Comp 6321 - Machine Learning - Assignment 3

Federico O'Reilly Regueiro

November 10^{th}, 2016

Question 1: Midterm preparation question

Propose an adequate learning algorithm for each instance.

1.a 1000 samples, 6-dimensional continuous space, classify ∼100 examples.

curse of dimensionality - fixed error rate e n $(c/e)^{(}(d+4)/4)$

non-parametric? still 92 % accuracy is possible...

1.b Clasifier for children in special-ed, justified to the board before it's implemented.

One of the easiest classification algorithms to explain in layman's terms is decision trees; since the method should be justified to the board, this would probably be an adequate choice. Furthermore, given that the stakes for such a classification are very high, an ensemble approach such as random forests / bagging could increase the classifier's performance by diminishing avoiding overfitting.

1.c Binary classification, train with very large data-set of products / customer preferences. Input - 1 million bits - other clients' preferences. Frequent updates.

A recommender system of this nature could use naive bayes in the similar way to document classification. In this case, the features of each product are the clients who have shown interest in the product. And the training relies on the trends in clients' preferences accross all products. NB could work well given the size of the dataset and the need for frequent updates. However, a drawback to this approach is that the recommender assumes feature independence while relying on the underlying relation between customer preferences, which in turn implies feature-dependency. A similar problem arises in document classification, where the presence of words in a document cannot really be considered independent, yet naive bayes performs well.

1.d 40 attributes, discrete and continuous, some have noise; only about 50 labeled observations.

With few examples and a fair amount of features, the curse of dimensionality haunts this classifiation. The presence of noise and the need for some sort of reduction in dimensionality might be well served by logistic regression with L1 regularization extensible to a kernel approach if the classes do not seem linearly separable. K-fold cross-validation, with a relatively small k given the dataset size, would be necessary in order to find the appropriate rate for regularization.

Question 2: Properties of entropy

2.a Compute the following for (X, Y): $p(0,0) = 1/3, p(0,1) = 1/3, p(1,0) = 0, p(1,1) = 1/3$.

i $H[x] = -\sum_x p(x)log_2(p(x)) = -\frac{1}{3}log_2\left(\frac{1}{3}\right) - \frac{2}{3}log_2\left(\frac{2}{3}\right) = .9182$

ii $H[y] = -\sum_y p(y)log_2(p(y)) = -\frac{1}{3}log_2\left(\frac{1}{3}\right) - \frac{2}{3}log_2\left(\frac{2}{3}\right) = .9182$

iii $H[y|x] = -\sum_x p(x)H[Y|X=x] = -\frac{2}{3}\left(\frac{1}{2}log_2\left(\frac{1}{2}\right) + \frac{1}{2}log_2\left(\frac{1}{2}\right)\right) = \frac{2}{3}$

iv $H[x|y] = -\sum_y p(x)H[X|Y=y] = -\frac{2}{3}\left(\frac{1}{2}log_2\left(\frac{1}{2}\right) + \frac{1}{2}log_2\left(\frac{1}{2}\right)\right) = \frac{2}{3}$

v $H[x,y] = -\sum_x \sum_y p(x,y)log_2(p(x,y)) = 3\left(-\frac{1}{3}log_2\left(\frac{1}{3}\right)\right) = 1.5849$

vi $I[x,y] = \sum_x \sum_y p(x,y)log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) = H[x] - H[x|y] = 0.2516$

2.b Prove maximum entropy in a discrete distribution happens in U

We wish to find:

$$\arg\max_{p_n} \sum_{n=1}^{N} p_n log(p_n)$$

With constraints:

$$1 - \sum_{n=1}^{N} p_n = 0$$

$$p_i \geq 0, \forall i \in \{1, 2, \ldots, N\}$$

We use Lagrange for maximization with constraints with a lagrangian multiplier only for the first constraint[1]:

$$\mathcal{L}(p_1, p_2, \ldots, p_n, \lambda) = \sum_{n=1}^{N} p_n log(p_n) - \lambda(1 - \sum_{n=1}^{N} p_n)$$

[1] The second constraint is satisfied by the solution.

