

Mining Block I/O Traces for Cache Preloading with Sparse Temporal Non-parametric Mixture of Multivariate Poisson*

Lavanya Sita Tekumalla[†]

Chiranjib Bhattacharyya[†]

Abstract

Existing caching strategies, in the storage domain, though well suited to exploit short range spatio-temporal patterns, are unable to leverage long-range motifs for improving hitrates. Motivated by this, we investigate novel Bayesian non-parametric modeling(BNP) techniques for count vectors, to capture long range correlations for cache preloading, by mining Block I/O traces. Such traces comprise of a sequence of memory accesses that can be aggregated into high-dimensional sparse correlated count vector sequences.

While there are several state of the art BNP algorithms for clustering and their temporal extensions for prediction, there has been no work on exploring these for correlated count vectors. Our first contribution addresses this gap by proposing a DP based mixture model of Multivariate Poisson (DP-MMVP) and its temporal extension(HMM-DP-MMVP) that captures the full covariance structure of multivariate count data. However, modeling full covariance structure for count vectors is computationally expensive, particularly for high dimensional data. Hence, we exploit sparsity in our count vectors, and as our main contribution, introduce the Sparse DP mixture of multivariate Poisson(Sparse-DP-MMVP), generalizing our DP-MMVP mixture model, also leading to more efficient inference. We then discuss a temporal extension to our model for cache preloading.

We take the first step towards mining historical data, to capture long range patterns in storage traces for cache preloading. Experimentally, we show a dramatic improvement in hitrates on benchmark traces and lay the groundwork for further research in storage domain to reduce latencies using data mining techniques to capture long range motifs.

1 Introduction

Bayesian non-parametric modeling, while well explored for mixture modeling of categorical and real valued data, has not been explored for multivariate count data. We explore BNP models for sparse correlated count vectors to mine block I/O traces from enterprise storage servers for *Cache Preloading*.

Existing caching policies in systems domain, are either based on eviction strategies of removing the least relevant data from cache (Ex: Least Recently Used a.k.a LRU) or read ahead strategies for sequential access patterns. These strategies are well suited for certain types of workloads where nearby memory accesses are correlated in extremely short intervals of time, typically in milli-secs. However, often in real workloads, we find correlated memory accesses spanning long intervals of time (See fig 1), exhibiting no discernible correlations over short intervals of time (see fig 2).

There has been no prior work on analyzing trace data to *learn long range access patterns* for **predicting future accesses**. We explore caching alternatives to automatically learn long range spatio-temporal correlation structure by analyzing the trace using novel BNP techniques for count data, and exploit it to pro-actively preload data into cache and improve *hitrates*.

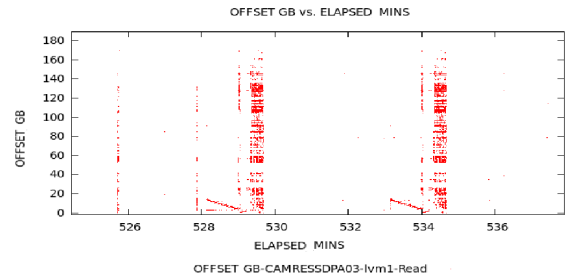


Figure 1: LBA accesses vs time plot for MSR trace(CAMRESWEBA03-lvm2): A Zoomed out(coarser view) shows repeating access patterns over a time span of minutes.

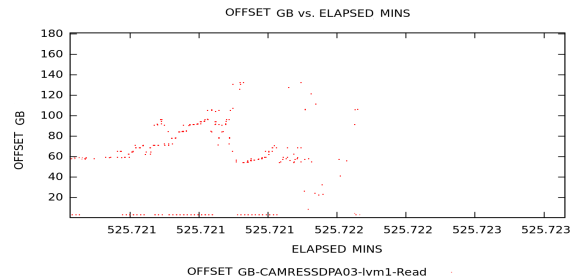


Figure 2: A zoom-in access level view of this trace, into one of these patterns, over a time span of ~ 100 milli-secs, does not show any discernible correlation. Also note the sparsity of data; during any interval of time, only a small subset of memory locations are accessed

*This work was done in collaboration with NetApp, Inc.

[†]Indian Institute of Science

Capturing long range access patterns in Trace Data: Block I/O traces comprise of a sequence of memory block access requests (often spanning millions per day). We are interested in mining such traces to capture spatio-temporal correlations arising from repetitive long range access patterns (see fig 1). For instance, every time a certain file is read, a similar sequence of accesses might be requested.

We approach this problem of capturing the long-range patterns by taking a more aggregated view of the data to understand longer range dependencies. We partition both the memory and time into discrete chunks and constructing histograms over memory bins for each time slice (spanning several seconds) to get a coarser view of the data. Hence, we aggregate the data into a sequence of count vectors, one for each time slice, where each component of the count vector records the count of memory access in a specific bin (a large region of memory blocks) in that time interval. Thus, a trace is transformed into a sequence of count vectors.

Thus, the components of count vector instances after aggregation are correlated within each instance since memory access requests are often characterized by *spatial correlation*, where adjacent regions of memory are likely to be accessed together. This leads to a rich covariance structure. Further, due to the long range temporal dependencies in access patterns (fig 1), the sequence of aggregated count vectors are also temporally correlated. Count vectors thus obtained by aggregating over time and space are also sparse, where only a small portion of memory is accessed in any time interval (fig 2). Hence a small subset of count vector dimensions have significant non zero values. Modeling such sparse correlated count vector sequences to understand their spatio-temporal structure remains unaddressed.

Modeling Sparse Correlated Count Vector Sequences: A common technique for modeling temporal correlations are Hidden Markov Models(HMMs) which are mixture model extensions for temporal data. Owing to high variability in access patterns inherent in storage traces, finite mixture models do not suffice for our application since the number of mixture components varies often based on the type of workload being modeled and the kind of access patterns. BNP techniques address this issue by automatically adjusting the number of mixture components based on complexity of data. Non-parametric clustering with Dirichlet Process(DP) mixtures and their temporal variants have been extensively studied over the past decade [15], [16]. However, to the best of our knowledge we are not aware of such models in the context of count data, particularly temporally correlated sparse count vectors.

