传感器网络中自适应滑动窗口的高效 Top-k 查询技术

郑吉平[1,2],宋保利[1],王海翔[1],王永阁[1]
(1.南京航空航天大学计算机科学与技术学院,江苏南京 210016;
2.南京大学计算机软件新技术国家重点实验室,江苏南京 210093)

摘 要: 在传感器节点上安装动态窗口的过滤机制是无线传感器网络 Top-k 查询处理研究的一个重要方向.然而,已有过滤窗口算法会产生很大的窗口更新代价.本文针对过滤窗口更新频繁产生巨大能量消耗的问题,提出基于高斯过程回归预测的自适应滑动窗口 Top-k 查询处理算法 FUGPR.当过滤窗口发生变化时,对传感器网络节点读数进行预测,评估窗口更新前后的代价来决定过滤窗口是否更新,从而减少了频繁更新窗口带来的巨大能量消耗.实验表明,本文提出的 FUGPR 算法无论在真实传感器网络环境的数据集上还是模拟的传感器网络环境数据集上都可以有效地减少由于过滤窗口更新带来的能量消耗.

关键词: 过滤窗口; 无线传感器网络; Top-k; FUGPR
中图分类号: TP392 **文献标识码:** A **文章编号:** 0372-2112 (2015)10-2117-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2015.10.034

Energy-Efficient Top-k Query Techniques Based on Adaptive Filters in Wireless Sensor Networks

ZHENG Ji-ping[1,2], SONG Bao-li[1], WANG Hai-xiang[1], WANG Yong-ge[1]
(1. *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu* 210016, *China*;
2. *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu* 210093, *China*)

Abstract: Adopting the filtering mechanism of dynamic filtering windows installed on sensor nodes to process top-k queries is an important research direction in wireless sensor networks. Existing algorithms based on filters consume a vast amount of energy on filter updating. As updating filters consume a large amount of energy, a top-k query processing algorithm adopting adaptive filters named FUGPR based on Gaussian process regression is provided. When the filters change, the sensor readings are predicted to calculate the updating costs of filters, then FUGPR decides whether the filters need to be updated or not. Thus, the energy consumption for updating filters is decreased. Experimental results show that our approach can reduce energy consumption efficiently for updating filters on real and simulated datasets.

Key words: filtering windows; wireless sensor networks; Top-k; FUGPR(Filter Updating based on Gaussian Process Regression)

1 引言

随着微电子技术、计算机技术、网络技术以及无线通信技术的进步,体积小、成本低、功能多的传感器得到了快速的发展,无线传感器网络技术得到了深入研究和广泛应用.传感器节点由电池供电,能量有限,无线通信在传感器网络能量消耗中居首要地位.同时,无线传感器网络产生大量数据,如果把这些数据都传送到基站进行集中式处理需要进行大量的数据传输.因此,在执行无线传感器网络数据查询时,减少传感器网络中的无线通信,可以有效地延长网络的生命周期.面对传感器网络中产生的大量数据,人们往往只对其中的前 k 个最大(或最小)的读数感兴趣,因此 Top-k 查询[1]作为在不确定数据和关系数据库中广泛使用的一种聚集查询技术,在无线传感器网络中有着广泛的应用.例如 Top-k 查询可以有效地监控环境、生态变化等.

对于传感器网络环境下的 Top-k 查询处理,人们已进行了一些研究工作.为了减少传感器网络中的无线通信,Madden 等人[2]引入了聚集技术 TAG,但该方法存在大量冗余数据的传输,并非真正能量有效.Silberstein 等人[3]利用线性规划的方法提出 Top-k 查询处理算法 PROSPECTORLP-LF 和 PROSPECTORLP + LF.每次查询时,

收稿日期:2014-01-22;修回日期:2015-01-28;责任编辑:李勇锋
基金项目:江苏省自然科学基金(No.BK2014086);中央高校基本科研业务费专项资金(No.NS2015095);南京航空航天大学研究生创新基地(实验室)开放基金(No.KFJJ201461)

在一定的能量消耗限制下,这两种算法都最小化返回结果集中所丢失的 Top-k 集中的个数,尽可能的减少传感器网络中的无线通信.但是该方法不能保证用户查询的精确性,当精度很高时,能量消耗是巨大的.Chen 等人[4]提出了分位数过滤算法 QF(Quantile Filter),该算法虽然也可以避免大量冗余数据的上传,但是每个节点与父节点都要进行通信,传感器网络的无线通信次数仍然很大.Abbasi 等人[5]以最小化监控查询中的能量消耗为目的,提出了 MOTE(Model-based Optimization Technique)方法,它基于模型优化的技术在节点上设置过滤条件,但如何取得最优的过滤条件进行 Top-k 结果的维护,仍然是一个 NP 问题.Wu 等人[6]提出运用 Range Caching 方法处理传感器网络中的 Top-k 查询,为每个传感器节点设置宽度同为 α 的过滤窗口,若数据落在窗口内则节点不向基站发送数据.Range Caching 方法由于节点之间的过滤窗口可能存在重叠,基站向节点请求读数的通信能量消耗巨大.此外,用户对确定过滤窗口范围 α 的取值也存在困难.鉴于 Range Caching 存在的问题,Wu 等人提出 FILA (FILter-based monitoring Approach)算法[7]对传感器网络进行 Top-k 查询处理.FILA 算法基于过滤思想,为每一个节点分配一个过滤窗口,避免了大量冗余数据的传输,但是该算法会产生很大的窗口更新代价.Mai 等人[8]提出了通过预测方法减小窗口更新代价的 DAFM(Distributed Adaptive Filter based Monitoring)算法,但是,由于传感器节点数据的复杂性,节点读数在时间上的线性回归预测方法存在一定的局限性.

