A Gaussian Process-Bayesian Bernoulli Mixture Model for Multi-Label Active Learning

Weishi Shi ∗ Dayou Yu ∗ Qi Yu
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
{ws7586,dy2507,qi.yu}@rit.edu

Abstract

Multi-label classification (MLC) allows complex dependencies among labels, making it more suitable to model many real-world problems. However, data annotation for training MLC models becomes much more labor-intensive due to the correlated (hence non-exclusive) labels and a potentially large and sparse label space. We propose to conduct multi-label active learning (ML-AL) through a novel integrated Gaussian Process-Bayesian Bernoulli Mixture model (GP-\(B^2M\)) to accurately quantify a data sample’s overall contribution to a correlated label space and choose the most informative samples for cost-effective annotation. In particular, the \(B^2M\) encodes label correlations using a Bayesian Bernoulli mixture of label clusters, where each mixture component corresponds to a global pattern of label correlations. To tackle highly sparse labels under AL, the \(B^2M\) is further integrated with a predictive GP to connect data features as an effective inductive bias and achieve a feature-component-label mapping. The GP predicts coefficients of mixture components that help to recover the final set of labels of a data sample. A novel auxiliary variable based variational inference algorithm is developed to tackle the non-conjugacy introduced along with the mapping process for efficient end-to-end posterior inference. The model also outputs a predictive distribution that provides both the label prediction and their correlations in the form of a label covariance matrix. A principled sampling function is designed accordingly to naturally capture both the feature uncertainty (through GP) and label covariance (through \(B^2M\)) for effective data sampling. Experiments on real-world multi-label datasets demonstrate the state-of-the-art AL performance of the proposed model.

1 Introduction

In multi-label classification (MLC), each data instance may be associated with more than one label. Such a rich representation of labels can encode more complex data-label distributions that arise in many real-world problems [1, 2, 3, 4]. As a simple yet powerful tool, binary relevance machines (BRMs) transform an MLC problem into multiple binary problems and train independent binary classifiers for each label [5]. Such a transformation gives BRMs the flexibility to leverage state-of-art binary classifiers (e.g., deep neural networks and SVMs). However, applying BRMs to a correlated and potentially large label space poses key challenges. First, many real-world multi-label datasets contain a large number of labels. Training one predictor per label incurs a prohibitive cost. Second, despite an overall large label space, each data instance is usually assigned limited labels. Many labels are relatively rare and their appearances depend not only on the features but also the occurrence of other labels. Predicting these “complex” labels directly using independent binary classifiers is fundamentally difficult due to the limited positive data instances and weaker direct dependency on the features. Correlations among labels provide important auxiliary information to enhance multi-label

*Equal contribution

In this paper, we propose a novel Gaussian Process-Bayesian Bernoulli Mixture (GP-B²M) model to achieve cost-effective sampling for multi-label active learning (ML-AL). In ML-AL, since labels are not mutually exclusive as in the single label setting, all the labels should be considered collectively when designing an active sampling function so that a data sample’s overall contribution to the entire label space can be accurately measured. However, since only limited training data instances are available for an ML-AL model, how to accurately model label correlations and hence quantify a data sample’s overall informativeness using very sparse labels under AL poses a grand challenge. Existing efforts that explicitly model label correlations usually focus on limited types of correlations such as pairwise \([5, 9]\), conditional \([10, 11]\), or full correlation in a subset of labels \([12, 13]\). Consequently, those methods may miss some important label correlations. Label correlations can also be captured through a latent embedding \([7, 8]\). While these methods can scale to a large label space, they usually require a decent number of training labels to compute an accurate embedding, making them less suitable for ML-AL. Furthermore, the learned embedding has no semantic meanings, which cannot be used to interpret the discovered label dependencies \([\dagger]\).

The proposed GP-B²M model addresses the limitations of existing methods to fundamentally advance ML-AL. In particular, the B²M encodes label correlations using a Bayesian Bernoulli mixture of label clusters. Since labels are highly sparse in ML-AL, a predictive GP is further integrated to learn a distribution of mixture coefficients that connect data features with the label clusters. Thus, the label clusters can be regarded as a global pattern of label co-occurrences discovered from both the training labels and data features to address label sparsity. In this novel feature-component-label mapping, data features serve as an inductive bias to learn accurate mixture components of labels, where data samples with similar features should be mapped to similar mixture components, which in turn lead to a similar set of labels. Such an inductive bias allows the discovery of label relationships from limited labels with the support of feature relationships, which is essential for a sparse label space in ML-AL.

