A

Final Exam

GU4241/GR5241 Fall 2016

Name

UNI

Problem 0: UNI (2 points)

Write your name and UNI on the exam book **and** on the first page of the problem sheet. After the exam, please put the problem sheet into the exam book and return **both** to us.

Problem 1: Short questions (2+2+3+3+4+4 points)

Please briefly EXPLAIN you answers. Short explanation (about one sentence) is sufficient.

(a) (**True/False**) The number of nodes in a decision tree can be larger than the number of the features in the data to train that tree.

(b) (**True/False**) The number of nodes in a decision tree can be larger than the number of data points used to train that tree.

(c) Why is the dual formulation of the SVM optimization problem so important when we work with a kernel (say, the RBF kernel)?

(d) Why can Newton's method be slower than gradient descent in high dimensions, even though it converges at a faster rate?

(e) Can regularization reduce training mean squred error (MSE)? What about test MSE?

(f) Considier a soft-margin, linear support vector machine:

$$\begin{aligned} \min_{\mathbf{v}_H, b, \xi} \quad & \|\mathbf{v}_H\|^2 + C\sum_{i=1}^{n} \xi_i^2 \\ \text{s.t.} \quad & y_i(\langle \mathbf{v}_H, x_i \rangle - b) \geq 1 - \xi_i, \quad \text{for } i = 1, \dots, n \\ & \xi_i \geq 0, \quad \text{for } i = 1, \dots, n \end{aligned}$$

- For increasing C, will the margin increase or decrease, and why?
- For increasing C, will $\|\mathbf{v}_H\|$ increase or decrease, and why?
- For increasing C, will training error increase or decrease, and why?
- For increasing C, should test error increase or decrease, and why?

Solution:

(a) True. A feature can be used to split a node multiple times.

(b) False. The stopping rule is that the number of observations in each node is smaller than certain threshold.

(c) In the dual problem, we only need to compute $K(\mathbf{x}, \mathbf{x}')$ instead of calculating the inner product in the feature space. This is important because for some kernels, such as the radial kernel, the feature space is implicit and infinite-dimensional.

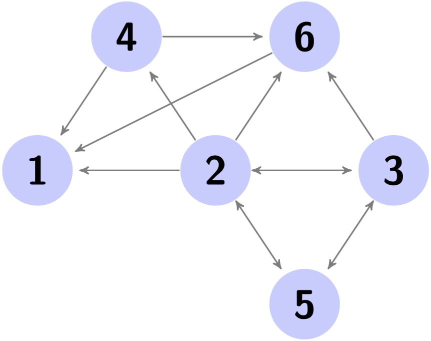
(d) Because in the Newton's method, we need to compute the Hessian matrix within each iteration. The computational cost of this step is expensive, especially when the dimension is high.

(e) Regularization cannot reduce training MSE. Regularization moves the result away from the least squares solution, which by definition has the minimum training MSE. But may reduce test MSE.

(f)
- For increasing C, the margin will decrease. This is because we penalize more for the points which cross the margin. The margin has to decrease to avoid making errors.
- For increasing C, $\|\mathbf{v}_H\|$ will increase, because $1/\|\mathbf{v}_H\|$ corresponds to the margin.
- For increasing C, the training error will decrease. This is because errors cost more.
- Test error may increase, as we risk overfitting by being oversensitive to errors.

Problem 2: Page Rank (10 points)

A directed graph G has the set of nodes $\{1, 2, 3, 4, 5, 6\}$ with the edges arranged as follows.



(a) The adjacency matrix of the graph is defined to be a matrix A with

$$A_{ji} = \begin{cases} \frac{1}{\text{degree of node } i} & \text{if } i \text{ links to } j, \\ 0 & \text{otherwise.} \end{cases}$$

Write down the adjacency matrix of this graph.

(b) In PageRank algorithm, the ranking of the webpages is given by the invariant distribution of a Markov Chain with transition matrix

$$\mathbf{p} = (1-\alpha)A + \frac{\alpha}{d}\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

If we set $\alpha = 0.2$ here, then *explicitly* write the PageRank equations. Denote the PageRank of node a by $r(a)$. (Please write the equations explicitly. DO NOT use matrix representation since we only have six nodes.)

