Exercises for *Statistical Learning*

Chapter 1. Introduction

Paper exercises

1. Give the differential entropy of the Gaussian distribution: $p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

2. Give the K-L divergence of two Gaussian distributions: $p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ and $q(x) = \frac{1}{\sqrt{2\pi}\tau} e^{-\frac{(x-\nu)^2}{2\tau^2}}$.

3. Using the Lagrange dual method, prove that given $x \in \{1,2,\dots,N\}$, the discrete uniform distribution $\forall x, p(x) = \frac{1}{N}$ has the maximal entropy.

Programming exercises

1. Choose a pair of parameters (μ, σ^2) to generate N random numbers that follow the Gaussian distribution, then calculate the mean and variance of the random numbers and compare with the ground-truth (μ, σ^2). Set N to 100, 1000, 10000, 100000… to observe the change of results.
2. Learn how to use constrained nonlinear optimization functions (e.g. in Python-SciPy or MATLAB), select one to solve the following example problem:

$$\min_{x_1, x_2} 10 - x_1^2 - x_2^2$$

$$\text{subject to } x_1^2 - x_2 \leq 0, x_1 + x_2 = 0$$

Additional exercises

1. State more than 3 examples in your familiar field to argue the utility of statistical learning methods in practice.
2. Are there any other methods for machine learning beyond statistical approaches? You may refer to the textbook “Machine Learning” written by T. M. Mitchell.

Chapter 2. Data Analysis

Paper

1. For the Gaussian distribution with parameters (μ, σ^2), give the mode, median and percentiles at 25%, 75%, 90%, 95%.

2. Give the mean, mode, and median of the log-normal distribution: $p(x) = \frac{1}{x\sqrt{2\pi}\sigma} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$.

3. Give the PDF of power-law distribution that is defined on $[x_{min}, +\infty)$.

4. Calculate the ratio between the volume of a d-dimensional hypersphere with radius r and the volume of a d-dimensional hypercube with edge length $2r$. Show that the ratio approaches to 0 as d increases to infinity.

5. Prove that the covariance matrix of the transformed data D' in the Karhunen–Loève transform is a diagonal matrix.

Programming

1. Calculate the covariance matrix and the correlation coefficient matrix of the Iris data set. Perform the Karhunen–Loève transform on this data set.

2. Plot 2-dimensional histograms, with every pair of the 4 variables in the Iris data set.

3. Plot boxes to show the distributions of the 4 variables in the Iris data set.

4. Perform dimensionality reduction on the Iris data set using PCA, LLE, and ISOMAP to draw some 2-dimensional scatter plots.

Additional

1. Among empirical CDF and histogram, which one do you prefer to use for probabilistic modeling? Why?

2. State more than 3 examples of feature selection or feature extraction in your familiar field.

3. How can we evaluate and compare the performance of different dimensionality reduction methods, given a real-world data set? Please use your results obtained in the Programming Exercise 4 to analyze the performance of PCA, LLE, and ISOMAP on the Iris data set.

4. What is stem plot and how useful is it? You may refer to textbooks about exploratory data analysis.

Chapter 3. Supervised Learning

3.1 Supervised Learning Theory

Paper

1. Given training samples: $(\boldsymbol{x}_n, y_n), n = 1,2,\dots,N$, where $y_n \in \{-1,1\}$. Assume we use the linear classifier: $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + b)$, and our objective is to minimize the empirical risk: $R_{emp}(\boldsymbol{w}, b) = -\frac{1}{N}\sum_{n\in\mathcal{N}}[y_n \times (\boldsymbol{w}^T\boldsymbol{x} + b)]$, where $\mathcal{N} = \{n|y_n \neq f(\boldsymbol{x}_n)\}$. Design a training algorithm by the gradient descent method.

2. Similar to Paper Exercise 1, assume the linear classifier is now $f(\boldsymbol{x}) = \text{sign}(\sum_{j=1}^{N}\alpha_j y_j \boldsymbol{x}_j \cdot \boldsymbol{x} + b)$. Design a training algorithm.