Poisson distribution is a natural prior for counts and the Multivariate Poisson(MVP) for correlated count vectors. However, owing to the structure of multivariate Poisson and its computational intractability [17] non parametric mixture modeling for multivariate count vectors has received less attention. Hence, we first bridge this gap by paralleling the development of *DP based non-parametric mixture models and their temporal extensions for multivariate count data* along the lines of those for Gaussians and multinomials .

Modeling the full covariance structure using the MVP is often computationally expensive. Hence, we further exploit the sparsity in data and introduce *sparse mixture models for count vectors* and their temporal extensions. We propose a sparse MVP Mixture modeling the covariance structure over a select subset of dimensions for each cluster. We are not aware of any prior work that models sparsity in count vectors.

The proposed predictive models showed dramatic hitrate improvement on several real world traces. At the same time, these count modeling techniques are of independent interest outside the caching problem as they can apply to a wide variety of settings such as text mining, where often counts are used.

Contributions: Our first contribution, is the DP based non-parametric mixture of Multivariate Poisson (DP-MMVP) and its temporal extensions (HMM-DP-MMVP) which capture the full covariance structure of correlated count vectors. Our next contribution, is to exploit the sparsity in data, and proposing a novel technique for non parametric clustering of sparse high dimensional count vectors with the sparse DP-mixture of Multivariate Poisson (Sparse-DP-MMVP). This methodology not only leads to a better fit for sparse multidimensional count data but is also computationally more tractable than modeling full covariance. We then discuss a temporal extension, Sparse-HMM-DP-MMVP, for cache preloading. We are not aware of any prior work that addresses non-parametric modeling of sparse correlated count vectors.

As our final contribution, we take the first steps in outlining a framework for cache preloading to capture long range spatio-temporal dependencies in memory accesses. We perform experiments on real-world benchmark traces showing dramatic hitrate improvements. In particular, for the trace in Fig 1, our preloading yielded a 0.498 hitrate over 0.001 of baseline (without preloading), a 498X improvement (trace MT2: Tab 2).

2 Related Work

BNP for sparse correlated count vectors: Poisson distribution is a natural prior for count data. But the multivariate Poisson(MVP) [9] has seen limited use due to its computational intractability due to the

complicated form of the joint probability function [17]. There has been relatively little work on MVP mixtures [8, 2, 13, 14]. On the important problem of designing MVP mixtures with an unknown number of components, [13] is the only reference we are aware of. The authors explore MVP mixture with an unknown number of components using a truncated Poisson prior for the number of mixture components, and perform uncollapsed RJMCMC inference. DP based models are a natural truncation free alternative that are well studied and amenable to hierarchical extensions [16, 4] for temporal modeling which are of immediate interest to the caching problem. To the best of our knowledge, there has been no work that examines a truncation free non-parametric approach with DP-based mixture modeling for MVP. Another modeling aspect we address is the sparsity of data. A full multivariate emission density over all components for each cluster may result in over-fitting due to excessive number of parameters introduced from unused components. We are not aware of any work on sparse MVP mixtures. There has been some work on sparse mixture models for Gaussian and multinomial densities [11] [18]. However they are specialized to the individual distributions and do not apply here. Finally, we investigate sparse MVP models for temporally correlated count vectors. We are not aware of any prior work that investigates mixture models for temporally correlated count vectors.

Cache Preloading: Preloading has been studied before [20] in the context of improving cache performance on enterprise storage servers for the problem of cache warm-up, of a cold cache by preloading the most recently accessed data, by analyzing block I/O traces. Our goal is however different, and more general, in that we are seeking to improve the cumulative hit rate by exploiting long ranging temporal dependencies even in the case of an already warmed up cache. They also serve as an excellent reference for state of the art caching related studies and present a detailed study of the properties of MSR traces. We have used the same MSR traces as our benchmark. Fine-grained prefetching techniques to exploit short range correlations [6, 12], some specialized for sequential workload types [5] (SARC) have been investigated in the past. Our focus, however is to work with general non-sequential workloads, to capture long range access patterns, exploring prediction at larger timescales. Improving cache performance by predicting future accesses based on modeling file-system events was studied in [10]. They operate over NFS traces containing details of file system level events. This technique is not amenable for our setting, where the only data source is block I/O traces, with no file system level data.

3 A framework for Cache Preloading based on mining Block I/O traces

In this section we briefly describe the caching problem and describe our framework for cache preloading.

3.1 The Caching Problem: Application data is usually stored on a slower persistent storage medium like hard disk. A subset of this data is usually stored on **cache**, a faster storage medium. When an application makes an I/O request for a specific block, if the requested block is in cache, it is serviced from cache. This constitutes a **cache hit** with a low application latency (in microseconds). Else, in the event of a **Cache miss**, the requested block is first retrieved from hard disk into cache and then serviced from cache leading to much higher application latency (in milliseconds).

Thus, the application’s performance improvement is measured by $hitrate = \frac{\#cachehits}{\#cachehits + \#cachemisses}$.

3.2 The Cache Preloading Strategy: Our strategy involves observing a part of the trace \mathcal{D}^{lr} for some period of time and deriving a model, which we term as **Learning Phase**. We then use this model to keep predicting appropriate data to place in cache to improve hitrates in **Operating Phase** at the end of each time slice (ν secs) for the rest of the trace \mathcal{D}^{op} .

In terms of the *execution time of our algorithms*, while the learning phase can take a few hours, the operational phase, is designed to run in time much less than the slice length of ν secs. In this paper we restrict ourselves to learning from a fixed initial portion of the trace. In practice, the learning phase can be repeated periodically, or even done on an online fashion.

Data Aggregation: As the goal is to improve hitrates by preloading data exploiting *long range* dependencies, we capture this by *aggregating* trace data into count vector sequences. We consider a partitioning of addressable memory (LBA Range) into M equal bins. In the learning phase, we divide the trace \mathcal{D}^{lr} into T_{lr} fixed length time interval slices of length ν seconds each. Let $A_1, \dots, A_{T_{lr}}$ be the set of actual access requests in each interval of ν seconds. We now aggregate the trace into a sequence of T_{lr} count vectors $X_1, \dots, X_{T_{lr}} \in \mathbf{Z}^M$, each of M dimensions. Each count vector X_t is a histogram of accesses in A_t over M memory bins in the t th time slice of the trace spanning ν seconds.

