ORF 505: Statistical Analysis of Financial Data

Tuesday Precept 11: Final Review Session

Dec 3, 2024

Lecturer: Qishuo Yin *Scribe: Qishuo Yin*

1 Comprehensive Coding Task on the Stocks Dataset

```
library(Rsafd)
stock_logs = log(stocks) # Load the stocks dataset and take the logarithm
lrets = apply(stock_logs, 2, diff) # Get the difference between each time
```

Listing 1: set up dataset

1.1 Plot the QQ plot

```
qqnorm(lrets[, 9]) # Plot the QQ plot
```

Listing 2: QQ plot

1.2 Fit and use Generalized Pareto Distributions (GPD)

```
EST <- NULL
UNIF <- NULL
GAUSS <- NULL
for ( J in 1:dim(lrets)[2])
{
    # Fit a GPD
    z.est <- fit.gpd(lrets[, J], plot = FALSE)

    # Transform each marginal distribution into a uniform distribution
    # by computing the estimated c.d.f. on the data
    UNIF <- pgpd(z.est, lrets[, J])

    # Transform the uniform marginals so-obtained into N(0,1) Gaussian random variables
    # by evaluating the quantile function of the standard Gaussian distribution on these uniform variates
    GAUSS <- cbind(GAUSS, qnorm(UNIF))
}
```

Listing 3: GPD

1.3 Graphical LASSO

```
# Compute the covariance of the Gaussian
COV <- cov(GAUSS)

# Fit the graphical LASSO by taking rho = 0.45
library(glasso)
glasso_rets <- glasso(COV, rho = 0.45, nobs = dim(lrets)[1])

# Estimate the precision matrix of the Gaussian
prec <- glasso_rets$wi

# Plot the dependence graph
library(qgraph)
qgraph(wi2net(prec), labels = colnames(stocks), layout = "spring", title = "GLASSO for the log-returns")
```

Listing 4: Graphical LASSO

1.4 Principal Component Analysis (PCA)

```
# Conduct the PCA on the dataset
pca_train = princomp(GAUSS)
dim(pca_train$loadings)

# Plot the standard deviations of the projections on the loadings
plot(pca_train$sdev, main = "Standard deviations of the projections on the loadings")

# Plot the proportion of the variance explained by the successive principal components
prop = cumsum( pca_train$sdev * pca_train$sdev / sum(pca_train$sdev * pca_train$sdev))
plot(prop, main = "Proportion of the variance explained by the loadings")
```

Listing 5: PCA

2 Reinforcement Learning Summary Problem

Consider the following simple model in which an agent maximizes their expected utility aggregated over a long time horizon. For practical purposes, we assume that this horizon T is ∞ and we use a discount factor $\gamma = 0.95$ to discount the future daily utilities to today.

On any given day, the utility of the agent is calculated in the following way: on any given day the agent can be `satisfied` or `worried` about the state of their investment. As a result, they can `downsize` their investment, `keep` their portfolio unchanged, or `scale up` their inverstments, the daily untility being a random consequence of the state and the action of the agent according to the following rules:

- If the agent decides to `downsize`, then no matter how they feel, their utility will increase by 1, the agent will have equal chances to be `satisfied` or `worried` tomorrow;

- If the agent is `satisfied` and decides to `scale up` their investments, their utility will increase by 5, and tomorrow they will have 37.5% chance to remain `satisfied`, and 62.5% chance to become `worried`;

- If the agent is `worried` and decides to `scale up`, their utility will add 0.5, and tomorrow they will have 45% chance to be `satisfied` and 55% chance to be `worried`;

- If the agent is `satisfied` and decides to `keep` their portfolio unchanged, their utility will increase by 3, and they will have a 60% chance to remain `satisfied` the following day, and a 40% chance to become `worried`;

- If the agent is `worried` and `keep` their investments unchanged, their utility will increase by 2, and they will have a 47.5% chance to become `satisfied` the following day, and 52.5% chance to remain `worried`.

2.1 Question (a)

Formulate the above problem as a MDP (Markov Decision Process) by clearly identifying the state space, the action space, the transition probability, the reward function, and define the value function to be optimized.

[Solution]

1. State space: $\mathcal{S} = \{\text{satisfied } (s),\ \text{worried } (w)\}$
2. Action space: $\mathcal{A} = \{\text{scale up } (s), \text{keep } (k), \text{downsize } (d)\}$
3. Transition probability:

$$P(\cdot \mid s, d) = P(\cdot \mid w, d) = \begin{cases} \frac{1}{2}, & s' = s \\ \frac{1}{2}, & s' = w \end{cases}$$

$$P(\cdot \mid s, s) = \begin{cases} 0.375, & s' = s \\ 0.625, & s' = w \end{cases}$$

$$P(\cdot \mid w, s) = \begin{cases} 0.45, & s' = s \\ 0.55, & s' = w \end{cases}$$

$$P(\cdot \mid s, k) = \begin{cases} 0.6, & s' = s \\ 0.4, & s' = w \end{cases}$$

$$P(\cdot \mid w, k) = \begin{cases} 0.475, & s' = s \\ 0.525, & s' = w \end{cases}$$

4. Reward function: The utility $r(s, a, s') \in \mathbb{R}$, specifically is as follows:

$$
\begin{aligned}
r(s, d) = r(w, d) &= r_0 + 1 \\
r(s, s) &= r_0 + 5 \\
r(w, s) &= r_0 + 0.5 \\
r(s, k) &= r_0 + 3 \\
r(w, k) &= r_0 + 2
\end{aligned}
$$

The value function is defined as:

$$
V^{\pi} = \mathbb{E}_t \left(\sum_{t=0}^{\infty} 0.95^t \, r(S_t, A_t, S_{t+1}) \right)
$$

2.2 Question (b)

Give in detail the steps in pseudo code of an algorithm, to actually compute this optimal value function and an optimal policy of the above problem.

[Solution]

Algorithm 1 optimal value function and an optimal policy

Initialize the state $\mathcal{S}$.
repeat
Choose action using ϵ-greedy

$$
\begin{cases} \epsilon, & \text{random action} \\ 1 - \epsilon, & \text{greedy action} \end{cases}
$$

Take action, get $r(s)$ and S'.
Update Q-value.
Update current state to new state.
until convergence

References