Figure 1 shows three mixture components learned from synthetic data designed with complex label correlations, including geometric, cardinality, overlapping, and hierarchical (see Figure 1(c) for definitions). For example, the hierarchical labels \((E_1 - E_4)\) show some interesting but quite complicated correlations that may exist in many real-world data. If \(E_1\) represents a common disease and \(E_2, E_3\) represent some less common ones that may co-occur with \(E_1\) (30% of the time), then \(E_4\) corresponds to a rather rare disease that only co-occurs with \(E_3\) but not \(E_2\). In Figure 1(c), both components 1 and 2 show a high chance of \(E_4\). It is also clear that while both \(E_1\) and \(E_3\) are very likely to appear in these components, the chance of seeing \(E_2\) is much lower, reflecting its exclusive relationship with \(E_3\). Interestingly, while both components 1 and 2 cover \(E_4\), they also show complementary information. In component 1, \(E_4\) mostly appears in the non-overlapping regions, which is indicated by smaller \(O_1\) and \(O_4\); whereas in component 2, it’s more likely to

\(^\dagger\)Correlation sometimes refers to the degree to which a pair of random variables are linearly related. However, in the broadest sense, both correlation and dependence refer to any statistical relationship between two random variables, so they are used interchangeably in the rest of the paper.
occur in overlapping regions $O_1$ and $O_4$. Finally, in component 3, it is less likely to observe $E_4$. Consequently, the chance to see $E_2$ is significantly higher.

The above example demonstrates that the learned mixture components accurately capture complex label correlations that are critical for active data sampling. They are also interpretable, which can help to unveil important relationships among labels. Our key contribution is threefold: (i) We propose a novel GP-Bayesian Bernoulli mixture model to discover meaningful label correlations from limited labels by encoding the inductive bias from data features as Bayesian priors to learn from both labels and features while ensuring consistency. (ii) We introduce a set of auxiliary latent variables to achieve a fully conjugate feature-component-label mapping in the Bayesian model to support efficient end-to-end posterior inference. The number of mixture components is also dynamically adjusted during Bayesian inference so that the model complexity is automatically calibrated according to the size of training data, which is critical for AL. (iii) The model outputs the predictive distribution that provides both the label prediction and their correlations in the form of a label covariance matrix. We design a novel active sampling function that integrates both feature uncertainty and label covariance to quantify a data sample's overall contribution to a correlated label space. Extensive experiments on both synthetic and real-world multi-label data and comparison with competitive models demonstrate that the proposed GP-B$^2$M achieves the state-of-the-art active learning performance.

2 Related Work

Due to the wide adoption of BRMs for MLC, a number of AL models have been developed based upon BRMs. For example, the estimated reduction of a BRM loss function has been used as an uncertainty criterion for data sampling [14]. Uncertainty from individual SVMs in BRMs has also been integrated to compute a sampling score, where label correlation is used to reduce the complexity of the active query rather than improve active sampling [15]. Label inconsistency provides an alternative way to incorporate label correlation into BRMs [16] for data sampling. This has been further extended through label ranking [17]. As discussed earlier, AL models built upon BRMs do not systematically capture label correlations, which may lead to inaccurate uncertainty measures for ML-AL.

A few existing models capture label correlations explicitly or through a latent embedding to support active multi-label sampling. For example, the approximate entropy of the predicted labels from a Bernoulli Mixtures model (CBM) is used for data sampling [18]. However, the quality of the uncertainty estimation relies on an external multi-class classifier used to predict the component coefficients. Both model selection and parameter tuning for the classifier make it difficult for AL. One fundamental limitation is that CBM was originally designed for MLC (instead of AL) by predicting a distinct set of label clusters for each data sample [6]. Thus, different from the proposed GP-B$^2$M model, no global label clusters are discovered to capture the label correlations, making it unsuitable for multi-label AL. Gao et al propose a correlation aware active sampling method for transfer learning task [19]. However, the method uses a kernel function to measure the label similarity between two data instances which cannot work well for a large and sparse label space. Compressed sensing (CS) has been employed to learn a latent embedding of the label space to capture potential label correlations, which can then be used to design a sampling function. However, since the latent space is continuous, the labels are further assumed to be drawn from a Gaussian distribution to ensure conjugacy, which violates the binary nature of the labels [20]. Furthermore, CS requires an additional step to recover the predicted label from the latent code, which is less efficient for AL. Active sampling is also sensitive to the recovery process and the recovery quality is usually low at the beginning of AL due to the lack of training data [21].

The proposed GP-B$^2$M model systematically addresses the key limitations of existing methods through a well integrated Bayesian framework that supports a fully conjugate feature-component-label mapping and end-to-end posterior inference for cost-effective ML-AL.

