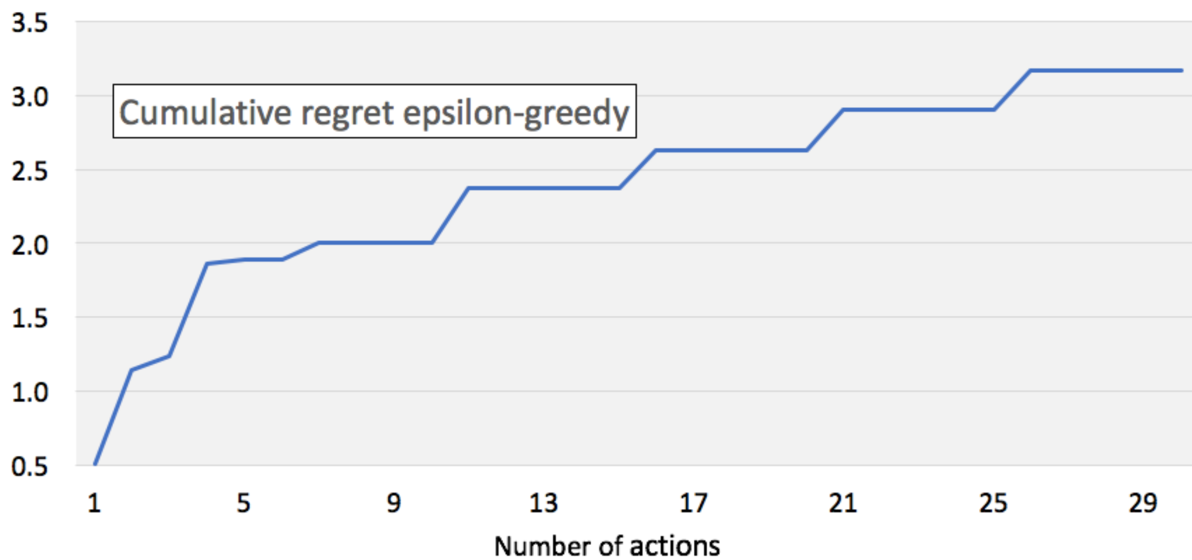
Chapter 14: Multiarmed Bandits



$$E[R] = T\mu^* - \sum_{t=0}^{T-1} \mu_{a_t}$$

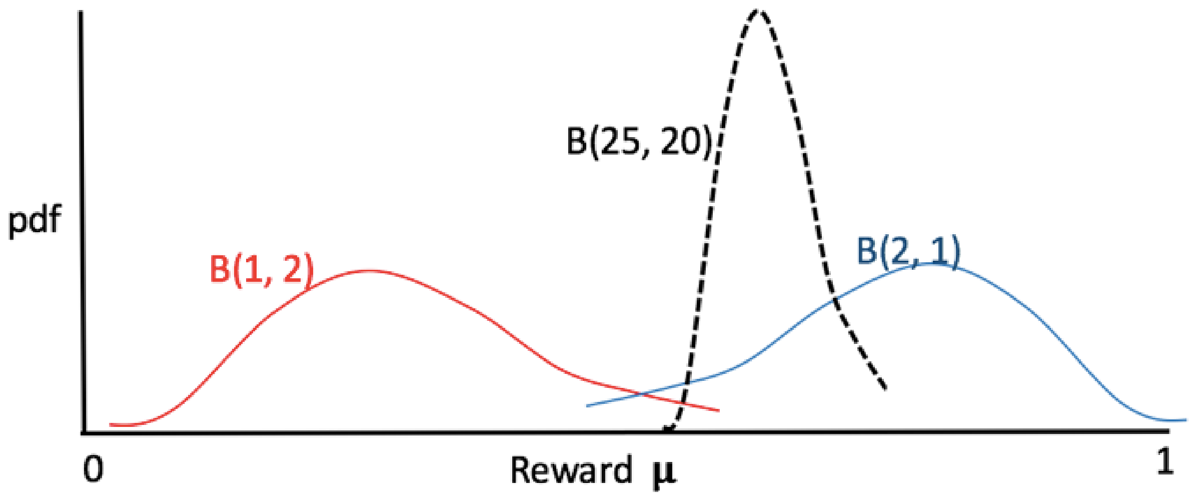


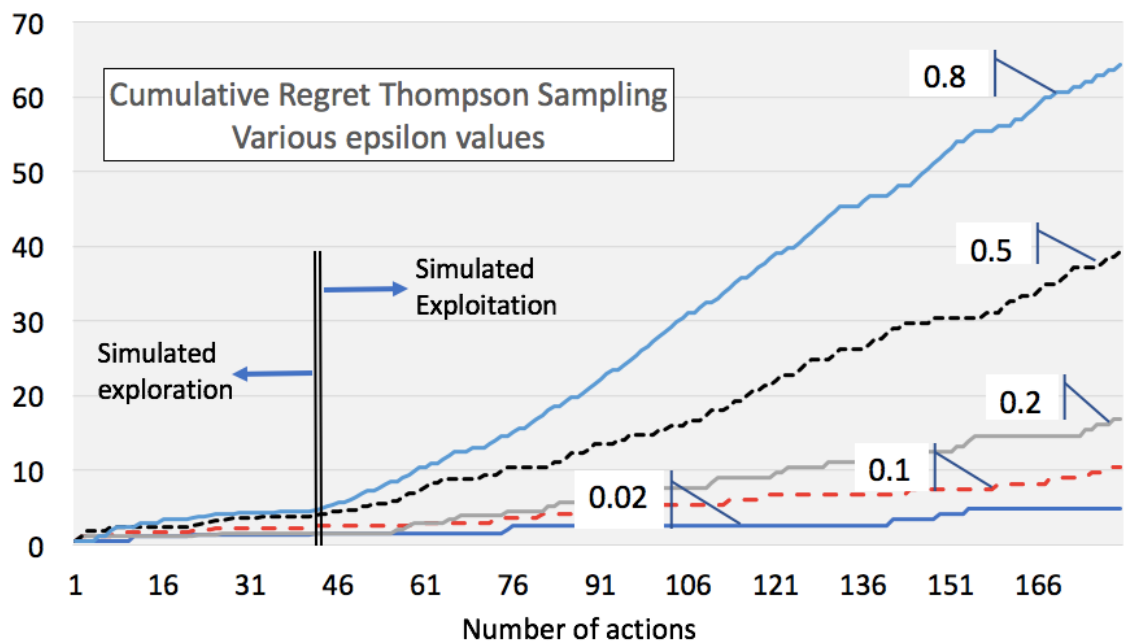
$$E[R] = T\varepsilon\left(\mu^* - \frac{1}{K}\right)\sum_{j=0}^{K-1}\mu_j$$

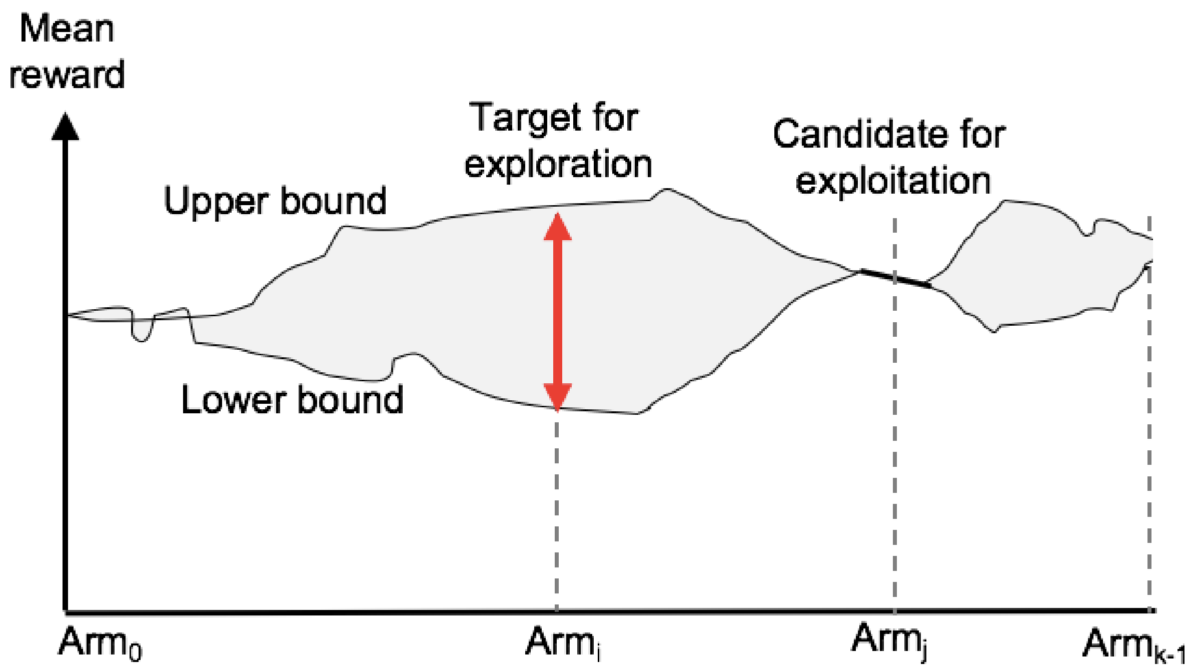


$$m = \arg\max_{0 \le j < k} f\left(. \mid w_j\right)$$

$$f\left(x \mid w\right) = \frac{1}{1 + e^{-\sum x_i w_j}}$$

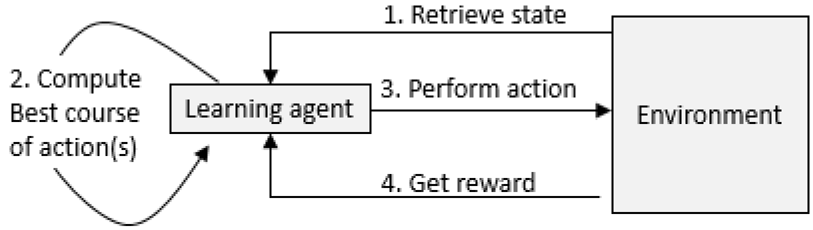

B(25, 20)
pdf
B(1, 2)
B(2, 1)
0
Reward μ
1


Cumulative Regret Thompson Sampling
Various epsilon values
70
60
50
40
30
20
10
0
0.8
0.5
0.2
0.1
0.02
Simulated
Exploitation
Simulated
exploration
1
16
31
46
61
76
91
106
121
136
151
166
Number of actions



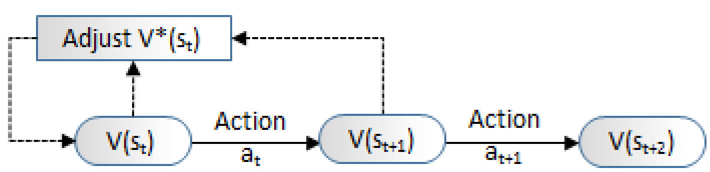
$$l = \arg\max_{0 \le j < k} f\left\{ \mu_j + \sqrt{2\frac{\log t}{n_j}} \right\}$$

