Written Exam
Introduction to Linear and Integer Programming (DM825)

Department of Mathematics and Computer Science
University of Southern Denmark

Thursday, June 15, 2012, 10:00–13:00, U49 and U49B

The rules for the exam are explained in the document Guidelines for the Exam available from the link:
http://www.imada.sdu.dk/~marco/DM825/DM825_Samlet_vejl_stud.pdf
The content of that document is assumed known.

The exam consists of 6 tasks and relative subtasks distributed on 8 pages.

The weight in the evaluation of each task and subtask is given in points. The total sum is 105 points even though the grades will be defined from 0 to 100. The amount of points assigned to the subtasks are not proportional to the difficulty of the question (rather the inverse).
A few subtasks are dependent on previous subtasks. One can buy an hint to some subtasks asking to the exam vigilantes to call the instructor. The cost of an hint to a subtask is equivalent to the half of the credit of the subtask. That is, if answered correctly after an hint, the subtask will count for half of its credit.

Remember to justify all your statements. You may refer to results from the textbooks or the lecture slides. In particular, it is possible to justify a statement by saying that it derives trivially from a result in the textbook (if this is true!). You may use all methods or extensions that have been used in the exercise sheets, published during the course. However, it is not allowed to answer a subtask exclusively by reference to an exercise seen during the course. Reference to other books (outside the course material) is not accepted as answer to a task!

You may write your answers in Danish or in English.

Task 1

Assume that we have a training set consisting of examples $(\vec{x}^i, y^i)$ for $i = 1, \ldots, n$. The task is a binary classification problem so each label y_i is either -1 or $+1$.
In training a *hard margin SVM with bias*, the final classifier is $\hat{y} = \text{sign}(\vec{\theta} \cdot \vec{x} + \theta_0)$ where the parameters $\vec{\theta}$ and θ_0 solve the following primal and dual optimization problems:

Primal: find $\vec{\theta}, \theta_0$ that
minimize $\frac{1}{2} \parallel \vec{\theta} \parallel^2$
subject to $y_i(\vec{\theta} \cdot \vec{x}_i + \theta_0) \geq 1, i = 1, \ldots, n$

Dual: find α_i that
maximize $\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\alpha_i \alpha_j y_i y_j \vec{x}_i \cdot x_j$
subject to $\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$

Subtask 1.a (10 points)

Assume that $n = 4$, and that $x_1 = (1, 1)^T$, $x_2 = (2, 2)^T$, $x_3 = (-1.5, -1.5)^T$, and $x_4 = (4, 4)^T$. We now train an SVM with bias, and in addition *with slack variables*. Show that for any labeling of the four training examples, the optimal parameter vector $\hat{\vec{\theta}} = (\hat{\theta}_1, \hat{\theta}_2)^T$ has the property that $\hat{\theta}_1 = \hat{\theta}_2$.

Solution The SVM solution is of the form $\vec{\theta} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$. Since all the points have $x_{i1} = x_{i2}$, this property has to hold for $\vec{\theta}$ as well.

Subtask 1.b (10 points)

Consider the SMO algorithm applied to training a hard-margin SVM with slack variables and a bias variable. Initially all dual variables α_i for $i = 1, \ldots, n$ are set to 0. At each step in the SMO algorithm, two variables α_i and α_j are chosen with the special heuristic that $y_i = y_j$; these two variables are optimized in the usual way for SMO. What solution will you find with this constrained SMO procedure? Justify your answer showing the steps of the SMO algorithm.

Solution Consider first modifying α_i and α_j and set all the other α_k's to zero. The dual constraint $\sum_{i=1}^{n} \alpha_i y_i = 0$ then requires that $y_i \alpha_i + y_j \alpha_j = 0$ since the other α_k's are zero. When the labels agree, this implies that $\alpha_i + \alpha_j = 0$. Since α's are positive, we can only have $\alpha_i = \alpha_j = 0$. As a result, the SMO algorithm won't change any α's from zero.