对于基于过滤窗口的方法,在过滤窗口更新阶段,当数据波动频繁时,会导致过滤窗口更新频繁,从而消耗大量的传感器网络能量,不利于延长传感器网络的生命周期.本文提出基于高斯过程回归预测的窗口更新机制从而实现高效 Top-k 查询处理的算法 FUGPR (Filter Updating based on Gaussian Process Regression).通过对节点未来读数的预测,评估窗口更新代价,决定是否向相关节点发送过滤窗口更新,从而减少由于数据波动频繁所产生的不必要的窗口更新代价.实验结果表明,本文所提出的 FUGPR 算法无论在真实传感器网络环境的数据集上还是模拟的传感器网络环境都有良好的表现,极大地减少了传感器网络的能量消耗,从而延长了网络的生存周期.

2 Top-k 查询及窗口过滤方法

2.1 问题定义

传感器网络中共有 N 个节点,对所有传感器节点进行标号,组成的集合记为 I,$I=\{1,2,\cdots,N\}$.传感器每隔一个固定的时间片刻采集其所在地域的物理现象,如温度、电压、光照、湿度等.节点 $n_i(n_i\in I)$感知的数据记为 v_i.每隔一个时间片刻,用户对传感器网络进行 Top-k 查询,要求返回由读数最高的 k 个传感器节点组成的排序列表 $R=\langle n_1,n_2,\cdots,n_k\rangle$,其中 $\forall\, i<j, v_i\geqslant v_j$,并且 $\forall\, l\neq i(i=1,2,\cdots,k), v_l\leqslant v_k$.

Top-k 查询结果集由基站维护,最终返回给终端用户.本文方法的优化目标是,在保证查询结果正确性的同时,最小化传感器节点的能量消耗,从而延长传感器网络的生存周期.

FILA 是目前传感器网络基于过滤思想进行 Top-k 查询的典型算法.FILA 算法在初始阶段,得到所有传感节点的读数集合$\{v_1,v_2,\cdots,v_N\}$,其中 $v_1\geqslant v_2\geqslant\cdots\geqslant v_N$.对当前时刻 Top-$k$ 结果集中的节点分别设置一个过滤窗口,其它非 Top-k 结果集的节点设置一个共同的过滤窗口,因此,只需要计算 $k+1$ 个过滤窗口.为了保证算法的正确性,所有节点过滤窗口$[l_i,u_i]$之间应不重叠.过滤窗口$[l_1,u_1],[l_2,u_2],\cdots,[l_k,u_k],[l_{k+1},u_{k+1}]$满足:

$$\begin{cases} v_1\leqslant u_1 \\ v_{i+1}\leqslant u_{i+1}\leqslant l_i\leqslant v_i,(1\leqslant i\leqslant k) \\ l_{k+1}\leqslant v_N \end{cases} \tag{1}$$

为了最大化过滤窗口的过滤能力,往往把 u_1和 l_{k+1}分别设置成 $+\infty$和 $-\infty$;且除了 u_1和 l_{k+1}以外,令 $u_{i+1}=l_i$.

2.2 已有窗口过滤方法存在缺陷

FILA 算法在一定程度上避免了冗余数据的上传,节省了能量,但是仍然存在大量不必要的通讯代价.在过滤窗口更新阶段,当数据波动频繁时,会导致过滤窗口更新频繁,从而消耗大量的能量.如图 1 所示.

在图 1(a)中,t_i时刻未采集数据前,Top-3 结果集为$\{v_1,v_4,v_3\}$,采集数据后如图 1(b)所示,Top-3 结果集为$\{v_1,v_3,v_4\}$,即 v_3跳进 v_4的过滤窗口中,此时需要为相关节点更新过滤窗口,基站通过计算重新调整各节点的过滤窗口如图 1(c)所示.由图 1(d)可知,在 t_{i+1}时刻传感器节点重新采集数据后,v_3值再次发生变化,回

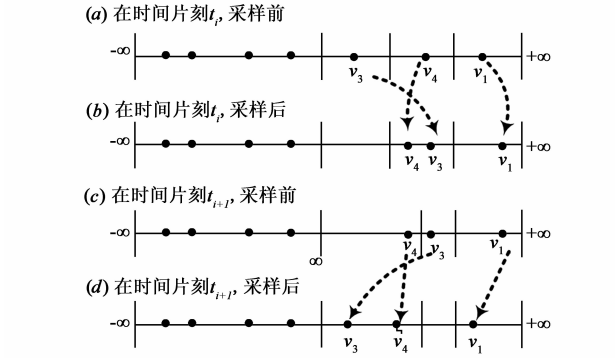


图1 频繁地更新过滤窗口

到更新前的过滤窗口中.对于这样波动频繁的数据,如果每次都对节点进行窗口更新,会造成很大的窗口更新代价.