Chapter 15: Reinforcement Learning



$$R_t \sum_{k=0}^{+\infty} \gamma^k r_{t+1+k}$$
$$V^\pi(s_t) = E\{R_t \mid s_t\}$$

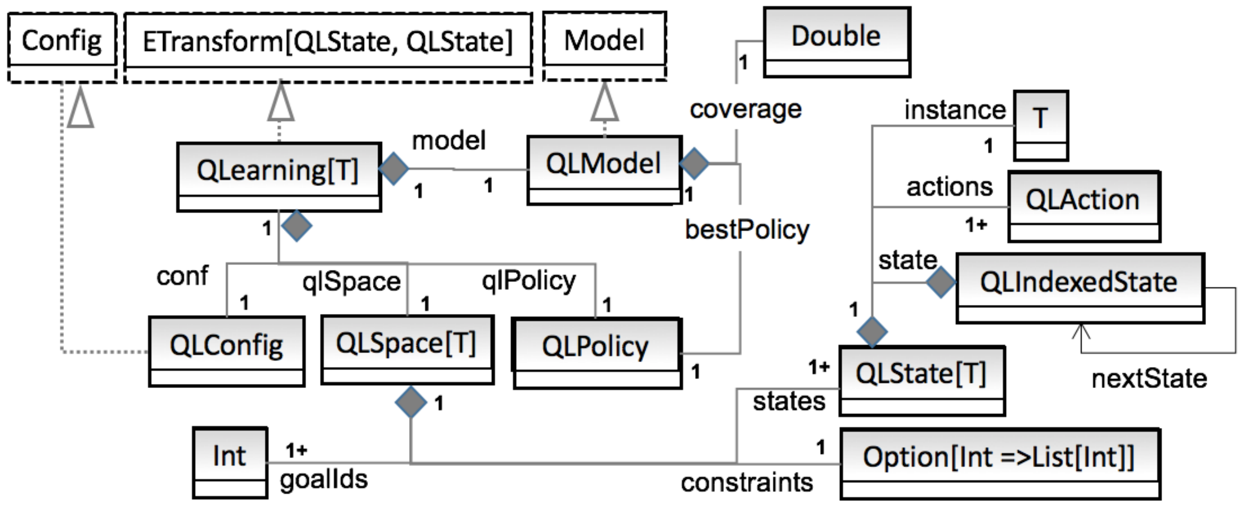
$$V^\pi(s_t) = \sum_{a \in A} \pi_t \sum_k \{p_k(r_k + \gamma \cdot V^\pi(s_k))\}$$
$$V^*(s_t) = \max_\pi V^\pi(s_t)$$

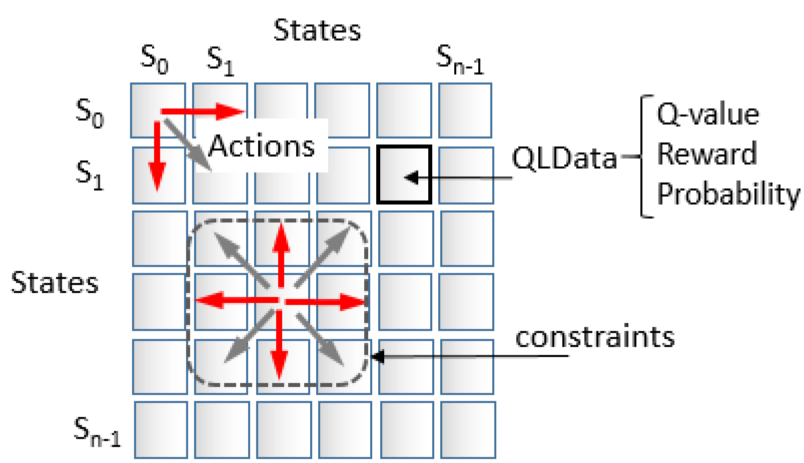


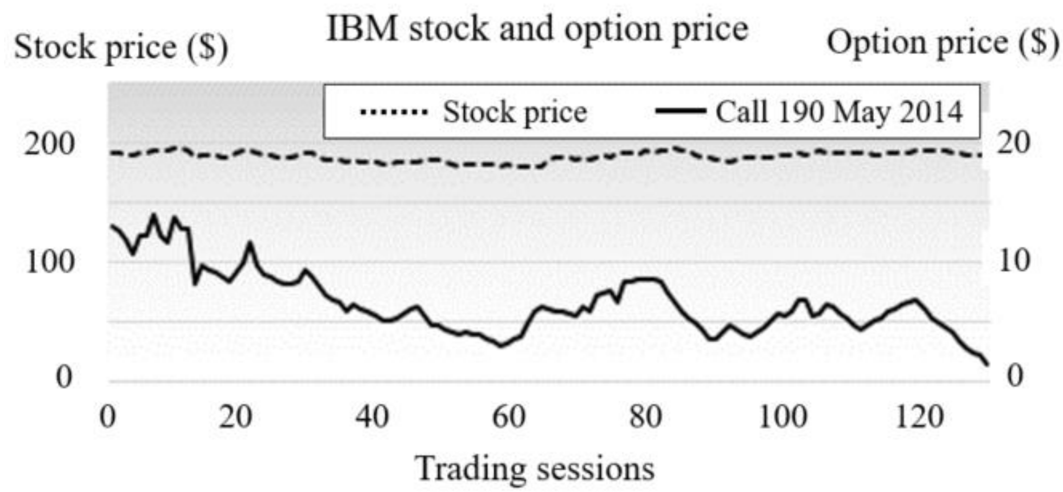
$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$
$$\hat{V}^\pi(s_t) = V^\pi(s_t) + \alpha\, \delta_t$$

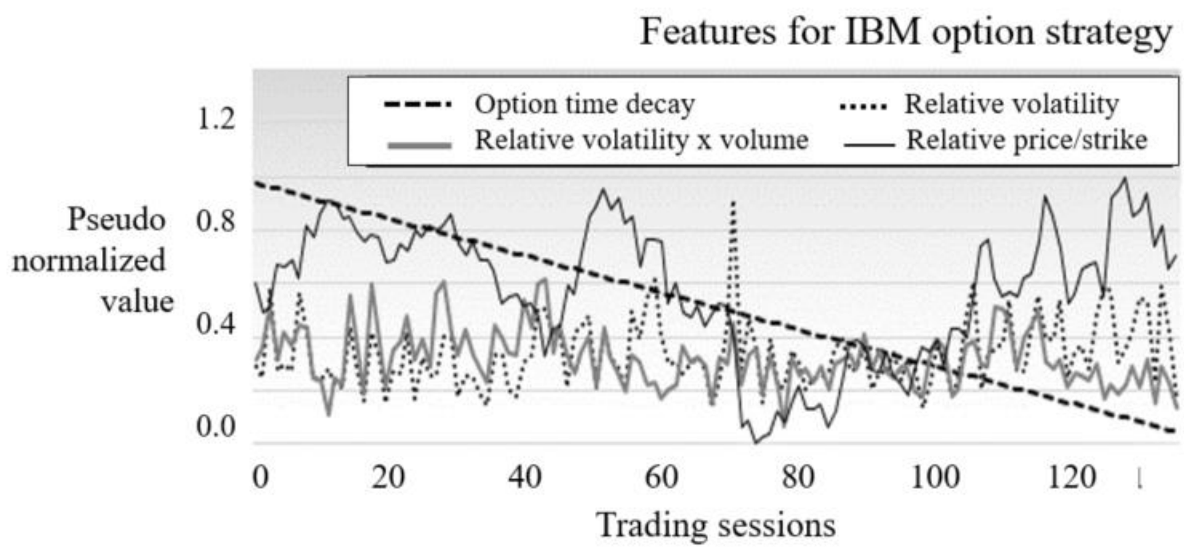
$$Q_t^\pi = Q^\pi(s_t, a_t) = E(R_t \mid s_t, a_t)$$

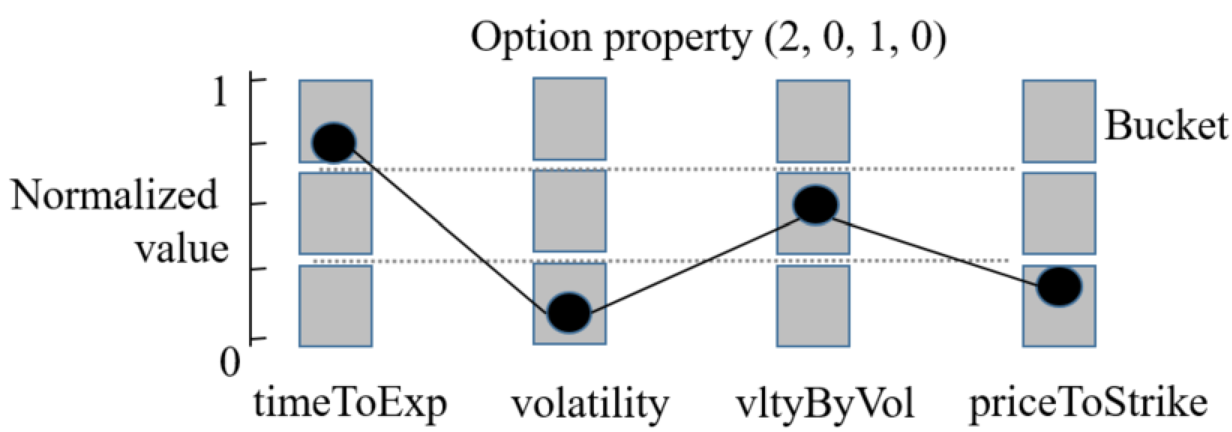
$$\tilde{Q}_t^\pi = Q_t^\pi + \alpha\left[r_{t+1} + \gamma \max_{a_t+1} Q_{t+1}^\pi - Q_t^\pi\right] Q_t^\pi = Q^\pi(s_t, a_t)$$