Solution:

(a)

$$A = \begin{bmatrix} 1/6 & 1/5 & 0 & 1/2 & 0 & 1 \\ 1/6 & 0 & 1/3 & 0 & 1/2 & 0 \\ 1/6 & 1/5 & 0 & 0 & 1/2 & 0 \\ 1/6 & 1/5 & 0 & 0 & 0 & 0 \\ 1/6 & 1/5 & 1/3 & 0 & 0 & 0 \\ 1/6 & 1/5 & 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

In order to make $\mathbf{p}$ a proper transition matrix, we set the first column to be $(1/6, \dots, 1/6)^T$.

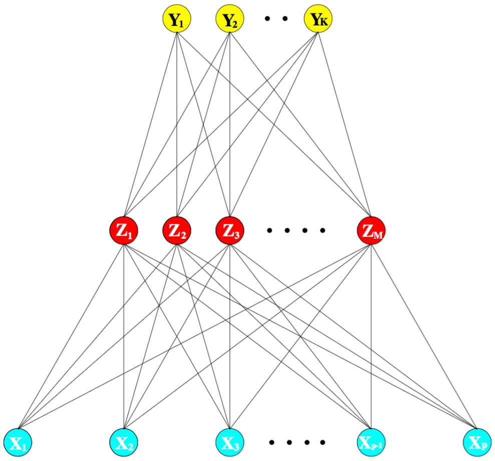
(b) The page ranks equations are:

$$\begin{aligned}
r(1) &= 0.8(\frac{1}{6}r(1) + \frac{1}{5}r(2) + \frac{1}{2}r(4) + r(6)) + \frac{0.2}{6} \\
r(2) &= 0.8(\frac{1}{6}r(1) + \frac{1}{3}r(3) + \frac{1}{2}r(5)) + \frac{0.2}{6} \\
r(3) &= 0.8(\frac{1}{6}r(1) + \frac{1}{5}r(2) + \frac{1}{2}r(5)) + \frac{0.2}{6} \\
r(4) &= 0.8(\frac{1}{6}r(1) + \frac{1}{5}r(2)) + \frac{0.2}{6} \\
r(5) &= 0.8(\frac{1}{6}r(1) + \frac{1}{5}r(2) + \frac{1}{3}r(3)) + \frac{0.2}{6} \\
r(6) &= 0.8(\frac{1}{6}r(1) + \frac{1}{5}r(2) + \frac{1}{3}r(3) + \frac{1}{2}r(4)) + \frac{0.2}{6}
\end{aligned}$$

Problem 3: Neural Networks(10 points)

Assume that we fit a single layer hidden neural network in a regression problem on $\mathbb{R}^p$. Recall our model is:

$$\begin{aligned} Z_m &= \sigma(\alpha_{0m} + \alpha_m^T X), \quad m = 1, \ldots, M. \\ f_k(X) &= \beta_{0k} + \beta_k^T Z, \quad k = 1, \ldots, K. \end{aligned}$$



The parameters of the model are $\theta = \{\alpha_{0m}, \alpha_m, \beta_{0k}, \beta_k\}$ (each α_m is a p-dimensional vector and β_k is an M-dimensional vector. We use gradient descent to minimize the squared error loss

$$R(\theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2.$$

A gradient update at the $(r+1)$st iteration has the form

$$\begin{aligned} \beta_{km}^{(r+1)} &= \beta_{km}^{(r)} - \frac{\partial R}{\partial \beta_{km}^{(r)}}, \\ \alpha_{ml}^{(r+1)} &= \alpha_{ml}^{(r)} - \frac{\partial R}{\partial \alpha_{ml}^{(r)}}. \end{aligned}$$

An issue of neural networks is that they have too many weights and will overfit the data. Therefore, regularization is necessary. Instead of minimizing the emprical risk $R(\theta)$, we add a penalty $J(\theta)$ to it with the form

$$J(\theta) = \sum_{k,m} \beta_{km}^2 + \sum_{m,l} \alpha_{ml}^2.$$

Now the object function of the optimization problem becomes

$$R(\theta) + \lambda J(\theta).$$

Write down the gradient update for this regularized problem. (You DO NOT NEED to calculate $\frac{\partial R}{\partial \beta_{km}}$ and $\frac{\partial R}{\partial \alpha_{ml}}$)

