Learning Set Cardinality in Distance Nearest Neighbours

Christos Anagnostopoulos
School of Computing Science, University of Glasgow
G12 8QQ, Glasgow, UK
Email: christos.anagnostopoulos@glasgow.ac.uk

Peter Triantafillou
School of Computing Science, University of Glasgow
G12 8QQ, Glasgow, UK
Email: peter.triantafillou@glasgow.ac.uk

***Abstract*—Distance-based nearest neighbours (*d*NN) queries and aggregations over their answer sets are important for exploratory data analytics. We focus on the Set Cardinality Prediction (SCP) problem for the answer set of *d*NN queries. We contribute a novel, query-driven perspective for this problem, whereby answers to previous *d*NN queries are used to learn the answers to incoming *d*NN queries. The proposed novel machine learning (ML) model learns the dynamically changing query patterns space and thus it can focus only on the portion of the data being queried. The model enjoys several comparative advantages in prediction error and space requirements. This is in addition to being applicable in environments with sensitive data and/or environments where data accesses are too costly to execute, where the data-centric state-of-the-art is inapplicable and/or too costly. A comprehensive performance evaluation of our model is conducted, evaluating its comparative advantages versus acclaimed methods (i.e., different self-tuning histograms, sampling, multidimensional histograms, and the power-method).**

***Keywords*-Query-driven set cardinality prediction; distance nearest neighbors analytics; hetero-associative competitive learning; local regression vector quantization.**

I. Introduction

Given a multi-dimensional (m-d) data space, analysts often wish to provide a focal data point in the space and a radius around the focal point, determining a subspace of interest. *d*NN queries do exactly this. They return the subset of data points (answer set) from a data space, whose distance from the query (focal) point are within a specified distance threshold (radius). Such queries are common in exploratory analytics, and applications like spatial data management GIS, CAD, bioinformatics, etc. Frequently, analysts are in search of answers to aggregation operators over such *d*NN subspaces. Imagine exploratory analytics based on a stream of such aggregation operators over *d*NN subspaces being issued, until the analyst locates the exact subspace of interest. Note that the analyst issuing these *d*NN subspace aggregation queries does not require to know the set of actual data points in each subspace, but just the aggregate (at least not until the final subspace of interest is found). The answer set cardinality prediction (SCP) of *d*NN queries is a typical aggregation operator. Hence, *d*NN SCP is important for query-driven data exploration. In addition, *d*NN SCP amounts to estimating the selectivity size of *d*NN queries, which is in its own right important for *d*NN query optimization. The selectivity of a *d*NN query is just the fraction of the answer set cardinality out of the total number of points in the dataset. Efficient optimization of complex queries relies heavily on accurate SCP, i.e., prediction of the cardinality of the intermediate answer set of a *d*NN query drives the evaluation of different execution plans, deciding if /which indices to use, the order to perform operations, etc.

Fundamentally, this work represents a drastic departure from the state-of-the-art methods, which are data-driven in the sense that they require to access the raw data and construct structures and synopses (e.g., histograms and samples) which will be used to answer *d*NN SCP. Data-driven approaches, in general, bear the high costs for constructing and maintaining synopses and related structures. Furthermore, the state-of-the-art approaches, as we shall qualitatively and quantitatively elaborate shortly, are inefficient and error-prone when used to handle *d*NN SCP queries. Finally, and perhaps more importantly, data-driven approaches for *d*NN SCP are not applicable in environments with sensitive data, prohibiting the access scans over raw-data and knowledge of raw-data updates required to build and maintain such structures. Similarly, in situations where accesses to raw data are costly money-wise (e.g., when datasets are maintained in the cloud) it is highly desirable to accurately solve the *d*NN SCP problem based only on the answers to a small set of previous *d*NN SCP queries. Enter our work, which offers a machine learning (ML) model for *d*NN SCP, based on answers of previously executed *d*NN SCP queries.

II. Related Work & Contribution

Consider a set $\mathcal{B}$ of d-dim. real-valued data points $\mathbf{x} = [x_1, \ldots, x_d] \in \mathbb{R}^d$. The result of any *d*NN query depends on the underlying distance metric (norm). The most widely used metric is the p-norm (L_p).

Definition 1: The p-norm (L_p) distance between two points $\mathbf{x}$ and $\mathbf{x}'$ from $\mathbb{R}^d$ for $1 \leq p < \infty$, is $\|\mathbf{x} - \mathbf{x}'\|_p = (\sum_{i=1}^{d} |x_i - x'_i|^p)^{\frac{1}{p}}$ and for $p = \infty$, is $\|\mathbf{x} - \mathbf{x}'\|_\infty = \max_{i=1,\ldots,d}\{|x_i - x'_i|\}$.

Definition 2: Given $\mathbf{x} \in \mathbb{R}^d$ and $\theta \in \mathbb{R}$, $\theta > 0$, a *d*NN query is represented by the vector $\mathbf{q} = (\mathbf{x}, \theta)$, which retrieves all points $\mathbf{x}' \in \mathcal{B}$ such that $\|\mathbf{x} - \mathbf{x}'\|_p \leq \theta$.

Definition 3: Given a *d*NN query $\mathbf{q} = (\mathbf{x}, \theta)$, $y \in \mathbb{N}$ is the cardinality of the answer set: $|\{\mathbf{x}' \in \mathcal{B} : \|\mathbf{x} - \mathbf{x}'\|_p \leq \theta\}|$.