3.3 Learning Phase (Learning a latent variable model): The input to the learning phase is a set of sparse count vectors $X_1, \dots, X_{T_{lr}} \in \mathbf{Z}^M$, correlated within and across instances obtained from a block I/O trace as described earlier. These count vectors can often be intrinsically grouped into cohesive clusters which arise as a result of long range access patterns (see Figure 1) that repeat over time albeit with some randomness.

Hence we would like to explore unsupervised learning techniques based on clustering for these count vectors, that capture temporal dependencies between count vector instances and the correlation within instances.

Hidden Markov Models(HMM), are a natural choice of predictive models for such temporal data. In a HMM, latent variables $Z_t \in \{1, \dots, K\}$ are introduced that follow a markov chain. Each X_t is generated based on the choice of Z_t , inducing a clustering of count vectors. In the learning phase, we learn the HMM parameters, denoted by θ .

Owing to the variability of access patterns in trace data, a fixed value of K is not suitable for use in realistic scenarios motivating the use of non-parametric techniques of clustering. In section 4 we propose the HMM-DP-MMVP, a temporal model for non-parametric clustering of correlated count vector sequences capturing their full covariance structure, followed by the Sparse-HMM-DP-MMVP in section 5, that exploits the sparsity in count vectors to better model the data, also leading to more efficient inference.

As an outcome of the learning phase, we have a HMM based model with appropriate parameters, that provides predictive ability to infer the next hidden state on observing a sequence of count vectors. However, since the final prediction required is that of memory accesses, we maintain a map from every value of hidden state k to the set of all raw access requests from various time slices during training that were assigned latent state k . $H(k) = \cup_{\{t|Z_t=k\}} A_t$, for $\forall k$.

3.4 The Operating Phase (Prediction for Preloading): Having observed $\{X_1, \dots, X_{T_r}\}$ aggregated from \mathcal{D}^{lr} , the learning phase learns a latent variable model. In the Operating Phase, as we keep observing \mathcal{D}^{op} , after the time interval t' , the data is incrementally aggregated into a sequence $\{X'_1, \dots, X'_{t'}\}$. At this point, we would like the model to predict the best possible choice of blocks to load into cache for interval $t' + 1$ with knowledge of aggregated data $\{X'_1, \dots, X'_{t'}\}$.

This prediction happens in two steps. In the first step, our HMM based model Sparse-HMM-DP-MMVP infers hidden state $Z'_{t'+1}$ from observations $\{X'_1, \dots, X'_{t'}\}$, using a Viterbi style algorithm as follows. (3.1)

$$(X'_{t'+1}, \{Z'_r\}_{r=1}^{t'+1}) = \underset{(X'_{t'+1}, \{Z'_r\}_{r=1}^{t'+1})}{\operatorname{argmax}} p(\{X'_r\}_{r=1}^{t'+1}, \{Z'_r\}_{r=1}^{t'+1} | \theta)$$

Note the slight deviation from usual Viterbi method as $X'_{t'+1}$ is not yet observed. We also note that alternate strategies based on MCMC might be possible based on Bayesian techniques to infer the hidden state $Z'_{t'+1}$. However, *in the operating phase, the execution time becomes important* and is required to be much smaller than ν , the slice length. Hence we explore such a Viterbi

based technique, that is quite efficient and runs in a very small fraction of ν in practice for each prediction. The algorithm is detailed in the supplementary material.

In the second step, having predicted the hidden state $Z'_{t'+1}$, we would now like to load the appropriate accesses. Our prediction scheme consists of loading all accesses defined by $H(Z'_{t'+1})$ into the cache (with H as defined previously).

4 Mixture Models with Multivariate Poisson for correlated count vector sequences

We now describe non-parametric temporal models for correlated count vectors based on the MVP [9] mixtures. MVP [9] distributions are natural models for understanding multi-dimensional count data. There has been no work on exploring DP-based mixture models for count data or for modeling their temporal dependencies.

Hence, we first parallel the development of non-parametric MVP mixtures along the lines of DP based multinomial mixtures [16]. To this end we propose DP-MMVP, a DP based MVP mixture and propose a temporal extension HMM-DP-MMVP along the lines of HDP-HMM[16] for multinomial mixtures.

However a more interesting challenge lies in designing algorithms of scalable complexity for high dimensional correlated count vectors. We address this in our next section(5) by introducing the Sparse-MVP that exploits sparsity in data. DP mixtures of MVP and their sparse counterparts lead to different inference challenges addressed in section 6.

4.1 Preliminaries: We first recall some definitions. A probability distribution $G \sim DP(\alpha, H)$, when $G = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}$, $\beta \sim GEM(\alpha)$, $\theta_k \sim H$, $k = 1 \dots$ where H is a diffused measure. Probability measures G_1, \dots, G_J follow **Hierarchical Dirichlet process**(HDP)[16] if

$$G_j \sim DP(\alpha, G_0), j = 1 \dots J \text{ where } G_0 \sim DP(\alpha, H)$$

HMMs are popular models for temporal data. However, for most applications there are no clear guidelines for fixing the number of HMM states. A DP based HMM model, **HDP-HMM**, [16] alleviates this need. Let X_1, \dots, X_T be observed data instances. Further, for any $L \in \mathbb{Z}$, we introduce notation $[L] = \{1, \dots, L\}$. The HDP-HMM is defined as follows. $\beta \sim GEM(\gamma)$

$$\pi_k | \beta, \alpha_k \sim DP(\alpha_k, \beta), \text{ and } \theta_k | H \sim H, k = 1, 2, \dots$$

$$Z_t | Z_{t-1}, \pi \sim \pi_{Z_{t-1}}, \text{ and } X_t | Z_t \sim f_{Z_t}(\theta_k), t \in [T]$$

Commonly used base distributions for H are the multivariate Gaussian and multinomial distributions. There has been no work in exploring correlated count vector emissions. In our setting, we explore MVP emissions with H being an appropriate prior for parameter θ_k of the MVP distribution.