3　窗口更新算法 FUGPR

本文基于高斯过程回归预测的窗口更新机制,提出 Top-k 查询处理算法 FUGPR.FUGPR 算法共设置 $k+1$ 个互不相临且不重叠的过滤窗口.前 k 个过滤窗口分别安装在当前 Top-k 结果集中的节点上,其它所有非 Top-k 结果集的节点安装第 $k+1$ 个过滤窗口.当传感器节点感知的数据落在其过滤窗口中,节点不向基站发送数据,否则向基站发送当前采集的数据.基站根据节点新采集的数据,重新估计过滤窗口;当过滤窗口变化时,利用高斯过程回归预测方法,对传感器网络节点读数进行预测.由于高斯过程回归(Gaussian Process Regression,GPR)方法[9]所预测的节点读数具有概率意义,可以计算出所预测节点的读数分别落在新旧过滤窗口的概率值.进一步计算出过滤窗口的更新代价,供用户决定是否立即更新过滤窗口,从而减少了频繁更新窗口带来的巨大能量消耗.

3.1　高斯过程回归预测

节点读数在时间上存在一定的相关性,可以用节点过去时刻的读数预测将来的读数.对节点前后时刻的数据进行回归.输入为前一时刻读数,输出为后一时刻的读数,即数据集 $H_D=\{(x_i^{(l)},y_i^{(l)});l=1,2,\cdots,m$ 且 $y_i^{(l)}=x_i^{(l+1)}\}$.但这样单维的输入输出过于简单,且仅仅根据当前时刻读数去预测下一时刻读数精度不够.

节点发送 m 个字节的数据所消耗的总能量为 $\alpha_s+\beta_s m$,其中 α_s 表示每次建立通信连接发送方消耗的能量,β_s 表示发送每一字节的数据消耗的能量;接收 m 个字节的数据所消耗的总能量为 $\alpha_r+\beta_r m$,其中 α_r 表示每次建立通信连接接收方消耗的能量,β_r 表示接收每一字节的数据消耗的能量.对于 MICA2 型传感器节点 $\alpha_r\leqslant 0.4\alpha_s$,$\beta_r\leqslant 0.4\beta_s$[10],为了定量计算,本文取 $\alpha_r=0.4\alpha_s$,$\beta_r=0.4\beta_s$.α_s,β_s,α_r 和 β_r 的具体取值如表 1 所示.

表 1　符号典型取值

符号	取值
α_s	0.645 mJ
β_s	0.0144 mJ/byte
α_r	0.258 mJ
β_r	0.00576 mJ/byte

由表 1 可知,建立通信连接消耗的能量相比传输少量字节数据消耗的能量是巨大的.因此,减少传输次数,每次传输适当多发送一些数据,可以有效的减少节点的能量消耗.当节点读数落在过滤窗口时,抑制节点读数上传;当节点读数违背过滤窗口时,节点选择过去 p 个时间片刻的从未传送过的读数打包传送给基站.p 一般在基站根据历史数据求得,若 p 值过大,传输的字节数过多,能量消耗依然很大,因此 p 的取值不宜过大,本文 p 的取值从 1~5,根据 p 的取值不同分别进行回归预测,所求得的高斯过程回归误差的方差最小,则认为该 p 的取值是比较合适的.在基站我们可以对数据集 H_D 进行处理,输入为前 p 时刻的读数,输出为后一时刻的读数,令:

$$\boldsymbol{X}^{(t)}=\begin{bmatrix}x^{(t)}\\x^{(t-1)}\\\vdots\\x^{(t-p+1)}\end{bmatrix},\quad y^{(t)}=x^{(t+1)}\tag{2}$$

则:$H_D=\{(X^{(l)},y^{(l)});l=p,p+1,\cdots,p+m-1\}$,通过 $x^{(1)},x^{(2)},\cdots,x^{(p+m)}$ 共 $p+m$ 个历史数据构造出 m 个回归样本点.以节点过去几个时刻感知的数据作为输入,后一时刻感知的数据作为输出,通过监督学习确定映射函数:

$$y=f(X)+N(0,\sigma^2)\tag{3}$$

通常,利用参数化回归,即先假定映射函数的具体类型,通过训练集来求得最好的函数参数.线性回归就是最典型的参数化回归方法.Mai 等人提出的 DAFM[8] 方法中采用的就是线性回归.线性回归预测模型定义为以下形式:

$$y=b+\theta X+\varepsilon\tag{4}$$

其中 e 为随机误差项.通常可根据最小二乘法对参数 b、θ 的值进行估计.误差项 e 服从高斯分布 $N(0,\sigma^2)$.