Given a data (sub)space of $\mathbb{R}^d$ most approaches for predicting the cardinality of the answer set use some form of m-d histograms. Histograms partition the data space into buckets by inspecting the (possibly huge) underlying set $\mathcal{B}$ and then estimate the probability density function (pdf) $p(\mathbf{x})$. In histograms, the estimation of $p(\mathbf{x})$ is highly exploited for SCP limited under L_∞ like in [1], [2]. However, histograms do not scale well with big datasets. Histograms need be periodically rebuilt to incorporate updates (i.e., $p(\mathbf{x})$ is updated), increasing the overhead of this approach [9]. Further, the local uniformity assumption rarely holds in real datasets. Hence, histogram-based approaches can be both expensive and error-prone. Self-tuning histograms (STHs) [3], [12], [11] were introduced to alleviate some of the above problems. STHs estimate $p(\mathbf{x})$ from scratch; starting with no buckets and relying only on the cardinality provided by the execution of queries, referred to as query feedback records (QFR): given a query $\mathbf{q}$ with output y, STHs learn the conditional pdf $p(\mathbf{x}|y,\mathbf{q})$. Fundamentally, STHs limitations stem from the need to estimate $p(\mathbf{x}|y,\mathbf{q})$ thus having to deal with the (i) high data dimensionality, (ii) real valued domains, (iii) data variability, and (iv) the need to make assumptions of the statistical dependencies of data. Other histogram-based SCP methods utilize wavelets [4], singular value decomposition [5], value transformations [6], and entropy-based [8]. Overall, STHs and the other advanced histogram-based approaches, are associated with the above-mentioned problems of building and maintenance costs and of lack of support for evaluating the volume of the hypersphere of radius θ with center $\mathbf{x}$ under L_p, $p \neq \infty$. There are SCP methods for *d*NN queries under any L_p. In [19] methods use the $\mathcal{B}$ set's fractal dimensions relying on the assumption that the density of the number of points given a point $\mathbf{x}$ follows the power-law distribution with radius θ. Sampling methods [7] have been also proposed for SCP. They share the common idea to evaluate the query over a small subset of $\mathcal{B}$ and extrapolate the observed cardinality. In conclusion, approaches from related work are data-centric, since they require explicit access of the data to construct and maintain their structures, which is not applicable to our problem, where any knowledge on the underlying data (e.g., $p(\mathbf{x})$) is not available for private and cost-execution reasons.

Contribution. One should rest on a definitely query-driven approach to deal with our case. The first central point is that we dynamically learn the query patterns $p(\mathbf{q})$. The second central point is that we extract knowledge on how a query $\mathbf{q}$ and its answer set cardinality (output) y are associated by incrementally learning the association $\mathbf{q} \rightarrow y$. To this end, we introduce a novel Machine Learning (ML) model that learns the unknown $p(\mathbf{q})$ and, in parallel, the unknown $p(y|\mathbf{q})$ for *d*NN queries. The third central point is that the $\mathbf{q} \rightarrow y$ association is learned without relying on the underlying data distribution $p(\mathbf{x})$, which is considered totally unknown/unavailable/inaccessible. Then, we predict the answer set cardinality $\hat{y}$ of an *unseen d*NN query without actually executing it. Our model swiftly adapts and learns on-the-fly *new* query patterns.

Our query-driven model is based only on the pairs $(\mathbf{q}, y)$ and given unseen query $\mathbf{q}$ returns a $\hat{y}$ that is close to the actual y. In learning phase, our model operates in two dimensions. First, it adaptively quantizes the query pattern space to discover prototypes of query patterns. This is achieved by proposing a *conditional* unsupervised competitive learning (UCL) scheme. Second, the model locally associates prototype queries with their output cardinality. This is achieved by a novel local hetero-associative competitive learning (HCL) scheme based on stochastic gradient descent. In prediction phase, upon an unseen query, the model projects it onto a prototype and through local quantized regression predicts its cardinality. The major contributions are:

- An novel incremental UCL ML model for quantizing the *d*NN query pattern space under any L_p;
- A novel incremental HCL ML model based on *local* regression over *regions* of the *d*NN query pattern space;
- We provide comprehensive experimental results showcasing the benefits of our model vis-à-vis the data-centric approaches: GenHist histogram [1], ISOMER [11], a learning framework for STHs [13] under L_∞, the power-method in [19] under any L_p and a sampling method for SCP.

III. The *d*NN Set Cardinality Learning Model

Overview. In the remainder, we assume normalized points, i.e., the domain of each dimension x_k is scaled to $[0,1]$, and scalar $\theta \in [0,1]$, thus the query space is $\mathbf{q} \in \mathbb{Q} \subseteq [0,1]^d \times [0,1]$. Consider a ML model $\mathcal{M}$ that estimates the SCP function $\mathcal{F} : \mathbb{Q} \rightarrow \mathbb{N}$ given training pairs $(\mathbf{q}, y)$ drawn from the unknown $p(\mathbf{q}, y)$, i.e., $y = \mathcal{F}(\mathbf{q})$. Two queries $\mathbf{q}, \mathbf{q}' \in \mathbb{Q}$ are similar if their corresponding points $\mathbf{x}, \mathbf{x}'$ and scalar θ, θ' are equally-weighted close under L_p and L_1, respectively.

Definition 4: The normalized $L_{(p,1)}$ distance between queries $\mathbf{q} = (\mathbf{x}, \theta)$ and $\mathbf{q}' = (\mathbf{x}', \theta')$ is $\|\mathbf{q} - \mathbf{q}'\|_{(p,1)} = \frac{1}{2}\left(d^{-\frac{1}{p}}\|\mathbf{x} - \mathbf{x}'\|_p + |\theta - \theta'|\right)$, with $d^{-\frac{1}{p}}$ be a normalization factor since $0 \leq \|\mathbf{x} - \mathbf{x}'\|_p \leq d^{\frac{1}{p}}$.

Definition 5: Given model $\mathcal{M}$ and query $\mathbf{q}$, we define $J(y, \hat{y})$ as the absolute J_1 and square J_2 loss between estimated $\hat{y}$ and actual y: $J_1(y, \hat{y}) = |y - \hat{y}|$, $J_2(y, \hat{y}) = (y - \hat{y})^2$.