The Multivariate Poisson(MVP): Let $\bar{a}, \bar{b} > 0$. A random vector, $X \in \mathbf{Z}^M$ is Multivariate Poisson(MVP) distributed, denoted by $X \sim MVP(\Lambda)$, if

$$X = Y1_M \text{ Alternately, } X_j = \sum_{l=1}^M Y_{jl}, \forall j \in [M]$$

$$\text{where } \forall j \leq l \in [M], \lambda_{l,j} = \lambda_{j,l} \sim \text{Gamma}(\bar{a}, \bar{b})$$

$$(4.2) \quad Y_{j,l} \sim \text{Poisson}(\lambda_{j,l})$$

and 1_M is a M dimensional vector of all 1s. It is useful to note that $E(X) = \Lambda 1_M$, where Λ is an $M \times M$ symmetric matrix with entries $\lambda_{j,l}$ and $\text{Cov}(X_j, X_l) = \lambda_{j,l}$. Setting $\lambda_{j,l} = 0, j \neq l$ yields $X_i = Y_{i,i}$ which we refer to as the **Independent Poisson (IP)** model as $Y_{i,i}$ for each dimension i are independently Poisson distributed.

4.2 DP Mixture of Multivariate Poisson (DP-MMVP): In this section we define DP-MMVP, a DP based non-parametric mixture model for clustering correlated count vectors. We propose to use a DP based prior, $G \sim DP(\alpha, H)$, where H is a suitably chosen Gamma conjugate prior for the parameters of MVP, $\Lambda = \{\Lambda_k : k = 1, \dots\}$, k being cluster identifier. We define DP-MMVP as follows.

$$\lambda_{kjl} \sim \text{Gamma}(\bar{a}, \bar{b}), \forall j \leq l \in [M], k = 1, \dots$$

$$\beta \sim GEM(\alpha) \text{ and } G = \sum_{k=1}^{\infty} \beta_k \delta_{\Lambda_k}$$

$$Z_t | \beta \sim \text{Mult}(\beta) \forall t \in [T]$$

$$(4.3) \quad X_t | Z_t \sim MVP(\Lambda_{Z_t}), \forall t \in [T]$$

where T is the number of observations and $(\Lambda_k)_{jl} = (\Lambda_k)_{lj} = \lambda_{kjl}$. We also note that the **DP Mixture of Independent Poisson (DP-MIP)** can be similarly defined by restricting $\lambda_{k,j,l} = 0, \forall j \neq l, k = 1, \dots$

4.3 Temporal DP Mixture of MVP (HMM-DP-MMVP): DP-MMVP model does not capture temporal correlations that are useful for prediction problem of cache preloading. To this end we propose HMM-DP-MMVP, a temporal extension of the previous model, as follows. Let $X_t \in \mathbf{Z}^M, t \in [T]$ be a temporal sequence of correlated count vectors.

$$\lambda_{kjl} \sim \text{Gamma}(\bar{a}, \bar{b}) \forall j \leq l \in [M], k = 1, \dots$$

$$\beta \sim GEM(\gamma) \quad \pi_k | \beta, \alpha_k \sim DP(\alpha_k, \beta) \forall k = 1, \dots$$

$$Z_t | Z_{t-1}, \pi \sim \pi_{Z_{t-1}}, \forall t \in [T]$$

$$(4.4) \quad X_t | Z_t \sim MVP(\Lambda_{Z_t}), \forall t \in [T]$$

The HMM-DP-MMVP incorporates the HDP-HMM structure into DP-MMVP in equation (4.3). This model can again be restricted to the special case of diagonal covariance MVP giving rise to the HMM-DP-MIP by extending the DP-MIP model. The HMM-DP-MIP models the temporal dependence, but not the spatial correlation coming from trace data.

5 Modeling with Sparse Multivariate Poisson:

We now introduce the *Sparse Multivariate Poisson (SMVP)*. Full covariance MVP, defined with $\binom{M}{2}$ latent variables (in Y) is computationally expensive during inference for higher dimensions. However, vectors X_t emanating from traces are often very sparse with only a few significant components and most components close to 0. While there has been work on sparse multinomial[19] and sparse Gaussian[7] mixtures, there has been no work on sparse MVP Mixtures. We propose the SMVP by extending the MVP to model sparse count vectors. We then extend this to Sparse-DP-MMVP for a non-parametric mixture setting and finally propose the temporal extension Sparse-HMM-DP-MMVP.

5.1 Sparse Multivariate Poisson distribution (SMVP): We introduce the SMVP as follows. Consider an indicator vector $b \in \{0, 1\}^M$, that denotes whether a dimension is active or not. Let $\hat{\lambda}_j \geq 0, \forall j \in [M]$ and $b \in \{0, 1\}^M$. We define $X \sim SMVP(\Lambda, \hat{\lambda}, b)$ as: $X = Y1_M$ where $\forall j \leq l \in [M]$

$$(5.5) \quad Y_{j,l} \sim \text{Poisson}(\lambda_{j,l}) b_j b_l + \text{Poisson}(\hat{\lambda}_j) (1 - b_j) \delta(j, l)$$

where Λ is a symmetric positive matrix with $(\Lambda)_{jl} = \lambda_{jl}$. If $b_j = 1, b_l = 1$ then $Y_{j,l}$ is distributed as $\text{Poisson}(\lambda_{j,l})$. However if $b_j = 0$, variables $Y_{j,j}$ are distributed as $\text{Poisson}(\hat{\lambda}_j)$. The selection variables b_j decide if the j th dimension is active. Otherwise we consider any emission at the j th dimension noise, modulated by $\text{Poisson}(\hat{\lambda}_j)$, independent of other dimensions. Parameter $\hat{\lambda}_j$ is close to zero for the extraneous noise dimensions and is common across clusters.

With Sparse-MVP, we are defining a full covariance MVP for a subset of dimensions while the rest of the dimensions are inactive and hence modeled independantly (with a small mean to account for noise). The full covariance MVP is a special case of Sparse-MVP where all dimensions are active.

5.2 DP Mixture of Sparse Multivariate Poisson: In this section, we propose the **Sparse-DP-MMVP**, extending our DP-MMVP model. For every mixture component k we introduce an indicator vector $b_k \in \{0, 1\}^M$. Hence, $b_{k,j}$ denotes whether a dimension j is active for mixture component k .

