

240 A Appendix

241 A.1 Rough work for claims

Claim 1.

$$\beta \geq 1 - \frac{(\lambda_c \lambda_\epsilon)^{\delta-r_0}}{\exp(\delta-r_0)^2} \left(\frac{\delta^\delta}{r_0^{r_0}} \right)^2 \quad (13)$$

Proof.

$$\beta = 1 - \prod_{r_0}^{\delta} P_{\text{error}}(r)^{dr} = 1 - \exp \left(\int_{r_0}^{\delta} \ln P_{\text{error}}(r) dr \right) \quad (14)$$

$$\geq 1 - \exp \left(\int_{r_0}^{\delta} \ln \lambda_c \lambda_\epsilon r^2 dr \right) \quad \text{Lower bound from Eq. 9} \quad (15)$$

$$= 1 - \exp \left(r \ln(\lambda_c \lambda_\epsilon r^2) - 2r \Big|_{r_0}^{\delta} \right) \quad (\text{see Appendix A.1: Claim 2}) \quad (16)$$

$$= 1 - \frac{(\lambda_c \lambda_\epsilon)^{\delta-r_0}}{\exp(\delta-r_0)^2} \left(\frac{\delta^\delta}{r_0^{r_0}} \right)^2 \quad (\text{see Appendix A.1: Claim 3}) \quad (17)$$

242

□

Claim 2.

$$\int \ln(ax^2) dx = x \ln(ax^2) - 2x + C$$

243 *Proof.* Use integration by parts. Let:

$$f(x) = ax^2 \quad u = \ln f(x) \quad dv = dx$$

$$f'(x) = 2ax \quad du = f'(x)/f(x) dx \quad v = x$$

244 Also note:

$$\frac{x f'(x)}{f(x)} = \frac{2ax^2}{ax^2} = 2$$

245 So the original problem can be integrated by parts:

$$\begin{aligned} \int \ln(ax^2) dx &= \int u dv \\ &= uv - \int v du \\ &= x \ln f(x) - \int \frac{x f'(x)}{f(x)} dx \\ &= x \ln(ax^2) - 2x + C \end{aligned}$$

246

□

Claim 3.

$$\exp \left(r \ln(\lambda_c \lambda_\epsilon r^2) - 2r \Big|_{r_0}^{\delta} \right) = \frac{(\lambda_c \lambda_\epsilon)^{\delta-r_0}}{\exp(\delta-r_0)^2} \left(\frac{\delta^\delta}{r_0^{r_0}} \right)^2$$

Proof.

$$\begin{aligned}
\exp \left(r \ln(\lambda_c \lambda_\epsilon r^2) - 2r \Big|_{r_0}^\delta \right) &= \exp \left(\delta \ln(\lambda_c \lambda_\epsilon \delta^2) - r_0 \ln(\lambda_c \lambda_\epsilon r_0^2) - 2\delta + 2r_0 \right) \\
&= (\lambda_c \lambda_\epsilon \delta^2)^\delta (\lambda_c \lambda_\epsilon r_0^2)^{-r_0} \exp(-2(\delta - r_0)) \\
&= \frac{(\lambda_c \lambda_\epsilon)^\delta}{(\lambda_c \lambda_\epsilon)^{r_0}} \frac{\delta^{2\delta}}{r_0^{2r_0}} \frac{1}{\exp(\delta - r_0)^2} \\
&= \frac{(\lambda_c \lambda_\epsilon)^{\delta - r_0}}{\exp(\delta - r_0)^2} \left(\frac{\delta^\delta}{r_0^{r_0}} \right)^2
\end{aligned}$$

247

□

Claim 4.

$$\frac{(\lambda_c \lambda_\epsilon)^{\delta(1-z)}}{\exp(1-z)^{2\delta}} \left(\frac{\delta^\delta}{\delta^{\delta z} z^{\delta z}} \right)^2 = \left(\left(\frac{\delta \sqrt{\lambda_c \lambda_\epsilon}}{e} \right)^{1-z} \frac{1}{z^z} \right)^{2\delta}$$