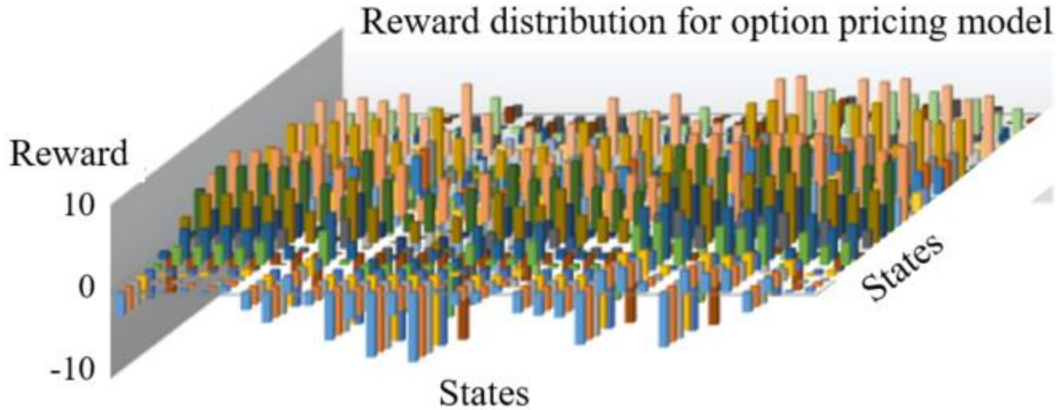

Config
ETransform[QLState, QLState]
Model
Double
coverage
instance
T
model
QLearning[T]
QLModel
actions
QLAction
bestPolicy
state
QLIndexedState
conf
qlSpace
qlPolicy
QLConfig
QLSpace[T]
QLPolicy
QLState[T]
nextState
states
Int
goalIds
Option[Int =>List[Int]]
constraints

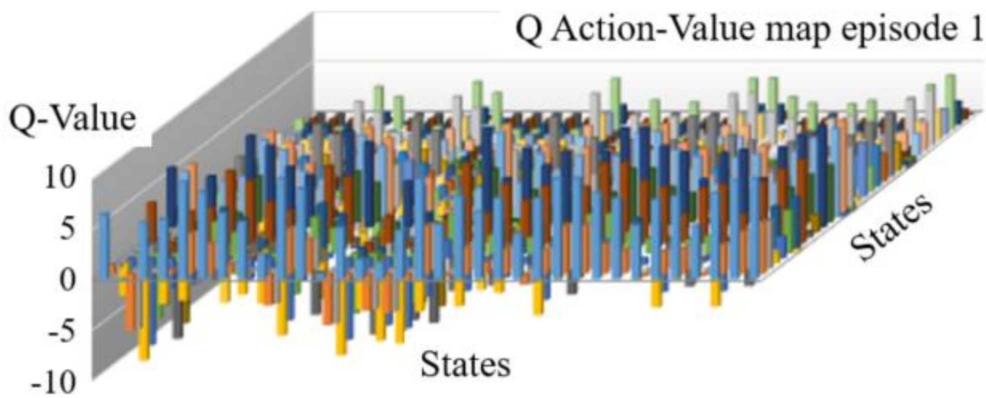

States
S0
S1
Sn-1
Actions
QLData
Q-value
Reward
Probability
States
constraints

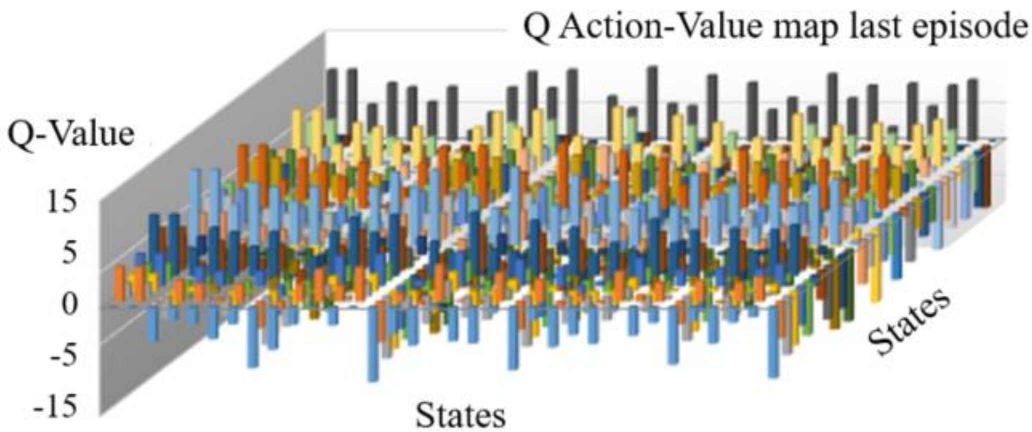

IBM stock and option price
Stock price ($)
Option price ($)
Stock price
Call 190 May 2014
200
100
0
20
10
0
0
20
40
60
80
100
120
Trading sessions

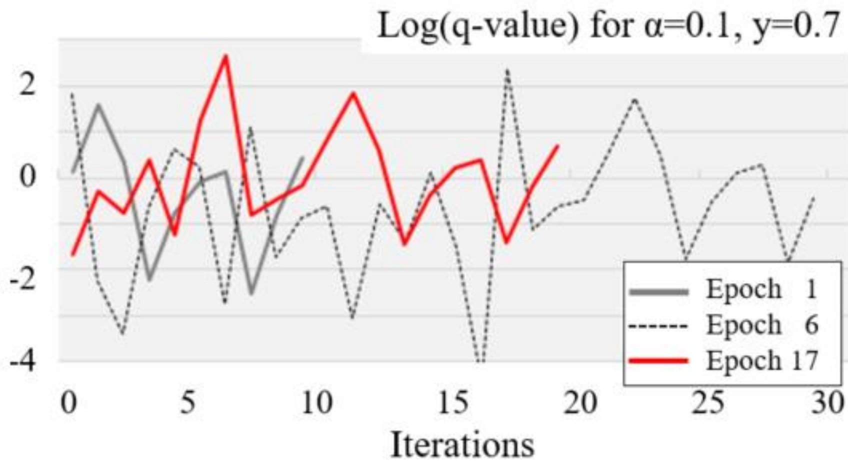

Features for IBM option strategy
Option time decay
Relative volatility
Relative volatility x volume
Relative price/strike
Pseudo normalized value
1.2
0.8
0.4
0.0
0
20
40
60
80
100
120
Trading sessions

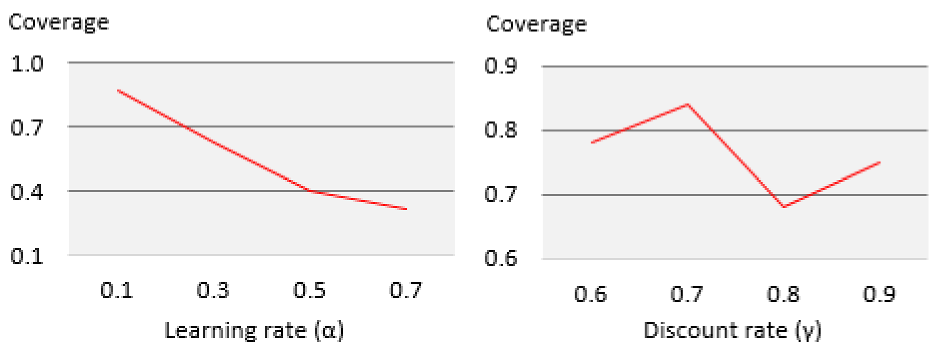

Option property (2, 0, 1, 0)
1
0
Normalized value
Bucket
timeToExp
volatility
vltyByVol
priceToStrike

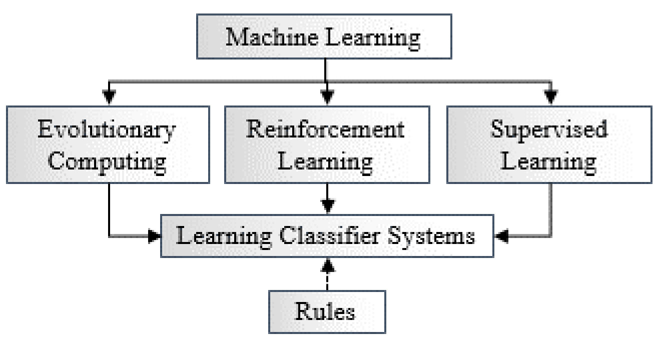

Reward distribution for option pricing model
Reward
10
0
-10
States
States

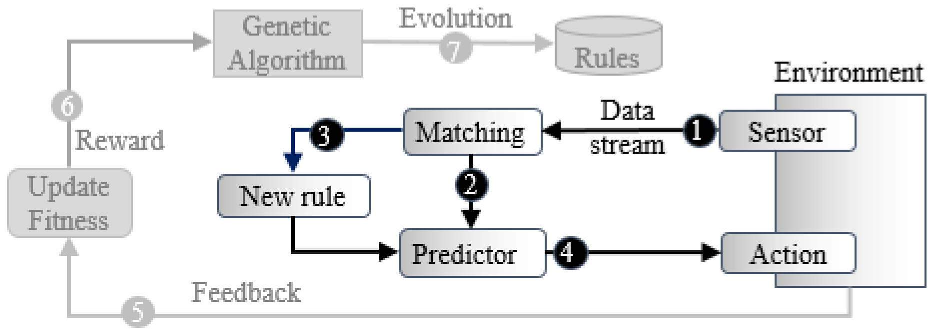

Q Action-Value map episode 1
Q-Value
10
5
0
-5
-10
States
States

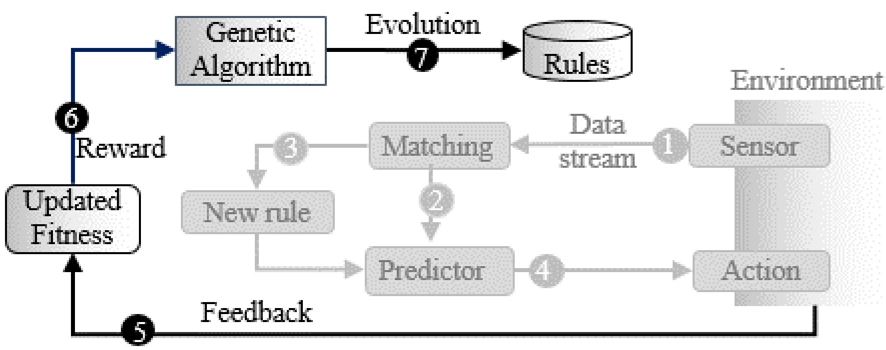

Q Action-Value map last episode
Q-Value
15
5
0
-5
-15
States
States

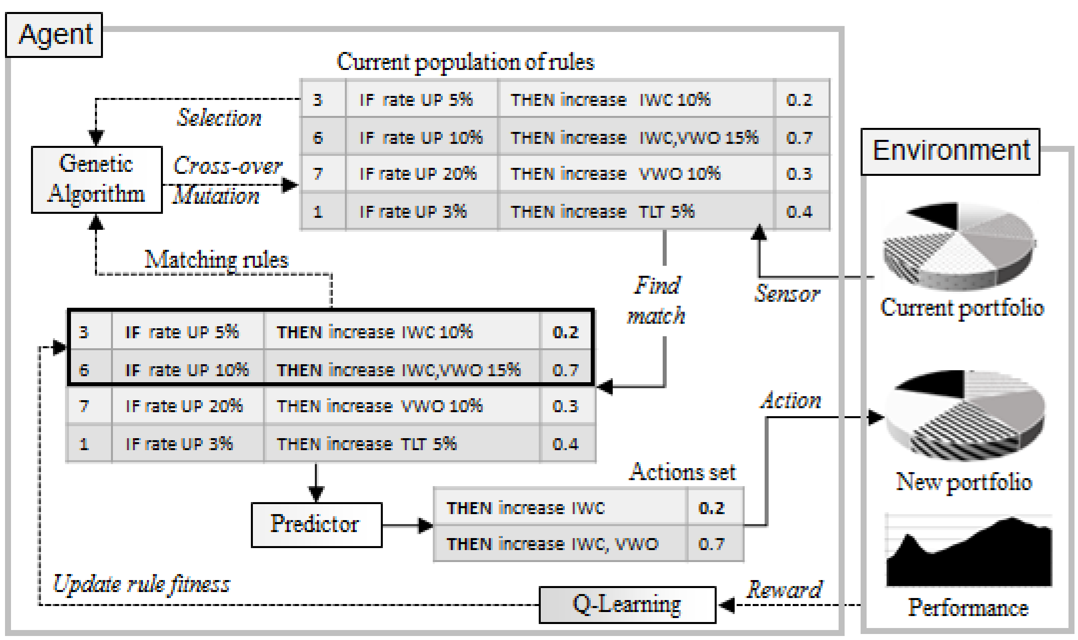

Log(q-value) for α=0.1, y=0.7
2
0
-2
-4
0
5
10
15
20
25
30
Iterations
Epoch 1
Epoch 6
Epoch 17