And by setting the gradient of the Lagrangian function to 0

$$\nabla_{p_1,p_2,\ldots p_N,\lambda}\mathcal{L}(p_1,p_2,\ldots,p_n,\lambda = 0$$

We are thus left with a system:

$$\begin{aligned}
\frac{\partial_{\mathcal{L}}}{\partial_{p_1}}\sum_{n=1}^{N}p_n log(p_n){-}\lambda(1-\sum_{n=1}^{N}p_n) = 0\\
\frac{\partial_{\mathcal{L}}}{\partial_{p_2}}\sum_{n=1}^{N}p_n log(p_n){-}\lambda(1-\sum_{n=1}^{N}p_n) = 0\\
\vdots\\
\frac{\partial_{\mathcal{L}}}{\partial_{p_N}}\sum_{n=1}^{N}p_n log(p_n){-}\lambda(1-\sum_{n=1}^{N}p_n) = 0\\
\frac{\partial_{\mathcal{L}}}{\partial_{\lambda}}\lambda(1{-}\sum_{n=1}^{N}p_n) = 0
\end{aligned}$$

Which in turn yields:

$$\begin{aligned}
log(p_1) + 1{-}\lambda p_1 = 0\\
log(p_2) + 1{-}\lambda p_2 = 0\\
\vdots\\
log(p_N) + 1{-}\lambda p_N = 0\\
1 - \sum_{n=1}^{N}p_n = 0 \qquad (1)
\end{aligned}$$

From which:

$$\lambda = \frac{\log(p_1)+1}{p_1} = \frac{\log(p_2)+1}{p_2} = \ldots \frac{\log(p_N)+1}{p_N} \qquad (2)$$

it is clear from equations 1 and 2 that $p_1 = p_2 = \ldots p_N = \frac{1}{N}$, which is precisely a discrete uniform distribution.

2.c Show that T_1 wins

The notes show two possible tests for a decision tree. T1, where the left child has $[20+, 10-]$ posible outcomes in its sub-trees and the right node has $[10+, 0-]$. T2, on the other hand, yields: $left = [15+, 7-]; right = [15+, 3-]$.

The best choice should yield the maximum mutual information or information gain $I[p, T_n]$,$n \in \{1, 2\}$. So for T_1:

$$
\begin{aligned}
H[p] =& -\frac{1}{4}log_2\left(\frac{1}{4}\right) - \frac{3}{4}log_2\left(\frac{3}{4}\right) = 0.8112 \\
H[p|T_1 = t] =& -\frac{2}{3}log_2\left(\frac{2}{3}\right) - \frac{1}{3}log_2\left(\frac{1}{3}\right) = 0.9182 \\
H[p|T_1 = f] =& 0 \\
H[p|T_1] =& p(T_1 = t)H[p|T_1 = t] + p(T_1 = f)H[p|T_1 = f] \\
=& 0.6887 \\
I[p, T_1] =& H[p] - H[p|T_1] = 0.1225
\end{aligned}
$$

Whereas for T_2 we have:

$$
\begin{aligned}
H[p|T_2 = t] =& -\frac{15}{22}log_2\left(\frac{15}{22}\right) - \frac{7}{22}log_2\left(\frac{7}{22}\right) = 0.9024 \\
H[p|T_2 = f] =& -\frac{15}{18}log_2\left(\frac{15}{18}\right) - \frac{3}{18}log_2\left(\frac{3}{18}\right) = 0.65002 \\
H[p|T_2] =& p(T_2 = t)H[p|T_2 = t] + p(T_2 = f)H[p|T_2 = f] \\
=& \frac{22}{40}0.9024 + \frac{18}{40}0.65002 = 0.7888 \\
I[p, T_2] =& H[p] - H[p|T_2] = 0.02245
\end{aligned}
$$

From which we can see that we gain much more information from knowing the result of T_1 than by knowing the result of T_2.

