Referee Report

Solving maximum-entropy sampling problems using factored masks

by S. Burer and J. Lee

This paper continues a long line of research by Lee and co-authors on the maximum-entropy sampling problem. The idea in this paper is to try to evaluate the "masked spectral bound" from [3] in a more sophisticated way, and to incorporate the resulting bound in a complete Branch and Bound algorithm. The results are somewhat mixed. On the one hand the bound computations show clear improvement over the primitive approach used in [3], but unfortunately the final Branch and Bound code cannot solve to optimality some problems that were previously solved using another bounding methodology. The paper is generally quite well written but would benefit from a minor revision.

- This paper uses $z(C, s)$ and $\xi_{C,s}(\cdot)$ while [3] used $z(C, s, n)$ and $\xi_C(\cdot)$. The paper also just uses ξ at several points in the text.
- At the end of the second paragraph of the Introduction there should be a brief description of the different bounds that have been used in past work. There should also be a "Notation" paragraph at the end of the Introduction. As it stands now some notation ($\succeq$, $\succ$, e, $\mathrm{Diag}(\cdot)$) is never defined.
- In Oppenheim's inequality $B \succeq 0$ should be $B \succ 0$. The last of the "interrelated points" on p.2 doesn't make much sense - basically the authors are saying that you can always hope to get lucky later on. On the other hand the authors could mention that by suitably initializing the search for a mask they can be assured to do at least as well as with other known bounds, like the eigenvalue or diagonal bounds. The last sentence in the following paragraph ("Considering that we ...") needs to be made into two sentences.
- The last sentence in Section 1 could be changed to something like "They did not, however, consider efficient evaluation of the bound or use of the bound in a complete Branch-&-Bound algorithm."
- On page 3, writing $f(V) = g(h(V))$ might be better than $f = g \circ h$ given the other use of $\circ$ to denote Hadamard product. The definition of Σ is not clear; I presume that $\Sigma_{ll} = \lambda_l$, $l = 1, \ldots, s$. The use of a componentwise operation as in d^3 is never defined. Presumably $\mathrm{diag}(VV')^{-1/2}$ is $[\mathrm{diag}(VV')]^{-1/2}$, again using the componentwise operation. Defining $D = \mathrm{Diag}(d)$, the expression for ∇f could be written more clearly as $D\nabla g(\bar{V}) - [D^3 \circ \nabla g(\bar{V})V']V$.
- Would Algorithm 1 be unchanged if W in step 2 was replaced with $h(W)$, as in Algorithm 2? If so this would be worth mentioning, since it would help to clarify the difference between Algorithms 1 and 2. (Then $d = e$ would also always hold, which is a nice simplification.) If it is *not* true then explicit mention of this should be made.
- On page 4, the paragraph beginning "Anstreicher and Lee [3]..." could be incorporated in parentheses after "... are practical." in the preceeding paragraph. Also the "alternative interpretation" at the bottom of the page is not at all clear.

- On page 5 it would be better to say that that the bounds based on the original and complementary problems "sometiems yield" or "can yield" very different values. A reference for the "greedy algorithm" ([11]?) would be nice. The communication from Lee to himself is certainly very private.

- The comparisons in Figures 4 and 5 are good but the text above Figure 4 needs to be expanded to better describe what is being shown. The conclusion seems to be that Algorithm AS performs pretty well compared to Algorithm 1 for the first few iterations, but eventually Algorithm 1 pulls ahead. One point that should definitely be considered is if Algorithm AS can ever "catch up," or if Algorithm 1 is sometimes obtaining a bound that is unattainable using Algorithm AS. Comparing Figure 5 to [3, Figure 2] it appears that for some s (for example $s = 21$) the bound obtained by Algorithm 1 after 100 iterations is better than that obtained by Algorithm AS after 1000 iterations.

- The caption for Figure 7 should be written out in full, as for Figure 6.

- The comparisons between Algorithms 1 and 2 are interesting. The superior performance of Algorithm 1 in the Branch and Bound code doesn't seem to be adequately explained by the possible effect of the warm-start strategy. I also really don't understand the warm-start strategy. Are the *first* $(s - f)^2$ entries of V used regardless of the indecies of the variables that are actually in the subproblem? This doesn't seem to make much sense.

- At the bottom of page 10 use $N \setminus (U \cup F)$ instead of $N \setminus F \setminus U$.

- The discussion of the Branch and Bound results in Figures 8 and 9 needs to be expanded in two ways. First there should be more discussion of what is being accomplished, since the problems are not being solved to optimality for all s. Comparing with the root bounds for these same problems from [3, Figures 2&3] shows that the gaps are being substantially reduced, although for the $n = 124$ problems the final gaps for most s are still extremely high. On the other hand the paper is remiss in failing to mention that the $n = 63$ problems were successfully solved by the NLP-based algorithm from [2]. For example the problem with $n = 63$, $s = 31$ was solved on a single HP9000/735 CPU running at 125 MHz in 62615 seconds. Using only the clock speed for normalization this would be about 3300 seconds running at 2.4 GHz, suggesting that the NLP-based algorithm would solve these problems for each s in under 3600 seconds. This observation must be added to the paper.

- On page 14 a reference should be given for the strong-Wolfe linesearch.

- In [15] *maxim* should be *maximum.* (I will refrain from conjecturing the source of this typo.)