The proposed model $\mathcal{M}$ learns the unknown $p(\mathbf{q})$ and unknown $p(\mathbf{q}, y)$ through UCL and HCL. The model learns in an on-line fashion by performing two parallel stochastic learning tasks: (i) quantization of the query space $\mathbb{Q}$, i.e., estimation of $p(\mathbf{q})$, by incrementally adapting query prototypes; (ii) on-line quantized regression (OQR) over $\mathbb{Y} \subseteq \mathbb{N}$, i.e., estimation of $p(\mathbf{q}, y)$. The overall idea is to partition the query space $\mathbb{Q} \equiv \cup_{i=1}^{N} \mathbb{Q}_i$ into finite regions $\mathbb{Q}_i$ by discovering their prototypes $\mathbf{q}_i$. Simultaneously, in each

region locally, we associate $\mathbb{Q}_i$ with a region $\mathbb{Y}_i \subseteq \mathbb{Y}$ and, correspondingly, identify the cardinality prototype y_i of $\mathbb{Y}_i$. The y_i prototype is used for prediction.

Conditional Unsupervised Competitive Learning. Our goal is finding the best possible approximation (in $L_{(p,1)}$) of a $\mathbb{Q}$-valued random $\mathbf{q}$ out of a set $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^{N}$ of (finite) N query prototypes. UCL considers a closest neighbor projection of $\mathbf{q}$ to a prototype $\mathbf{q}_j$, which represents the j-th partition $\mathbb{Q}_j$ satisfying $\mathbb{Q}_j \subset \{\mathbf{q} \in \mathbb{Q} : \|\mathbf{q} - \mathbf{q}_j\|_{(p,1)} = \min_{\mathbf{q}_i \in \mathcal{Q}} \|\mathbf{q} - \mathbf{q}_i\|_{(p,1)}\}$. In our case, the number of prototypes $N > 0$ is completely unknown and not necessarily constant. Prototype $\mathbf{q}_j$ of partition $\mathbb{Q}_j$ being the closest to query $\mathbf{q}$ is the *winning* prototype of $\mathbf{q}$, i.e., $j = \arg\min_{\mathbf{q}_i \in \mathcal{Q}} \|\mathbf{q} - \mathbf{q}_i\|_{(p,1)}$. The expected quantization error is: $\mathcal{E}_1(\mathbf{q}_1, \ldots, \mathbf{q}_N) = \mathbb{E}\left[\min_{\mathbf{q}_i \in \mathcal{Q}} \|\mathbf{q} - \mathbf{q}_i\|_{(p,1)}\right]$. The proposed UCL incrementally minimizes $\mathcal{E}_1$ with the presence of a random query $\mathbf{q}$ by updating the wining $\mathbf{q}_j \in \mathcal{Q}$. The key problem is to decide an appropriate $N = |\mathcal{Q}|$ value. In the literature a variety of UCL methods exists however not suitable for incremental implementation, because the cardinality of $\mathcal{Q}$ (resolution of quantization) must be supplied in advance. We propose an incremental, conditional UCL (CUCL) method over $L_{(p,1)}$ (a) in which the prototypes are sequentially trained directly for incoming patterns and (b) is adaptively growing, i.e., increases N if a conditional criterion holds true. Given that N is not available a-priori, CUCL minimizes $\mathcal{E}_1$ with respect to a threshold value ρ, which determines the current number of prototypes N. CUCL initiates the $\mathbb{Q}$ partitioning with a unique (random) prototype, i.e., $N = 1$. Upon the presence of a series of pattern pairs, CUCL locally updates the winning $\mathbf{q}_j$ if the condition $\|\mathbf{q} - \mathbf{q}_j\|_{(p,1)} \leq \rho$ holds true, thus, quantizes the local region $\mathbb{Q}_j$. Otherwise, $\mathbf{q}$ is currently considered as a *new* prototype and is inserted in $\mathcal{Q}$, thus, increasing N by one. CUCL leaves the random pattern pairs to self-determine the resolution of quantization. Evidently, high ρ would result to coarse vector quantization (low resolution) while low ρ yields a fine-grained quantization. Parameter ρ is associated with the stability-plasticity dilemma also known as *vigilance* in Adaptive Resonance Theory [14]. In our case vigilance ρ represents a threshold of similarity between random patterns and prototypes, thus, guiding CUCL in determining when a new prototype should be formed. To give a physical meaning to ρ, it is expressed through a set of percentages $a_k \in (0, 1)$ and $a_\theta \in (0, 1)$ of the value ranges of each dimension x_k, $k = 1, \ldots, d$ and θ, respectively. Then $\rho = \|[a_1, \ldots, a_d]\|_p + a_\theta$ and if we let $a_k = a_\theta = a, \forall k$, then $\rho = a(d^{\frac{1}{p}} + 1)$. A high a value over high dimensional data result to a low number of prototypes and vice versa. In a 2-dim. Euclidean space (i.e., $p = 2$, $d = 2$) the region of a query prototype $\mathbf{q}_j \in [0, 1]^3$ is represented by a cylinder with radius $a\sqrt{2}$ and height $2a$. The problem for CUCL over $L_{(p,1)}$ is to determine the update rules for prototypes in $\mathcal{Q}$.