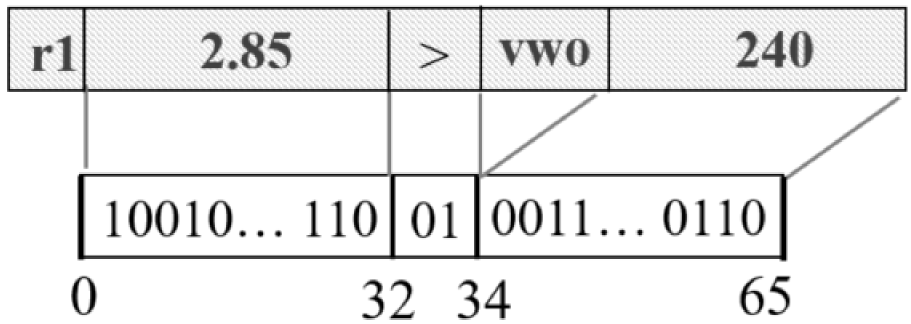

Coverage
1.0
0.7
0.4
0.1
0.1
0.3
0.5
0.7
Learning rate (α)
Coverage
0.9
0.8
0.7
0.6
0.6
0.7
0.8
0.9
Discount rate (γ)


Machine Learning
Evolutionary Computing
Reinforcement Learning
Supervised Learning
Learning Classifier Systems
Rules


Genetic Algorithm
Evolution
7
Rules
Environment
6
Reward
3
Matching
Data stream
1
Sensor
New rule
2
Update Fitness
Predictor
4
Action
Feedback
5


Genetic Algorithm
Evolution
7
Rules
Environment
6
Reward
3
Matching
Data stream
1
Sensor
New rule
2
Updated Fitness
Predictor
4
Action
Feedback
5


Agent
Current population of rules
3 IF rate UP 5% THEN increase IWC 10% 0.2
6 IF rate UP 10% THEN increase IWC,VWO 15% 0.7
7 IF rate UP 20% THEN increase VWO 10% 0.3
1 IF rate UP 3% THEN increase TLT 5% 0.4
Selection
Genetic Algorithm
Cross-over
Mutation
Environment
Matching rules
Find match
Sensor
Current portfolio
3 IF rate UP 5% THEN increase IWC 10% 0.2
6 IF rate UP 10% THEN increase IWC,VWO 15% 0.7
7 IF rate UP 20% THEN increase VWO 10% 0.3
1 IF rate UP 3% THEN increase TLT 5% 0.4
Action
New portfolio
Actions set
Predictor
THEN increase IWC 0.2
THEN increase IWC, VWO 0.7
Update rule fitness
Q-Learning
Reward
Performance

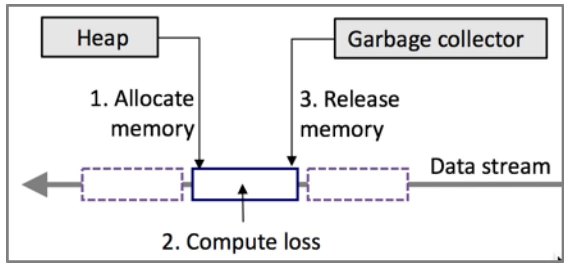

r1
2.85
>
vwo
240
10010… 110
01
0011… 0110
0
32
34
65

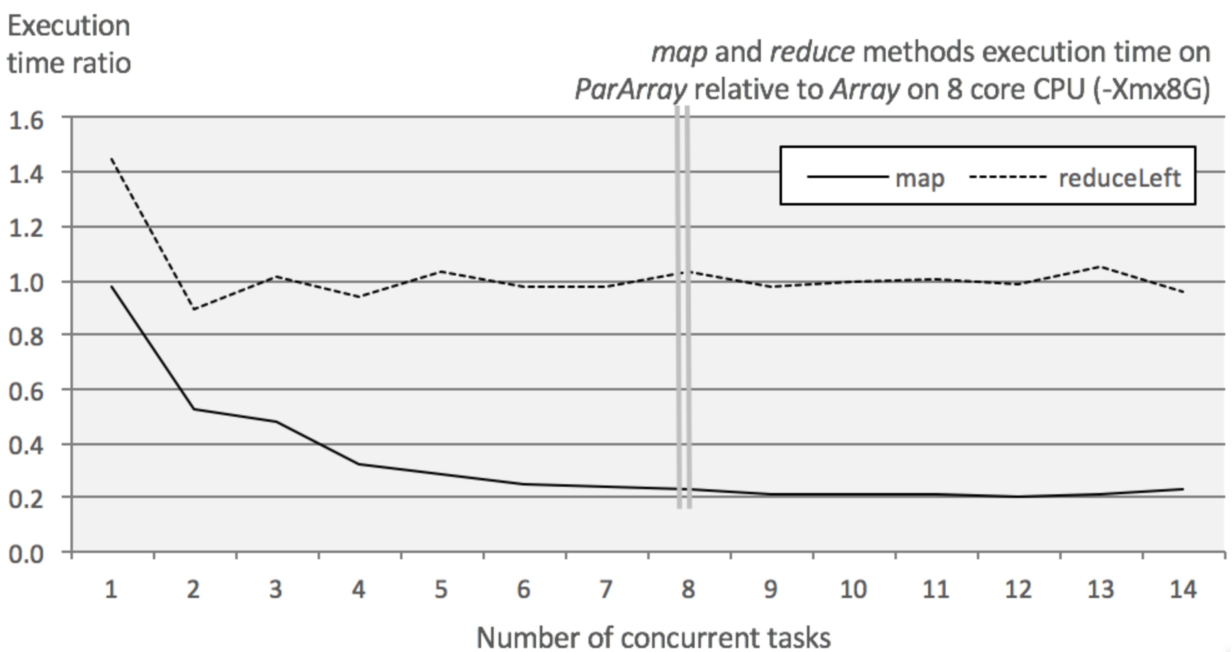
Chapter 16: Parallelism in Scala and Akka

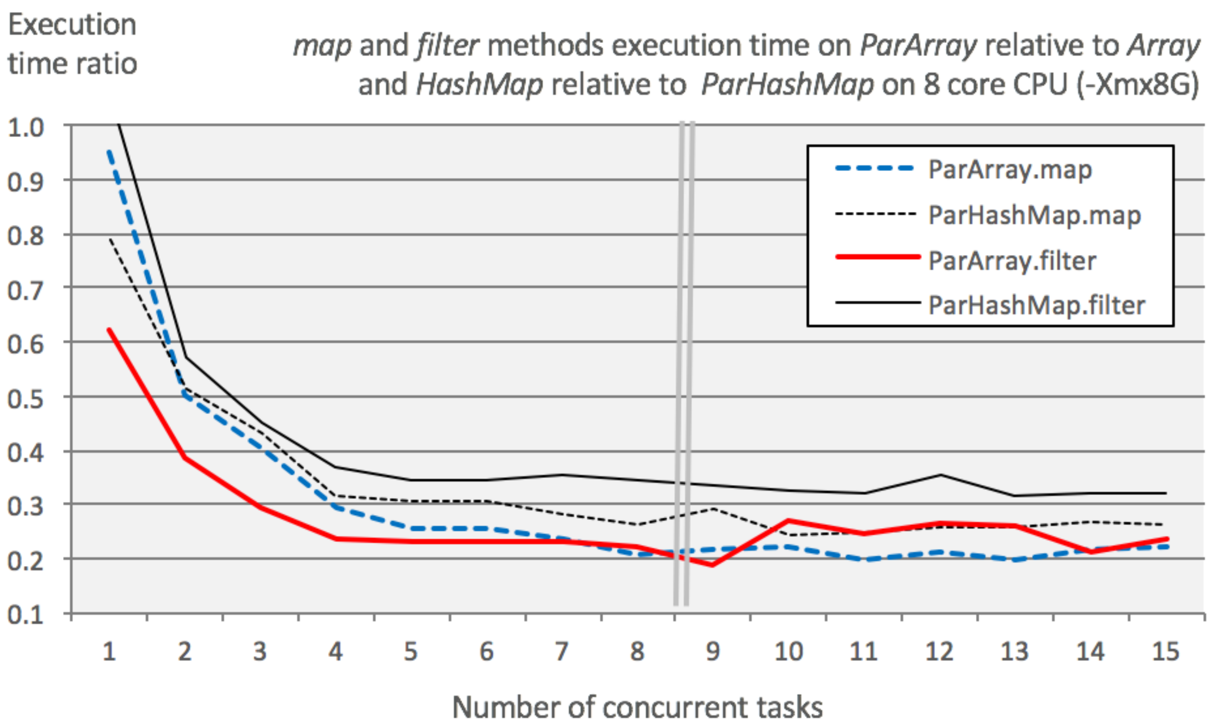
Spark	**Partitioner, Accumulator**: *org.apache.spark* **Broadcast**: *org.apache.spark.broadcast* **Resilient datasets**: *org.apache.spark.rdd._* **Data frame**: *org.apache.spark.sql._* **Caching, Shuffling**: *org.apache.spark._* **Listeners:** *org.apache.spark.scheduler._* **Serialization**: *org.apache.spark.serializer*
Akka	**Actors, Supervisors**: *akka.actors._* **Remote actors**: *akka.remote* **Type actors**: *akka.actors._* **Mailbox management**: *akka.mailbox._* **Clusters**: *akka.cluster._* **Dispatchers**: *akka.dispatch* **Events management**: *akka.event._* **Routing, Broadcast**: *akka.routing* **Persistency**: *akka.persistence._*
Scala	**Scheduler**: *scala.actors.scheduler* **Concurrency**: *scala.concurrent* **Par. collections**: *scala.collection.parallel*
JVM	**Threads, executors:** *java.util.concurrent.**