3.1.1　GPR 预测

高斯过程把多元高斯分布拓展到无限维,任意 n 个观测值构成的集合可以看作是某个多元高斯分布的一个采样点.假设每一个观测值的均值为零,则其中一个观测值对另一个观测值的影响主要取决于它们之间的协方差函数 $k(x,x')$.

假设训练集 H_D 共有 m 个数据点,由协方差函数可以构造 $m\times m$ 阶正定的协方差矩阵:

$$\boldsymbol{K}=\begin{bmatrix}k(x^{(1)},x^{(1)}) & k(x^{(1)},x^{(2)}) & \cdots & k(x^{(1)},x^{(m)})\\k(x^{(2)},x^{(1)}) & k(x^{(2)},x^{(2)}) & \cdots & k(x^{(2)},x^{(m)})\\\vdots & \vdots & \ddots & \vdots\\k(x^{(m)},x^{(1)}) & k(x^{(m)},x^{(2)}) & \cdots & k(x^{(m)},x^{(m)})\end{bmatrix}\tag{5}$$

设新的输入为 x_*,则令 $\boldsymbol{K}_*$ 为:

$$\boldsymbol{K}_*=[k(x_*,x_1)\quad k(x_*,x_2)\quad\cdots\quad k(x_*,x_m)],\quad \boldsymbol{K}_{**}=k(x_*,x_*)\tag{6}$$

$\boldsymbol{y}=[y^{(1)}\quad y^{(2)}\quad\cdots\quad y^{(m)}]$,则 $\boldsymbol{y}$ 与 y_* 的联合先验分布为:

$$\begin{bmatrix}\boldsymbol{y}\\ y_*\end{bmatrix}\sim N\left(0,\begin{bmatrix}\boldsymbol{K} & \boldsymbol{K}_*^{\mathrm{T}}\\ \boldsymbol{K}_* & \boldsymbol{K}_{**}\end{bmatrix}\right) \tag{7}$$

由式(7)可以计算出后验分布:

$$y_*|\boldsymbol{y}\sim N(\boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{y},\boldsymbol{K}_{**}-\boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{K}_*^{\mathrm{T}}) \tag{8}$$

y_* 的均值为:

$$y'_*=\boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{y} \tag{9}$$

y_* 的方差 σ^2 为:

$$\delta^2=\boldsymbol{K}_{**}-\boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{K}_*^{\mathrm{T}} \tag{10}$$

即:

$$y_*\sim N(\boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{y},\boldsymbol{K}_{**}-\boldsymbol{K}_*\boldsymbol{K}^{-1}\boldsymbol{K}_*^{\mathrm{T}}) \tag{11}$$

当给定新的输入 x_* 时,以及一定的置信水平 1-δ 时,可以估计出一个置信区间:

$$\boldsymbol{I}_C=(\boldsymbol{y}'_*-z_{\delta/2}\boldsymbol{\sigma},\boldsymbol{y}'_*+z_{\delta/2}\boldsymbol{\sigma}) \tag{12}$$

3.1.2 GPR 训练

GPR 可以选择不同的协方差函数,根据训练数据的不同特性,改造协方差函数,以便更好地进行回归预测.本文采用常用的均方指数协方差函数

$$k(\boldsymbol{x}_p,\boldsymbol{x}_q)=\sigma_f^2\exp\left(-\frac{1}{2}(\boldsymbol{x}_p-\boldsymbol{x}_q)^{\mathrm{T}}P^{-1}(\boldsymbol{x}_p-\boldsymbol{x}_q)\right) \tag{13}$$

通常协方差函数中要加入高斯白噪声部分与其它协方差函数组成复合协方差函数一起使用,高斯噪声协防差函数以 covNoise 表示,如式(14):

$$k(\boldsymbol{x}_p,\boldsymbol{x}_q)=\sigma_n^2\delta(\boldsymbol{x}_p,\boldsymbol{x}_q) \tag{14}$$

因此复合以后的协方差函数为:

$$k(x_p,x_q)=\sigma_f^2\exp\left(-\frac{1}{2}(\boldsymbol{x}_p-\boldsymbol{x}_q)^{\mathrm{T}}P^{-1}(\boldsymbol{x}_p-\boldsymbol{x}_q)\right)+\sigma_n^2\delta(\boldsymbol{x}_p,\boldsymbol{x}_q) \tag{15}$$

式(13)中若 $P=\begin{pmatrix}\lambda^{-2} & & \\ & \ddots & \\ & & \lambda^{-2}\end{pmatrix}$,称协方差函数为 covSEiso.假设 $\boldsymbol{x}_p$、$\boldsymbol{x}_q$ 是 d 维列向量,令 $\boldsymbol{x}_p-\boldsymbol{x}_q=[r_1,r_2,\cdots r_d]^{\mathrm{T}}$,令 $P=\begin{pmatrix}l_1^2 & & \\ & \ddots & \\ & & l_d^2\end{pmatrix}$,则协方差函数称为:covSEard.