Subtask 1.c (10 points)

Consider the kernel $K(\vec{x}, \vec{z}) = \vec{x} \cdot \vec{z} + 4(\vec{x} \cdot \vec{z})^2$ where the vectors $\vec{x}$ and $\vec{z}$ are 2-dimensional. This kernel is equal to an inner product $\vec{\phi}(x) \cdot \vec{\phi}(z)$ for some definition of ϕ. What is the function ϕ?

Solution Since $\vec{x} \cdot \vec{z} = (x_1 z_1)^2 + 2(x_1 x_2)(z_1 z_2) + (x_2 z_2)^2$ we have

$$
\begin{aligned}
K(\vec{x},\vec{z}) &= x_1z_1 + x_2z_2 + 4(x_1z_1)^2 + 8(x_1x_2)(z_1z_2) + 4(x_2z_2)^2 \\
&= [x_1, x_2, 2x_1^2, 2\sqrt{2}x_1x_2, 2x_2^2] \cdot [z_1, z_2, 2z_1^2, 2\sqrt{2}z_1z_2, 2z_2^2]
\end{aligned}
$$

Thus $\phi(\vec{x}) = [x_1, x_2, 2x_1^2, 2\sqrt{2}x_1x_2, 2x_2^2]$

Subtask 1.d (5 points)

Consider training an SVM with slack variables, but with no bias variable. The kernel used is $K(\vec{x},\vec{z})$; it has the property that for any two points x_i and x_j in the training set, $-1 < K(\vec{x}_i,\vec{x}_j) < 1$. $K(\vec{x}_i,\vec{x}_i) < 1$ as well. There are n points in the training set. Show that for having all dual variables α_i non-zero (i.e., all points in the training set become support vectors) the slack-variable constant C must be chosen to be $C \leq \frac{1}{n}$.

Solution The SVM solution is of the form $\vec{\theta} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$ and in the absence of bias the KKT condition becomes:

$$\alpha_i[y_i(\vec{x}_i^T\vec{\theta} + \theta_0) - (1 - \xi_i)] = 0$$

$$\alpha_i[y_i(\sum_{i=1}^{n} \alpha_i y_i \vec{x}_i \vec{x}_i^T) - (1 - \xi_i)] = 0$$

Then a point is a support vector if $\alpha_i > 0$ which leaves the condition that:

$$y_i(\sum_{j=1}^{n} \alpha_i y_i \vec{x}_i \vec{x}_j^T) - (1 - \xi_i) = 0$$

with the Kernel trick:

$$y_i(\sum_{j=1}^{n} \alpha_i y_i K(\vec{x}_i, \vec{x}_j^T)) - (1 - \xi_i) = 0$$

If a point is a support vector and correctly classified, ie, $\xi_i = 0$:

$$y_i(\sum_{j=1}^{n} \alpha_i y_i K(\vec{x}_i, \vec{x}_j^T)) = 1$$

If a point is a support vector and wrongly classified, ie, $\xi_i > 0$:

$$y_i(\sum_{j=1}^{n} \alpha_i y_i K(\vec{x}_i, \vec{x}_j^T)) + \xi_i = 1$$

$$y_i(\sum_{j=1}^{n} \alpha_i y_i K(\vec{x}_i, \vec{x}_j^T)) < 1$$

Hence

$$y_i(\sum_{j=1}^{n} \alpha_i y_i K(\vec{x}_i, \vec{x}_j^T)) \leq 1$$

$$y_i(\sum_{j=1}^{n} \alpha_i y_i K(\vec{x}_i, \vec{x}_j^T) \leq \sum_{j=1}^{n} \alpha_i |K(\vec{x}_i, \vec{x}_j)^T| \leq \sum_{j=1}^{n} \alpha_i \leq 1$$

We know that $0 < \alpha_i \leq C$ hence

$$\sum_{j=1}^{n} \alpha_i \leq \sum_{j=1}^{n} C \leq 1$$

$$C \leq \frac{1}{n}$$

Task 2

Let's consider a test T for an event A. The frequency of *false negatives* is 0.01, and the frequency of *false positives* is 0.

