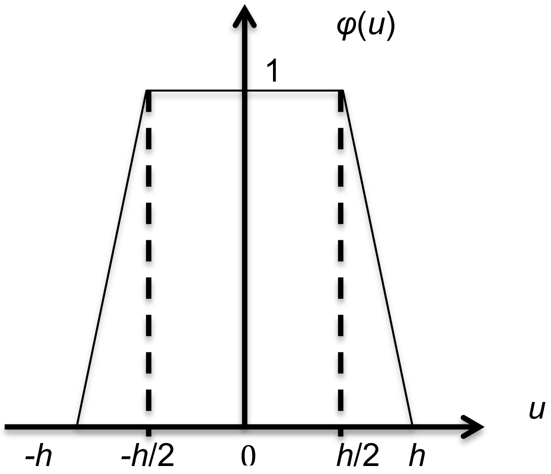
1. We perform Parzen Window density estimation, using trapezoidal window functions given (in unnormalized form) in the figure below: (15 pts)



Choose $h = 1$. Assume that we have the following data

$$\mathcal{D}_{\omega_1} = \{0, 2, 5\}$$
$$\mathcal{D}_{\omega_2} = \{4, 7\}$$

   (a) Sketch or plot the Parzen window estimates of the pdfs $p(x|\omega_1)$ and $p(x|\omega_2)$. Please label pertinent values on both axes.

   (b) Estimate the prior probabilities based on frequency of occurrence of the prototypes in each class.

   (c) Use the estimates you have developed in above to find the decision boundaries and regions for a Bayes minimum-error classifier based on Parzen windows. Only the part of feature space where at least one density is nonzero need to be classified.

2. We perform k-nearest neighbor density estimation, using $k = 2$. Assume that you are given the following training set for a 2-class problem with one feature: (20 pts)

$$\mathcal{D}_{\omega_1} = \{2, 5\}$$
$$\mathcal{D}_{\omega_2} = \{4, 7\}$$

   (a) Sketch or plot the k-nearest neighbors estimates of the pdfs $p(x|\omega_1)$. Please label pertinent values on both axes. Also give the density estimates algebraically, for each region in feature space.

   (b) Estimate the prior probabilities based on frequency of occurrence of the prototypes in each class.

   (c) Use the estimates you have developed in above to find the decision boundaries and regions for a Bayes minimum-error classifier based on k-nearest neighbors.

(d) Derive a classifier based on using KNN as a discriminative technique that estimates $p(\omega_i|x)$ directly using nearest neighbors, and compare it to the classifier you obtained in 2c. If there are ties, break them in favor of ω_2.

3. Consider the following training data set:

$$\begin{aligned}
\mathbf{x}_1 &= [1,0]^T, z_1 = -1\\
\mathbf{x}_2 &= [0,1]^T, z_2 = -1\\
\mathbf{x}_3 &= [0,-1], z_3 = -1\\
\mathbf{x}_4 &= [-1,0]^T, z_4 = 1\\
\mathbf{x}_5 &= [0,2]^T, z_5 = 1\\
\mathbf{x}_6 &= [0,-2]^T, z_6 = 1\\
\mathbf{x}_7 &= [-2,0]^T, z_7 = 1
\end{aligned}$$

Use following nonlinear transformation of the input vector $\mathbf{x} = [x_1, x_2]^T$ to the transformed vector $\mathbf{u} = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x})]^T$: $\varphi_1(\mathbf{x}) = x_2^2 - 2x_1 + 3$ and $\varphi_2(\mathbf{x}) = x_1^2 - 2x_2 - 3$. What is the equation of the optimal separating "hyperplane" in the $\mathbf{u}$ space? (15 pts)

4. Consider the following training data set : (25 pts)

$$\begin{aligned}
\mathbf{x}_1 &= [0,0]^T, z_1 = -1\\
\mathbf{x}_2 &= [1,0]^T, z_2 = 1\\
\mathbf{x}_3 &= [0,-1], z_3 = 1\\
\mathbf{x}_4 &= [-1,0]^T, z_4 = 1
\end{aligned}$$

Note that in the following, you need to use equations that describe $\mathbf{w}$ and give rise to the dual optimization problem.

(a) Write down the dual optimization problem for training a Support Vector Machine with this data set using the polynomial kernel function

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^2$$

(b) Solve the optimization problem and find the optimal λ_i's using results about quadratic forms and check the results with Wolfram Alpha or any software package.

(c) Show that the equation of the decision boundary in a kernel SVM $\mathbf{w}^T\mathbf{u} + w_0 = 0$ can be represented as $g(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i z_i \kappa(\mathbf{x}_i, \mathbf{x}) + w_0$.

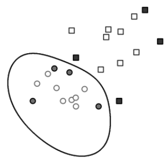
(d) We learned that for vectors that do not violate the margin[1] (i.e. $z_j(\mathbf{w}^T\mathbf{u}_j + w_0) - 1 > 0$), the Lagrange multiplier is zero, i.e. $\lambda_j = 0$. On the other hand, for

[1] For simplicity, consider Kernel SVM with hard margins, i.e. no slack variables.