3 GP-B$^2$M for Multi-Label Active Learning

In this section, we first describe the proposed GP-B$^2$M model by introducing key latent variables along with their conditional dependencies. We then present a novel posterior inference process by augmenting the original model using a set of auxiliary variables to resolve the non-conjugate prior and likelihood. A principled sampling function is introduced in the end for cost-effective ML-AL.
Data Samples
Θ = Σ

where

This is achieved through a mapping function

softmax) are non-conjugate with the prior distribution

term

p

key challenge that prevents us from using the standard mean field variational inference (MF-VI) is the

Directly maximizing this likelihood is intractable due to the interplay of the latent variables.

Posterior inference of latent variables in the two phases are jointly performed by maximizing the log

learned mixture components

π

feature space with the label mixture components as a way to encode the feature related inductive bias.

label vectors. One key innovation lies in using the latent indicator variables

parameters of the beta distributions so that updated mixture components can best recover the true

F

of the mixture components by learning a distribution of latent functions:

probability of assigning

x

l

labels, where

X

a mixture process. Mixture component

k

y

to component

k

in (1). As the most typical forms of

K

Label clusters

L

Correlated Labels

N Data Samples

GP

B

M

1

2

3

4

C

1

C

2

C

K

Compress

Generate (Phase II)

Predict (Phase I)

Figure 2: (a) The GP-B\(^2\text{M}\) framework for ML-AL; (b) Graphical model of GP-B\(^2\text{M}\).

### 3.1 The Bayesian Bernoulli Mixture Model

Let \(X = \{x_1, \ldots, x_N\}\) denote a training set with \(N\) data samples and \(Y = \{y_1, \ldots, y_N\}\) denote the labels, where \(y_n \in \{0, 1\}^L\). The proposed GP-B\(^2\text{M}\) model assumes there are \(K\) mixture components, \(\Theta = \{\theta_k\}_{k=1}^K\), shared by all the data samples and the label vector of each sample is generated from a mixture process. Mixture component \(k\) is a result of \(L\) Bernoulli experiments governed by two parameters \(a_{kl}\) and \(b_{kl}\): \(\theta_l \sim \prod_{l=1}^L \text{Beta}(a_{kl}, b_{kl})\), where \(\theta_{kl}\) denotes the probability of assigning label \(l\) to component \(k\). The indicator variable \(z_{nk}\) denotes whether component \(k\) is assigned to sample \(x_n\), where \(z_{nk} \sim \text{Cat}(\pi_n)\), \(\pi_{nk} = h^{(k)}(f_n)\), \(h^{(k)}\) is a mapping function that outputs the probability of assigning \(x_n\) to mixture component \(k\), and \(f_n = (f_n^{(1)}, \ldots, f_n^{(K)})^T\) are GP latent functions for sample \(x_n\) with \(f_n^{(k)} = f^{(k)}(x_n)\). The final label vector associated with \(x_n\) is drawn from the mixture distribution given by \(p(y_n|\Theta) = \sum_{k=1}^K \pi_{nk} \theta_k\). Figure 2(b) shows the graphical model of this generative process.

The GP-B\(^2\text{M}\) model essentially adopts a two-phase learning process, where these two phases are seamlessly integrated (see Figure 2(a)). In phase I, it predicts the probabilistic assignment of the mixture components by learning a distribution of latent functions: \(F = \{f^{(k)}\}_{k=1}^K\), where \(f^{(k)} = (f^{(k)}_1, \ldots, f^{(k)}_N)^T\). In phase II, these predicted mixture assignments are used to refine the parameters of the beta distributions so that updated mixture components can best recover the true label vectors. One key innovation lies in using the latent indicator variables \(Z = \{z_{nl}\}_{l=1}^L\) to link the feature space with the label mixture components as a way to encode the feature related inductive bias.

This is achieved through a mapping function \(h_n = (h_n^{(1)}, \ldots, h_n^{(K)})^T\), with \(h_n^{(k)} = h^{(k)}(f_n)\):

\[
p(z_{nk} = 1|f_n) = \pi_{nk} = h^{(k)}(f_n), h^{(k)} \in [0, 1], \sum_{k=1}^K h^{(k)} = 1, \quad p(f^{(k)}) = \mathcal{N}(f^{(k)}; \mathbf{m}_k, \Sigma_k) \tag{1}
\]