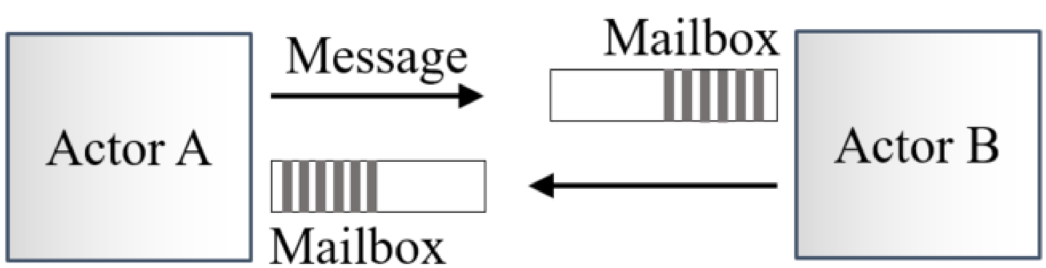
$$\mu_n = \frac{1}{n}\sum_{i=0}^{n-1} x_i$$

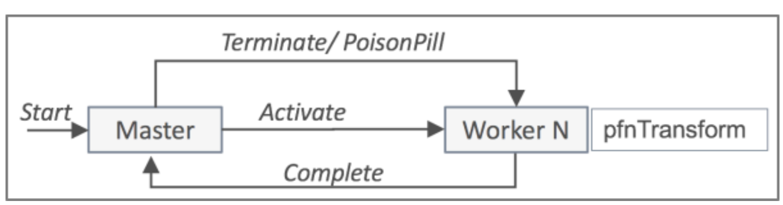
$$\mu_n = \left(1 - \frac{1}{n}\right)\mu_{n-1} + \frac{x_n}{n}$$

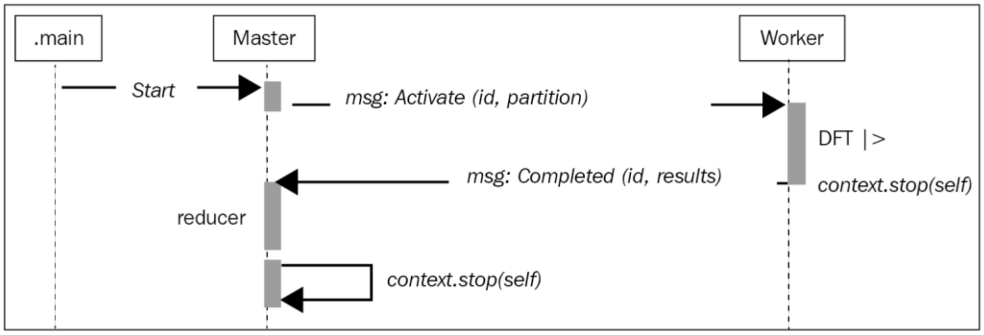


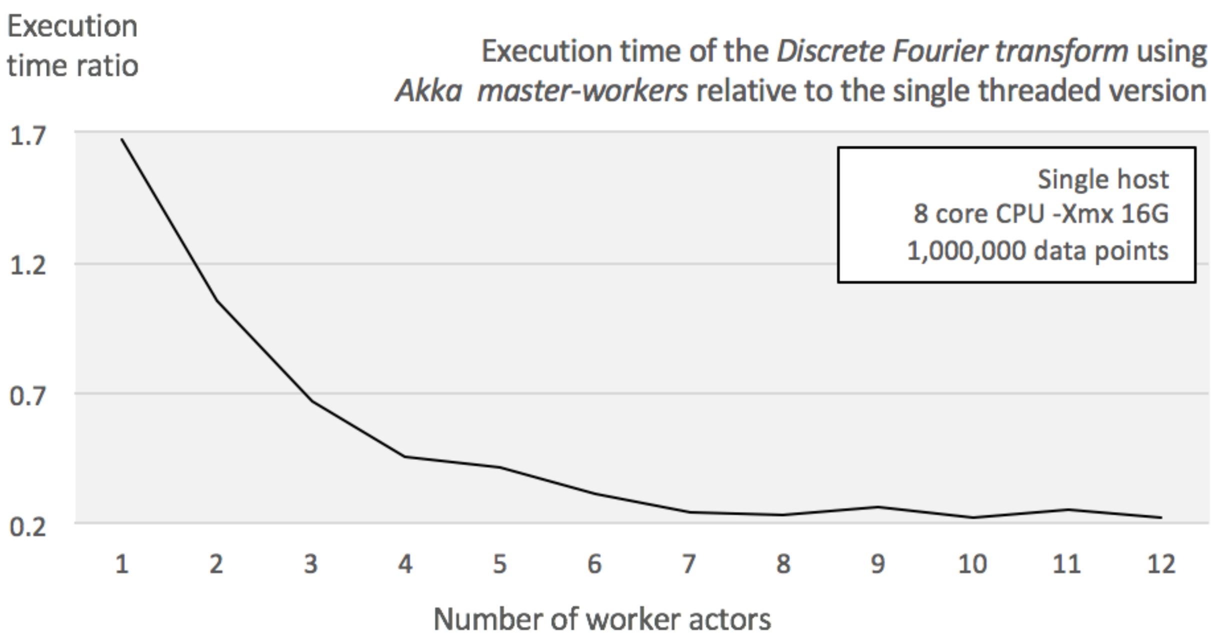


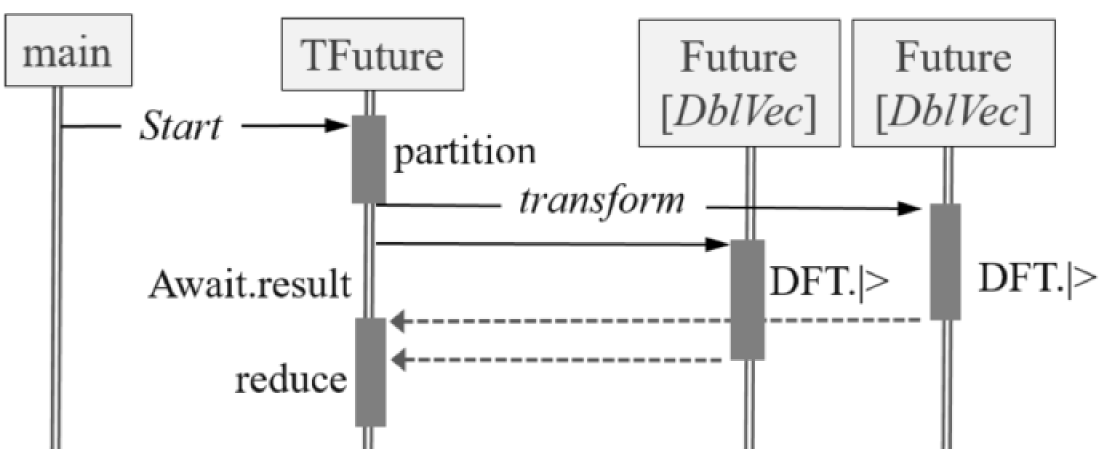

Execution
time ratio
map and filter methods execution time on ParArray relative to Array
and HashMap relative to ParHashMap on 8 core CPU (-Xmx8G)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
ParArray.map
ParHashMap.map
ParArray.filter
ParHashMap.filter
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Number of concurrent tasks

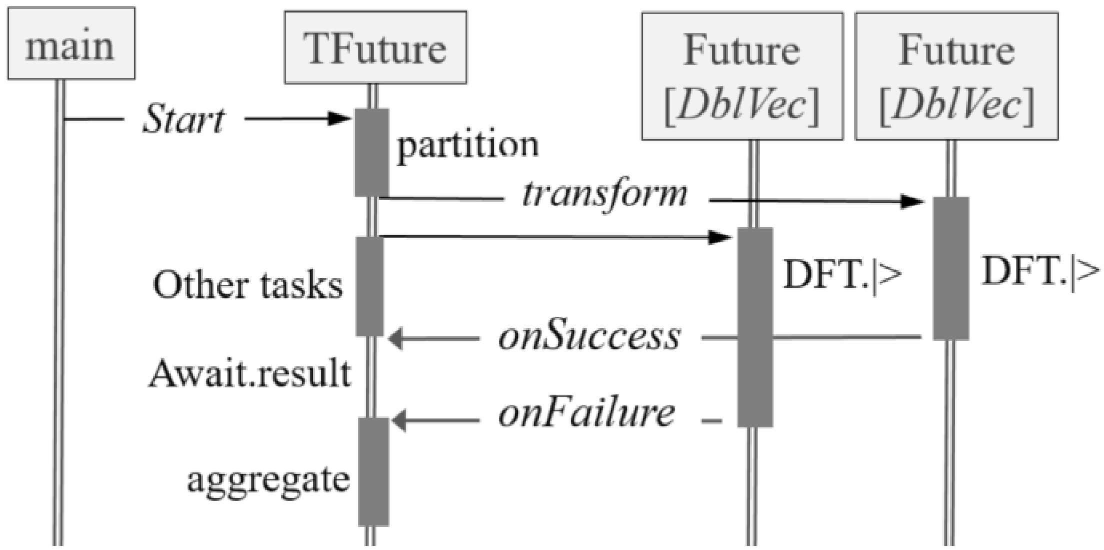

Actor A
Message
Mailbox
Mailbox
Actor B


Terminate/ PoisonPill
Start
Master
Activate
Worker N
pfnTransform
Complete

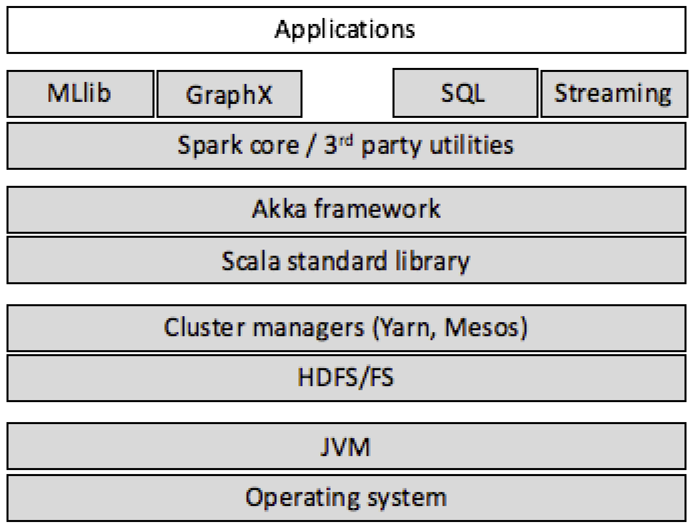

.main
Master
Worker
Start
msg: Activate (id, partition)
DFT |>
msg: Completed (id, results)
context.stop(self)
reducer
context.stop(self)