Proof.

$$\begin{aligned}
\frac{(\lambda_c \lambda_\epsilon)^{\delta(1-z)}}{\exp(1-z)^{2\delta}} \left(\frac{\delta^\delta}{\delta^{\delta z} z^{\delta z}} \right)^2 &= \left(\frac{\sqrt{\lambda_c \lambda_\epsilon}}{e} \right)^{2\delta(1-z)} \left(\frac{\delta^{\delta(1-z)}}{z^{\delta z}} \right)^2 \\
&= \left(\frac{\delta \sqrt{\lambda_c \lambda_\epsilon}}{e} \right)^{2\delta(1-z)} \frac{1}{z^{2\delta z}} \\
&= \left(\left(\frac{\delta \sqrt{\lambda_c \lambda_\epsilon}}{e} \right)^{1-z} \frac{1}{z^z} \right)^{2\delta}
\end{aligned}$$

248

□

249 A.2 Negative result: probabilistic core-sets

250 Instead of using a deterministic algorithm to compute core-sets, we score random batches on their
 251 likelihood of being a subset of the optimal core-set. The goal is for the concatenation of the best-
 252 scoring batches with the training set to result in a set of elements that are spread out on the feature
 253 space and occur in dense regions, minimizing δ by definition.

254 Suppose we are interested in classifying whether a real number is positive or not. Figure 8 shows
 255 the unlabelled data and the result of learning a Gaussian mixture model (GMM) over the features.
 256 We then use the trained GMM to estimate feature probabilities, which are required for computing
 257 modified batch-BALD scores (see Appendix A.3). Figure 9 shows that higher scores indicate features
 258 that are likely spread apart. For random batches sampled from this toy dataset, Figure 10 shows that
 259 the distribution of scores form a long right tail that contains the most likely core-set centers.

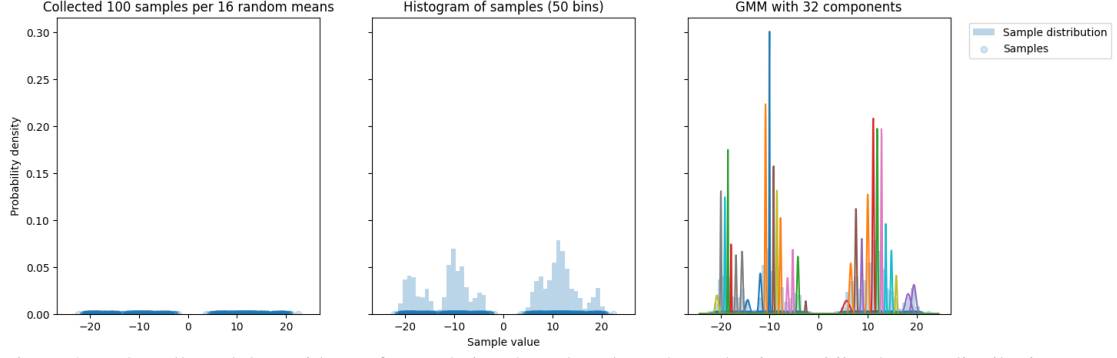


Figure 8: *Left*: collected data with one feature being the value along the real axis. *Middle*: the true distribution of the input features. *Right*: Gaussian mixture with 32 components fit to the collected data. We purposefully overfit the GMM because the distribution of features is not known a priori and we would like high resolution for computing joint information later.