where \(\Sigma_k = [K(x_n, x_m)]\) is a covariance matrix and \(K(\cdot, \cdot)\) is a kernel function, and we set \(\mathbf{m}_k = \mathbf{0}\) with out losing generality. From the Bayesian perspective, this is equivalent to placing a Dirac delta prior over \(\pi_{nk} : \pi_{nk} \sim \delta(\pi_{nk} - h^{(k)}(f_n))\), where the inductive bias is encoded by the prior distribution. By introducing \(h^{(k)}(f_n)\), we essentially convert a multi-label problem into a multi-class problem as \(\pi_{nk}\) encodes the probability of assigning \(x_n\) to each of the \(K\) components. Specifically, given the learned mixture components \(\theta_k\)’s, for a test data sample \(x_*\), we predict the component assignments \(\pi_*\) using the trained GP. The final labels are obtained as \(p(y_*|\Theta) = \sum_{k=1}^K \pi_{nk} \theta_k\).

Posterior inference of latent variables in the two phases are jointly performed by maximizing the log marginal likelihood of the observed multiple labels for all training samples:

\[
\ln p(Y|X) = \ln \int \int \sum_Z \prod_n \prod_l \prod_k p(f^{(k)}) p(\theta(k)) p(z_{nk}|f_n) p(y_{nl}|z_{nk}, \theta(k)) dF d\Theta \tag{2}
\]

Directly maximizing this likelihood is intractable due to the interplay of the latent variables. So we turn to optimizing the evidence lower bound (ELBO) of the log marginal:

\[
\mathcal{L}(q) = \int q(\Theta, Z, F) \ln p(Y, Z, F|\Theta, X) d\Theta dZ dF, \quad q(\Theta, Z, F) \text{ is the variational distribution. However, a key challenge that prevents us from using the standard mean field variational inference (MF-VI) is the term } p(z_{nk}|f_n), \text{ defined by the mapping function } h^{(k)} \text{ in (1). As the most typical forms of } h^{(k)} \text{ (e.g., softmax) are non-conjugate with the prior distribution } p(f^{(k)}) \text{, which is a Gaussian, the variational posterior } q(f^{(k)}) \text{ cannot be derived analytically.}
3.2 Auxiliary Variables based Variational Inference

We propose to resolve the non-conjugate mapping function in the complete data likelihood by introducing a number of auxiliary latent variables such that the augmented complete data likelihood becomes conjugate. Auxiliary variables have been used in MCMC based inference, such as slice sampling \cite{rippoll} and Hamiltonian MCMC \cite{vehtari}, with improved sampling efficiency. The basic idea of auxiliary variables based variational inference (AV-VI) is to apply the following transformation: 

\[
p(x) = \int_y p(x|y)p(y)dy,
\]

where \( p(x) \) is a target function that is difficult to compute (e.g., non-conjugate) during VI. If the conditional likelihood \( p(x|y) \) is still non-conjugate, this process will continue until a conjugate conditional is achieved.

A key identity that we leverage to achieve a conditional likelihood conjugate to a Gaussian prior \( p(f(k)) \) is to convert a logistic sigmoid function as a scale mixture of Gaussian’s \cite{minka} where the mixture is defined by a Pólya-Gamma distribution \( p(\omega) = PG(\omega|b,0) \),

\[
\frac{(e^f)^a}{(1 + e^f)^b} = 2^{-b}e^{\kappa f} \int_0^\infty e^{-\omega f^2} p(\omega) d\omega
\]

where \( b \geq 0, \kappa = a - \frac{b}{2} \). However, a sigmoid function is only suitable for binary classification, making it infeasible for a mapping function that outputs the assignments for \( K \geq 2 \) components. Thus, we adopt the logistic-softmax function \cite{minka} as our mapping function:

\[
h_n^{(k)} = p(z_{nk} = 1|f_n) = \frac{\sigma(f_n^{(k)})}{\sum_{j=1}^{K} \sigma(f_n^{(j)})}
\]

To handle the summation in \cite{minka}, we introduce random variables \( \lambda_1: \lambda_N \) and use identity \( \frac{1}{x} = \int_0^\infty e^{-\lambda x} d\lambda \) so that

\[
p(z_{nk} = 1|f_n, \lambda_n) = \sigma(f_n^{(k)}) \prod_{j=1}^{K} e^{-\lambda_n \sigma(f_n^{(j)})}
\]

where \( p(\lambda_n) \propto 1_{(0,\infty)}, \forall n \in [1, N] \). By leveraging the moment generation function of the Poisson distribution \( \text{Po}(\lambda) \), we introduce random variables \( \Upsilon = \{ \upsilon_1, ..., \upsilon_N \} \), where \( \upsilon_n = (\upsilon_{n1}, ..., \upsilon_{nK})^T \), to convert the exponential term in \cite{minka}, which leads to