Subtask 2.a (10 points)

A company producing semiconductor chips has a test for processors under the suspicion of been flawed, e.g., processors that require a particular technology that is known to be risky. The test has the above characteristics in case of a flaw the test returns a certificate for it, that is, the wrongly computed operation. Experience says that 20% of the processors under suspicion do in fact contain some flaws in their functions. A suspicious processor has a negative test, that is, the test indicates that the processor has no flaw. What is the probability that the processor is nevertheless flawed?

Solution

	A = flawed	A = no
T = yes	0.99	0
T = no	0.01	1

Tabel 1: Conditional probabilities $P(T \mid A)$ characterizing test T for A.

$$\begin{aligned} p(A = \mathit{flawed} \mid T = no) &= \frac{p(T = no \mid A = \mathit{flawed}) p(A = \mathit{flawed})}{\sum_A p(T = no \mid A) p(A)} \\ &= \frac{0.01 * 0.2}{0.01 * 0.2 + 1 * 0.8} = 0.0025 \end{aligned}$$

Subtask 2.b (10 points)

The company decides to test all processors independently of the producing technology, and use the same test. It is estimated that one out of 1,000 processors is flawed. A processor has a negative test result. What is the probability that the processor is nevertheless flawed?

Solution

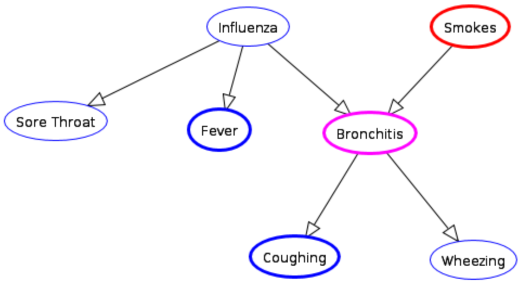
$$
\begin{aligned}
p(A = flawed \mid T = no) &= \frac{p(T = no \mid A = flawed)p(A = flawed)}{\sum_A p(T = no \mid A)p(A)} \\
&= \frac{0.01 * 0.001}{0.01 * 0.001 + 999/1000} = 1 \times 10^{-5}
\end{aligned}
$$

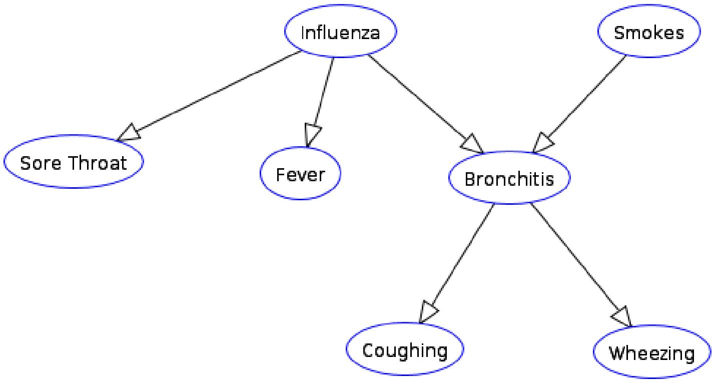
Task 3 (10 points)

Consider the Bayesian Network in Figure 1.
Answer and provide a justification to the following questions:

Is Smokes conditionally independent of Bronchitis given Fever and Coughing?

Solution No, there is a path from Smoke to Bronchitis and observing Bronchities or one of its descendants will explain Smoke.

