• take home exam
• please write neatly for full credit
• notes and books are allowed
• electronic devices are not necessary
• Work alone – do not consult with ANYONE except the instructor
• Proofs are required everywhere but Problems 1 and 3.1.
• DUE: Thursday February 21 at the beginning of lecture OR submitted on Catalyst by 11:20 (Hard deadline)

• Do Well!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>Decision regions</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Stochastic gradient</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Boosting</td>
</tr>
<tr>
<td>4</td>
<td>2.5</td>
<td>Convexity</td>
</tr>
<tr>
<td>5</td>
<td>3.5</td>
<td>Entropy and information</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>17</strong></td>
<td><strong>points</strong></td>
</tr>
</tbody>
</table>
Problem 1 – Decision regions

1.1 The figure on the left depicts a decision boundary in $\mathbb{R}^2$. Mark the boxes beside the classifier families below, if this decision boundary and the corresponding decision regions could (a) be obtained, or (b) be approximated arbitrarily closely with the respective classifier, given appropriate training sets.

\[
\begin{array}{l}
\hline
\text{Exact} & \text{Appr} \\
\hline
\text{1-nearest neighbor classifier} & \square \quad \square \\
\text{linear classifier} & \square \quad \square \\
\text{quadratic classifier} & \square \quad \square \\
\text{decision tree} & \square \quad \square \\
\text{2 layer neural network} & \square \quad \square \\
\hline
\end{array}
\]

Assume that the data set can be as large or as small as you need it be. No proofs or examples are required, but you are encouraged to use the working space to create examples.
Problem 2 – Stochastic gradient for additive model

Consider fitting the additive model

\[ f(x) = \sum_{k=1}^{m} \beta_k b_k(x) \]  

where \( b_{1:m} \) are fixed functions of \( x \) and the unknown parameters are \( \beta_{1:m} \).

The cost function is \( L(y, f) = e^{-yf} \).

The training set is \( \mathcal{D} = \{(x^1, y^1), \ldots, (x^N, y^N)\} \).

2.1 Write the expression of the training set cost \( \hat{L}(f) \) for \( f \) given by (1) as a function of \( \beta_{1:m}, b_{1:m} \) and the data.

2.2 Write the expression of the gradient of \( L(y, f(x)) \) w.r.t. \( \beta \), or alternatively the partial derivative \( \frac{\partial L(y, f(x))}{\partial \beta_j} \), for \( f \) given by (1).

2.3 Write the expression of the gradient of \( \hat{L} \) w.r.t. \( \beta \), or alternatively the partial derivative \( \frac{\partial \hat{L}}{\partial \beta_j} \), for \( f \) given by (1).

2.4 Is the cost \( \hat{L} \) convex in \( \beta \)?

2.5 Assume that \( \hat{L} \) is \( \lambda \)-strongly convex as a function of \( \beta \) with \( \lambda \) known. (In fact, \( \hat{L} \) is not always strongly convex, but it can be made so by imposing additional conditions on \( b_{1:m}(x) \) and the data.)

Write in pseudocode a stochastic gradient descent algorithm to minimize \( \hat{L} \). Write the \( \beta^k \) update as a function of the data, the \( b_k \) functions, the parameters \( \beta \) and other algorithm dependent constants.
3. Problem 3 – Boosting

Denote by \( B \) a base classifier family, with \( b \in B \) defined on \( \mathbb{R}^n \) taking values in \( \{ \pm 1 \} \), and let \( f^m(x) = \sum_{k=1}^{m} \beta_k b_k(x) \) the boosted classifier obtained by the \textsc{AdaBoost} algorithm on a data set \( D = \{ (x^i, y^i), i = 1 : N \} \) sampled i.i.d. from a distribution \( P_{XY} \).

3.1 Give the mathematical definition and name the following expressions

<table>
<thead>
<tr>
<th>Example</th>
<th>Formula</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{L}_{01}(B) )</td>
<td>( \frac{1}{N} \sum_{i=1}^{N} L_{01}(y^i, b(x^i)) ) OR ( \frac{1}{N} \sum_{i=1}^{N} 1_{y^i b(x^i) &gt; 0} )</td>
<td>classification error of ( b ) on the training set</td>
</tr>
</tbody>
</table>

3.2 Assume that \( B \) is a family of weak classifiers, i.e. for any training set \( D \) of size \( N \) and for any weights on \( D \), there is a classifier in \( B \) that has training error no larger than \( \delta \), with \( \delta < \frac{1}{2} \).

Find the number of boosting steps \( k_0 \) sufficient to drive the training error of \( f^k \) to 0, i.e. the minimum \( k_0 \) to guarantee that for \( k > k_0 \) the training error of \( f^k \) is 0.
Problem 4 – Concave lower bound to convex function

Let $\text{dom } f = \text{dom } g \subseteq \mathbb{R}^n$, $f$ convex, $g$ concave, $f(x) \geq g(x)$ for all $x$ in the common domain. Thus, $g$ is a concave lower bound for the convex $f$, and $f$ is convex upper bound for the concave $g$. We want to show that in this situation, an affine upper/lower bound exists as well.

Prove that there is an affine function $h(x) = w^T x + b$ so that $g(x) \leq h(x) \leq f(x)$ for all $x$ in $\text{dom } f$.

Problem 5 – Optimal discretization of a continuous variable

Suppose we have a continuous random variable $Y \in [0, 1)$ with density $f(y) > 0$ on $[0, 1)$. We want to discretize $Y$ into $m$ levels. That is, we define another random variable $X \in \{1, \ldots, m\}$ by

$$X = i, \text{ iff } Y \in [a_{i-1}, a_i) \text{ for } i = 1, \ldots, m$$

(2)

with $0 = a_0 < a_1 < a_2 < \ldots < a_m = 1$.

5.1 What is the conditional entropy $H(X|Y)$?

5.2 Show how to set $a_1, \ldots, a_{m-1}$ so that $X$ preserves the maximum information about $Y$, when $f$ and $m$ are given.

Hints: the proof is short if you introduce the right notation; you may want to consider $F(y) = \int_0^y f(u)du$.

Notes: At a more careful analysis, you will note that the result is more general, i.e. it applies to, or can be generalized for, any r.v. $Y \in \mathbb{R}$. 