3. Give the closed-form expression of $\widehat{\boldsymbol{w}}$ in polynomial curve fitting.

4. Define a loss function when the objective of learning is to maximize the *precision*. Define another loss function when the objective is to maximize the *recall*.

5. Assume we use 0-1 loss function for binary classification and the hypothesis space consists of a finite number (say M) of functions. Prove that the following inequality is satisfied with probability $1-\eta$: $R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{1}{2N}(\log M - \log\eta)}$, where N is the number of training samples. Hint: use the Hoeffding's inequality (http://en.wikipedia.org/wiki/Hoeffding%27s_inequality).

6. Some functions look quite simple but cannot be represented by linear classifiers. Show that $f(\boldsymbol{x}) = \frac{1}{2}(\text{sign}(\boldsymbol{w}^T\boldsymbol{x} + b) + 1)$ cannot represent the XOR function. What about the XNOR function?

Additional

1. What are the model, strategy, and algorithm of the *k*-NN classifier?

2. If the hypothesis space consists of a finite number of functions, how can we do structural risk minimization?

3. What is *k*-d tree and how can it be utilized to speed up the *k*-NN classifier? You may refer to textbooks on advanced data structures.

3.2 Linear Regression

Paper

1. Let the prior be $p(w) = \mathcal{N}(w|\mu, \sigma^2)$ and the likelihood function be $p(t|w) = \mathcal{N}(t|wx, \tau^2)$. Give the posterior $p(w|t)$.

2. Prove that the Beta distribution is the conjugate prior of the Bernoulli distribution, i.e. if the prior follows the Beta distribution: $p(w) = \frac{w^{\alpha-1}(1-w)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$, where $\mathrm{B}(\alpha, \beta)$ is the Beta function, and the likelihood function is $p(t|w) = \begin{cases} 1-w & t=0 \\ w & t=1 \end{cases}$, then the posterior $p(w|t)$ also follows the Beta distribution.

3. Consider a weighted regression problem. Each training sample $(\boldsymbol{x}_n, y_n)$ has a positive weight r_n, and thus the weighted empirical risk is $R_{emp}(\boldsymbol{w}) = \frac{1}{2N}\sum_{n=1}^{N} r_n(y_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2$. Give its solution $\widehat{\boldsymbol{w}}$. Add quadric regularization and define the structured risk as $R_{str}(\boldsymbol{w}) = \frac{1}{2N}\sum_{n=1}^{N} r_n(y_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$. Give its solution.

4. Give the equivalent kernel function corresponding to the basis function $\phi_0(\mathbf{x}) = 1$.

5. Give the algorithm details for estimating the parameters α and β for linear regression with the Bayesian model selection.

Programming

1. Randomly generate 100 datasets, each of which consists of 25 points that are samples of the function $y = \sin(2\pi x)$ plus Gaussian noise. Perform Ridge regression on each dataset with 7th order polynomial with different values of λ. Observe the results with respect to λ.

2. Choose one dataset that you have generated in Programming Exercise 1. Perform Ridge regression with 3rd, 4th, …, 9th order polynomials. Estimate α and β using the algorithm that you have designed in Paper Exercise 5, and then calculate the model evidence term $\ln p(\mathbf{t}|\alpha, \beta)$. Which order of polynomial shall be used? Observe the results of β with respect to the order of polynomial.

Additional

1. Let the original signal be $x(t)$, the observed signal $y(t)$ is disturbed by additive white Gaussian noise, i.e. $y(t) = x(t) + n(t)$. State that denoising—to recover $x(t)$ from $y(t)$—is an ill-posed problem. How to tackle this problem with regularization? You may refer to textbooks on advanced digital signal processing.

2. There are two approaches in statistics, known as the Frequentists and the Bayesians. What are the major differences between these two approaches? Which one do you prefer, why? You may refer to textbooks on advanced statistics.