在给定协方差函数的形式后,高斯过程回归模型的特性就被协方差函数中的超参数唯一决定.以协方差函数 covSEard + covNoise 为例,超参数 $\theta=\{l,\sigma_f,\sigma_n\}$,其中 $l=\{l_1,\cdots,l_d\}$.一般通过极大似然法求得超参数求得.假设输入为 X,输出为 y,当概率密度函数 $p(\theta|X,y)$取最大值,即最大化 $p(\theta|X,y)$,θ 的取值即为所求.根据贝叶斯公式得:

$$p(\theta|y,X)=\frac{p(y|X,\theta)p(\theta)}{p(y|X)} \tag{16}$$

$p(X|y)$可看作是常数;假设 θ 在其值域的任意取值是等可能的,即 θ 服从均匀分布,$p(\theta)$也可看作是常数.则最大化 $p(\theta|X,y)$等价于最大化 $p(y|X,\theta)$,即:

$$\arg\max_\theta p(\theta|X,y)=\arg\max_\theta p(y|X,\theta) \tag{17}$$

由 $y\sim N(0,K)$,得:

$$p(y|X,\theta)=\frac{1}{(2\pi)^{n/2}|K|^{1/2}}\exp\left(-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{K}^{-1}\boldsymbol{y}\right) \tag{18}$$

令 $L(\theta))=\log p(y|X,\theta)$,则:

$$L(\theta)=-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}K^{-1}\boldsymbol{y}-\frac{1}{2}\log|K|-\frac{n}{2}\log 2\pi \tag{19}$$

对超参数求偏导,然后可以利用梯度下降法求得超参数.

3.2 窗口更新策略优化

当传感器节点的值违背其过滤窗口时,节点向基站发送数据.当传感器节点更新的读数跃入其它节点的过滤窗口时,在基站 Top-k 结果集变得不可确定.因此基站需要探测其它节点的读数来重新估计 Top-k 结果集.对于主动向基站传送数据以及基站向探测节点请求读数,这些节点需要把当前及过去共 p 个读数打包传送给基站,基站进行高斯过程回归预测.预测未来读数的均值及方差.对于没有更新的传感器读数,其未来的预测值及方差仍采用最近更新读数后通过高斯过程回归计算得到的预测值和方差.

假设节点 n_i第 t_{n+1}时间片刻的预测值为 v_i^{pre},方差为 σ_i^2,则其真实值 $v_i\sim N(v_I^{\mathrm{pre}},\sigma_i^2)$,设其概率密度函数为 $f_i(x)$.假设节点 n_i已有的过滤窗口为 $F_i^{\mathrm{old}}=[l_I^{\mathrm{old}},u_i^{\mathrm{old}}]$,通过计算待更新的过滤窗口为 $F_i^{\mathrm{new}}=[l_I^{\mathrm{new}},u_i^{\mathrm{new}}]$.则真实值 v_i落在 F_i^{old} 的概率为:

$$P_i^{\mathrm{old}}=\int_{l_i^{\mathrm{old}}}^{u_i^{\mathrm{old}}}f_i(x)\mathrm{d}x \tag{20}$$

真实值 v_i落在 F_i^{new} 的概率为:

$$P_i^{\mathrm{new}}=\int_{l_i^{\mathrm{new}}}^{u_i^{\mathrm{new}}}f_i(x)\mathrm{d}x \tag{21}$$

令 r_i表示在最近的时间片刻中节点 n_i的读数落在过滤窗口中的平均个数.对于 r_i的计算采用以下方式:已运行 FILA 算法 100 个时间片刻,为方便描述,现在只考虑第 80 ~ 100 时间片刻,假设节点 n_i读数违背过滤窗口是在第 84,91,94,100 时间片刻,那么节点落在过滤窗口中的时间片刻数分别为:91 − 84 − 1 = 6,94 − 91 − 1 = 2,100 − 94 − 1 = 5.则节点 n_i的读数在最近的时间片刻中落在其过滤窗口的平均个数 $r_i=(6+2+5)/3$.

计算过滤窗口更新后的传感器网络执行 Top-k 查询的代价的 $cost_{\mathrm{new}}$,以及过滤窗口维持现状的的代价

$cost_{old}$.若 $cost_{new} < cost_{old}$则把新的过滤窗口发送给传感器节点;若 $cost_{new} \geqslant cost_{old}$,则不需要进行过滤窗口的更新. $cost_{new}$和 $cost_{old}$的计算方式如下:

$$cost_{new} = |N_u| + |N| \times s - \sum_{n_i \in I} P_i^{new} \times r_i \quad (22)$$

$$cost_{old} = |N| \times s - \sum_{n_i \in I} P_i^{old} \times r_i \quad (23)$$

式(22)中$|N_u|$表示需要更新过滤窗口的节点个数,式(22)和式(23)中 s 代表将来的 s 个时间片刻,$|N|$代表传感器网络中节点的个数. $cost_{new}$和 $cost_{old}$的大小比较结果不受$|N| \times s$ 的影响,所以对于这一部分无需计算,也不需要考虑 s 具体取何值.