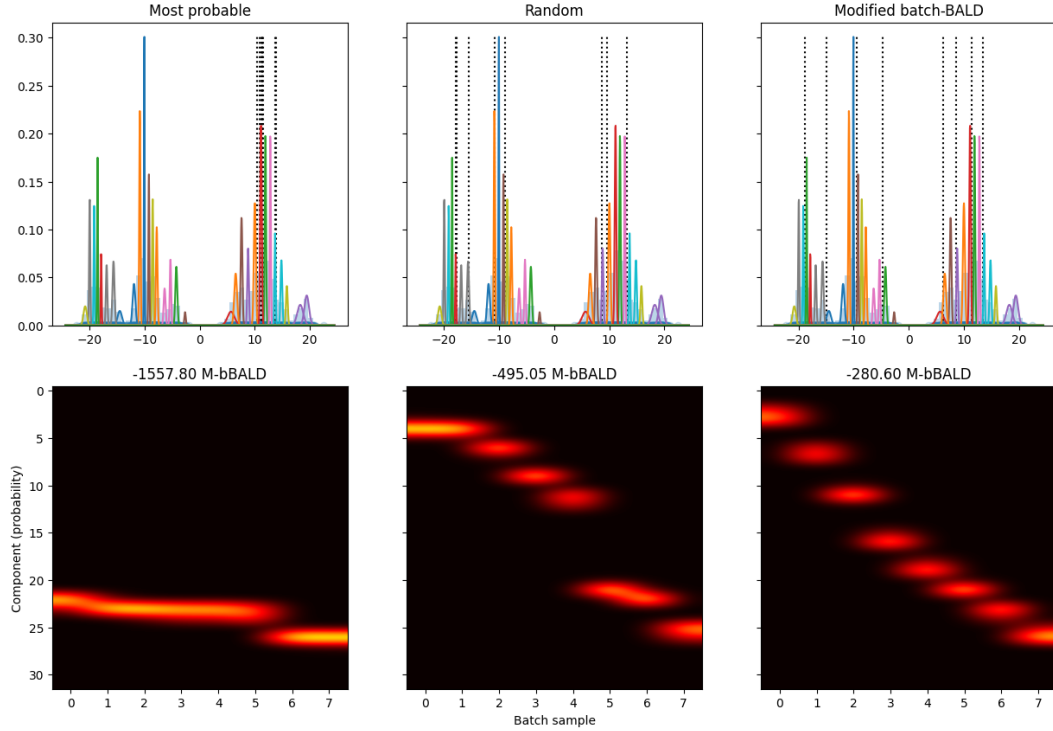


Figure 9: Given the trained Gaussian mixture model from Figure 8, we estimate the probability per component for each element. We use the probabilities to compute a modified batch-BALD joint information score (M-bBALD), which is positively correlated with entropy across the components. The batches with the highest scores occur at dense regions but are spread out across the feature distribution. We want to avoid redundant labelling of elements in the left column. *Top row*: each figure contains eight dotted lines that represent the locations of elements in three different batches. *Bottom row*: corresponding probabilities per GMM component for each element.

260 In the subsequent iterations, we first construct a new unlabelled data pool that contains features that
 261 have low probability to have appeared in the labelled pool, according to a fitted GMM on the labelled
 262 pool. Then, we fit a new GMM on this modified unlabelled pool and repeat the selection algorithm to
 263 search for the batch with the highest modified batch-BALD score when combined with the existing
 264 training data. We also experiment with interpolating the modified batch-BALD score with the least
 265 confidence acquisition metric.