3.3 Linear Classification

Paper

1. Give the distance from the point $\boldsymbol{x}_0$ to the decision boundary $\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$.
2. Give the maximum likelihood solution to the likelihood function:
$p(x_1, \dots, x_N, t_1, \dots, t_N|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^{N}[\pi\mathcal{N}(x_n|\mu_1, \Sigma)]^{t_n}[(1-\pi)\mathcal{N}(x_n|\mu_2, \Sigma)]^{1-t_n}$.
3. Prove that the Bernoulli distribution belongs to the exponential family. Give its variance according to the properties of the exponential family.
4. Does log-normal distribution belong to the exponential family? What about the power-law distribution?
5. Let a random variable $x \in \{A, B, C, D, E\}$. Observed that $P(A) + P(B) = 0.3$ and $P(A) + P(C) = 0.5$. Estimate $P(x)$ according to the maximum entropy principle.

Programming

1. Implement the learning algorithms for perceptron in both the original form and the dual form. Test your program with the following training data: $\boldsymbol{x}_1 = (3,3)^T, y_1 = 1, \boldsymbol{x}_2 = (4,3)^T, y_2 = 1, \boldsymbol{x}_3 = (1,1)^T, y_3 = 0$, and verify these data are linearly separable.

Additional

1. State the relations and differences between binary classification, multi-class classification, and multi-label classification.
2. What is the Akaike information criterion (AIC) and how does it compare with the Bayesian information criterion (BIC)? You may refer to textbooks on advanced statistics.

3.4 Generative Bayesian Approaches

Paper

1. Give the PDF for the Bayes network shown in Fig. 3.4.1. Draw the corresponding factor graph.

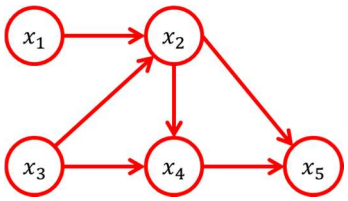

Fig. 3.4.1

2. If the PDF satisfies $p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_1, x_3)p(x_5|x_3, x_4)$, draw the corresponding Bayes network, Markov random field, and factor graph.

3. Represent the Logistic regression problem using the Bayes network.

4. Give the PDF for the Bayes network shown in Fig. 3.4.2, which is the famous latent Dirichlet allocation (LDA).

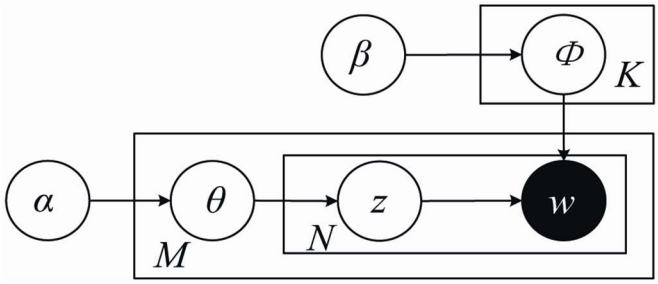

Fig. 3.4.2

5. Given the Bayes network shown in Fig. 3.4.3, (1) are x_5, x_6 conditionally independent, given x_1, x_3? (2) what about x_5, x_6 given x_2, x_3? (3) what about x_2, x_5 given x_6, x_7?

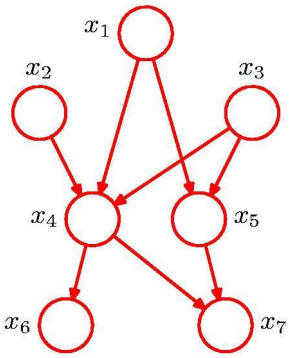

Fig. 3.4.3

6. Give the Markov blanket for each random variable in the Bayes network shown in Fig. 3.4.3.

7. Give one possible PDF for the Markov random field shown in Fig. 3.4.4 and draw the corresponding factor graph.

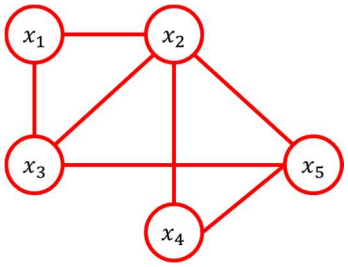

Fig. 3.4.4

8. Prove that no Bayes network is equivalent to the Markov random field shown in Fig. 3.4.5.

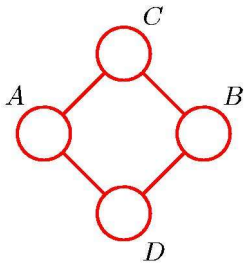

Fig. 3.4.5

9. Consider the toy example of object recognition. If we observed an irregular, yellow object, what are the posterior probabilities of the 3 categories?