Question 3: Kernels

Suppose $k_1(\boldsymbol{x}, \boldsymbol{z})$ and $k_2(\boldsymbol{x}, \boldsymbol{z})$ are valid kernels over $\mathbb{R}^n \times \mathbb{R}^n$. Prove or disprove that the following are valid kernels.

Use Mercer's theorem regarding the Gram matrix[2] or the fact that a kernel can be expressed as $k(x, z) = \phi(\boldsymbol{x})^T\phi(\boldsymbol{z})$.

preliminaries

From Mercer, we know for each $k_1(\boldsymbol{x}, \boldsymbol{z})$ and $k_2(\boldsymbol{x}, \boldsymbol{z})$ we have corresponding kernel matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ which are symmetric and positive semi-definite.

For both $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$:

Symmetry:

$$\boldsymbol{M}_i = \boldsymbol{M}_i^T \tag{3}$$

Positive semidefiniteness:

$$\boldsymbol{x}^T\boldsymbol{M}_i\boldsymbol{x} \geq 0 \tag{4}$$

$$|\boldsymbol{M}_i| \geq 0 \tag{5}$$

[2] Equivalently known as the kernel matrix.

3.a $k(\boldsymbol{x}, \boldsymbol{z}) = ak_1(\boldsymbol{x}, \boldsymbol{z}) + bk_2(\boldsymbol{x}, \boldsymbol{z}), a, b > 0;\ a, b \in \mathbb{R}$

Firstly, we establish that if $k(\boldsymbol{x}, \boldsymbol{z})$ is a valid kernel, then $ak(\boldsymbol{x}, \boldsymbol{z})$ is also a valid kernel $\forall a > 0;\ a \in \mathbb{R}$:

We know that for a square matrix $\boldsymbol{A}$ of size $n \times n$, $|a\boldsymbol{A}| = a^n\,|A|$. And, since $a \geq 0$, we know that $a^n \geq 0$. Thus equation 5 holds for both of our summands. Additionally, since the scalar multiplication of a symmetric matrix yields another symmetric matrix, both summands are are symmetric and therefore valid kernels.

Now, let us say:

$$ak_1(\boldsymbol{x}, \boldsymbol{z}) = k_1'(\boldsymbol{x}, \boldsymbol{z})$$

and

$$bk_2(\boldsymbol{x}, \boldsymbol{z}) = k_2'(\boldsymbol{x}, \boldsymbol{z})$$

are both valid kernels with kernel matrices $\boldsymbol{M}_1'$ and $\boldsymbol{M}_2'$. The addition of two symmetric matrices yields a symmetric matrix, so we need to check for positive semi-definiteness.

Since both $\boldsymbol{M}_1'$ and $\boldsymbol{M}_2'$ are symmetric we can write:

$$\boldsymbol{M}_1' = \boldsymbol{U}^T \boldsymbol{\Lambda}_{\boldsymbol{U}} \boldsymbol{U}$$
$$\boldsymbol{M}_2' = \boldsymbol{V}^T \boldsymbol{\Lambda}_{\boldsymbol{V}} \boldsymbol{V}$$

and using equation 4:

$$(\boldsymbol{x}^T \boldsymbol{U}^T \boldsymbol{\Lambda}_{\boldsymbol{U}} \boldsymbol{U} \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{V}^T \boldsymbol{\Lambda}_{\boldsymbol{V}} \boldsymbol{V} \boldsymbol{x}) \geq 0$$
$$\boldsymbol{x}^T (\boldsymbol{U}^T \boldsymbol{\Lambda}_{\boldsymbol{U}} \boldsymbol{U} + \boldsymbol{V}^T \boldsymbol{\Lambda}_{\boldsymbol{V}} \boldsymbol{V}) \boldsymbol{x} \geq 0$$
$$\boldsymbol{x}^T (\boldsymbol{M}_1' + \boldsymbol{M}_2') \boldsymbol{x} \geq 0$$

Which proves that $k(\boldsymbol{x}, \boldsymbol{z}) = ak_1(\boldsymbol{x}, \boldsymbol{z}) + bk_2(\boldsymbol{x}, \boldsymbol{z}), a, b > 0; a, b \in \mathbb{R}$ is a valid kernel.