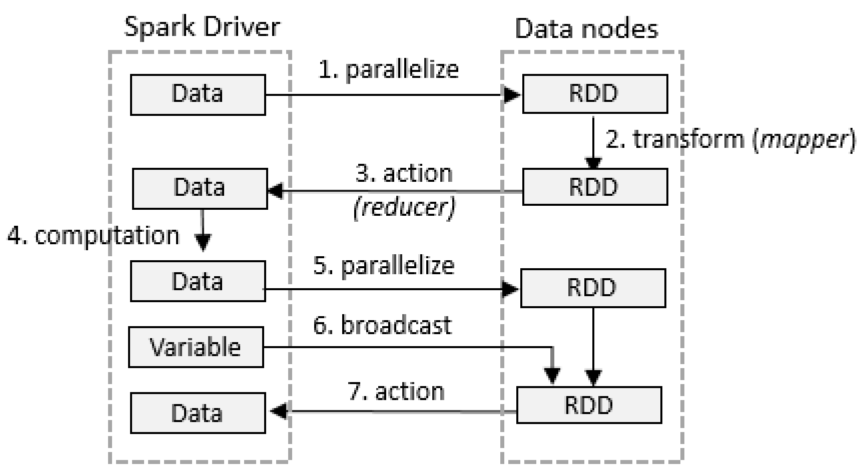

Execution time ratio
Execution time of the Discrete Fourier transform using Akka master-workers relative to the single threaded version
Single host
8 core CPU -Xmx 16G
1,000,000 data points
1.7
1.2
0.7
0.2
1
2
3
4
5
6
7
8
9
10
11
12
Number of worker actors

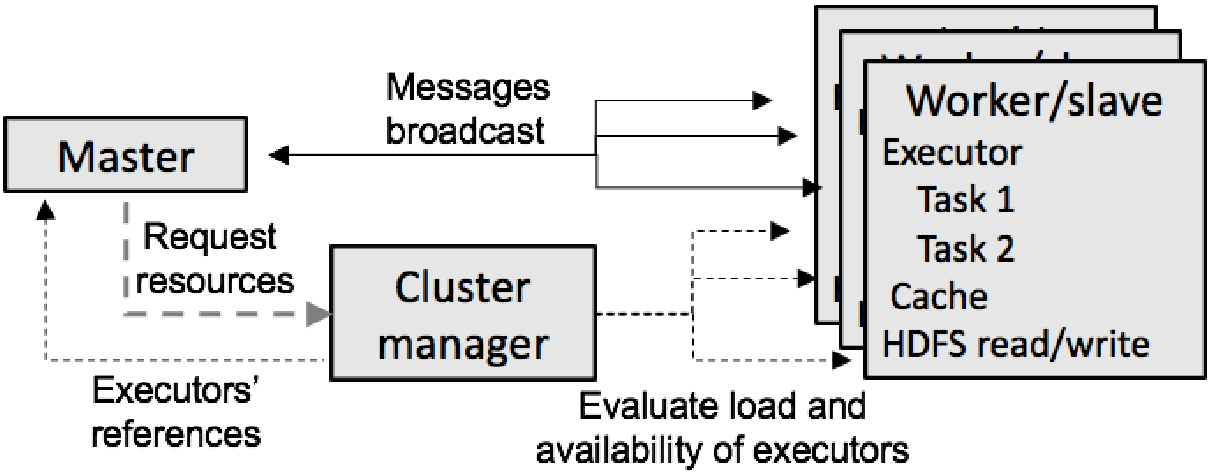

main
TFuture
Future [DblVec]
Future [DblVec]
Start
partition
transform
Await.result
DFT.|>
DFT.|>
reduce


main
TFuture
Future [DblVec]
Future [DblVec]
Start
partition
transform
Other tasks
DFT.|>
DFT.|>
onSuccess
Await.result
onFailure
aggregate

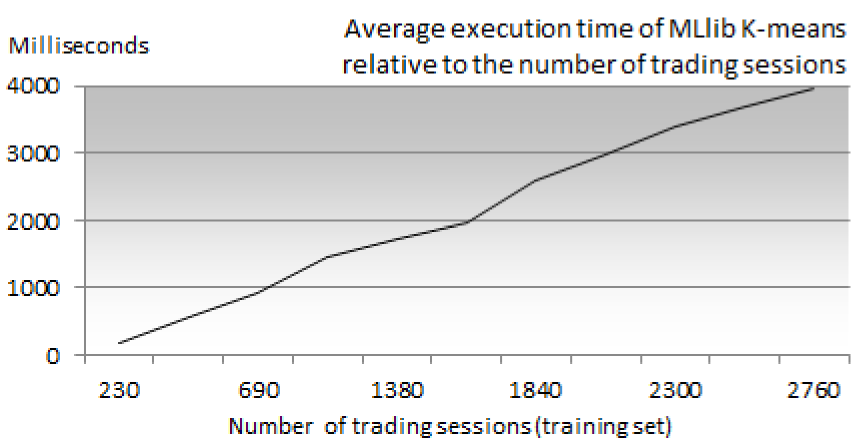
Chapter 17: Apache Spark MLlib

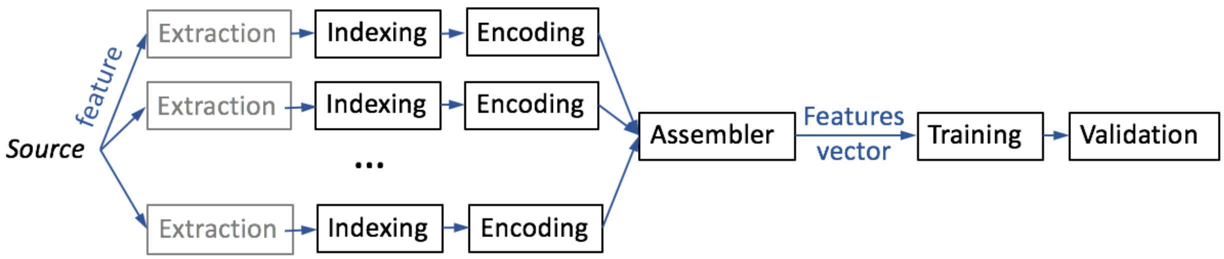





Master
Messages broadcast
Request resources
Cluster manager
Executors' references
Evaluate load and availability of executors
Worker/slave
Executor
Task 1
Task 2
Cache
HDFS read/write


Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
16/12/13 15:21:40 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform...
16/12/13 15:21:41 WARN SparkContext: Use an existing SparkContext, some configuration may not take
Spark context Web UI available at http://10.1.2.194:4040
Spark context available as 'sc' (master = local[*], app id = local-1481671300908).
Spark session available as 'spark'.
Welcome to
version 2.0.1
Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_66)
Type in expressions to have them evaluated.
Type :help for more information.
scala>


Average execution time of MLlib K-means relative to the number of trading sessions
Milliseconds
4000
3000
2000
1000
0
230
690
1380
1840
2300
2760
Number of trading sessions (training set)




```
root
 |-- date: string (nullable = true)
 |-- asset: string (nullable = true)
 |-- region: integer (nullable = true)
 |-- agent: string (nullable = true)
 |-- dateIndex: double (nullable = true)
 |-- assetIndex: double (nullable = true)
 |-- regionIndex: double (nullable = true)
 |-- agentIndex: double (nullable = true)
 |-- dateVector: vector (nullable = true)
 |-- assetVector: vector (nullable = true)
 |-- regionVector: vector (nullable = true)
 |-- agentVector: vector (nullable = true)
 |-- features: vector (nullable = true)
 |-- rawPrediction: vector (nullable = true)
 |-- probability: vector (nullable = true)
 |-- prediction: double (nullable = true)
```

date	asset	region	agent
07/05/2014	300ec90b	27	aa5
08/03/2014	23c9ab02	7	a08

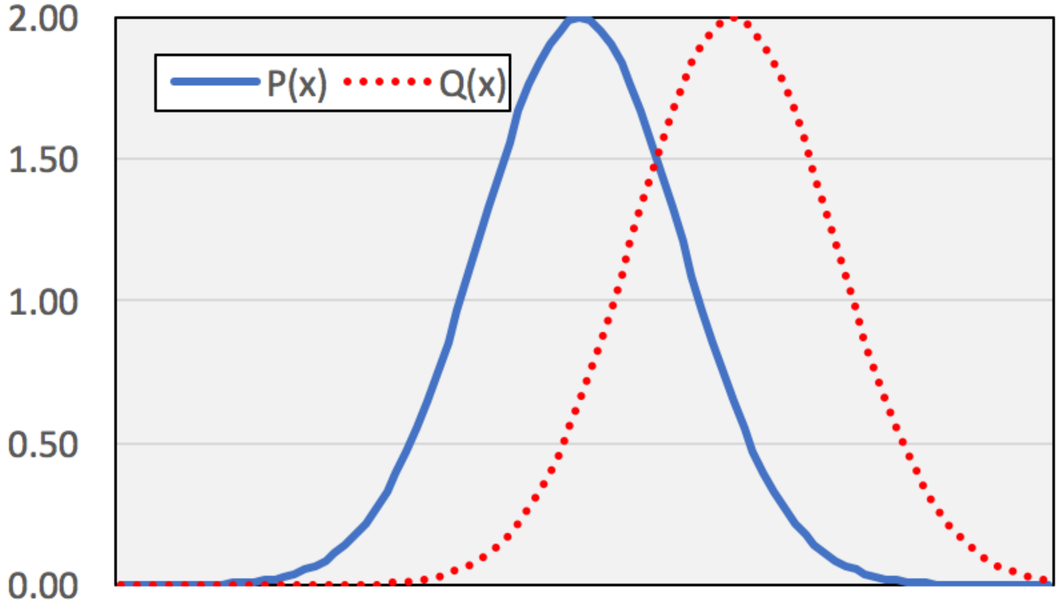
dateIndex	assetIndex	regionIndex	agentIndex
4.0	4.0	19.0	6.0
3.0	1.0	13.0	2.0