A natural prior for selection variables, $b_{k,j}$, is Bernoulli Distribution, while a Beta distribution is a natural conjugate prior for the parameter η_j of the Bernoulli. $\eta_j \sim \text{Beta}(a', b'), b_{k,j} \sim \text{Bernoulli}(\eta_j), j \in [M], k = 1, \dots$. The priors for parameters Λ and $\hat{\lambda}$ are again decided based on conjugacy, where $\forall j \leq l \in [M]$ $\lambda_{j,l}$ have a gamma prior, Let $\hat{a}, \hat{b} > 0$. We model $\hat{\lambda}$ to have a common Gamma prior for inactive dimen-

sions over all clusters. $\hat{\lambda}_j \sim \text{Gamma}(\hat{a}, \hat{b}), \forall j \in [M]$. The Sparse-DP-MMVP is defined as:

$$\begin{aligned} \eta_j &\sim \text{Beta}(a', b'), \quad b_{k,j} \sim \text{Bernoulli}(\eta_j), j \in [M], k = 1 \dots \\ \hat{\lambda}_j &\sim \text{Gamma}(\hat{a}, \hat{b}), j \in [M] \\ \lambda_{kjl} &\sim \text{Gamma}(\bar{a}, \bar{b}), \{j \leq l \in [M] : b_{k,j} = b_{k,l} = 1\}, k = 1, \dots \\ G &= \sum_{k=1}^{\infty} \beta_k \delta_{\Lambda_k}, \beta \sim \text{GEM}(\alpha) \\ \text{Cluster selection variables } Z_t | \beta &\sim \text{Mult}(\beta) \text{ and} \\ (5.6) \quad X_t | Z_t, \Lambda, \hat{\lambda}, b_{Z_t} &\sim \text{SMVP}(\Lambda_{Z_t}, \hat{\lambda}, b_{Z_t}), t \in [T] \end{aligned}$$

DP-MMVP is a special case of Sparse-DP-MMVP where all dimensions of all clusters are active.

5.3 Temporal Sparse Multivariate Poisson Mixture: We now define **Sparse-HMM-DP-MMVP**, by extending Sparse-DP-MMVP to also capture Temporal correlation between instances by incorporating HDP-HMM into the Sparse-DP-MMVP:

$$\begin{aligned} \eta_j &\sim \text{Beta}(a', b'), \quad b_{k,j} \sim \text{Bernoulli}(\eta_j), j \in [M], k = 1 \dots \\ \hat{\lambda}_j &\sim \text{Gamma}(\hat{a}, \hat{b}), j \in [M] \\ \lambda_{kjl} &\sim \text{Gamma}(\bar{a}, \bar{b}), \{j \leq l \in [M] : b_{k,j} = b_{k,l} = 1\}, k = 1, \dots \\ \beta &\sim \text{GEM}(\gamma) \text{ and } \pi_k | \beta, \alpha_k \sim \text{DP}(\alpha_k, \beta), k = 1, \dots \\ Z_t | Z_{t-1}, \pi &\sim \pi_{Z_{t-1}}, t \in [T] \\ (5.7) \quad X_t | Z_t, \Lambda_{Z_t}, \hat{\lambda}, b_{Z_t} &\sim \text{SMVP}(\Lambda_{Z_t}, \hat{\lambda}, b_{Z_t}), t \in [T] \end{aligned}$$

The plate diagram for the Sparse-HMM-DP-MMVP model is shown in figure 3. We again note that HMM-

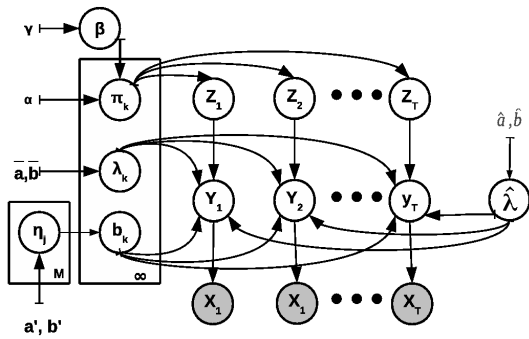


Figure 3: Plate Diagram : Sparse-HMM-DP-MMVP

DP-MMVP model described in section 4.3 is a restricted form of Sparse-HMM-DP-MMVP where $b_{k,j}$ is fixed to 1. The Sparse-HMM-DP-MMVP captures the spatial correlation inherent in the trace data and the long range temporal dependencies, at the same time exploiting sparseness, reducing the number of latent variables.

6 Inference

While inference for non-parametric HMMs is well explored [16][4], MVP and Sparse-MVP emissions introduce additional challenges due to the latent variables involved in the definition of the MVP and the introduction of sparsity in a DP mixture setting for the MVP.

We discuss the inference of DP-MMVP in detail in the supplementary material. In this section, we give a brief overview of collapsed Gibbs sampling inference for HMM-DP-MMVP and Sparse-HMM-DP-MMVP. More details are again in the supplementary material.

Throughout this section, we use the following notation: $Y = \{Y_t : t \in [T]\}$, $Z = \{Z_t : t \in [T]\}$ and $X = \{X_t : t \in [T]\}$. A set with a subscript starting with a hyphen(-) indicates the set of all elements except the index following the hyphen. The latent variables to be sampled are $Z_t, t \in [T]$, $Y_{t,j,l}, j \leq l \in [M], t \in [T]$ and $b_{k,j}, j \in [M], k = 1, \dots$ (For the sparse model). Further we have $\beta = \{\beta_1, \dots, \beta_K, \beta_{K+1} = \sum_{r=K+1}^{\infty} \beta_r\}$. and an auxiliary variable m_k is introduced as a latent variable to aid the sampling of β based on the direct sampling procedure from HDP[16]. Updates for m and β are similar to [4], as detailed in algorithm 1. The latent variables Λ and π are collapsed.

Sampling $Z_t, t \in [T]$ While the update for this variable is similar to that in [16][4], the likelihood term $p(Y|Z_t = k, Z_{-t}, Y_{-t}, \mathbf{b}^{\text{old}}, \bar{a}, \bar{b})$ differs due to the MVP based emissions. For the HMM-DP-MMVP, this term can be evaluated by integrating out the λ s. Similarly for Sparse-HMM-DP-MMVP when k is an existing component (for which b_k is known).

However, for the Sparse-HMM-DP-MMVP, an additional complication arises for the case of a new component, since we do not know the b_{K+1} value, requiring summing over all possibilities of b_{K+1} , leading to exponential complexity. Hence, we evaluate this numerically (see supplementary material for details). This process is summarized in algorithm 1.