4　实验测评

4.1　实验设置

实验数据集:

(1)Intel Lab Data:该数据集是由部署在伯克利大学英特尔实验室* 的 54 个传感器节点感知现实环境中的多个属性值得到.将传感器网络划分成多个区域,位置较近的节点划分到同一区域.本文对如何更合理地进行区域划分不做深入研究,仅根据节点位置信息以 m 均值算法对传感器网络进行区域划分.以伯克利大学英特尔实验室无线传感器网络为例,利用 m 均值算法把其划分成 10 个区域.图 2 为伯克利大学因特尔实验室的 54 个传感器节点区域划分结果,其中用深色节点表示在对传感器网络进行 m 均值划分时各区域的初始质心节点.传感器节点每 31s 感知一次数据,选择 2004 年 3 月 1 日的温度、湿度及电压作为实验数据,利用 Matlab 仿真实验来验证本文所提出算法的有效性.

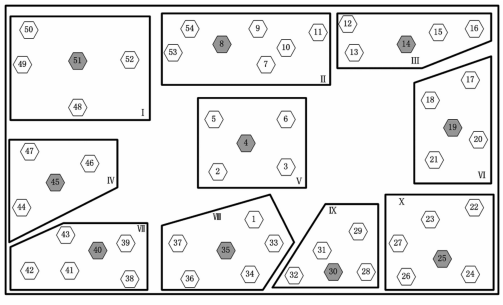
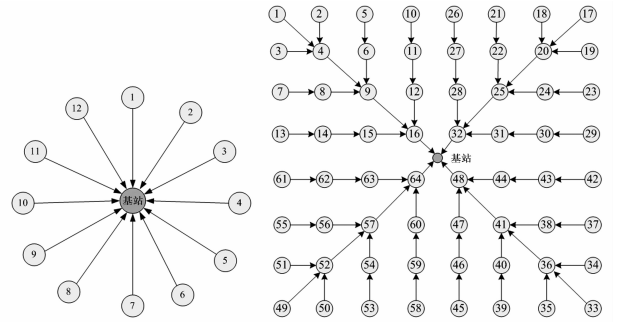


图2　伯克利大学英特尔实验室无线传感器网络划分图

(2)LEM Data:该实验数据集是由华盛顿大学 LEM(Live from Earth and Mars,LEM)计划** 采集的数据来模拟传感器读数.抽取从 2012 年 12 月 1 日到 2013 年 12 月 1 日采集的温度(Temperatures)、露点(Dewpoint)和海平面气压(Sea Level Pressure)作为实验数据.共 509174 条记录,中间有的时间片刻数据缺失,通过前后两个非缺失的数据取平均值补齐.每一个属性有 509174 条记录,把其分成多个小的数据段,每个数据段的大小为 3000 条记录,把这些小的数据段分配给各个传感器节点.

该实验模拟单跳和多跳网络,单跳网络共有 12 个传感器节点,如图 3(a)所示.多跳网络采用 8×8 以及 12×12 的网络,8×8 的网络拓扑结构如图 3(b)所示,并对节点进行编号;12×12 的网络拓扑结构类似于 8×8 的网络,限于文章篇幅以及为了论文的紧凑性,在此就不再给出.由于传感器网络节点读数存在空间相关性,即空间位置相近的节点读数有相似性,因而把时间相邻的数据段分配给空间位置相近的节点,这样可以更好的模拟真实的传感器网络.



(a) 单跳网络　　(b) 8×8多跳网络

图3　网络拓扑结构

实验中模拟的节点为 MICA2 型传感器节点,其能量消耗如表 1 所示.TAG 方法是最基本的聚集查询方法,作为能量消耗对比的基准.在 Cache 方法中,如果有两个 Top-k 候选节点的过滤窗口重叠时,基站需要向它们请求读数,以确定 Top-k 结果集.在实验中,对 Cache 方法来说,其过滤窗口 α 的取值从 0.3 到 3.2,并计算出不同取值时的能量消耗,选择能量消耗最小的 α 取值作为以后运行 Cahce 方法时的过滤窗口宽度.对于 FILA 方法,其窗口更新采用延迟更新策略.本文提出的 FUGPR 方法中高斯过程采用该领域最广泛采用的高斯过程 Matlab Toolbox*** 实现,其中协方差函数采用典型的 covSEard + covNoise 协方差函数.

4.2　实验结果

图 4~6 显示了在伯克利大学因特尔实验室真实的传感器网络环境下,TAG、Range Caching、FILA 以及本文提出的 FUGPR 算法执行 1000 次 Top-k 查询的能量消耗对比.由图 4~6 可见,FILA 算法通过窗口过滤机制,很大程度上了冗余数据的传输;本文的 FUGPR 算法在 FILA 算法的基础上,优化了窗口过滤机制,减少了 FILA 算法在更新过滤窗口带来的巨大能量消耗,降低了传感器节点的能量消耗,延长了网络的能量消耗.当 k 值

* http://db.lcs.mit.edu/labdata/labdata.html

** http://www-k12.atmos.washington.edu/k12/

*** http://www.gaussianprocess.org

逐渐增大时,FUGPR 算法更加明显优于 FILA 算法,这是因为 K 值增大时,违背节点过滤窗口的节点读数增多,而一旦出现某节点的过滤窗口发生变化,FILA 算法就需要对相应节点的过滤窗口进行更新;而 FUGPR 算法,从全局考虑去评估过滤窗口更新的代价,要么更新所有节点的过滤窗口,要么全不更新,减少了过滤窗口的更新次数,减少了网络的能量消耗.

