Kernel from Decision Stumps

When talking about non-uniform voting in aggregation, we mentioned that \( \alpha \) can be viewed as a weight vector learned from any linear algorithm coupled with the following transform:

\[
\Phi(x) = \left( g_1(x), g_2(x), \cdots, g_T(x) \right).
\]

When studying kernel methods, we mentioned that the kernel is simply a computational short-cut for the inner product \( \Phi(x)^T \Phi(x') \). In this problem, we mix the two topics together using the decision stumps as our \( g_i(x) \).

(1) (10%) Assume that the input vectors contain only integers between (including) \( L \) and \( R \).

\[
g_{s,t,\theta}(x) = \text{sign}(s \cdot x_i - \theta).
\]

Two decision stumps \( g^{(1)} \) and \( g^{(2)} \) are defined as the same if \( g^{(1)}(x) = g^{(2)}(x) \) for every \( x \in \mathcal{X} \). Two decision stumps are different if they are not the same. Argue that there are only finitely-many different decision stumps for \( \mathcal{X} \) and list all of them for the case of \( d = 2 \) and \( (L, R) = (1, 6) \).

(2) (10%) Let \( \mathcal{G} = \{ \text{ all different decision stumps for } \mathcal{X} \} \). Since \( \mathcal{G} \) is finite, we can enumerate each hypothesis \( g \in \mathcal{G} \) by some index \( t \). Define

\[
\Phi_{ds}(x) = \left( g_1(x), g_2(x), \cdots, g_t(x), \cdots, g_{|\mathcal{G}|}(x) \right).
\]

Derive a simple equation that evaluates \( K_{ds}(x, x') = \Phi_{ds}(x)^T \Phi_{ds}(x') \) efficiently.

The result can be easily extended to the case when \( \mathcal{X} \) is an arbitrary box in \( \mathbb{R}^d \) as well.
Power of Adaptive Boosting

Please consider the adaptive boosting (AdaBoost) algorithm shown on P.16 of Lecture 24. In the following problems, we will prove that AdaBoost can reach $E_{in}(G) = 0$ if $T$ is large enough and every hypothesis $g_t$ satisfies $\epsilon_t \leq \epsilon < \frac{1}{2}$.

3 (10%) Let $U^{(t)} = \sum_{n=1}^{N} u_n^{(t)}$ at the beginning of the $t$-th iteration. According to the AdaBoost algorithm, for $t \geq 1$, prove that

$$U^{(t+1)} = \frac{1}{N} \sum_{n=1}^{N} \exp \left( \frac{y_n \sum_{\tau=1}^{t} \alpha_\tau g_\tau(x_n)}{\epsilon_t} \right).$$

4 (10%) By the result in the previous problem, prove that $E_{in}(G) \leq U^{(T+1)}$.

5 (10%) According to the AdaBoost algorithm above, for $t \geq 1$, prove that $U^{(t+1)} = U^{(t)} \cdot 2 \sqrt{\epsilon_t(1-\epsilon_t)}$.

6 (10%) Using $0 \leq \epsilon_t \leq \epsilon < \frac{1}{2}$, for $t \geq 1$, prove that $\sqrt{\epsilon_t(1-\epsilon_t)} \leq \sqrt{\epsilon(1-\epsilon)}$.

7 (10%) Using $\epsilon < \frac{1}{2}$, prove that $\sqrt{\epsilon(1-\epsilon)} \leq \frac{1}{2} \exp \left( -2 \left( \frac{1}{2} - \epsilon \right)^2 \right)$.

8 (10%) Using the results above, prove that $U^{(T+1)} \leq \exp \left( -2T \left( \frac{1}{2} - \epsilon \right)^2 \right)$.

9 (10%) Using the results above, argue that after $T = O(\log N)$ iterations, $E_{in}(G) = 0$.

Experiments with Adaptive Boosting

Implement the AdaBoost algorithm with decision stumps (i.e., use $A_{ds}$ as $A_b$). Run the algorithm on the following set for training:

http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/hw6/hw6_train.dat

and the following set for testing:

http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/hw6/hw6_test.dat

Use a total of $T = 300$ iterations. Let $G_t(x) = \text{sign} \left( \sum_{\tau=1}^{t} \alpha_\tau g_\tau(x) \right)$. Plot $E_{in}(G_t)$, $E_{out}(G_t)$, and $U^{(t)}$ (see the definition above) as functions of $t$ on the same figure. Briefly state your findings.

10 (10%, *) for $E_{in}(G_t)$

11 (10%, *) for $E_{out}(G_t)$

12 (10%, *) for $U^{(t)}$

13 (10%, *) for your explanation of the findings

14 (10%, *) Plot the training examples $(x_n, y_n)$ and mark the positive/negative examples clearly. Then, mark the vectors with top 10 numerical values of $u_n^{(T)}$. Briefly state your findings.

Experiments with Unpruned Decision Tree (*)

15 (10%, *) Implement the C&RT algorithm introduced in class for numerical, non-missing features only, without pruning. Run the algorithm on the following set for training:

http://www.csie.ntu.edu.tw/~htlin/course/ml13fall/hw6/hw6_train.dat

and the following set for testing:
Show your binary tree $G$ (by graph or by writing down the if-then-else).

(16) (10%, *) Report $E_{in}(G)$ and $E_{out}(G)$.

(17) (10%, *) Implement the Bagging algorithm and couple it with your C&RT to make a preliminary random forest. Produce 100 trees with Bagging. Plot a histogram of $E_{out}(G_t)$, where each $G_t$ is one of the trees. Compare your result with $E_{out}(G)$ above and report your findings.

(18) (10%, *) Let $\bar{G}_t$ be the uniform aggregation of the first $t$ trees in your random forest above. Plot $E_{in}(\bar{G}_t)$ and $E_{out}(\bar{G}_t)$ on the same figure. Report your findings.

Yes, A Lighter Homework :-)

(19) (10%) Which one of our lectures do you like most? Why?

(20) (10%) Which one of our lectures do you like least? Why?

Bonus: Kernel Shifting

For a valid kernel $K$, consider a new kernel

$$K(x, x') = K(x, x') + c$$

for some positive $c$. It is not difficult to see that $K$ is also a valid kernel.

(21) (Bonus 10%) Argue that for the dual of soft-margin SVM, using $\tilde{K}$ instead of $K$ yields exactly the same solution and exactly the same hypothesis.

(22) (Bonus 10%) Use the result above to simplify your kernel in (2).