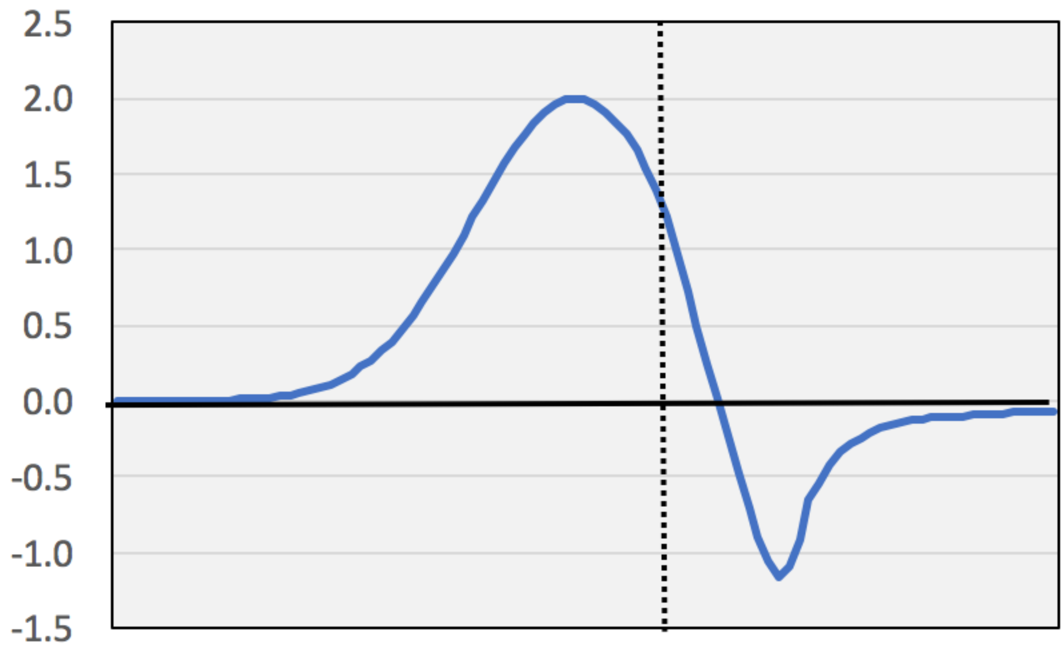
dateVector	assetVector	regionVector	agentVector	features
(16,[4],[1.0])	(14,[4],[1.0])	(20,[19],[1.0])	(15,[6],[1.0])	(65,[4,20,49,56],...
(16,[3],[1.0])	(14,[1],[1.0])	(20,[13],[1.0])	(15,[2],[1.0])	(65,[3,17,43,52],...

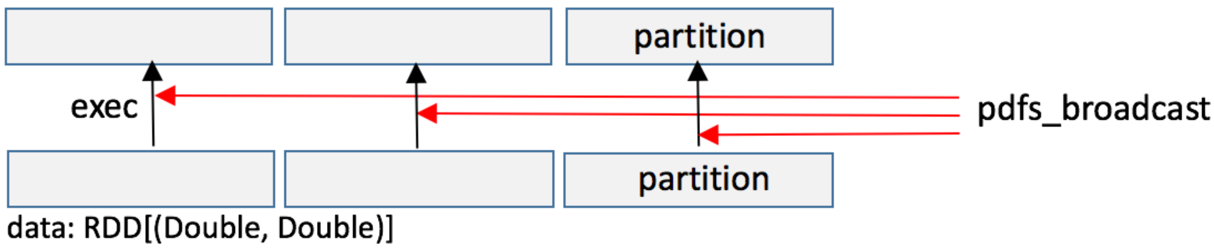
$$TPR = \frac{TP}{TP + FN} \quad TFP = \frac{TP}{TP + FP}$$

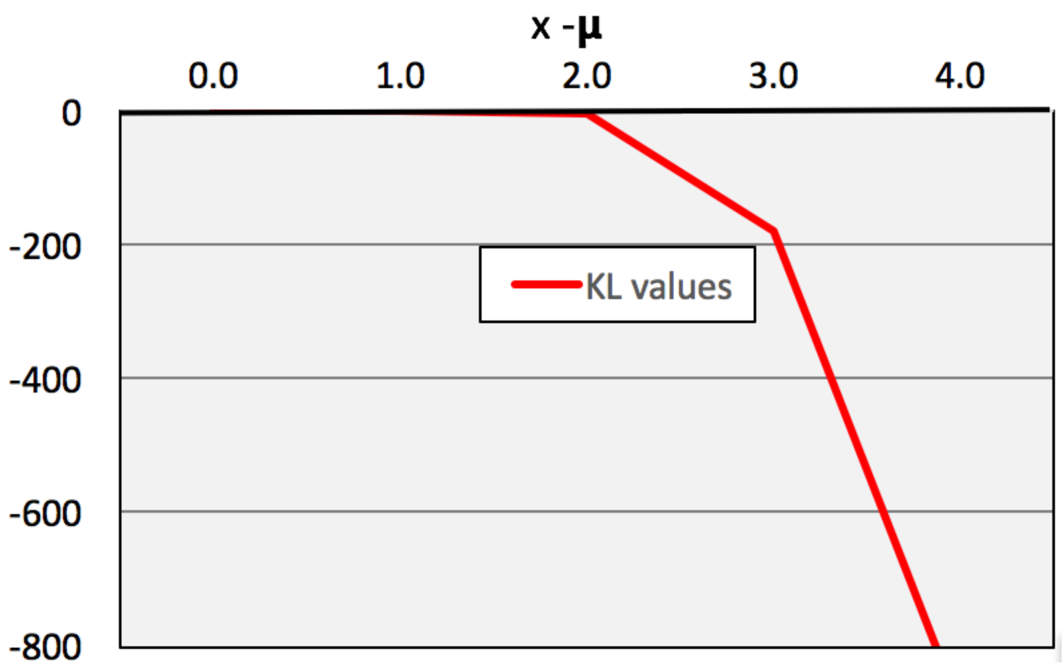
$$D_{KL}(P \parallel Q) = -\int_{-\infty}^{+\infty} p(x).log\frac{p(x)}{q(x)}$$

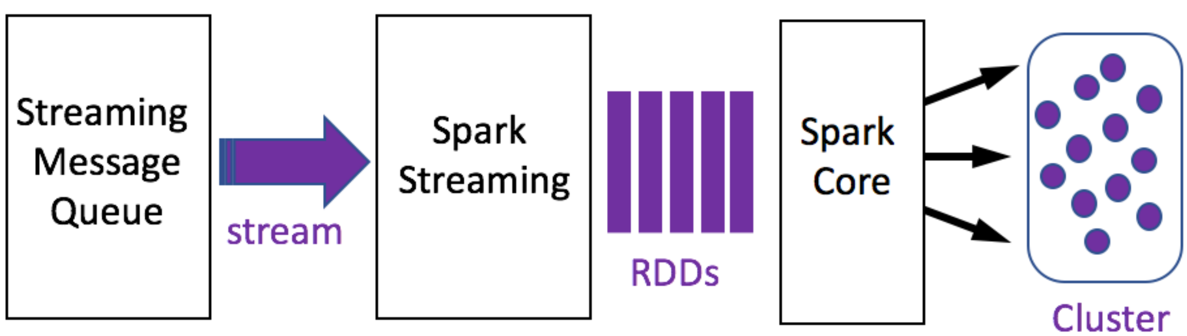
$$D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$$


2.00
1.50
1.00
0.50
0.00
P(x)
Q(x)

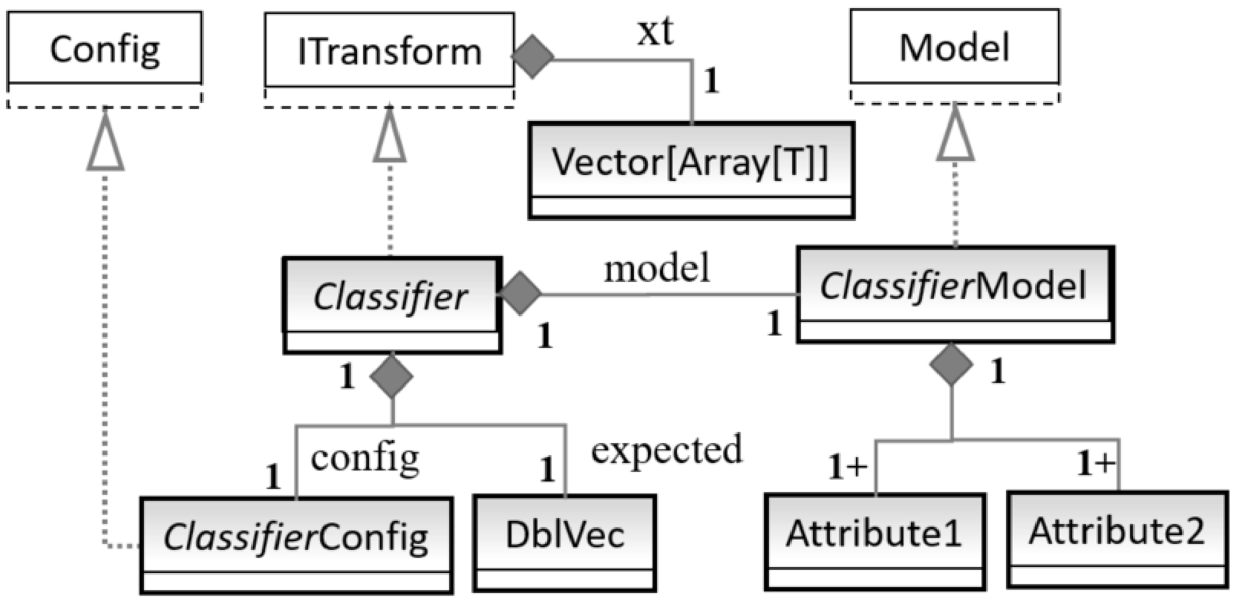

2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5


partition
exec
pdfs_broadcast
partition
data: RDD[(Double, Double)]


x -µ
0.0
1.0
2.0
3.0
4.0
0
-200
-400
-600
-800
KL values


Streaming Message Queue
stream
Spark Streaming
RDDs
Spark Core
Cluster

Appendix A: Basic Concepts



$$J(f)=\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n}\end{bmatrix}$$

$$H(f)=\begin{bmatrix}\frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}\\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2}\end{bmatrix}$$

$$f(x,y)=x^2y+e^{-y} \quad J(f)=\left[2xy, x^2-e^{-y}\right] \quad H(f)=\begin{bmatrix}2y & 2x\\ 2x & e^{-y}\end{bmatrix}$$

$$x_{(t+1)} = x_{(t)} - \gamma \nabla F(a)$$

$$Ax = b \rightarrow \sum_{i=0}^{n-1} \alpha_i p_i x^* = b;\ p_i . p_j = 0$$

$$F(x) = \sum_{i=0}^{n-1} f_i(x),\ x_{t+1} = x_t - \alpha \sum_{i=0}^{n-1} \nabla f_i(x)$$

$$F(x_t + \Delta x) - F(x_t) \approx F'(x).\Delta x + F''(x_t) \rightarrow x_{t+1} = x_t - \frac{F'(x_t)}{F''(x_t)}$$

$$H_t p_t = -\nabla F(x_t),\ \ x_{t+1} = x_t + \alpha_t p_t$$

$$x_{t+1} = x_t + \Delta x_t;\ \nabla F(x_t) + \Delta G_t = \Delta(\nabla F(x_t))$$