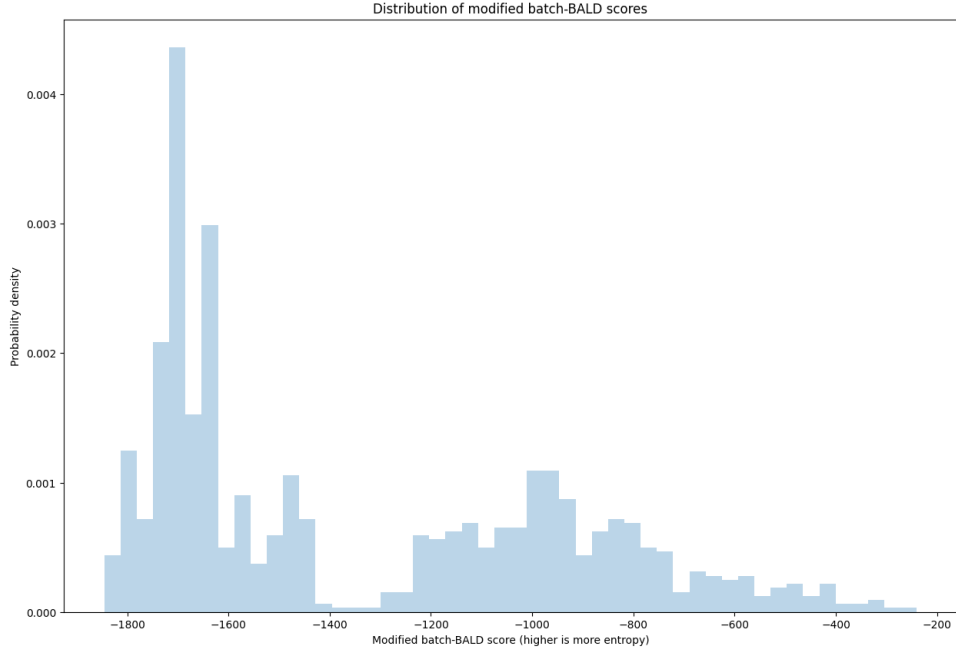


Figure 10: The distribution of modified batch-BALD scores for randomly sampled batches have a long right tail from which we mine for likely core-set elements. Range of scores depends on the number of components in the Gaussian mixture model.

266 We plot test accuracy versus number of labelled points for random acquisition (random), maximum
 267 entropy (max-entropy), least certainty (min-max-probs), probabilistic core-set (probabilistic-coreset)
 268 and an exploitative version of probabilistic core-set that interpolates with least certainty at a 9:1 ratio
 269 (probabilistic-coreset-exploitive-0.1). Figures 11 and 12 show that there is modest improvement
 270 of the core-set variants from the random baseline in both toy datasets, although its significance is
 271 unknown. The entropy and least certainty methods performed poorly.

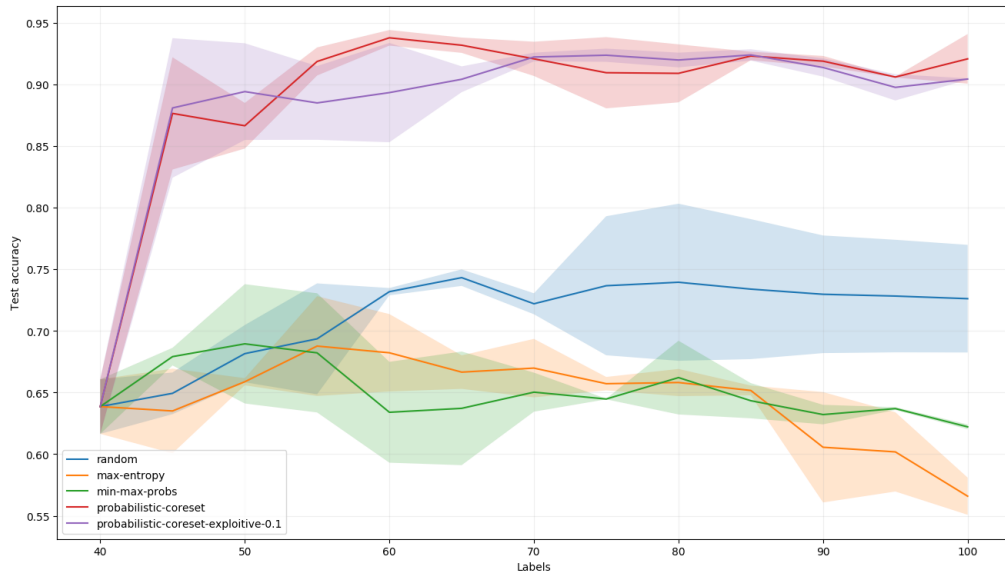


Figure 11: In the toy experiment with 0.5 standard deviation, clusters were mostly separable and core-set variants dominated all baselines. Shaded area represents one standard deviation.