On-line Quantized Regression. OQR learns the joint $p(\mathbf{q}, y)$ by defining an associative set of cardinality prototypes $\mathcal{Y}$ given the quantized $\mathbb{Q}$ space defined by query prototypes. Although OQR requires CUCL to priorly quantize $\mathbb{Q}$, we introduce an on-line, parallel HCL scheme, in which CUCL progressively optimizes $\mathcal{E}_1$, while at the same time OQR learns the specificities of a quantized, localized regression model over the current associated domain $\mathbb{Y}$. By having quantized $\mathbb{Q}$, OQR quantizes $\mathbb{Y}$ through a set $\mathcal{Y} = \{y_1, \ldots, y_N\}$ of prototypes y_i, each one corresponding to (query) prototype $\mathbf{q}_i \in \mathcal{Q}$. OQR learns localized regression models over each region $\mathbb{Y}_i$ of prototype y_i, thus, providing a framework for SCP. Consider a random pair $(\mathbf{q}, y)$ and the corresponding winning $\mathbf{q}_j$. The associated y is used to implicitly quantize the corresponding $\mathbb{Y}_j$ by updating the associated prototype $y_j \in \mathcal{Y}$ (corresponding to $\mathbf{q}_j$). Nonetheless, the adaptation of y_j is not a mere quantization of $\mathbb{Y}_j$, but also refers to an incrementally localized learning of a regression model over the cardinalities reside in $\mathbb{Y}_j$. To this end, the quantization and regression error is expressed through $J(y, \mathcal{F}(\mathbf{q}, y_j))$, i.e., loss between actual y and localized predicted $\hat{y} = \mathcal{F}(\mathbf{q}, y_j)$. Observe that $\mathcal{F}(\mathbf{q}, y_j)$ depends not only on $\mathbf{q}$ but also on the associated prototype y_j of the winning $\mathbf{q}_j$. This provides us with the flexibility to introduce two classes of SCP functions $\mathcal{F}$: (1) one which takes into account statistical information of the quantized $\mathbb{Y}_j$ space; in this case $\mathcal{F}(\mathbf{q}, y_j) = y_j$ with $j = \arg\min_{\mathbf{q}_i \in \mathcal{Q}} \|\mathbf{q} - \mathbf{q}_i\|_{(p,1)}$, and (2) one which takes into account the dependency between cardinality and query in the space $\mathbb{Y}_j$; in this case $\mathcal{F}(\mathbf{q}, y_j; \mathbf{w}_j)$ refers to a parametric regression function with y_j and $\mathbf{q}$ being the dependent and predictor variable, respectively, while $\mathbf{w}_j$ is the parameter of the localized regression model, still needs to be trained. OQR minimizes $\mathcal{E}_2(y_1, \ldots, y_N) = \mathbb{E}\left[J(y, \mathcal{F}(\mathbf{q}, y_j))|\mathbf{q}_j\right]$ subject to $j = \arg\min_{\mathbf{q}_i \in \mathcal{Q}} \|\mathbf{q} - \mathbf{q}_i\|_{(p,1)}$.

Learning Model. Since we deal with incrementally, parallel learning of $p(\mathbf{q})$ and $p(\mathbf{q}, y)$, both tasks (CUCL and OQR) upon the presence of a random pair $(\mathbf{q}, y)$ minimize $\mathcal{E}(\mathcal{Q}; \mathcal{Y}) = \mathcal{E}_1(\mathcal{Q}) + \mathcal{E}_2(\mathcal{Y})$. Given a series of pattern pairs, $\mathcal{M}$ estimates the unknown model parameter $\alpha = \{\mathbf{q}_1, \ldots, \mathbf{q}_N\} \cup \{y_1, \ldots, y_N\}$ with $\mathbf{q}_j \in \mathcal{Q}$ being associated with $y_j \in \mathcal{Y}$. The model progressively minimizes $\mathcal{E}$ though stochastic gradient descent.

Theorem 1: Given pattern pair $(\mathbf{q}, y)$, model $\mathcal{M}$ converges if the components $\mathbf{x}_j = [x_{j1}, \ldots, x_{jd}]$ and θ_j of winning prototype $\mathbf{q}_j$ are updated as

$$\begin{aligned}
\Delta x_{jk} &= \eta_j \frac{|x_k - x_{jk}|^{p-2}(x_k - x_{jk})}{\|\mathbf{x} - \mathbf{x}_j\|_p^{p-1}}, \text{ if } 1 \leq p < \infty, \\
\Delta x_{jk} &= \begin{cases} \eta_j \text{sgn}(x_k - x_{jk}), \text{ if } |x_k - x_{jk}| = \max\limits_{\ell=1,\ldots,d} \{|x_\ell - x_{j\ell}|\} \\ 0, \text{ otherwise.} \end{cases} \\
&\text{if } p = \infty, \text{ and} \\
\Delta\theta_j &= \eta_j \text{sgn}(\theta - \theta_j)
\end{aligned} \tag{1}$$

$k = 1, \ldots, d$, $\mathrm{sgn}(\cdot)$ is the signum function, and $\eta_j \in (0,1)$ is the adaptive learning rate of j-th prototype depending on the current update step.

Proof: The expected quantization error depends on the j-th winner prototype, i.e., $\mathcal{E}_1 = \int_{\mathcal{X}} \|\mathbf{x} - \mathbf{x}_j\|_p \mathrm{d}P(\mathcal{X}) + \int_{\mathcal{S}} |\theta - \theta_j| \mathrm{d}P(\mathcal{S})$ being taken over an infinite sequence of $\mathcal{X} = \{\mathbf{x}_j(1), \mathbf{x}_j(2), \ldots\}$ and $\mathcal{S} = \{\theta_j(1), \theta_j(2), \ldots\}$; $P(\mathcal{X})$ and $P(\mathcal{S})$ is the distribution of $\mathcal{X}$ and $\mathcal{S}$, respectively. Based on Robbins-Monro stochastic optimization [15], the stochastic sample at step t, $E_1(t) = \|\mathbf{x}(t) - \mathbf{x}_j(t)\|_p + |\theta(t) - \theta_j(t)|$ should decrease with random query $\mathbf{q}(t) = (\mathbf{x}(t), \theta(t))$ by descending in the negative direction of the gradient descent w.r.t. $\mathbf{x}_j$ and θ_j. The update rules are: $\Delta\mathbf{x}_j(t) = -\eta_j(t)\frac{\partial E_1(t)}{\partial \mathbf{x}_j(t)}$ and $\Delta\theta_j(t) = -\eta_j(t)\frac{\partial E_1(t)}{\partial \theta_j(t)}$ where $\eta_j(t)$ is a step-size hyperbolic schedule. Both partial derivatives are expressed in the closed form in (1) for each p-norm. ■