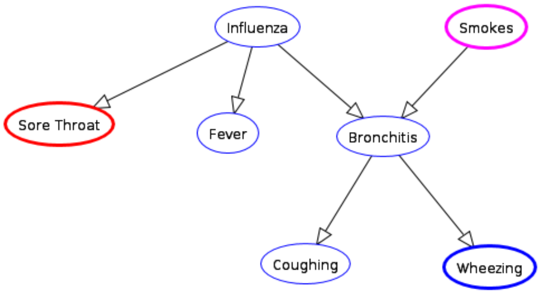




Figur 1:

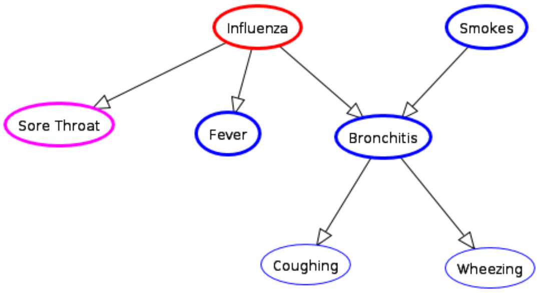
Is Sore Throat conditionally independent of Smokes given Wheezing?

Solution No, there is an head to head connection and one of the descendant is observed.



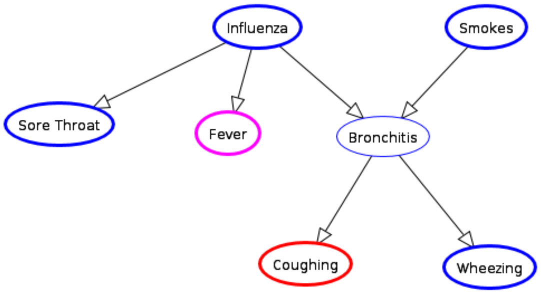
Is Influenza conditionally independent of Sore Throat given Smokes, Bronchitis and Fever?

Solution No, observing Sore Throat tells us something about Influenza



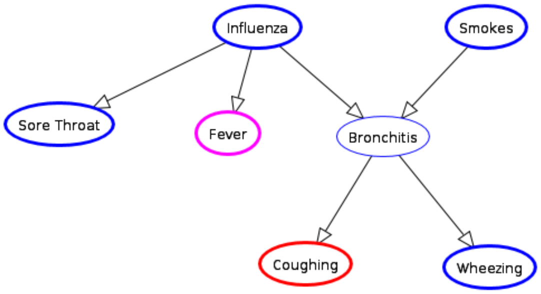
Is Coughing conditionally independent of Fever given Wheezing, Influenza, Smokes and Sore Throat?

Solution Yes, the only path is blocked.



Is Sore Throat conditionally independent of Coughing given Smokes?

Solution False, there is a path not blocked.



Task 4

Design a Bayesian network that can be used to recognize handwritten digits $0, 1, 2, \ldots, 9$ from scanned, pixelated images like the ones in Figure 2. Assume for simplification a 9×9 grid.

Subtask 4.a (5 points)

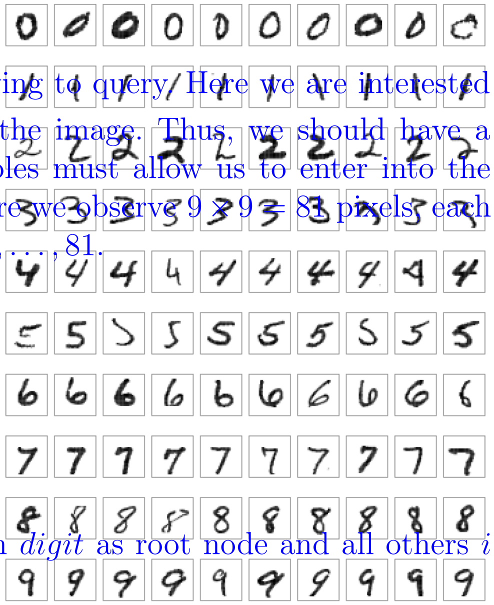
Explain what are hidden and observation variables.