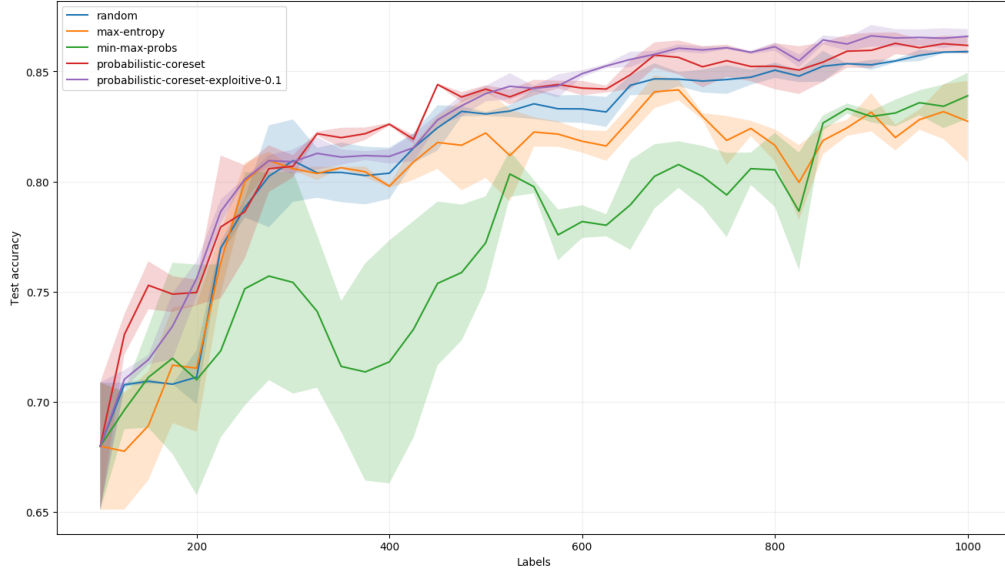


Figure 12: In the toy experiment with 1 standard deviation, clusters overlapped substantially and probabilistic core-set methods formed a modest upper bound in accuracy over all baselines. Shaded area represents one standard deviation.

272 Figure 13 shows examples of data points acquired in the toy experiments by probabilistic core-set
 273 versus the points evaluated to be informative by maximum entropy (Figure 14) and least confidence
 274 (Figure 15). Whereas the core-set variants prioritized covering the input space, the uncertainty-based
 275 methods focused on areas of overlapping clusters, which are prone to error and hard to classify.

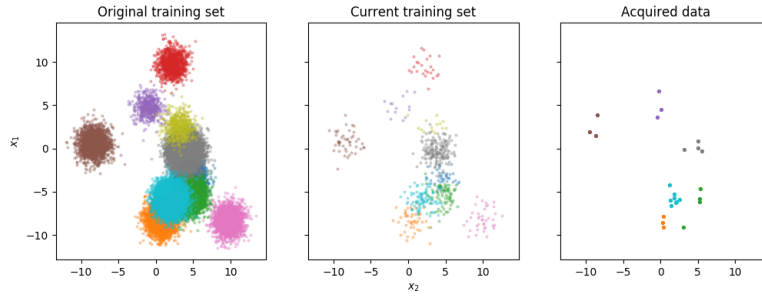


Figure 13: Batch-BALD effectively maximized the distance between elements of selected batches.