图 7~9 显示了 k 取不同值时,TAG、Range Caching、FILA 以及本文提出的 FUGPR 算法在单跳网络执行 1000 次 Top-k 查询后的能量消耗对比.分别对不同属性值如温度、露点及海平面气压进行实验.FILA 通过设置不重叠的过滤窗口,可以过滤大量的节点读数,尤其是 k 取值比较小的时候;但是当 k 值较大时,基站向节点探测读数及窗口更新,能量消耗代价较大,在露点属性值的实验中当 k 值取 9 时,FILA 的能量消耗甚至超过了 TAG 聚集方法,这是因为 FILA 算法虽然过滤掉部分节点的读数,但当数据波动频繁时,基站探测节点的读数和对节点的过滤窗口进行更新所消耗的能量甚至会超出其之前因过滤部分无用节点读数所节省下的能量.本文的 FUGPR 算法优化了过滤窗口的更新策略,节省了大量的传感器网络能量.

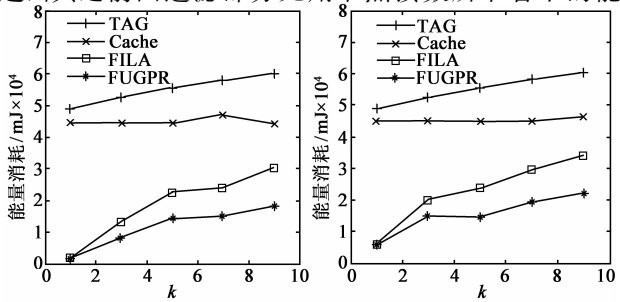


图4 Intel Lab网络能耗对比(温度)

图5 Intel Lab网络能耗对比(湿度)

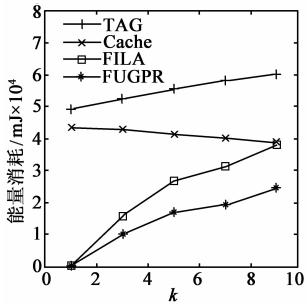


图6 Intel Lab网络能耗对比(电压)

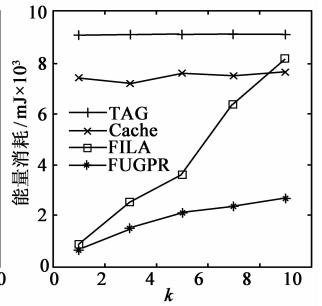


图7 单跳网络能耗对比(温度)

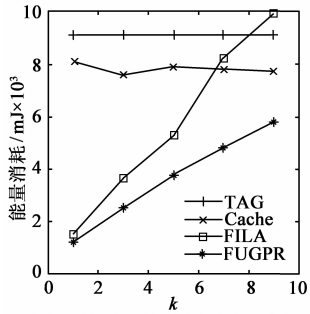


图8 单跳网络能耗对比(露点)

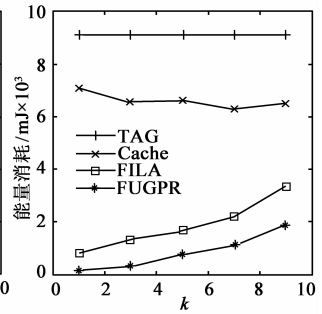


图9 单跳网络能耗对比(海平面气压)

图 10~12 和图 13~15 分别显示了 k 取不同值时,TAG、Range Caching、FILA 以及本文提出的 FUGPR 算法在多跳网络的的能量消耗对比.由图 10~12 和图 13~15 可见,本文提出的 FUGPR 方法在 8×8 和 12×12 的多跳网络以及多种属性值下执行 1000 次 Top-k 查询后的能量消耗都优于 TAG、Cache 和 FILA 方法.FUGPR 算法可以减少传感器节点读数波动频繁的影响,降低节点的能量消耗,从而延长了网络生命周期.

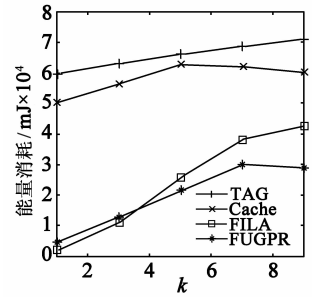


图10 8×8网络能耗对比(温度)

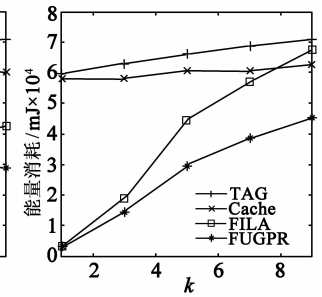


图11 8×8网络能耗对比(露点)