Sampling $Y_{t,j,l}, j \leq l \in [M], t \in [T]$: The $Y_{t,j,l}$ latent variables in the MVP definition, differentiate the inference procedure of an MVP mixture from standard inference for DP mixtures. Further, the large number of $Y_{t,j,l}$ variables $\binom{n}{2}$ also leads to computationally expensive inference for higher dimensions motivating sparse modeling. In, Sparse-HMM-DP-MMVP, only those $Y_{t,j,l}$ values are updated for which $b_{Z_t,j} = b_{Z_t,l} = 1$.

We have, for each dimension j , $X_{t,j} = \sum_{l=1}^M Y_{t,j,l}$. To preserve this constraint, suppose for row j , we sample $Y_{t,j,l}, j \neq l$, $Y_{t,j,j}$ becomes a derived quantity as $Y_{t,j,j} = X_{t,j} - \sum_{p=1, p \neq j}^M Y_{t,p,j}$. We also note that, updating the value of $Y_{t,j,l}$ impacts the value of only two other random variables i.e $Y_{t,j,j}$ and $Y_{t,l,l}$. The final update for $Y_{t,j,l}, j \neq l$ can be obtained by integrating out Λ .

(full expression in alg 1, more details: Appendix B).

Algorithm 1: Inference: Sparse-HMM-DP-MMVP
 Inference steps(The steps for HMM-DP-MMVP are similar and are shown as alternate updates in brackets)

Repeat until convergence
for $t = 1, \dots, T$ **do**
 // Sample Z_t from

$$p(Z_t = k | Z_{-t, -(t+1)}, z_{t+1} = l, \mathbf{b}^{\text{old}}, X, \beta, Y; \alpha, \bar{a}, \bar{b})$$

$$\propto p(Z_t = k, Z_{-t, -(t+1)}, t+1 = l | \beta; \alpha, \bar{a}, \bar{b})$$

$$p(Y | Z_t = k, Z_{-t}, Y_{-t}, \mathbf{b}^{\text{old}}; \bar{a}, \bar{b})$$
 // Case 1: For HMM-DP-MMVP (with $b_{k,j} = 1 \forall j, \forall k$)
 // and Sparse-HMM-DP-MMVP For existing k

$$p(Y | Z_t = k, Z_{-t}, Y_{-t}, \mathbf{b}^{\text{old}}; \bar{a}, \bar{b}) \propto \prod_{j \in [M]} F_{k,j,l}^{b_{k,j} b_{k,l}} \prod_{j \in [M]} \hat{F}_j$$

$$\text{Where } F_{k,j,l} = \frac{\Gamma(\bar{a} + S_{k,j,l})}{(\bar{b} + n_k)^{(\bar{a} + S_{k,j,l})} \prod_{\bar{t}: Z_{\bar{t}}=k} Y_{\bar{t},j,l}!}$$

$$\text{and } \hat{F}_j = \frac{\Gamma(\hat{a} + \hat{S}_j)}{(\hat{b} + \hat{n}_j)^{(\hat{a} + \hat{S}_j)} \prod_{t,j: b_{Z_t,j}=1} Y_{t,j,j}!}$$
 , With $\hat{S}_j = \sum_t Y_{t,j,j} (1 - b_{Z_t,j})$, $\hat{n}_j = \sum_t (1 - b_{Z_t,j})$ and $S_{k,j,l} = \sum_{\bar{t}} Y_{\bar{t},j,l} \delta(Z_{\bar{t}}, k)$, for $j \leq l \in [M]$
 // Case 2: For Sparse-HMM-DP-MMVP for new k ,
 // compute following numerically, where $\mathbf{b}^{\text{old}} = \{b_1, \dots, b_K\}$

$$p(Y | \mathbf{b}^{\text{old}}, Z_t = K+1, Z_{-t}, Y_{-t}; \bar{a}, \bar{b}) = \sum_{b_{K+1}} p(b_{K+1} | \mathbf{b}^{\text{old}}, \eta)$$

$$p(Y | \mathbf{b}^{\text{old}}, b_{K+1}, Z_t = K+1, Z_{-t}, Y_{-t}; \bar{a}, \bar{b})$$
for $j \leq l \in [M]$ **do**
if $b_{Z_t,j} = b_{Z_t,k} = 1$ **then**
 // Sample $Y_{t,j,l}$ from

$$p(Y_{t,j,l} | Y_{-t,j,l}, Z, \bar{a}, \bar{b}) \propto F_{k,j,l} F_{k,j,j} F_{k,l,l}$$
 Set $Y_{t,j,j} = X_{t,j} - \sum_{j=1}^M Y_{t,j,j}$
 Set $Y_{t,l,l} = X_{t,l} - \sum_{l=1}^M Y_{t,l,l}$
for $j = 1, \dots, M, k = 1, \dots, K$ **do**
 // Sample $b_{k,j}$ (for Sparse-HMM-DP-MMVP) from

$$p(b_{k,j} | b_{-k,j}, Y, Z) \sim p(b_{k,j} | b_{-k,j}) p(Y | b_{k,j}, b_{-k,j}, Z; \bar{a}, \bar{b})$$

$$p(b_{k,j} | b_{-k,j}) \propto \frac{c_j^{-k} + b_{k,j} + a' - 1}{K + a' + b' - 1}$$
 where $c_j^{-k} = \sum_{k \neq j, \bar{k}=1}^K b_{k,j}$ is the number of clusters (excluding k) with dimension j active
for $k = K, \dots, M, k = 1, \dots, K$ **do**
 $m_k = 0$
for $i = 1, \dots, n_k$ **do**
 $u \sim \text{Ber}(\frac{\alpha \beta_k}{i + \alpha \beta_k})$, if $(u == 1) m_k + +$
 $[\beta_1 \beta_2 \dots \beta_K \beta_{K+1}] | m, \gamma \sim \text{Dir}(m_1, \dots, m_K, \gamma)$

Update for $b_{k,j}$: For Sparse-HMM-DP-MMVP, the update for $b_{k,j}$ is obtained by integrating out η to evaluate $p(b_{k,j} | b_{-k,j})$ and computing the likelihood term by integrating out $\lambda, \hat{\lambda}$ as before. This is shown in algorithm 1 (see supplementary material for more details).