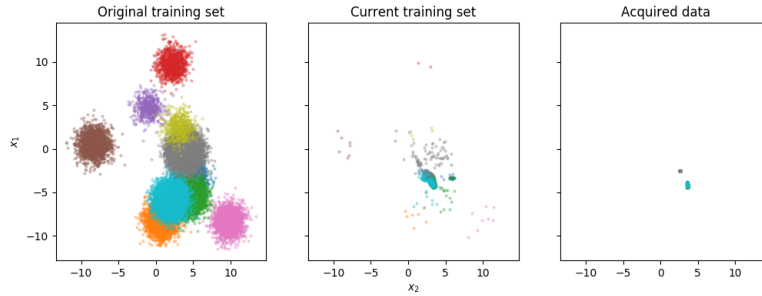


Figure 14: Maximizing entropy resulted in concentrated sampling in the most uncertain regions.

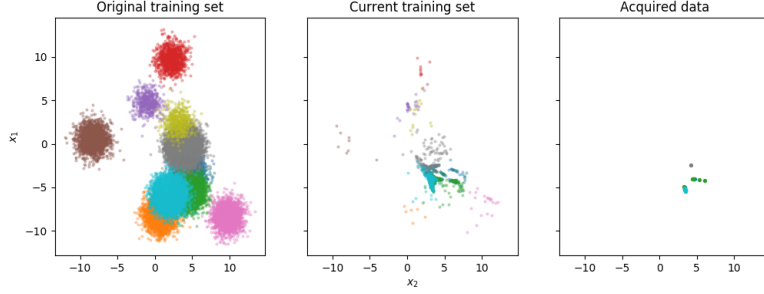


Figure 15: Least confidence also concentrates sampling along uncertain regions.

276 The poor performance of entropy and uncertainty methods for large batch acquisition in a noisy
 277 classification dataset agrees with existing work [11, 5, 10]. The cause of this is wasteful labelling
 278 requests in uncertain regions of features that turned out to be inseparable. In contrast, core-set variants
 279 and random acquisition are successful because they covered the majority of the input space.

280 The effect of increasingly difficult separability on acquisition function efficiency is clear in the toy
 281 data with 0.5 versus 1 standard deviation. When multiple class distributions overlap substantially,
 282 their joint distribution density is sampled more frequently under the core-set variants, which is
 283 harmful because those samples do not improve test accuracy for noisy class boundaries. This suggests
 284 that class inseparability may play some role in the poor performance of the core-set variants.

285 Overall, probabilistic core-sets barely improved from random acquisitions and cost more computation
 286 than Algorithm 2. Like Sener and Savarese [10], we also conclude with the belief that any method
 287 that depends on distributional density sampling will have difficulty exceeding random sampling at
 288 an unknown test because of the obvious fact that i.i.d. samples are already well-represented in the
 289 target distribution. Then, the main beneficial effect of these density sampling techniques is to reduce
 290 redundancy, but this may be a rare phenomenon in the typical high dimensional representations of
 291 under-determined and nonlinear classification tasks.

292 A.3 Batch-BALD evaluates the mutual information of batches of data

293 Given a distribution of model parameters, Bayesian active learning by disagreement (BALD) evaluates
 294 the information of a single data point as its marginal entropy penalized with the average entropy
 295 across the parameter distribution [5]. Intuitively, this selects for samples that elicit low overall
 296 certainty from the Bayesian model, but high individual certainty from the competing hypotheses
 297 sampled from its parameter distribution. Naive application of BALD to a batch of data may lead to
 298 the overestimation of mutual information between elements within the batch [5]. On the other hand,
 299 Batch-BALD scores their joint information [5].

300 The Batch-BALD information metric is useful for identifying likely and different core-set centers in
 301 two important but different ways from its original setting. First, we fit a GMM and sample its means
 302 θ from $P(\theta)$, which we assume to be uniform. We use these Gaussian means to estimate $P(y|\mathbf{x}, \theta)$.
 303 Second, since there may exist multiple means that cover the same peak, optimizing for batch BALD
 304 identifies peaks with high overall certainty that have low likelihood of intersecting with other peaks.