\[
p(z_{nk} = 1|f_n, \upsilon_n) = \sigma(f_n^{(k)}) \prod_{j=1}^{K} (\sigma(-f_n^{(j)})^{\upsilon_{nj}})
\]

where \( \upsilon_{nk} \sim \text{Po}(\upsilon_{nk} | \lambda_n) \). Finally, using \cite{minka} and introducing the Pólya-Gamma random variables \( \Omega = \{ \omega_1, ..., \omega_N \} \), where \( \omega_n = (\omega_{n1}, ..., \omega_{nK})^T \), leads to

\[
p(z_{nk} = 1|f_n, \upsilon_n, \omega_{nk}) = \prod_{k=1}^{K} 2^{-(z_{nk} + \upsilon_{nk})} \exp \left\{ \frac{(z_{nk} - \upsilon_{nk})f_n^{(k)}}{2} - \frac{(f_n^{(k)})^2}{2} \omega_{nk} \right\}
\]

where \( \omega_{nk} \sim PG(\omega_{nk}|\upsilon_{nk}, 0) \). Figure \cite{minka} shows the graphical model with the auxiliary variables (\( x_n \)'s are omitted from the graph to keep the notation uncluttered).

We proceed by defining a variational distribution with auxiliary variables:

\[
q(\Theta, Z, F, \Lambda, \Upsilon, \Omega) = q(\Theta)q(Z)q(F)q(\Lambda)q(\Upsilon)q(\Omega)
\]

The optimal variational distribution can be obtained by computing the moments of component variational distributions using some important properties of the main and auxiliary variables and iterating until convergence. The optimal variational distributions of the main latent variables are summarized in the following theorem.

**Theorem 1.** With the auxiliary random variables and the transformed complete conditional likelihood given in \cite{minka}, the optimal components of the variational distribution as specified by \cite{minka} are given by

- **Component assignments** \( \hat{q}(Z) = \prod_n \prod_k \hat{q}(z_{nk}) \):

\[
\hat{q}(z_{nk}) = \text{Cat}(z_{nk}|\hat{\phi}_{nk}) ; \quad \hat{\phi}_{nk} \propto \exp \left\{ \sum_{l=1}^{L} \left[ y_{nl}(\psi(\hat{a}_{kl}) - \psi(\hat{a}_{kl} + b_{kl})) \right] + \frac{m_{nk}}{2} \right\}
\]

where \( \psi(\cdot) \) is the digamma function and \( m_{nk} \) is \( n \)-th element of mean of \( \hat{q}(f^{(k)}) \) defined in \( \hat{q}(F) \).
Figure 3: Graphical model with auxiliary variables

- Bernoulli mixture components $q(\Theta) = \prod_k \prod_i q(\theta_{ki})$

  
  $q(\theta_{ki}) = \text{Beta}(\theta_{ki} | \delta_{ki}, \beta_{ki}); \quad \widehat{a}_{ki} = a_{ki} + \sum_{n=1}^{N} \phi_{nk} y_{nl}, \quad \widehat{b}_{ki} = b_{ki} + \sum_{n=1}^{N} \phi_{nk}(1 - y_{nl})$  

- GP latent functions $q(F) = \prod_k q(f(k))$:  

  $q(f(k)) = \mathcal{N}(f_k|m_k, \Sigma_k); \quad m_k = \frac{1}{2} \Sigma_k ((\Phi_k - \mathbb{E}(\xi_k)) | \xi_k)), \quad \Sigma_k = (\Sigma_k^{-1} + \text{diag}(\mathbb{E}(q(\omega_k, v_k) | \omega_k)))^{-1}$

  where $\nu_k = (\nu_{1k}, ...., \nu_{Nk})^T, \omega_k = (\omega_{1k}, ..., \omega_{Nk})^T$, and $q(\omega_k, v_k) = q(\omega_k | v_k)q(v_k)$ is the optimal variational distribution for these auxiliary variables.

The specific forms of the auxiliary variational distributions $q(\lambda)$ and $q(\Theta, \Omega)$ are provided in Appendix B as part of the detailed proof of the theorem.

Model interpretation. The optimal variational distributions are fairly intuitive. Interpreting these distributions can reveal some key insights on how the proposed GP-B^2M model leverages the data features as an effective inductive bias to discover semantically coherent components from a sparse label space. First, from (9), the component assignment of data sample $(x_n, y_n)$ is determined by two terms: the first term indicates how all its labels $y_n$ are correlated with the component and the second term reflects how likely to categorize the features $x_n$ into the component. Second, from (10), since the component assignments are further utilized to compute the Bernoulli mixture components, the optimal components naturally aggregate both label and feature information to ensure semantic consistency as a result of using data features as the inductive bias. Last, from (11), the GP latent function value on a component increases with a positive component assignment and decreases with a ‘negative’ assignment, captured by the Poisson auxiliary variables $v_k$.