Train time Complexity Comparison: Sparse-HMM-DP-MMVP vs HMM-DP-MMVP: The inference procedure for both models is similar, with

different updates shown in Algorithm 1. For the HMM-DP-MMVP all dimensions are active for all clusters. We sample $\binom{M}{2}$ random variables for the symmetric matrix Y_t in this step for each $t \in [T]$. On the other hand, for Sparse-HMM-DP-MMVP with \bar{m}_k active components in cluster k , we sample only $\binom{\bar{m}}{2}$ random variables which is a significant improvement when $\bar{m} \ll M$.

7 Experimental Evaluation

We perform experiments on benchmark traces, to evaluate our models, in terms of likelihood and also evaluate their effectiveness for the caching problem, by measuring hit rates using our predictive models.

7.1 Datasets: We perform experiments on diverse enterprise workloads : 10 publicly available real world Block I/O traces (MT 1-10), commonly used benchmark in storage domain, collected at Microsoft Research Cambridge[3] and 1 NetApp internal workload (NT1). See dataset details and choice of traces in Appendix C.

We divide the available trace into two parts \mathcal{D}^{lr} that is aggregated into T_{lr} count vectors and \mathcal{D}^{op} that is aggregated into T_{op} count vectors. In our initial experimentation, for aggregation, we fix the number of memory bins $M=10$ (leading to 10 dim count vectors), length of time slice $\nu=30$ seconds. Further, we use a test train split of 50% for both experiments such that $T_{lr} = T_{op}$. (Later, we also perform some experiments to study the impact of some of these parameters with $M = 100$ dimensions on some of the traces.)

7.2 Experiments: We perform two types of experiments, to understand how well our model fits data in terms of likelihood, the next to show how our model and our framework can be used to improve cache hit rates.

7.2.1 Experiment Set 1: Likelihood Comparison: We show a likelihood comparison between HMM-DP-MMVP, Sparse-DP-MMVP and baseline model HMM-DP-MIP. We train the three models using the inference procedure detailed in section 6 on \mathbf{T}_{lr} and compute Log-likelihood on the held out test trace \mathbf{T}_{op} . The results are tabulated in Table 1.

Results: We observe that the Sparse-HMM-DP-MMVP model performs the best in terms of likelihood, while the HMM-DP-MMVP outperforms HMM-DP-MIP by a large margin. Poor performance of HMM-DP-MIP clearly shows that spatial correlation present across the M dimensions is an important aspect and validates the necessity for the use of a Multivariate Poisson model over an independence assumption between the dimensions. Superior performance of Sparse-HMM-DP-MMVP over HMM-DP-MMVP is again testimony to the fact that there exists inherent sparsity in the data and this is better modeled by the sparse model.

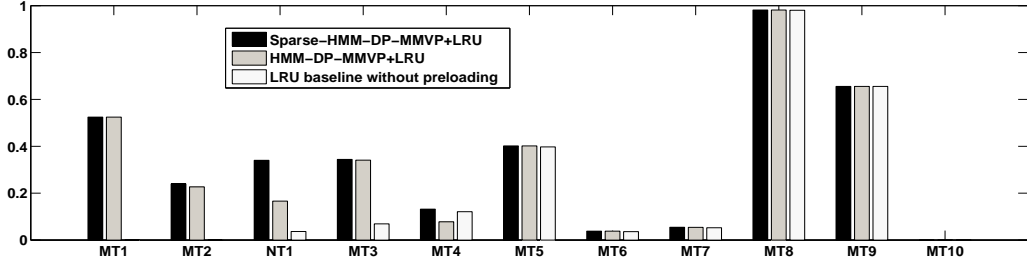


Figure 4: Comparison of Hitrates against LRU baseline without preloading : On MT1 we show the highest hitrate increase of 0.52 vs baseline 0.0002. On MT10 we show the least improvement where both baseline and our model yield 0 hitrate.

Table 1: Log-likelihood on held out data: Sparse-HMM-DP-MMVP model fits data with Best Likelihood

Trace Name	HMM-DP IP ($\times 10^6$)	HMM-DP MMVP ($\times 10^6$)	Sparse-HMM DP-MMVP ($\times 10^6$)
NT1	-24.43	-20.18	-19.35
MT1	-14.02	-8.12	-8.10
MT2	-0.46	-0.44	-0.27
MT3	-0.09	-0.08	-0.06
MT4	-1.15	-1.01	-0.93
MT5	-20.23	-12.70	-12.61
MT6	-69.12	-50.86	-50.53
MT7	-87.25	-84.97	-80.24
MT8	-2.44	-2.33	-2.03
MT9	-12.45	-12.85	-11.53
MT10	-16.49	-13.52	-12.91

7.2.2 Experiment Set 2: Hitrates: We compute hitrate, on each of the 11 traces, with a baseline simulator without preloading and an augmented simulator with the ability to preload predicted blocks every $\nu = 30s$. Both the simulators use LRU for eviction. Off the shelf simulators for preloading are not available for our purpose and construction of the baseline simulator and that with preloading are described in detail in supplementary material- Appendix C.

Results: We see in the barchart in figure 4 prediction improves hitrates over LRU baseline without preloading. We see that our augmented simulator gives order of magnitude better hitrates on certain traces (0.52 with preloading against 0.0002 with plain LRU).

Effect of Training Data Size: We expect to capture long range dependencies in access patterns when we observe a sufficient portion of the trace for training where such dependencies manifest. Ideally we would like to run our algorithm in an online setting, where periodically, all the data available is used to update the model. Our model can be easily adapted to such a situation. In this paper, however, we experiment in a setting where we always train the model on 50% (see supplementary material for an explanation of the figure 50%) of available data for each trace and use this model to make predictions for the rest of the trace.

Effect of M (aggregation granularity): To understand the impact M (count vector dimensionality),

we pick some of the best performing traces from the previous experiment (barchart in figure 4) and repeat our experiment with $M=100$ features, a finer 100 bin memory aggregation leading to 100 dimensional count vectors. We find that we beat baseline by an even higher margin with $M=100$. We infer this could be attributed to the higher sensitivity of our algorithm to detail in traces leading to superior clustering. This experiment also brings to focus the *training time* of our algorithm. We observed that the Sparse-HMM-DP-MMVP outperforms HMM-DP-MMVP not only in terms of likelihood and hitrates but also in terms of training time. We fixed the training time to at most 4 hours to run our algorithms and report hitrates in table 2. We find that Sparse-HMM-DP-MMVP ran to convergence while HMM-DP-MMVP did not finish even a single iteration for most traces in this experiment. This corroborates our understanding that handling sparsity reduces the number of latent variables in HMM-DP-MMVP, improving inference efficiency translating to faster training time, particularly for higher dimensional count vectors.