Programming

1. Implement a naïve Bayes classifier for discrete input and output. Use the data given in Table 3.4.1 to train the classifier and test it with $x_1 = 1, x_2 = m$. Add Laplace smoothing into the classifier (with $\alpha = 1$), and test again.

Table 3.4.1 Training data set

x_1	1	1	1	1	1	2	2	2	2	2	3
x_2	*s*	*m*	*m*	*s*	*s*	*s*	*m*	*m*	*l*	*l*	*l*
y	-1	-1	1	1	-1	-1	-1	1	1	1	1

2. Implement a denoising algorithm for binary images with Markov random field and the ICM method. Test with randomly generated binary images plus random noise and observe the results with respect to β, η.

Additional

1. Give the rigid proof for the "Gold after doors" problem, i.e. to calculate the probabilities and to prove that the guesser's strategy should be changing to another door.

2. Study one of the approximate inference methods for graphical models, such as Monte Carlo sampling especially the Markov chain Monte Carlo (MCMC), Variational Bayesian, Expectation propagation, Generalized belief propagation, and so on.

3.5 Combining Models

Paper

1. Training data are given in Table 3.5.1. Generate a classification tree with the C4.5 algorithm (no pruning). Generate another classification tree with the AdaBoost algorithm with decision stump.

Table 3.5.1 Training data set

n	x_{n1} (age)	x_{n2} (occupied)	x_{n3} (has estate)	x_{n4} (credit)	y_n (class)
1	Young	No	No	Fair	No
2	Young	No	No	Good	No
3	Young	Yes	No	Good	Yes
4	Young	Yes	Yes	Fair	Yes
5	Young	No	No	Fair	No
6	Mid-life	No	No	Fair	No
7	Mid-life	No	No	Good	No
8	Mid-life	Yes	Yes	Good	Yes
9	Mid-life	No	Yes	Very Good	Yes
10	Mid-life	No	Yes	Very Good	Yes
11	Old	No	Yes	Very Good	Yes
12	Old	No	Yes	Good	Yes
13	Old	Yes	No	Good	Yes
14	Old	Yes	No	Very Good	Yes
15	Old	No	No	Fair	No

2. Training data are given in Table 3.5.2. Generate a regression tree with the CART algorithm (no pruning). Generate another regression tree with the boosting tree algorithm with decision stump. Training stops when mean squared error (MSE) is less than 0.1.

Table 3.5.2 Training data set

x_n	1	2	3	4	5	6	7	8	9	10
y_n	4.50	4.75	4.91	5.34	5.80	7.05	7.90	8.23	8.70	9.00

3. Using the CART algorithm to prune the regression tree that you have generated in Paper Exercise 2. Let $C(T)$ be mean squared error and $|T|$ be the number of leaf nodes. Draw illustrations of the pruned trees given different values of α.

Programming

1. One leaf node in a regression tree may indicate a regression function rather than a constant. Implement a regression tree generation algorithm when the leaf nodes indicate 3rd order polynomials. Test your program with the dataset that you had generated in the Programming Exercise 1 in Section 3.2, and compare the results with those of simple linear regression.

2. Generate a set of random numbers that follow a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Using bootstrap samples from this set to estimate the empirical posterior of μ and σ^2. Repeat and observe the accuracy of empirical posterior with respect to the number of bootstrap samples.

Additional

1. Is it possible that the AdaBoost algorithm overfits training data? How can we avoid such overfitting by regularization?
2. What are the relations between the empirical posterior achieved by bootstrap samples and the "real" posterior given by the Bayesian estimation? You may refer to textbooks on advanced statistics.