Solution:

$$\begin{aligned} \beta_{km}^{(r+1)} &= \beta_{km}^{(r)} - \left(\frac{\partial R}{\partial \beta_{km}^{(r)}} + 2\lambda \beta_{km}^{(r)} \right), \\ \alpha_{ml}^{(r+1)} &= \alpha_{ml}^{(r)} - \left(\frac{\partial R}{\partial \alpha_{ml}^{(r)}} + 2\lambda \alpha_{ml}^{(r)} \right). \end{aligned}$$

Problem 4: Decision Trees(10 points)

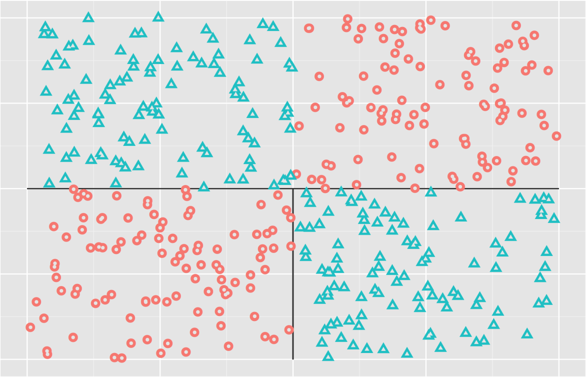
The standard method for fitting a decision tree involves:

- Growing the tree split by split. We maximize the reduction of the training error at each step until there are at most 5 samples per region.
- Pruning the tree to obtain a sequence of trees of decreasing size.
- Selecting the optimal size by cross-validation.

Consider the following alternative approach. Grow the tree split by split until the reduction in the training error produced by the next split is smaller than some threshold. This approach may lead to bad results because it is possible to make a split which does not decrease the error by much, and then make a second split which reduces the error significantly.

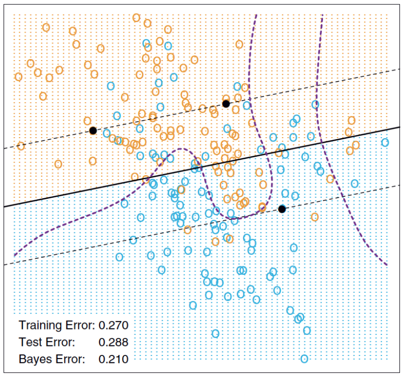
Draw an example dataset where this happens with two predictors X_1 and X_2, and a binary categorical response.

Solution: The figure shows the partition produced by a tree with 2 splits. The first split (horizontal line) barely reduces the classification error. However, the second split (vertical line) decreases the error significantly.

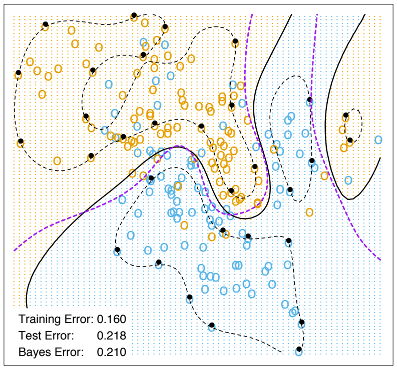


Problem 5: Decision boundaries (10 points)

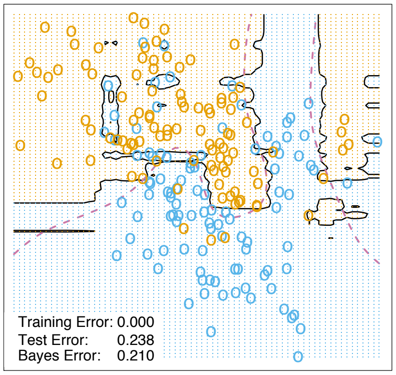
The following pictures, which we have all seen in class, show the output of several different classifiers. Recall that the thick line is the decision boundary determined by the classifier; you can ignore the dashed lines.



(a)



(b)



(c)

For each of the three pictures:

- Name at least one classifier which could have produced this solution. Explain why.
- Name at least one classifier which could not have produced the solution. Explain why not.

Solution:

	(a)	(b)	(c)
could be generated by	logistic regression or LDA	Bayes classifier	K-nearest-neighbor classifier
reason	linear boundary, class overlap	smooth, non-linear boundary	non-linear boundary
could not be generated by	K-nearest-neighbor classifier	any linear classifier	any linear classifier
reason	Trees or RF (smooth slope)	Trees or RF (smooth)	

Problem 6: The Kernel Trick(10 points)

In this problem, we will apply the kernel trick to ridge regression and derive kernel ridge regression.