Note, $\{\eta_j(t)\}$ defines a slowly decreasing sequence of learning rates $\eta_j \in (0,1)$ satisfying $\sum_{t=0}^{\infty} \eta_j(t) = \infty$ and $\sum_{t=0}^{\infty} \eta_j^2(t) < \infty$ [15]. Convergence in Theorem 1 means that $\mathcal{M}$ estimates the optimal α; see also Theorem 3 for stability and convergence analysis. The learning phase of OQR is processed in parallel with the learning phase of CUCL. Given a $(\mathbf{q}, y)$ the corresponding y_j is updated to estimate the function $\mathcal{F}(\mathbf{q}, y_j)$.

Quantization Model (QM): Here, $\mathcal{F}(\mathbf{q}, y) = y$, i.e., the prototype y_j is used for SCP. Given an unseen $\mathbf{q}$, the predicted $\hat{y} = y_j$ corresponds to the winning $\mathbf{q}_j$ of $\mathbf{q}$. The loss $J(y, \mathcal{F}(\mathbf{q}, y_j)) = J(y, y_j)$ drives the update rule for cardinality prototypes. We provide the adaptation rules of y_j for the two loss functions J_1 and J_2. Note, other loss functions can be also adopted, e.g., the λ-insensitive loss $J(y, y_j) = \max\{|y - y_j| - \lambda, 0\}, \lambda > 0$ or 0-1 loss $J(y, y_j) = I(y \neq y_j)$ with I be the indicator function. We adopt J_1 because it is widely used for SCP in [1], [10], [11].

Theorem 2: Given pattern pair $(\mathbf{q}, y)$ and $\mathcal{F}(\mathbf{q}, y) = y$, $\mathcal{M}$ converges if the associated prototype y_j is updated as

$$\Delta y_j = \eta_j \mathrm{sgn}(y - y_j) \text{ w.r.t. } J_1 \quad (2)$$

$$\Delta y_j = \eta_j (y - y_j) \text{ w.r.t. } J_2 \quad (3)$$

Proof: The proof is omitted here due to space limit. ■

The rationale behind the update rules (1) and (2) (or (3)) is that $(\mathbf{x}_j, \theta_j)$ is moved toward pattern $(\mathbf{x}, \theta)$ to capture the changes in query distribution and, simultaneously, the cardinality prototype moves toward y to represent by a degree of η_j the new cardinality value.

Theorem 3 (Convergence & Stability):

$$P(y_j = m_j) = 1 \text{ at equilibrium w.r.t. } J_1 \quad (4)$$

$$P(y_j = \bar{y}_j) = 1 \text{ at equilibrium w.r.t. } J_2 \quad (5)$$

where m_j and $\bar{y}_j$ is the median and mean, respectively, of the partition $\mathbb{Y}_j \subseteq \mathbb{Y}$ of the representative $y_j \in \mathcal{Y}$.

Proof: We report on the proof of (4). Let y_j be a prototype corresponding to $\mathbf{q}_j$, which the latter quantizes the $\mathbb{Q}_j$. Assume the image of $\mathbb{Q}_j$ to subspace $\mathbb{Y}_j$ via the estimated function $y = \mathcal{F}(\mathbf{q})$ and consider the median m_j of $\mathbb{Y}_j$, i.e., satisfying the inequalities: $P(y \geq m_j) = P(y \leq m_j) = \frac{1}{2}$. Suppose that y_j has reached equilibrium, i.e., $\Delta y_j = 0$, which holds with probability 1. By taking the expectations of both sides assuming a zero-mean property of the noise process and replacing Δy_j with the update rule (2) we obtain

$$E[\Delta y_j] = \int_{\mathbb{Y}_j} \mathrm{sgn}(y - y_j) \mathrm{d}P(y) = P(y \geq y_j) \int_{\mathbb{Y}_j} p(y) \mathrm{d}y - P(y < y_j) \int_{\mathbb{Y}_j} p(y) \mathrm{d}y = 2P(y \geq y_j) - 1.$$

Since $\Delta y_j = 0$ thus y_j is constant, then $P(y \geq y_j) = \frac{1}{2}$, which denotes, by definition of median, that y_j (at equilibrium) converges to the median m_j of $\mathbb{Y}_j$. The proof of (5) is omitted for space limitations. ■

Linear Model (LM): Here, $\mathcal{F}(\mathbf{q}, y; \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\theta}$ with $\boldsymbol{\theta} = [1, \theta]$. Fix a point $\mathbf{x}$. We notate as *local* $y_{\mathbf{x}}(\theta)$ as the number of points in $\mathcal{B}$ with distances in L_p no greater than θ from $\mathbf{x}$ in a (hyper)sphere of radius θ. Let us focus on a region $\mathbb{Y}_i$. We claim that if $\mathbb{Y}$ is partitioned into a finite (large) number of regions $\mathbb{Y}_i$, the local cardinality $y_{\mathbf{x}_i}(\theta)$ in $\mathbb{Y}_i$ linearly depends on radius θ. That is, we approximate the relation between θ and y in $\mathbb{Y}_i$ through a local line, i.e., $\hat{y} = w_{i0} + w_{i1}\theta$ or $\hat{y} = \mathbf{w}_i^\top \boldsymbol{\theta}$ with $\boldsymbol{\theta} = [1, \theta]$ and local parameter $\mathbf{w}_i = [w_{i0}, w_{i1}]$. In this case, given an unseen $\mathbf{q} = (\mathbf{x}, \theta)$, the predicted cardinality $\hat{y} = \mathcal{F}(\mathbf{q}, y_j; \mathbf{w}_j) = \mathbf{w}_j^\top [1, \theta]$ with $j = \arg\min_{\mathbf{q}_i \in \mathcal{Q}} \|\mathbf{q} - \mathbf{q}_i\|_{(p,1)}$. The update rules refer to learning the parameter $\mathbf{w}_j$ given that $\mathbf{q}_j$ is the winner of $\mathbf{q}$.