Solution The hidden variable(s) are those we are trying to query. Here we are interested in predicting which digit is actually represented by the image. Thus, we should have a variable $digit \in \{0, 1, \dots, 9\}$. The observation variables must allow us to enter into the Bayesian networks the observations that we make. Here we observe $9 \times 9 = 81$ pixels, each of which can be black or white: pixel $i \in \{b, w\}, i = 1, \dots, 81$.

Subtask 4.b (5 points)

Draw the graphical model so that the conditional independencies are (approximately) reasonable. Explain and motivate your choice.

Solution A naive Bayesian network would work with *digit* as root node and all others i pixels as leaves.



Figur 2:

Subtask 4.c (5 points)

Give one or more example of useful mediating variables (consider e.g. the desired invariance property with respect to translations, rotations, etc.)

Solution One could add another root representing an hidden variable for example transalition. In this way neighboring pixels would have a dependency conditional to the digit and to the type of transformation that has been applied.

Subtask 4.d (5 points)

Explain shortly how you would fill in the conditional probability tables given the samples in Figure 2.

Solution We could solve a counting problem derived from maximum likelihood approach. Thus we would have at the root a multinomial variable with equal probability for each outcome, ie. $\mu_k = 1/10$ and at each node representing a pixel we would have for each digit the ratio between the number of times the pixel is black on the corresponding row and the number of columns n of the matrix represented in Figure 2.

$$p_{1,1}(p = b \mid d = 9) = \frac{\sum_{i=1}^{n} I\{pixel^{i}_{1,1} = b\}}{n}$$

Task 5 (10 points)

Consider a binary classification problem with continuous-valued features. If we apply Gaussian discriminant analysis using the same covariance matrix Σ for both classes, then the resulting decision boundary is linear. Show that if we modeled the two classes using separate covariance matrices Σ_0 and Σ_1 (i.e., $x^i \mid y^i = b \sim N(\mu_b, \Sigma_b)$, for $b = 0$ or 1) the decision boundary would be quadratic.

Solution The decision boundary is given by the following equation

$$\log(p = 0 \mid x) = \log(p = 1 \mid x)$$

using Bayes rule:

$$\log(x \mid p = 0) + \log(p = 0) - \log p(x) = \log(x \mid p = 1) + \log(p = 1) - \log p(x)$$

$$C_0 - \frac{1}{2}(x - \mu_0)^T \Sigma_0^{-1}(x - \mu_0) = C_1 - \frac{1}{2}(x - \mu_1)^T \Sigma_1^{-1}(x - \mu_1)$$

This is a quadratic decision boundary in x.

Task 6 (10 points)

Consider a set of models of the form $p(y \mid \vec{x}, \vec{\theta}_h, h)$ in which $\vec{x}$ is the input vector, y is the response variable, h indexes the different models and θ_h is the set of parameters for model h. Suppose the models have prior probabilities $p(h)$ and that we are given a training set $X = \{\vec{x}_1, \dots, \vec{x}_N\}$ and $Y = \{y_1, \dots, y_N\}$. Write down the formulae needed to evaluate the predictive distribution $p(y \mid \vec{x}, X, Y)$ in which the model index is marginalized out.

Solution The required predictive distribution is given by

$$p(\vec{y} \mid \vec{x}, X, Y) = \sum_h p(h) \int p(\vec{y} \mid \vec{x}, \vec{\theta}_h, h) p(\vec{\theta}_h | X, Y, h) d\vec{\theta}_h$$

where

$$\begin{aligned} p(\vec{\theta}_h | X, Y, h) &= \frac{p(Y | X, \vec{\theta}_h, h) p(\vec{\theta}_h \mid h)}{p(Y \mid X, h)} \\ &\propto p(\vec{\theta} \mid h) \prod_{i=1}^{m} p(y_i | \vec{x}_i, \vec{\theta}, h) \\ &= p(\vec{\theta} \mid h) \prod_{i=1}^{m} (\sum_{z_h^i} p(y^i | \vec{x}^i, \vec{\theta}, h)) \end{aligned}$$