Consider a linear regression problem with n data points each in p dimensions, corresponding to the data matrix $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$ and response vector $\mathbf{y} \in \mathbb{R}^n$. Just as we extended a linear SVM to a nonlinear SVM with the kernel trick, we can do the same to create nonlinear kernel regression. Specifically, we want to find the solution to:

$$\arg\min_{\beta} \sum_{i=1}^{n} (y_i - \langle \beta, \phi(x_i)\rangle_{\mathcal{F}})^2 + \lambda \|\beta\|_{\mathcal{F}}^2,$$

where $\phi(\cdot)$ is the feature map and $\langle \cdot, \cdot \rangle_{\mathcal{F}} = k(\cdot, \cdot)$ in the usual "kernel trick" way.

A very important property of the solution is that β can be written in the form $\beta = \sum_{i=1}^{n} \alpha_i \phi(x_i) = \Phi^T \alpha$ with $\Phi = [\phi(x_1) \cdots \phi(x_n)]^T$ (this general fact in often called the representation theorem).

Then, using this property, write how to predict $\hat{f}(x^*)$ and a new point x^*. Your prediction $f(x^*)$ should be in terms of the kernel matrix $\mathbf{K} = \{\langle \phi(x_i), \phi(x_j)\rangle_{\mathcal{F}}\}_{i,j=1,\ldots,n}$, and elements in $\mathcal{F}$ should not appeaer explicitly in this prediction (since that could be infinite dimensional).

Solution: Assume that $\Phi = [\phi(x_1), \ldots, \phi(x_n)]^T$ is the feature matrix, where ϕ is the feature map.

$$\begin{aligned}
\sum_{i=1}^{n} (y_i - \langle \beta, \phi(x_i)\rangle_{\mathcal{F}})^2 + \lambda \|\beta\|_{\mathcal{F}}^2 &= \|y - \langle \beta, \Phi^T \rangle\|_2^2 + \lambda \|\beta\|_{\mathcal{F}}^2 \\
&= \|y - \langle \Phi^T \alpha, \Phi^T \rangle\|_2^2 + \lambda \|\Phi^T \alpha\|_{\mathcal{F}}^2 \\
&= y^T y - 2\alpha^T \Phi \Phi^T y + \alpha^T \Phi \Phi^T \Phi \Phi^T \alpha + \alpha^T \Phi \Phi^T \alpha \\
&= y^T y - 2\alpha^T K y + \alpha^T (K^2 + \lambda K)\alpha
\end{aligned}$$

Taking derivative, we have

$$2Ky = 2K(K + \lambda I)\alpha.$$

Therefore, the solution is

$$\alpha_{\text{KRR}} = (K + \lambda I)^{-1} y,$$

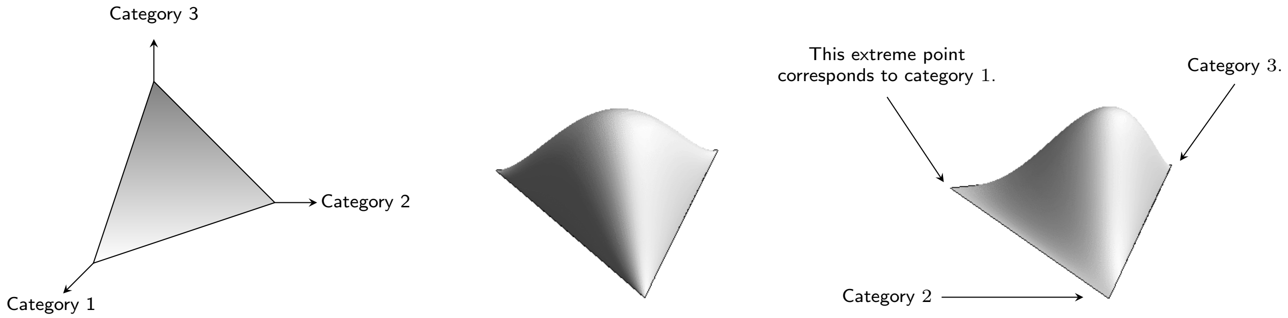
which has the same form as the ridge regression solution in the original space. Then we have $\beta_{\text{KRR}} = \Phi^T \alpha_{\text{KRR}}$, and thus:

$$\begin{aligned}
f(x^*) &= \langle \beta, \phi(x^*)\rangle_{\mathcal{F}} \\
&= \langle \Phi^T \alpha, \phi(x^*)\rangle_{\mathcal{F}} \\
&= \begin{bmatrix} k(x^*, x_1) \\ \vdots \\ k(x^*, x_n) \end{bmatrix} (K + \lambda I)^{-1} y.
\end{aligned}$$