Theorem 4: Given pattern pair $(\mathbf{q}, y)$ and $\mathcal{F}(\mathbf{q}, y; \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\theta}$, model $\mathcal{M}$ converges if the local parameter $\mathbf{w}_j$ of the associated winning prototype y_j is updated as

$$\Delta \mathbf{w}_j = \eta_j \mathrm{sgn}(y - \mathbf{w}_j^\top \boldsymbol{\theta}) \boldsymbol{\theta} \text{ w.r.t. } J_1 \quad (6)$$

$$\Delta \mathbf{w}_j = \eta_j (y - \mathbf{w}_j^\top \boldsymbol{\theta}) \boldsymbol{\theta} \text{ w.r.t. } J_2 \quad (7)$$

Proof: The proof is omitted here due to space limit. ■

Not surprisingly, (7) refers to a (local) stochastic gradient descent variant of the Least Mean Square method on $\mathbb{Y}_j$. The learning phase processes successive random pattern pairs until a termination criterion $T \leq \epsilon$, $\epsilon > 0$. T is the $L_{(p,1)}$ and L_1 norm between successive estimates of prototypes and cardinalities $T = \sum_{i=1}^{N} \left(\|\mathbf{q}_i(t) - \mathbf{q}_i(t-1)\|_{(p,1)} + |y_i(t) - y_i(t-1)| \right)$.

In prediction phase, $\mathcal{M}$ does not update any more the prototypes, thus, parameter α is left untouched. We then predict $\hat{y}$ given an unseen query $\mathbf{q} = (\mathbf{x}, \theta)$ through competition on the quantized $\mathbb{Q}$. Firstly, we find the winning $\mathbf{q}_j \in \mathcal{Q}$ under $L_{(p,1)}$; projection of $\mathbf{q}$ onto $\mathbb{Q}$. Secondly, we obtain the corresponding prototype $y_j \in \mathcal{Y}$. If we rely on QM, $\hat{y} = y_j$. If we rely on LM, $\hat{y} = w_{j0} + w_{j1}\theta$, since the local line is learned by the (θ, y) pairs in $\mathbb{Y}_j$ of those

queries projected onto $\mathbf{q}_j$. The prediction error is given by J_1 or J_2.

IV. Performance Evaluation

The model requires $O(dN)$ space to store the prototypes of $\mathcal{Q}$ and $\mathcal{Y}$. During CUCL, $\mathcal{M}$ finds the wining query prototype in $O(dN)$ time and then retrieves the associated cardinality prototype in $O(1)$. Since prototypes are updated during CUCL, the learning phase requires $O(d/\epsilon)$ [17] iterations. SCP requires $O(d \log N)$ assuming an one-nearest neighbor search for the wining prototype using a d-dim. tree structure over $\mathcal{Q}$. The update given a pair requires also $O(d \log N)$ time.

We will show that by extracting significant knowledge from the pairs $(\mathbf{q}, y)$ without relying on the underlying data, our approach achieves similar or even better prediction results than data-driven approaches. We study the performance of $\mathcal{M}$ over real datasets on SCP accuracy, required training patterns, and storage. We provide a comparative assessment with data-centric approaches (although inappropriate to our problem): (i) GenHist histogram [1], (ii) learning framework for STH [13], (iii) ISOMER STH [11], (iv) the power-method (PM) in [19], and (v) a random sampling method using reservoir sampling [20]. The relative percentage prediction error e is defined as $e(y, \hat{y}) = \frac{J_1(y,\hat{y})}{y} = \frac{|y-\hat{y}|}{y}$. This metric is used in [4] and [13] and adopted for comparison assessment.

Datasets & Workloads: The real dataset RS1 is taken from the UCI Machine Learning Repository (MLR)[1] containing $2 \cdot 10^6$ real points with $d \in \{2, \ldots, 6\}$. We use RS1 for comparison of $\mathcal{M}$ with PM and the sampling method. The real dataset RS2 refers to $0.5 \cdot 10^6$ 10-dim. real points from the UCI MLR[2]. We use RS2 for comparison with GenHist. To be aligned with the comparison with GenHist, all points in RS2 are normalized as in the GenHist paper [1]. RS3 refers to the Census dataset [18] from UCI MLR consisting of $2 \cdot 10^5$ 3-dim. points adopted for comparison with [13] and ISOMER. We generate certain sets of query patterns (workloads) for training $\mathcal{M}$, i.e., training set $\mathcal{T}$ and different evaluation set $\mathcal{E}$ thus assuring completely unseen queries with $|\mathcal{E}| = 100|\mathcal{T}|$. The sizes of $|\mathcal{T}|$ is defined as follows: $|\mathcal{T}| = \gamma|\mathcal{B}|$ is a very small fraction $\gamma \in [1‰, 1\%]$ of the dataset size $|\mathcal{B}|$. The number of prototypes $N \leq \gamma|\mathcal{T}|$ is then a percentage of $|\mathcal{T}|$. The converge threshold $\epsilon = 10^{-3}$ and the vigilance coefficient $a \in \{0.8, \ldots, 2\}\%$.