Time complexity. According to (11), posterior inference of GP-B^2M has the computational complexity of $O(N^3K)$ which is identical to training $K$ GPs. Since each component can be updated independently, we can parallelize the computation to further reduce the complexity to $O(N^2)$. We can further leverage sparse kernel machines (e.g., Sparse GP) to reduce the complexity if $N$ is large.

3.3 Multi-Label Active Sampling

Being a Bayesian model, GP-B^2M outputs the predictive distribution that provides both the label prediction and a label covariance matrix. As the covariance matrix captures both the uncertainty of individual labels and correlation of each pair of labels, it provides essential information to design a principled measure to quantify a data sample’s overall contribution to a correlated label space.

For each testing sample $x_*$, the predictive mean can be computed using the variational distributions:

$$
\mathbb{E}[y_* | x_*] \approx \sum_k \mathbb{E}_{p(F_*, X, Y, x_*) | \pi_*} \mathbb{E}_{q(\Theta)} [\theta_k]
$$

$$
p(F_* | X, Y, x_*) \approx \int p(F_* | X, F, x_*) q(F) dF, \quad \mathbb{E}_{q(\Theta)} [\theta_{kl}] = \frac{\alpha_{kl}}{(\alpha_{kl} + \beta_{kl})}
$$

where $\pi_* = p(z_* = 1 | f_*)$ is defined as a logistic-softmax function given in (10). Theorem 1 shows that $q(F)$, which approximates the true posterior $p(F | X, Y)$, is a Gaussian. Hence, $p(F_* | X, Y, x_*)$ is also a Gaussian. However, the logistic-softmax transformation makes predictive mean intractable.
We design a synthetic dataset with 18 labels that exhibit 4 distinct types of dependencies as defined through our active sampling by examining sampled data instances. We conduct extensive experiments on both synthetic and real-world multi-label data to demonstrate important properties of GP-B

The label covariance is computed using a point estimate of $\Lambda_k$, which is located deep in the hierarchy and appears much less than other labels. We use B$^2$M to capture complex label correlations and how they contribute to model training and sample them accordingly. Our final sampling function is given by: $x_s = \arg \max_{x_s} \ln[\text{cov}[y_s| x_s]] + \eta \sum_k \text{Var}[\pi_{sk}] / K$, where $\eta$ is used to balance between label covariance and predictive variance of data features. It can be dynamically updated to give a higher weight in the early stage of AL to the feature variance term for better exploration of the data space and then shift the focus to the label covariance term for effective fine-tuning of decision boundaries with a correct shape obtained through effective exploration.

4 Experiments

We conduct extensive experiments on both synthetic and real-world multi-label data to demonstrate: (1) important properties of GP-B$^2$M to capture complex label correlations and how they contribute to predict complex labels, (2) state-of-the-art ML-AL performance by comparing with existing competitive models, (3) impact of key model parameters through an ablation study, and (4) effectiveness of active sampling by examining sampled data instances.

4.1 Synthetic Data

We design a synthetic dataset with 18 labels that exhibit 4 distinct types of dependencies as defined in Figure 4(c). In the introduction, we show that three discovered mixture components precisely capture some rather complex label dependencies (e.g., hierarchical and exclusive) while being highly interpretable. For this dataset, the model discovers 10 components in total and we show some other components in Appendix D along with their interpretations. We further demonstrate how the discovered components contribute to the prediction of more complex and less frequent labels. We use $E4$ as an example, which is located deep in the hierarchy and appears much less than other labels.

Figure 4(a) shows the distribution of data samples whose labels contain $E4$. It can be seen that these samples are distributed across the entire $E1$ region (roughly corresponds to the shaded area in purple).
Note that, in addition to \( E_1, E_4 \) also depends on the exclusive relationship: \( E_3 \) AND NOT \( E_2 \) (\( E_2, E_3 \) are not shown in the figure to keep the distribution of \( E_4 \) clear). Figure 4 (b) shows the prediction result from BRMs, which has very high false positive and negative rates. The poor performance is also reflected by a low ROC-AUC (area under the receiver operating characteristic curve) score at 0.58 (slightly better than random guessing). It appears that BRMs only predict correctly samples in an area where \( E_4 \) samples are relatively dense while missing most others. This is because BRMs try to directly learn the feature-label mapping (by training independent binary predictors), which is usually weak for complex and less frequent labels, like \( E_4 \).