Problem 7: Conjugate priors (5+5 points)

(a) Consider a Bayesian model with exponential family likelihood $p(x|\theta)$ with sufficient statistic S and natural conjugate prior $q(\theta|\lambda, y)$. Suppose the prior family has the property that $\max_\theta q(\theta|\lambda, y) = \mathbb{E}[\Theta] = y$. Can you write the maximum a posteriori (MAP) estimator for the model as a simple function of the prior parameters and the data?
Hint: You can work with a speicial case: both the likelihood model and prior are Gaussian.

(b) Consider a multinomial distribution on the set of categories $\mathbf{X} = \{1, 2, 3\}$. Recall that the parameter space of this distribution (the set of all vectors $\theta = (\theta_1, \theta_2, \theta_3)$ with non-negative entries and $\theta_1 + \theta_2 + \theta_3 = 1$) can be plotted as the area within a triangle, with each corner corresponding to one category (left figure):



parameter space

prior

posterior

The plot in the middle shows the density of a natural conjugate prior for the multinomial on the parameter space; the plot on the right is the resulting conjugate posterior given a sample. Does the observed sample in this case consist of one data point in category 3, or of one data point each in category 1 and 2? Please explain your answer.

Solution:

(a) Since the prior is conjugate, the posterior is

$$\Pi(\theta|x_1, \ldots, x_n) = q\Big(\theta\,\Big|\,\lambda + n, y + \sum_{i=1}^{n} S(x_i)\Big)\ .$$

Since the maximum of q coincides with the mean, the MAP estimate is simply

$$\hat{\theta}_{\text{MAP}} = \arg\max_\theta\, q\Big(\theta\,\Big|\,\lambda + n, y + \sum_{i=1}^{n} S(x_i)\Big) = y + \sum_{i=1}^{n} S(x_i)\ .$$

(b) In a conjugate posterior, the posterior mean shifts towards the average of (the sufficient statistic of) the data points. Since the mean clearly shifts towards the extreme point corresponding to $k = 3$, we have observed a single data point with value 3, rather than two with values 1 and 2.

Problem 8: HMMs (10 points)

HMMs have several important applications in genetics, where many data sets consist of sequences. A DNA sequence is a sequence of amino acids. There are four acids, represented by the symbols A, C, G, and T. The sequence consists of *coding regions* (which actually encode information) and *non-coding regions*, for example:

$$\underbrace{\text{C C T A A G}}_{\text{coding}}\,\underbrace{\text{T T A G A G G A T T}}_{\text{non-coding}}\,\underbrace{\text{G A G T}}_{\text{coding}}$$

One of the simplest applications of HMMs is *gene finding*: Label the coding and non-coding regions in a given input sequence. We make some simplifying assumptions:

- At each symbol in coding region, the probability that the coding regions ends (i.e. that the next symbol belongs to a non-coding region) is p_c.
- At each symbol in a non-coding region, the probability that the non-coding region ends at the symbol is p_n.
- In a coding region, the probabilities of observing A, C, G, or T are respectively a_{A}, a_{C}, a_{G} and a_{T}.A
- In non-coding region, the probabilities are b_{A}, b_{C}, b_{G}, b_{T}.

Suppose all of these probabilities have already been determined from training data. Define an HMM which reads an input sequence and label each symbol as "coding" or "non-coding". To do so, please specify:

- The state space.
- The set of observations.
- The emission distributions.
- The transition matrix.

Solution:

- State space: $\mathbf{Z} = \{\text{coding}, \text{non-coding}\}$.
- Observation space: $\mathbf{X} = \{A, C, G, T\}$.
- Emission distributions:

$$\begin{aligned} P(X|Z = \text{coding}) &= \text{Multinomial}(a_{\mathsf{A}}, a_{\mathsf{C}}, a_{\mathsf{G}}, a_{\mathsf{T}}) \\ P(X|Z = \text{non-coding}) &= \text{Multinomial}(b_{\mathsf{A}}, b_{\mathsf{C}}, b_{\mathsf{G}}, b_{\mathsf{T}}) \end{aligned}$$