The workload WL1 for ISOMER, EquiHist and SpHist is generated exactly as in [13]. In WL1, center x_i, $i = 1, \ldots, d$ is selected uniformly at random from the data range. Then, each query is a d-dim. hyper-rectangle centered around $\mathbf{x}$ and with volume 2θ at most 20% of the total volume. For GenHist, we create two workloads WL2 and WL3 exactly as generated in [1]. WL2 contains queries whose points are chosen uniformly at random in the data domain. WL3 contains queries with points $\mathbf{x}$ such that $(x_i \leq o_i), \forall i$ for a randomly point $\mathbf{o} = [o_1, \ldots, o_d]^\top \in [0, 1]^d$. Comparison with the histogram-based approaches is obtained under L_∞.

Models Comparison: An approach to SCP is random sampling [7] where samples points from $\mathcal{B}$ randomly and uniformly without replacement, thus obtaining the random sample $\mathcal{B}'$. The cardinality values for $\mathcal{B}'$ are used as estimates of the cardinality values for the entire $\mathcal{B}$. We obtain $\mathcal{B}'$ by adopting the *reservoir sampling* algorithm [20]. ISOMER [11] uses the information-theoretic principle of maximum entropy to approximate $p(\mathbf{x})$ being consistent with the QFRs. The learning framework of STHs in [13] uses QFRs by introducing (i) the EquiHist algorithm, which learns a fixed size-bucket Equi-width histogram and (ii) the SpHist algorithm, which adopts Haar wavelets and compressed sensing for treating the histogram learning problem as a sparse-vector recovery problem. GenHist in [1] estimates $p(\mathbf{x})$ by allowing the buckets to overlap assuming that within each bucket, $p(\mathbf{x})$ is approximated by the average data density of the bucket. The power-method (PM) in [19] exploits the self-similar intrinsic dimensions of $\mathcal{B}$ assuming that the number of points within a radius θ of a given point $\mathbf{x} \in \mathcal{B}$ follows a local power law (LPLaw). SCP is achieved by pre-computing the LPLaw for a set $\mathcal{A} \subseteq \mathcal{B}$ of representative points (anchors) $\mathbf{x}_a$ randomly sampled from $\mathcal{B}$.

The error of the sampling method shown in Figure 1(left) is very higher than LM and QM for 2-dim. data from RS1 using $a = 0.01$, given exactly the same number of stored points. Here, the sample size is $|\mathcal{B}'| = |\mathcal{T}|$ and storage is represented as a percentage $\gamma \in [1‰, 8‰]$ of dataset size $|\mathcal{B}|$ with evaluation set size $|\mathcal{E}| = 100|\mathcal{T}|$. We compare LM and QM with EquiHist, SpHist and ISOMER using the same RS3 as used in [13]. The WL1 is generated with the exact same way as generated in [13] based on the L_∞ query generation model in [16]. We use $|\mathcal{T}| = 1\%|\mathcal{B}|$ and a different evaluation set $\mathcal{E}$ to compute the average error based on WL1; $|\mathcal{E}| = 100|\mathcal{T}|$. Figure 1(right) shows error against stored values for LM and QM (here, N corresponds to a fraction $\gamma|\mathcal{B}|$ with $\gamma \in [1‰, 8‰]$), SpHist, EquiHist, and ISOMER. Both LM and QM achieve significantly the lowest error than the other approaches. This is due to the fact that, as SpHist, EquiHist, and ISOMER attempt to tune a histogram with more QFRs (training samples) (as storage capacity increases the corresponding error decreases), LM and QM are trying to learn and maintain significant statistical information of those training pairs through prototypes, without focusing on learning the underlying data distribution. We obtain the same error for LM and QM with less training pairs, thus no need for higher storage.

We compare LM and QM with GenHist over RS2 as used in [1] and workloads WL2 and WL3 generated in exactly the same way as in [1] under L_∞. Figure 2 (left) shows the

[1] http://archive.ics.uci.edu/ml/machine-learning-databases/00235/

[2] kdd.ics.uci.edu/summary.data.type.html

error e vs. stored values (as a fraction of $|\mathcal{B}|$) for $d = 10$ using WL2 and WL3. We vary the training set size such that $|\mathcal{T}| = \gamma|\mathcal{B}|$ with $\gamma \in [1‰, 8‰]$. The set $\mathcal{E}$ is different with $\mathcal{T}$ and $|\mathcal{E}| = 100|\mathcal{T}|$. LM and QM outperform GenHist by achieving at most 51% lower error for WL2 and 70% lower error for WL3. This is attributed to the fact that LM and QM are trained to deal with the pattern space corresponding to WL2 and WL3. Moreover, an increase in N does not significantly contribute to better accuracy. Hence, we could utilize fewer prototypes to learn WL2 and WL3. We compare the performance of LM and QM with PM under L_2 with different distributions of θ (different μ_θ values) and storage capacity over 2-dim. data from RS1. Figure 2(right) shows the impact of the mean value of θ, $\mu_\theta \in [0.05, \ldots, 0.4]$ (with $\sigma_\theta^2 = 0.1$) on error for LM, QM and PM models. LM and QM are trained with $|\mathcal{T}| = 1\%|\mathcal{B}|$ and the size of the anchors set is also $|\mathcal{A}| = 1\%|\mathcal{B}|$ for PM. We create a different evaluation set $\mathcal{E}$ with $|\mathcal{E}| = 100|\mathcal{T}|$. Our models achieve very lower error than PM for all μ_θ values and utilizing at most 30% of $|\mathcal{T}|$. Our models do not depend on μ_θ, since their main purpose is to learn the association of θ with y given a region of points. On the other hand, PM stores all anchor points (randomly sampled from $\mathcal{B}$) and obtains error which progressively increases as μ_θ increases too. This is due to the fact that PM depends on θ (the major characteristic of the LPLaw) and the SCP is based on the LPLaw coefficients of each anchor point.