Different from BRMs, the proposed GP-B^2M learns mixture components that correctly capture the label correlations and the final labels can be recovered by combining the mixture components through their predicted coefficients. As discussed earlier, \( E_4 \) has a high chance to appear in either Component 1 or 2. Figure 4 (c) shows the predicted component assignments and the top component is highlighted. As can be seen, most \( E_4 \) samples are assigned Component 1 or 2 as their top component. Also, samples assigned to Component 1 (shown in blue) are mostly distributed in the non-overlapping geometric regions and those assigned to Component 2 (shown in yellow) are mostly in overlapping regions. By leveraging these components, GP-B^2M achieves much better prediction results as shown Figure 4 (d). Table 5 summarizes the AUC scores from BRMs and GP-B^2M on different types of labels. While both models achieve similar performance on some common labels (e.g., \( G \)), GP-B^2M significantly outperforms BRMs for more complex labels, where it is essential to capture important label correlations.

4.2 Real Data

Datasets and experiment settings. We choose five representative real-world multi-label datasets, including Delicious, Enron, Bibtex, Corel5K, and NUS-WIDE, from different application domains [26]. All datasets have a relatively large label space and high label sparsity (2–6%). Table 2 in Appendix D summarizes key properties of the pre-processed datasets. We randomly shuffle each dataset and partition them into three parts: training, testing, and candidate pool. We keep a minimum of one positive instance per label in the initial training partition as required by BRMs based AL models. To make sure each label is well represented in each partition, we remove extremely rare labels with label frequency less than 20. Since the remaining labels are still highly imbalanced, we use the ROC-AUC score to evaluate the model performance. All the baseline models share the same copy of the initial training set to make a fair comparison. Active learning stops after each model selects 500 samples.

Performance comparison. We include five competitive baselines for AL performance comparison:

- MMC samples instances that introduce the greatest change of the expected loss. During label prediction, it uses logistic regression to predict the number of labels for a new instance [14].
- Adaptive considers both the separation margin of an SVM and the label cardinality inconsistency and combines these two parts for data sampling [15].
- AUDI uses a label ranking mechanism, where a dummy label is used to separate the positive and negative labels. Its sampling function is based on a modified cardinality inconsistency measure [27].
- CVIRS combines the ranking on the magnitude of the difference margin in predictions and the label vector inconsistency for active sampling [17].
- CS-GP conducts active sampling in a compressed label space using a multi-output GP [21].

Figure 5 reports the AUC scores of all the models. For each curve, we present the average result along with the error bar from 3 trials of randomly initialized AL experiments. The proposed B^2M model achieves better AL performance consistently on all the datasets. For multiple datasets, GP-B^2M establishes a clear advantage in the early to middle stages of AL. While a few baselines eventually converge to a similar AUC score, they usually take more iterations (by consuming more labels) to reach a comparable performance as GP-B^2M. In addition, by comparing with random sampling with the proposed sampling function, we clearly demonstrate that the superior AL performance attributes to both the Bayesian mixture model and the effective active sampling. Note that the AUDI model runs much slower for Bibtex and Corel5K when both the number of features and candidate pool size become very large so we omit the results.

Analysis on label-wise improvement. To further justify why the proposed GP-B^2M outperforms the baselines, we offer a fine-grained analysis on the label-wise performance, which helps to achieve...
which balances label covariance and feature uncertainty for data sampling and (2) We further investigate the impact of two tunable parameters of the model: (1) Ablation study.

Figure 6: AL performance comparison (a)-(e); Ablation study on one example dataset (Enron): (f) shows the impact of $\eta$ and (g) shows the impact of effective components.

Figure 7: The label-wise average precision improvement over BRMs models.

a deeper insight on the performance gain. In fact, we reach a similar conclusion as the synthetic data experiments: while all models perform reasonably well on the most frequent labels, GP-B$^2$M clearly outperforms other baselines on less frequent labels, where considering the correlations with other labels can help to improve their predictions. Thus, in our fine-grained analysis, we first exclude the few most frequent labels and compute the Average Precision Improvement (API) of the $i$-th label,

$$API^{(i)} = \frac{AP^{(i)}_{B^2M} - AP^{(i)}_{BRMs}}{AP^{(i)}_{BRMs}} \times 100\%$$

where $AP^{(i)}_{B^2M}$ and $AP^{(i)}_{BRMs}$ denote the average precision of label $i$ provided by GP-B$^2$M and BRMs, respectively. Figure 7 demonstrates the label-wise $API$ on the NUS-WIDE and Delicious datasets as two illustrative examples. We observe that GP-B$^2$M performs significantly better than BRMs on labels with a moderate or low frequency. This is because those labels usually do not have sufficient positive instances for BRMs to learn independently. Furthermore, they may have a complex correlation with other labels. GP-B$^2$M effectively leverages label correlations to make better predictions on these labels. This further confirms the overall good performance of GP-B$^2$M.