Table 2: Hitrate after training 4 hours: $M=10, M=100$ (x indicates that inference did not complete one iteration)

Trace Name	S-HMM-DP-MMVP M=10	S-HMM-DP-MVP M=100	HMM-DP-MVP M=100	LRU without Preloading
NT1	0.340	0.592 (43 clusters)	0.34	0.0366
MT1	0.5245	0.710 (190 clusters)	x	0.0002
MT2	0.2397	0.498 (23 clusters)	x	0.0010
MT3	0.344	0.461 (53 clusters)	x	0.0366
Avg	0.362	0.565	-	0.0186

7.3 Discussion of Results: We observe both from table 1 and the barchart (fig 4) that HMM-DP-MMVP outperforms HMM-DP-MIP, and Sparse-HMM-DP-MMVP performs the best, outperforming HMM-DP-MMVP in terms of likelihood and hitrates, showing that traces indeed exhibit spatial correlation that is effectively modeled by the full covariance MVP and that handling sparsity leads to a better fit of the data.

The best results are tabulated in Table 2 where we observe that when using 100 bins Sparse-HMM-DP-

MVP model achieves an average hitrate of $h = 0.565$, 30 times improvement over LRU without preloading, $h = 0.0186$. On all the other traces, LRU without preloading is outperformed by the sparse-HMM-DP-MMVP, the improvement being dramatic for 4 of the traces. On computing average hitrate for the 11 traces in figure 4, we see 58% hitrate improvement.

Choice of Baselines: We did not consider a parametric baseline as it is clearly not suitable for our caching application. Traces have different access patterns with varying detail (leading to varying number of clusters: fig 2). A parametric model is clearly infeasible in a realistic scenario. Further, due to lack of existing predictive models for count vector sequences, we use a HMM-DP-MIP baseline for our models.

Extensions and Limitations: While we focus on capturing long range correlations, our framework can be augmented with other algorithms geared towards specific workload types for capturing short range correlations, like sequential read ahead and Sarc [5] to get even higher hitrates. We hope to investigate this in future.

We have shown that our models lead to dramatic improvement for a subset of traces and work well for the rest of our diverse set of traces. We note that there may not be discernable long range correlations present in all traces. However, we have shown, that when we can predict, the scale of its impact is huge. There are occasions, when the prediction set is larger than cache where we would have to understand the temporal order of predicted reads, to help efficiently schedule preloads. Cache size, prediction size and preload frequency, all play an important role to evaluate the full impact of our method. Incorporating our models within a full-fledged storage system involves further challenges, such as real time trace capture, smart disk scheduling algorithms for preloading, etc. These issues are beyond the scope of the paper, but form the basis for future work.

8 Conclusions

We have proposed DP-based mixture models (DP-MMVP, HMM-DP-MMVP) for correlated count vectors that capture the full covariance structure of multivariate count data. We have further explored the sparsity in our data and proposed models (Sparse-DP-MMVP and Sparse-HMM-DP-MMVP) that capture the correlation within a subset of dimensions for each cluster, also leading to more efficient inference algorithms. We have taken the first steps in outlining a preloading framework for leveraging long range dependencies in block I/O Traces to improve cache hitrates. Our algorithms achieve a 30X hitrate improvement on 4 real world traces, and outperform baselines on all traces.

References

[1] D. J. Aldous. In *École d’été de probabilités de Saint-*

Flour, XIII—1983, Lecture Notes in Math., pages 1–198. Springer, Berlin, 1985.

- [2] L. M. Dimitris Karlis. *Journal of Statistical Planning and Inference*, 2007.
- [3] A. R. Dushyanth Narayanan, A Donnelly. Write offloading: Practical power management for enterprise storage. *USENIX, FAST*, 2008.
- [4] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. A Sticky HDP-HMM with Application to Speaker Diarization. *Annals of Applied Statistics*, 2011.
- [5] B. S. Gill and D. S. Modha. Sarc: Sequential prefetching in adaptive replacement cache. In *UATC*, 2005.
- [6] M. M. Gokul Soundararajan and C. Amza. Context-aware prefetching at the storage server. In *USENIX ATC*. USENIX, 2008.
- [7] A. K. Jain, M. Law, and M. Figueiredo. Feature Selection in Mixture-Based Clustering. In *NIPS*, 2002.
- [8] D. Karlis and L. Meligkotsidou. Finite mixtures of multivariate poisson distributions with application. *Journal of Statistical Planning and Inference*, 137(6):1942–1960, June 2007.
- [9] K. Kawamura. The structure of multivariate poisson distribution. *Kodai Mathematical Journal*, 1979.
- [10] T. M. Kroeger and D. D. E. Long. Predicting file system actions from prior events. In *UATC*, 1996.
- [11] M. H. C. Law, A. K. Jain, and M. A. T. Figueiredo. Feature selection in mixture-based clustering. In *NIPS*, pages 625–632, 2002.
- [12] C. Z. S. S. M. Li, Z. and Y. Zhou. Mining block correlations in storage systems. In *FAST*. USENIX, 2004.
- [13] L. Meligkotsidou. Bayesian multivariate poisson mixtures with an unknown number of components. *Statistics and Computing*, 17, Iss 2, pp 93-107, 2007.
- [14] A. M. Schmidt and M. A. Rodriguez. Modelling multivariate counts varying continuously in space. In *Bayesian Statistics Vol 9*. 2010.
- [15] Y. W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, 2010.
- [16] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of American Statistical Association*, 2004.
- [17] P. Tsiamyrtzis and D. Karlis. Strategies for efficient computation of multivariate poisson probabilities. *Communications in Statistics (Simulation and Computation)*, Vol. 33, No. 2, pp. 271292, 2004.
- [18] B. D. M. Wang, Chong. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In *NIPS*.
- [19] C. Wang and D. Blei. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In *NIPS*. 2009.
- [20] Y. Zhang, G. Soundararajan, M. W. Storer, L. N. Bairavasundaram, S. Subbiah, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. Warming up storage-level caches with bonfire. In *USENIX Conference on File and Storage Technologies*. USENIX, 2013.