3.6 Artificial Neural Network

Paper

1. Design a two-layer perceptron network to perform the XOR function: $y = x_1 \, XOR \, x_2$.
2. Design the back propagation algorithm for multi-layer forward neural network, where the activation function is the Logistic sigmoid function: $h(a) = \frac{1}{1+e^{-a}}$.
3. Revise the back propagation algorithm if the activation function changes to: $h(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$.
4. Revise the back propagation algorithm if the activation function is the Logistic sigmoid and the optimization target change to: $\min_{\boldsymbol{w}} \frac{1}{N}\sum_{n=1}^{N} E_n(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$.

Programming

1. Implement the back propagation algorithm that you have designed in the Paper Exercise 2. Test the neural network with the data set that you had generated in the Programming Exercise 1 of Section 3.2. Increase the number of layers and the number of hidden-layer neurons to observe possible over-fitting.

Additional

1. What is Restricted Boltzmann machine (RBM) and how to train it?
2. Read the paper "Reducing the Dimensionality of Data with Neural Networks" and its supplemental materials. URLs for downloading:
paper (http://www.cs.toronto.edu/~hinton/science.pdf),
supplemental materials (http://www.cs.toronto.edu/~hinton/absps/science_som.pdf),
code (http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html).

3.7 Support Vector Machine

Paper

1. Given training data as in Table 3.7.1, calculate the learnt linear SVM, and draw an illustration to depict the decision boundary, margin boundary and support vector(s).

Table 3.7.1 Training data

$\boldsymbol{x}_n$	$(1,2)^T$	$(2,3)^T$	$(3,3)^T$	$(2,1)^T$	$(3,2)^T$
y_n	1	1	1	-1	-1

2. Linear SVM may be learnt by solving:

$$\min\left(\frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 + C\sum_{n=1}^{N}\xi_n^2\right)$$

$$\text{s.t.}\begin{cases}\forall n, y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \geq 1 - \xi_n \\ \forall n, \xi_n \geq 0\end{cases}$$

Give its dual form.

3. Draw a figure to display several loss functions for binary classification in the same coordinates, where the *x*-axis is $y \times f(x)$ and the *y*-axis is the loss. Loss functions include: 0-1 loss function, the loss function of perceptron, exponential loss function (AdaBoost), and surrogate loss function (SVM).

4. Prove that $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2\sigma^2})$ is a kernel function, i.e. for any two vectors $\boldsymbol{x}, \boldsymbol{y}$, $K(\boldsymbol{x}, \boldsymbol{y})$ can be written as $\boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{y})$, where $\cdot$ stands for inner product. Hint: use Taylor's expansion on the exponential function.

Programming

1. Learn how to use SVM light (http://svmlight.joachims.org/) or LIBSVM (http://www.csie.ntu.edu.tw/~cjlin/libsvm/), or other SVM software (e.g. in Python-scikit-learn).

2. Implement kernel PCA with Gaussian kernels. Test your implementation on the Iris data set and compare with the previous results in the Programming Exercise 4 of Chapter 2.

Additional

1. How to apply the kernel trick into Fisher's linear discriminant analysis?

3.8 Sparse Models

Paper

1. Solve the optimization problem:

$$\arg\min |\boldsymbol{x}|_1 , \text{s.t.} |\boldsymbol{x}|^2 = c$$

where c is a positive constant.

2. Solve the optimization problem:

$$\arg\min |\boldsymbol{x}|_p , \text{s.t.} |\boldsymbol{x}|_q = c$$

where c is a positive constant, $p > 0$, $q > 0$. Hint: consider the cases $p < q$ and $p > q$.

3. Prove that the following two optimization problems are equivalent:

$$J_1(\boldsymbol{x}) = |\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}|^2 + \lambda_1 |\boldsymbol{x}|_1 + \lambda_2 |\boldsymbol{x}|^2$$

$$J_2(\boldsymbol{x}) = |\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{A}}\boldsymbol{x}|^2 + c\lambda_1 |\boldsymbol{x}|_1$$

i.e. by choosing some a $\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{A}}$, we can achieve $\arg\min J_1(\boldsymbol{x}) = c(\arg\min J_2(\boldsymbol{x}))$.

4. According to the following description, design two algorithms to solve the optimization problem:

$$\arg\min |\boldsymbol{x}|_1 , \text{s.t.} \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

The first approach is gradient descent.