**Ablation study.** We further investigate the impact of two tunable parameters of the model: (1) $\eta$, which balances label covariance and feature uncertainty for data sampling and (2) $\rho$, which controls the effective number of mixture components. Limited by space, we use the Enron dataset as an example and report other results in Appendix D. Figure 6(f) compares the performance under different $\eta$ values. In early iterations, the label covariance guided sampling ($\eta$ is small) slightly falls behind the feature uncertainty guided sampling ($\eta$ is large) as the latter is more useful to explore
the feature space. Label covariance guided sampling gradually catches up and finally surpasses the feature uncertainty guided sampling. As shown next by the sampled instances, both criteria select informative instances that play complementary roles to improve the AL model. Figure 6(g) shows how the AL performance is affected by the effective number of mixture components that is automatically determined by an upper bound $K$ and component strength ratio $\rho$. For component $k$, we compute its total ‘effective posterior observations’ $g(\theta_k) = \sum_{l=1}^{L} a_{kl} + \sum_{l=1}^{L} b_{kl}$ and define a threshold as $\bar{g} = (\rho/K) \sum g(\theta_k)$. The effective components only include those with $g(\theta_k) \geq \bar{g}$. When the size of the training set is still small (early stage in AL), fewer components (large $\rho$) yield better results than more components (small $\rho$) by avoiding over-fitting. When more training labels are acquired, a more flexible model can better explain the label correlations thus has better performance.

**Examples of actively sampled instances.** To demonstrate the effectiveness of the proposed sampling function, we show images sampled by GP-B2M from the NUS-WIDE dataset. To explain the distinct nature of these sampled images and how they contribute to the model training, we also show some representative images from the initial training pool for comparison. These correspond to images in Figure 8(a)-(c) with common labels: ‘plants’ and ‘sky’. First, for sampled images with a large feature uncertainty, while ‘plants’ and/or ‘sky’ are predicted for those images, they look very different from the training images. In particular, although the labels of the image in Figure 8(e) contain both ‘plants’ and ‘sky’, there are no visible plants. For the image in Figure 8(f), the label ‘person’ is not present even though a person is visible in the image. These types of samples are significantly dissimilar to the initial training set, thus considered valuable to explore the feature space for effective sampling. Images in Figure 8(g)-(h) are samples based on a high label covariance. These images look similar to the training examples but their corresponding labels are somewhat different. Sampling these images can further improve the prediction of these labels and correlations there of, such as ‘plants’, ‘grass’, and ‘nature’. As these images bring in additional labels, they may also help the model discover more possible correlations, such as that between ‘plants’ and ‘flowers’ or ‘tree’.

5 Conclusion

We present a novel Gaussian Process-Bayesian Bernoulli Mixture (GP-B2M) model for cost-effective multi-label active learning. GP-B2M extracts global patterns of label correlations by learning from both (limited) training labels and data features. The mixture components, which are accurately learned from end-to-end and fully conjugate posterior inference, are capable of encoding complex label correlations while being highly interpretable. A novel sampling function is designed by combining feature uncertainty and label covariance, both of which can be obtained from the predictive distribution of the GP-B2M model. Experiments conducted on both synthetic and real data justify the important properties of the model and its state-of-the-art AL performance.
Acknowledgements

This research was supported in part by an NSF IIS award IIS-1814450 and an ONR award N00014-18-1-2875. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency. We would also like to thank the anonymous reviewers for their constructive comments.

References


Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] For performance comparison, we tried to include all the representative baselines based on our best knowledge. However, certain baselines may be designed for a specific purpose. As a result, some baseline cannot run properly for the datasets used in our experiments to generate reasonable results. We have made this clear in the performance comparison section of our experiments. Another potential limitation is that all the labels collected are assumed to be accurate, like most other active learning models. Dealing with imperfect labels is an important direction that is out of the scope of this paper.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We provided references of real-world applications (e.g., bioinformatics and medicine) with positive societal impacts. One potential negative impact is that the active learning model could be leveraged by malicious attackers as a tool to manipulate the supervised learning model. Since the model will be trained using much less data, if certain data samples are compromised, the impact may be larger than a typically trained supervised learning model.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]