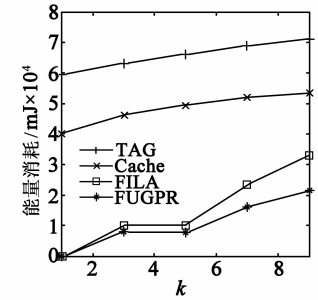


图12 8×8网络能耗对比(海平面气压)

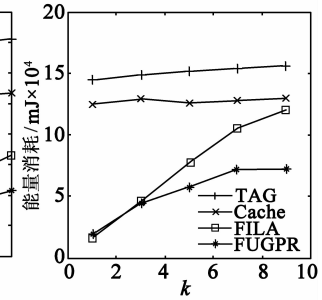


图13 12×12网络能耗对比(温度)

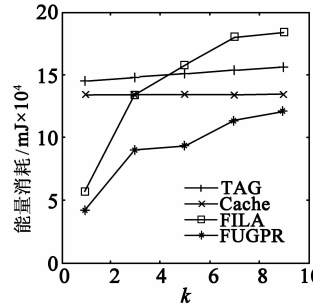


图14 12×12网络能耗对比(露点)

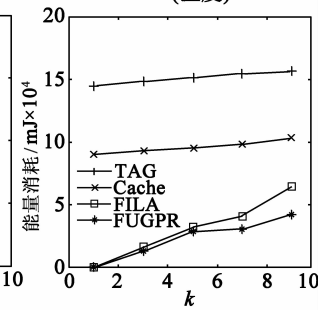


图15 12×12网络能耗对比(海平面气压)

5 结语

本文将高斯过程回归预测与 FILA 相结合,提出自适应过滤窗口更新机制从而实现高效 Top-k 查询处理的算法 FUGPR,FUGPR 算法减少了由于数据波动频繁所产生的不必要的窗口更新代价.实验表明,本文所提出的 FUGPR 算法无论在真实传感器网络环境的数据集

上还是模拟的传感器网络环境都极大地减少了传感器节点的能量消耗,延长了网络的生存周期.

参考文献

[1] Dylla M, Miliaraki I, Theobald M. Top-k query processing in probabilistic databases with non-materialized views[A]. Proceedings of the 29th International Conference on Data Engineering[C]. Washington: IEEE Press, 2013. 122 – 133.

[2] Madden S, Franklin M, Hellerstein J, et al. TAG: a tiny aggregation service for ad-hoc sensor networks[J]. ACM Special Interest Group on Operating Systems Review, 2002, 36(SI): 131 – 146.

[3] Silberstein A, Braynard R, Ellis C, et al. A sampling-based approach to optimizing Top-k queries in sensor networks[A]. Proceedings of the 22nd International Conference on Data Engineering[C]. Atlanta: IEEE Press, 2006. 68 – 78.

[4] Chen B, Liang W, Zhou R, et al. Energy-efficient Top-k query processing in wireless sensor networks[A]. Proceedings of the 19th ACM International Conference on Information and Knowledge Management[C]. New York: ACM Press, 2010. 329 – 338.

[5] AbbasiA, Khonsari A, Farri N. MOTE: efficient monitoring of Top-k set in sensor networks[A]. IEEE Symposium on Computers and Communications[C]. Riccione: IEEE Press, 2008. 957 – 962.

[6] Wu Min-ji, Xu Jian-liang, Tang Xue-yan. Processing precision-constrained approximate queries in wireless sensor networks [A]. Proceedings of International Conference on Mobile Data Management[C]. Nara: IEEE Press, 2006. 31 – 38.

[7] Wu Min-ji, Xu Jian-liang, Tang Xue-yan, et al. Top-k monitoring in wireless sensor networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(7): 962 – 976.

[8] Mai H, Lee Y, Lee K, et al. Distributed adaptive Top-k monitoring in wireless sensor networks[J]. The Journal of Systems and Software, 2011, 84(2): 314 – 327.

[9] C E Rasmussen, C K I Williams. Gaussian Processes for Machine Learning[M]. London: MIT Press, 2006.

[10] Silberstein A, Braynard R, Yang J. Constraint chaining: on energy-efficient continuous monitoring in sensor networks[A]. Proceedings of the ACM SIGMOD International Conference on Management of Data[C]. Chicago: ACM Press, 2006. 157 – 168.

作者简介

郑吉平　男,1979 年 10 月出生,江苏南京人.博士后、副教授、中国电子学会高级会员、中国计算机学会高级会员、IEEE 和 ACM 会员.2001 年在南京信息工程大学获得工学学士学位后,2004 年和 2007 年在南京航空航天大学分别获得工学硕士和工学博士学位,2007 年到 2009 年在清华大学计算机科学与技术博士后流动站从事研究工作.目前主要从事感知数据管理、Skyline 和 Top-k 查询处理、计算几何和蒙特卡罗方法等方面的研究工作.
E-mail: zhjcs@nuaa.edu.cn

宋保利　男,1987 年 2 月出生,山东临沂人.硕士研究生,主要从事 Top-k 查询处理、采样方法等方面的研究工作.
E-mail: songbaoli@nuaa.edu.cn