3.b $k(\boldsymbol{x}, \boldsymbol{z}) = ak_1(\boldsymbol{x}, \boldsymbol{z}) - bk_2(\boldsymbol{x}, \boldsymbol{z}), a, b > 0; a, b \in \mathbb{R}$

Suppose:

$$a = 1, b = 1, M_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, M_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

Both$\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ symetric, positive semi-definite matrices. Yet $\boldsymbol{M'} = a\boldsymbol{M}_1 - b\boldsymbol{M}_2$ would yield:

$$M_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The eigenvalues of which are $\lambda_1 = -1, \lambda_2 = 1$, making $\boldsymbol{M'}$ a non positive semi-definite matrix and thus $k(\boldsymbol{x}, \boldsymbol{z})$ is not a valid kernel.

3.c $k(\boldsymbol{x}, \boldsymbol{z}) = k_1(\boldsymbol{x}, \boldsymbol{z})k_2(\boldsymbol{x}, \boldsymbol{z})$

The kernel matrix $\boldsymbol{M'}$ of the product of two matrices $k_1(\boldsymbol{x}, \boldsymbol{z}), k_2(\boldsymbol{x}, \boldsymbol{z})$ is equivalent to the element-wise multiplication of the respective two kernel matrices $\boldsymbol{M'} = \boldsymbol{M}_1 \odot \boldsymbol{M}_2$. This is also known as the Hadamard product or the Schur product. The Schur product theorem states that said product of two positive semi-definite matrices is also positive semi-definite. It is trivial to show that symmetry is preserved under such conditions. Thus $k(\boldsymbol{x}, \boldsymbol{z}) = k_1(\boldsymbol{x}, \boldsymbol{z})k_2(\boldsymbol{x}, \boldsymbol{z})$ is a valid kernel.

3.d $k(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x})f(\boldsymbol{z}), where\ f : \mathbb{R}^n \to \mathbb{R}$

Here we rely on the fact that a kernel can be expressed as $k(x, z) = \phi(\boldsymbol{x})^T\phi(\boldsymbol{z})$ where $\phi(\boldsymbol{x})$ maps $\boldsymbol{x}$ onto an n-dimensional space.

It is trivial to see that if $n = 1$ and $\phi = f$, $f(\boldsymbol{x})f(\boldsymbol{z})$ constitutes a valid kernel sinc it can be expressed as $k(x, z) = \phi(\boldsymbol{x})^T\phi(\boldsymbol{z})$.

3.e $k(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x})p(\boldsymbol{z}), where\ p\ pdf$.

The same rationale as question 3.d applies here.

Question 4: Nearest neighbour vs decision trees, do boundaries coincide?

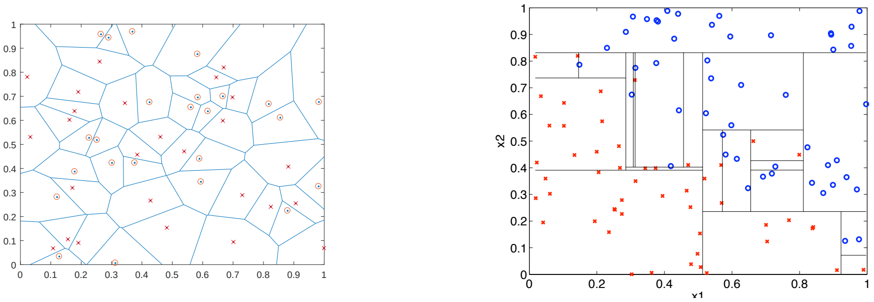
Boundaries do not necessarily coincide for these two classification strategies; moreover, in typical usage, they would tend to be non-coincidental but in some rare or contrived cases the boundaries might equate.