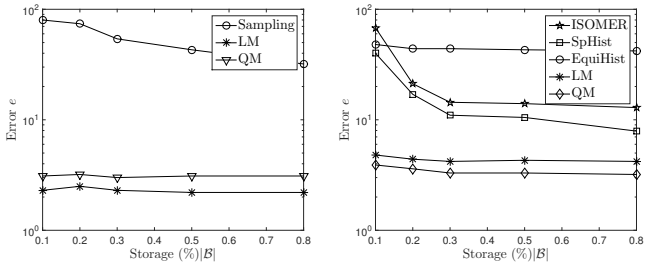

Figure 1. (Left) Error e for LM, QM and Sampling vs. storage (% of $|\mathcal{B}|$), $a = 0.01$, under 2-dim. data from RS1; (right) error e for ISOMER, SpHist, EquiHist, LM and QM against storage (% of $|\mathcal{B}|$), $a = 0.01$ using WL1 under 3-dim. data from RS3.

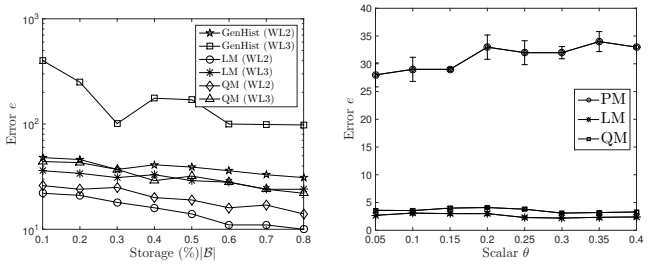

Figure 2. (Left) Error e for LM, QM and GenHist against storage (% of $|\mathcal{B}|$), $a = 0.01$ under 10-dim. data from RS2 using WL2 & WL3; (right) error e for LM, QM and PM against mean μ_θ ($\sigma_\theta^2 = 0.1$) with $|\mathcal{T}| = |\mathcal{A}| = 1\%|\mathcal{B}|$ under 2-dim. data from RS1.

V. Conclusions

We introduce a ML model for SCP of *d*NN queries achieving (i) incremental learning of the query patterns space, (ii) SCP of new queries based on previous *similar* queries, (iii) operating the above tasks in parallel, yielding an efficient model. A comprehensive evaluation showcased the model's robustness and that it significantly outperforms related works (based on m-d or self-tuning histograms, sampling, or the power method) which are data-driven. The proposed model is the only one that is applicable in environments including sensitive data, prohibiting access scans over raw-data and knowledge of raw-data updates (required by data-driven methods) and environments where accesses to raw data are costly money-wise (e.g., data maintained in the cloud).

References

[1] D. Gunopulos, G. Kolios, J. Tostras, C. Domeniconi, Selectivity estimators for multidimensional range queries over real attributes, VLDB J. 14(2):137–154, 2005.

[2] V. Poosala, P. Haas, Y. Ioannidis, E. Shekita, Improved histograms for selectivity estimation of range predicates, ACM SIGMOD'96, 294–305.

[3] A. Aboulnaga, S. Chaudhuri, Self-tuning histograms: building histograms without looking at data, ACM SIGMOD'99, 181–192.

[4] J. Vitter, M. Wang, B. Iyer, Data cube approximation and histograms via wavelets, ACM CIKM'98, 96–104.

[5] V. Poosala, Y. Ioannidis, Selectivity estimation without the attribute value independence assumption, VLDB'97, 486–495.

[6] J-H Lee, D-H Kim, C-W Chung, Multi-dimensional selectivity estimation using compressed histogram information, ACM SIGMOD'99, 205–214.

[7] F. Olken, D. Rotem, Random sampling from database files: a survey, SSDBM'90, 92–111.

[8] H. To, K. Chiang, C. Shahabi, Entropy-based histograms for selectivity estimation, ACM CIKM'13, 1939–1948.

[9] Y. Ioannidis, The history of histograms (abridged), VLDB'03, 19–30.

[10] N. Bruno, S. Chaudhuri, L. Gravano, STHoles: A multidimensional workload-aware histogram, ACM SIGMOD'01, 211–222.

[11] U. Srivastava, P. Haas, V. Markl, M. Kutsch, T. Tran, Isomer: Consistent histogram construction using query feedback', IEEE ICDE'06, 39.

[12] N. Bruno, S. Chaudhuri, Exploiting statistics on query expressions for optimization, ACM SIGMOD02, 263–274.

[13] R. Viswanathan, P. Jain, S. Laxman, A. Arasu, A Learning Framework for Self-Tuning Histograms, CoRR abs/1111.7295, 2011.

[14] G. Carpenter, S. Grossberg, D. Rosen, Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Networks, 4: 759–771, 1991.

[15] H. Robbins, S. Monro, A Stochastic Approximation Method, Ann. Mathem. Stat., 22(3):400–407, 1951.

[16] B-U Pagel, H-W Six, H Toben, P Widmayer, Towards an analysis of range query performance in spatial data structures, PODS'93, pp. 214–221.

[17] L. Bottou, O. Bousque, The Tradeoffs of Large Scale Learning, NIPS 2008, 161–168.

[18] C. Blake, E. Keogh, C. Merz, UCI repository of machine learning databases, 1998.

[19] Y. Tao, C. Faloutsos, D. Papadias, The Power-Law Method: A Comprehensive Estimation Technique for Multi-Dimensional Queries, CIKM'03, 83–90.

[20] J. Vitter. 1985. 'Random sampling with a reservoir'. ACM Trans. Math. Software 11 (1): 37–57.