- Transition matrix of the Markov chain:

$$\begin{pmatrix} p_{\text{coding}\to\text{coding}} & p_{\text{non-coding}\to\text{coding}} \\ p_{\text{coding}\to\text{non-coding}} & p_{\text{non-coding}\to\text{non-coding}} \end{pmatrix} = \begin{pmatrix} 1-p_c & p_n \\ p_c & 1-p_n \end{pmatrix}$$

Problem 9: MCMC (10 points)

Consider a distribution with a very simple "staircase" density p (the thick line in figure (i)). We divide the area under p into three "slices":

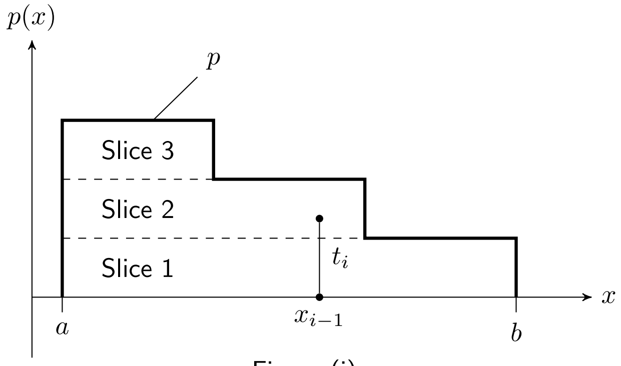


Figure (i)

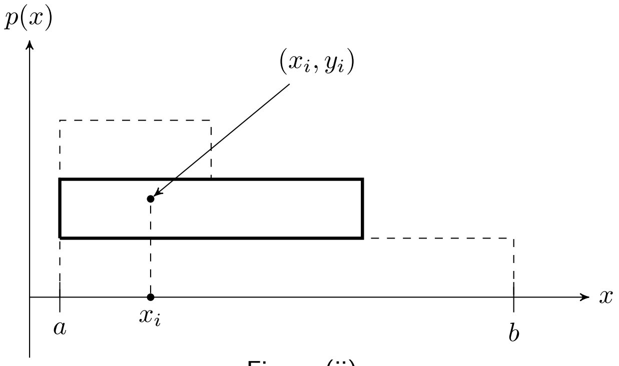


Figure (ii)

We define a Markov chain that generate samples $x_1, x_2, \ldots$ as follows. Start with any $x_1 \in [a, b]$. For each $i > 1$:

(a) Choose one of the slices which overlap the location of the previous sample x_{i-1} uniformly at random:

(1) $t_i \sim$ Uniform$[0, p(x_{i-1})]$ (2) Select the slice k which contains (x_{i-1}, t_i) .

(In figure (ii), this would be slice 2.)

(b) Regard slice k as a box and sample a point (x_i, y_i) uniformly from this box. Discard the vertical coordinate y_i and keep x_i as the ith sample.

Is this a valid sampler for p? More precisely: If we assume that the previous sample x_{i-1} is already distributed according to p (on the grounds that the Markov chain has converged), is x_i marginally distributed according to p? Please explain your answer.

Hint: This requires only the basic ideas we used to motivate rejection sampling. You do not need argue in terms of Markov chains, equilibria, etc.

Solution: If s_i is sampled according to p (ie if we choose the Markov chain version of the sampler), we obtain a valid sampler for p:

- Recall that, if we sample (x, y) *uniformly* from the area under the curve p and discard y, then x is marginally distributed according to p—that was how we motivated rejection sampling.
- If the sampler draws s_i from p and then spreads it vertically by a uniform coordinate t_i, we conversely obtain uniform samples (s_i, t_i) from the area under the curve.
- That means each slice gets selected with probability proportional to its volume. Since we then sample (x_i, y_i) uniformly from the chosen slice, (x_i, y_i) are again uniform on the area under p, and hence $x_i \sim p$.

Additional details (not required): To see that (s_i, t_i) are indeed uniform on the area under the curve, note that, if we sample $t_i \sim$ Uniform$[0, p(s_i)]$, then t_i has (conditional on s_i) constant density $\frac{1}{p(s_i)}$:

$$p(t_i|s_i) = \frac{1}{p(s_i)}$$

Since s_i is generated from p, the joint density is

$$p(s_i, t_i) \propto p(t_i|s_i)p(s_i) = \frac{p(s_i)}{p(s_i)} = 1,$$

so (s_i, t_i) is uniformly distributed over the area under the curve p.