Decision tree boundaries are typically composed of hyper-planes that are orthogonal to the features f_d chosen for each decision; boundaries pass through the midpoint between points neighboring on a projection along the axis of f_d[3]. Thus each segment of a decision-tree boundary can have one out of n directions for an n-dimensional space.

Conversely, boundaries for nearest-neibours correspond to a Voronoi tessellation, where each boundary segment corresponds to a hyper-plane running orthogonal to the line between the boundary's nearest neighbors and passing through the midpoint of such a line (thus the ensemble of said hyperplanes has a wide gammut of directions witin the space).

For an example, see figures 1a and 1b.

[3] We note that any function of an arbitrary number of features may be used as a decision or boundary segment but this is a somewhat contrived usage of the decision tree algorithm.



(a) Nearest-neighbour

(b) decision tree

Figure 1: A Voronoi tessellation has boundary segments in many different directions, perpendicular to the lines between any two nearest-neighbors whereas decision-tree boundary segments are typically perpendicular to any one of a given set of features or feature combinations

Question 5: Bayes rate

For the following univariate case where $P(\omega_i) = \frac{1}{c}$ and

$$P(x|\omega_i) = \begin{cases} 1 & 0 \leq x \leq \frac{cr}{c-1} \\ 1 & i \leq x \leq i + 1 - \frac{cr}{c-1} \\ 0 & otherwise \end{cases}$$

5.a Show that $P^* = r$

The minimal multi-class classification error rate P^* is given by:

$$P^* = 1 - \int \arg\max_i P(\omega_i) P(x|\omega_i) dx$$

And given the class density and probability, we can see that for any region with overlapping densities, the choice of any i will maximize. Additionally, we see that the constraints imposed by existing densities demand that $0 \leq r \leq \frac{c-1}{c}$.

This in turn implies that densities overlap only in $[0, \frac{cr}{c-1}]$ Thus:

$$\begin{aligned}
P^* =&1 - \int P(\omega_1)P(x|\omega_1)dx \\
=&1 - \frac{1}{c}\int_0^{\frac{cr}{c-1}} dx - \sum_{i=1}^{c}\frac{1}{c}\int_i^{i+1-\frac{cr}{c-1}} dx \\
=&1 - \frac{1}{c}\frac{cr}{c-1} - 1 - \frac{cr}{c-1} \\
=&\frac{cr-r}{c-1} \\
=&r
\end{aligned}$$

5.b Show the nearest-neighbor rate $P = P^*$

$$\begin{aligned}
LNN =&\int \left[1 - \sum_{i=1}^{c} P^2(\omega_i|x)\right] p(x)dx \\
=&\int \left[1 - \sum_{i=1}^{c} \left(\frac{P(x|\omega_i)P(\omega_i)}{p(x)}\right)^2\right] p(x)dx \\
=&\int p(x) - \sum_{i=1}^{c} \frac{P(x|\omega_i)^2 P(\omega_i)^2}{p(x)} dx \\
=&\int p(x) - \sum_{i=1}^{c} \frac{P(x|\omega_i)P(\omega_i)(P(x|\omega_i)P(\omega_i))}{p(x)} dx \\
=&\int p(x) - \sum_{i=1}^{c} \frac{P(x|\omega_i)P(\omega_i)p(x)}{p(x)} dx \\
=&\int p(x)dx - \frac{1}{c}\int_0^{\frac{cr}{c-1}} dx - \sum_{i=1}^{c}\frac{1}{c}\int_i^{i+1-\frac{cr}{c-1}} dx \\
=&1 - \frac{1}{c}\frac{cr}{c-1} - 1 - \frac{cr}{c-1} \\
=&\frac{cr-r}{c-1} \\
=&r
\end{aligned}$$

Question 6: Implementation