The second approach is via variable substitution. Let $\boldsymbol{x} = (x_1, \dots, x_D)^T$, $u_d = |x_d|, d = 1, \dots, D$, and $\boldsymbol{u} = (u_1, \dots, u_D)^T$. Then we may solve the following problem instead of the original one:

$$\arg\min \sum_{d=1}^{D} u_d , \text{s.t.} \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, -\boldsymbol{u} \le \boldsymbol{x} \le \boldsymbol{u}, \boldsymbol{u} \ge \boldsymbol{0}$$

which is a linear program.

Programming

1. First, implement the two algorithms that you have designed in the Paper Exercise 4. Second, generate a random D-dimensional sparse vector $\boldsymbol{x}$ (say there are K non-zero entries in the sparse vector; you may first choose the locations of these K non-zero entries randomly, and then generate the non-zero values randomly). Third, generate a random matrix $\boldsymbol{A}$ with size $M \times D$. Fourth, given $\boldsymbol{A}$ and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, use your implemented algorithms to solve $\boldsymbol{x}$, and compare with the ground-truth $\boldsymbol{x}$. Repeat with different values of K, M, D and observe the change of results.

Additional

1. Compare the *k*-NN method and sparse model-based method for face recognition.

Chapter 4. Beyond Supervised Learning

Paper

1. Draw some illustrations to show that different clustering approaches (including centroid-based, density-based, and connectivity-based) can give quite different results for the same clustering problem.

2. Consider the clustering problem: $\{\boldsymbol{x}_i \to q(\boldsymbol{x}_i), i = 1 \dots N, q(\boldsymbol{x}_i) \in \{1 \dots K\}\}$, where the center of the j-th cluster is $\boldsymbol{c}_j, j = 1 \dots K$. And the optimization target is to minimize

$$\sum_{i=1}^{N} |\boldsymbol{x}_i - \boldsymbol{c}_{q(\boldsymbol{x}_i)}|$$

How to solve this problem? Hint: imitate the *k*-means algorithm.

3. Given observed data points: -67, -48, 6, 8, 14, 16, 23, 24, 28, 29, 41, 49, 56, 60, 75. Estimate the parameters of two-component Gaussian mixture model.

4. Consider the "Laplace mixture model," which is quite similar to the Gaussian mixture model with the only replacement of the Gaussian distribution by the Laplace distribution,

$$p(x_i) = \sum_{j=1}^{K} w_j \mathcal{L}(x_i | \mu_j, b_j)$$

where the (one dimensional) Laplace distribution has PDF: $\mathcal{L}(x|\mu_j, b_j) = \frac{1}{2b_j} e^{-\frac{|x-\mu_j|}{b_j}}$. Design an algorithm to estimate the parameters of the Laplace mixture model, using maximum likelihood criterion and the EM approach.

5. Give the mathematical formulation of the problem for training nonlinear transductive support vector machine using kernels.

6. Fig. 4.1 shows a graph with 3 nodes and 5 edges and the edge weights are all 1.0. Calculate the ranks of the nodes, using the PageRank algorithm with damping factor β equal to 0.85 or 1.

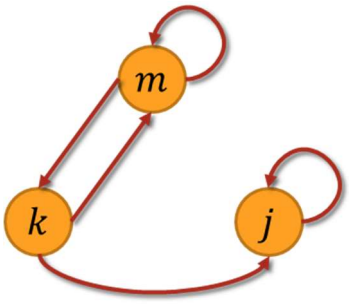

Fig. 4.1

Programming

1. Try with SVM light (http://svmlight.joachims.org/) or other SVM software to train the transductive support vector machine.

2. Randomly generate a graph with N nodes and $\frac{N(N-1)}{2}$ weights for edges, calculate the ranks for the nodes using PageRank. Increase N to observe the computational complexity of the PageRank algorithm.

Additional

1. Argue, with an example in your familiar field, that the cost of obtaining training data is often very high, and therefore semi-supervised learning is required.
2. What is the underlying principle of the Apriori algorithm for association analysis?