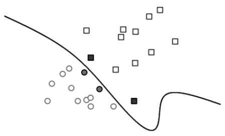
vectors on the margin ($z_j(\mathbf{w}^T\mathbf{u}_j + w_0) - 1 = 0$), $\lambda_j \neq 0$. Show that, consequently, one can find a vector $\mathbf{x}_j$ for which $\lambda_j \neq 0$ and calculate w_0 as $w_0 = 1/z_j - \sum_{i=1}^{N} \lambda_i z_i \kappa(\mathbf{x}_i, \mathbf{x}_j)$.

(e) Sketch the decision boundary for this data set based on parts (4c) and (4d).

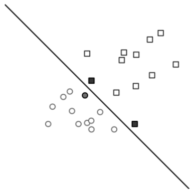
5. In the following figure, there are different SVMs with different decision boundaries. The training data is labeled as $z_i \in \{-1, 1\}$, represented as circles and squares respectively. Support vectors are drawn in solid circles. Determine which of the scenarios described below matches one of the 6 plots (note that one of the plots does not match any scenario). Each scenario should be matched to a unique plot. Explain your reason for matching each figure to each scenario. (10 pts)



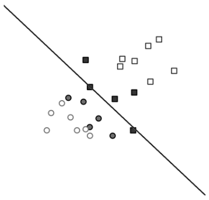
1



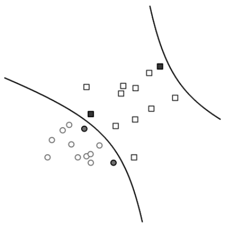
2



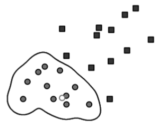
3



4



5



6

(a) A soft-margin linear SVM with $C = 0.02$

(b) A soft-margin linear SVM with $C = 20$

(c) A hard-margin kernel SVM with $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j + (\mathbf{x}_i^T\mathbf{x}_j)^2$

(d) A hard-margin kernel SVM with $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-5\|\mathbf{x}_i - \mathbf{x}_j\|^2)$

(e) A hard-margin kernel SVM with $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{5}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$

6. **Programming Part: Multi-class and Multi-Label Classification Using Support Vector Machines**

(a) Download the Anuran Calls (MFCCs) Data Set from: `https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs)`. Choose 70% of the data randomly as the training set.

(b) Each instance has three labels: Families, Genus, and Species. Each of the labels has multiple classes. We wish to solve a multi-class and multi-label problem. One of the most important approaches to multi-class classification is to train a classifier for each label. We first try this approach:

i. Research exact match and hamming score/ loss methods for evaluating multi-label classification and use them in evaluating the classifiers in this problem.

ii. Train a SVM for each of the labels, using Gaussian kernels and one versus all classifiers. Determine the weight of the SVM penalty and the width of the Gaussian Kernel using 10 fold cross validation.[2] You are welcome to try to solve the problem with both normalized[3] and raw attributes and report the results. (15 pts)

iii. Repeat 6(b)ii with $\mathscr{L}_1$-penalized SVMs.[4] Remember to normalize the attributes. (10 pts)

iv. Repeat 6(b)iii by using SMOTE or any other method you know to remedy class imbalance. Report your conclusions about the classifiers you trained. (10 pts)

v. Extra Practice: Study the Classifier Chain method and apply it to the above problem.

vi. Extra Practice: Research how confusion matrices, precision, recall, ROC, and AUC are defined for multi-label classification and compute them for the classifiers you trained in above.

[2] How to choose parameter ranges for SVMs? One can use wide ranges for the parameters and a fine grid (e.g. 1000 points) for cross validation; however,this method may be computationally expensive. An alternative way is to train the SVM with very large and very small parameters on the whole training data and find very large and very small parameters for which the training accuracy is not below a threshold (e.g., 70%). Then one can select a fixed number of parameters (e.g., 20) between those points for cross validation. For the penalty parameter, usually one has to consider increments in $\log(\lambda)$. For example, if one found that the accuracy of a support vector machine will not be below 70% for $\lambda = 10^{-3}$ and $\lambda = 10^6$, one has to choose $\log(\lambda) \in \{-3, -2, \dots, 4, 5, 6\}$. For the Gaussian Kernel parameter, one usually chooses linear increments,e.g. $\sigma \in \{.1, .2, \dots, 2\}$. When both σ and λ are to be chosen using cross-validation, combinations of very small and very large λ's and σ's that keep the accuracy above a threshold (e.g.70%) can be used to determine the ranges for σ and λ. Please note that these are very rough rules of thumb, not general procedures.

[3] It seems that this dataset is already normalized!

[4] The convention is to use $\mathscr{L}_1$ penalty with linear kernel.