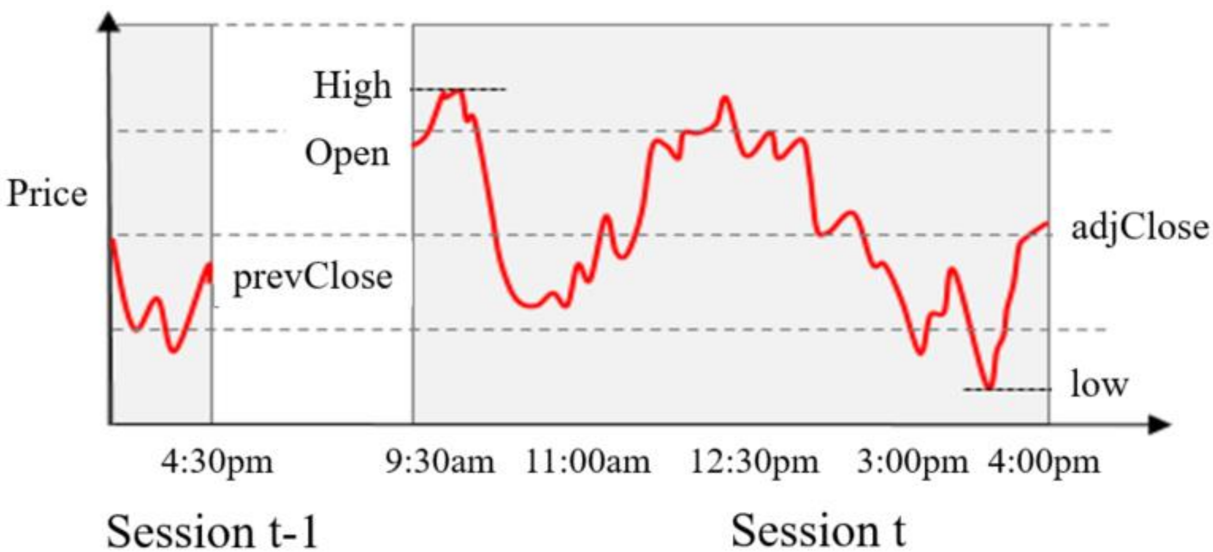
$$\mathcal{L}(w) = \sum_{i=0}^{m-1} r_i(w)^2;\ r_i = y_i - F(x_i, w)$$

$$w_{(t+1)} = w_{(t)} - \left\| \frac{\partial r_i(w_{(t)})}{\partial w_i} \right\|_{ij}^{-1} r(w_{(t)})$$

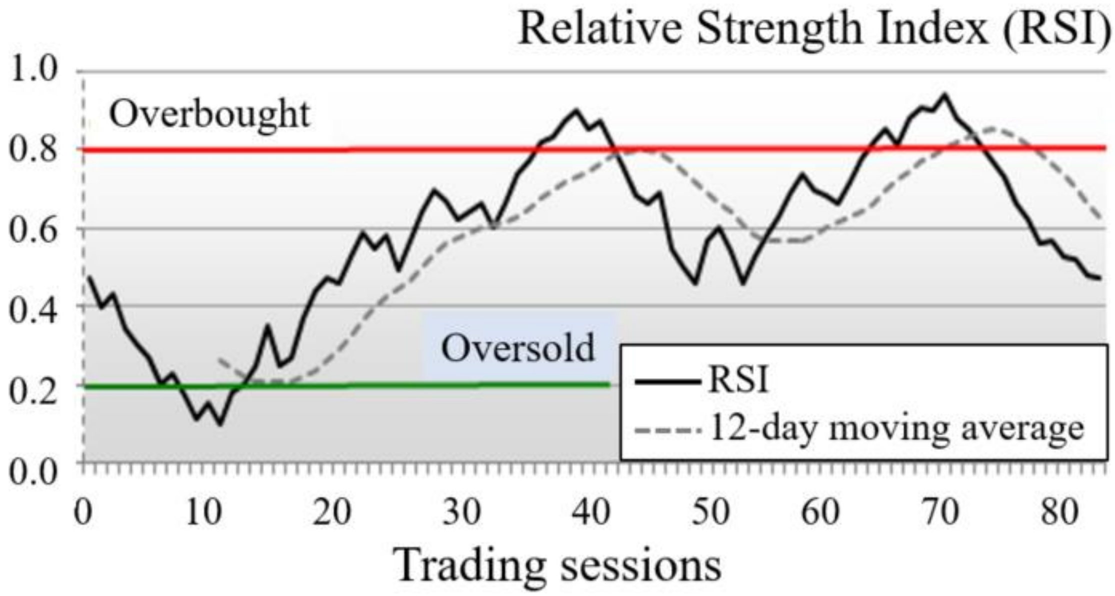
$$\mathcal{L}(w + \delta) \approx \sum_{i=0}^{m-1} \left(r_i(w) - \frac{\partial F(x_i, w)}{\partial w} \delta \right)^2$$

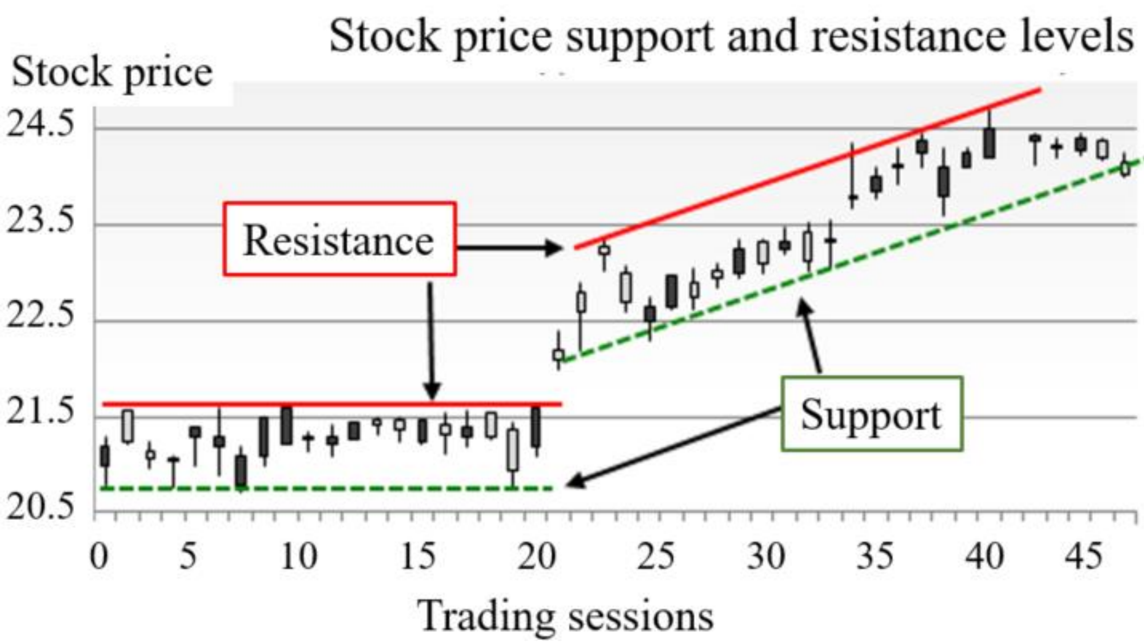
$$\mathcal{L}(x,\lambda) = f(x) + \lambda(g(x) - c)$$
$$\nabla_{x,\lambda}\mathcal{L}(x,y) = 0$$
$$\nabla\mathcal{L} = \left[\frac{\partial\mathcal{L}}{\partial x_i}, \frac{\partial\mathcal{L}}{\partial\lambda}\right]$$

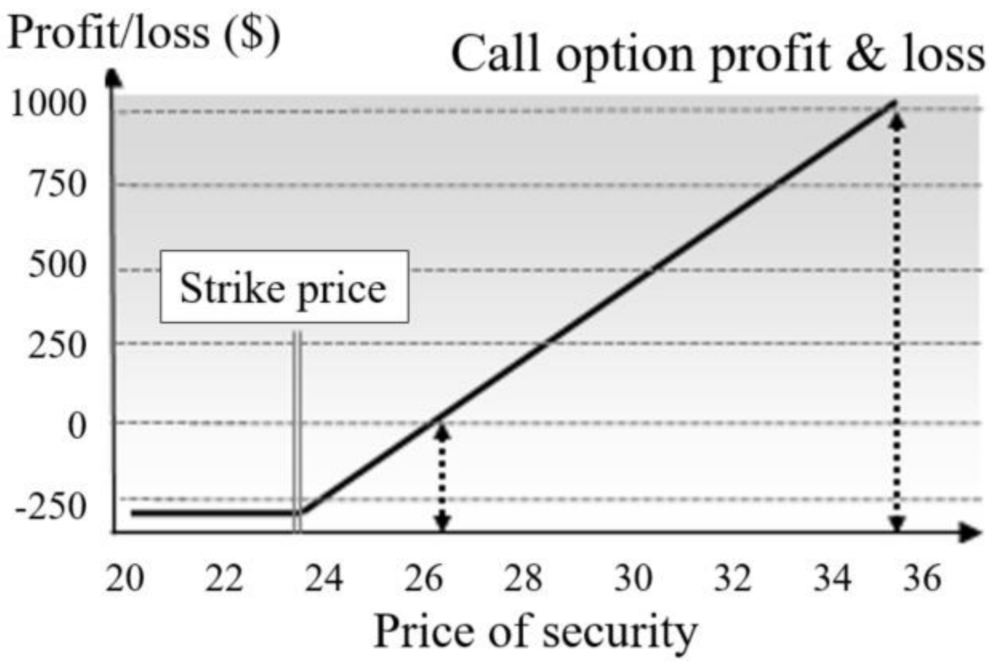
$$f(x,y) = x^2 + y^2 subject\, x - y = 2$$
$$\frac{\partial\mathcal{L}}{\partial x} = 2x + \lambda, \frac{\partial\mathcal{L}}{\partial y} = 2y - \lambda, \frac{\partial\mathcal{L}}{\partial y} = x - y - 2$$
$$x = 1, y = -1, \lambda = -2$$



$$U = p(t) - p(t-1), D = 0$$


Relative Strength Index (RSI)
Overbought
Oversold
RSI
12-day moving average
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70
80
Trading sessions


Stock price support and resistance levels
Stock price
Resistance
Support
24.5
23.5
22.5
21.5
20.5
0
5
10
15
20
25
30
35
40
45
Trading sessions


Profit/loss ($)
Call option profit & loss
1000
750
500
250
0
-250
Strike price
20
22
24
26
28
30
32
34
36
Price of security

∇