

# 受限路网中基于全局学习机制的在线轨迹预测

徐怀野<sup>1,2</sup> 丁治明<sup>1</sup> 刘奎恩<sup>1</sup> 许佳捷<sup>1</sup>

(中国科学院软件研究所 北京 100190)<sup>1</sup> (中国科学院研究生院 北京 100049)<sup>2</sup>

**摘 要** 受限路网中移动对象的轨迹预测已成为智能交通关注的热点,被广泛应用于应急保障、车辆导航等领域。但在仅知道移动对象近期轨迹的情况下,现有方法难以解决其未来路径的在线预测问题。提出一种新的在线轨迹预测方法 LPP,即通过全局学习机制发现最长频繁路径,构造快速访问结构 LPP-tree。基于移动对象近期轨迹可对未来运动路径进行快速在线预测。通过实验,验证了该方法的有效性。

**关键词** 受限路网,移动对象,在线轨迹预测,全局学习机制

中图法分类号 TP311 文献标识码 A

## Network-constrained On-line Path Prediction Based on Global Learning Mechanism

XU Huai-ye<sup>1,2</sup> DING Zhi-ming<sup>1</sup> LIU Kui-en<sup>1</sup> XU Jia-jie<sup>1</sup>

(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)<sup>1</sup>

(Graduate University of Chinese Academy of Sciences, Beijing 100049, China)<sup>2</sup>

**Abstract** The trajectory prediction of the moving object in the network-constrained has been the hot spot of the intelligent traffic's attention. And it has been widely used in the area of emergency security, GPS and so on. But if we only know the recent trajectory of the moving object, we couldn't predict its future trajectory with the existing methods. A trajectory prediction's method LPP(longest frequent path prediction) was put forward, which could construct the fast accessing structure LPP-tree through the global learning mechanism to find out the longest frequent trajectory. Based on the recent trajectory of the moving object, one could predict its future trajectory swiftly online. And the experiment proves the validity of this method.

**Keywords** Network-constrained, Moving object, On-line trajectory prediction, Global learning mechanism

### 1 前言

随着移动通信技术和全球定位技术的飞速发展,各种便携式移动通信设备被逐步安装到车辆等移动对象上,从而积累了各种移动对象的大量轨迹。利用这些轨迹挖掘出有价值的信息,然后再将其应用到交通管理中,这是智能化交通管理的一个方面。轨迹预测技术具有高度的研究价值和广泛的应用领域,在线预测移动对象未来的运动路径有利于很多服务信息的及时投递。例如,通过轨迹预测技术可以预测将要前往拥堵路段的汽车,服务商就能将相关路段的拥堵信息及时发给汽车司机,此时,司机就可以提前调整行驶路线,避开拥堵路段。可以说,移动对象的轨迹预测技术已经得到了人们日益广泛的重视。

现在,轨迹预测技术大体分为两类:基于欧氏空间的轨迹预测和基于受限路网的轨迹预测。基于欧氏空间的预测又分为基于向量的预测和基于模式的预测。基于向量的预测,主要是通过数学公式模拟移动对象的运动轨迹。文献[1]提出

的 RMF 方法能够模拟移动对象的短期运动趋势。基于模式的预测,主要是根据单个移动对象的历史轨迹,通过挖掘出它的轨迹模式来预测这个移动对象的未来运动路径。文献[2]提出了挖掘移动对象轨迹模式的 LM 算法。文献[3]提出了 LM+ 算法,用来增量挖掘轨迹模式。但是,基于欧氏空间的预测存在一定局限性,因为大多数移动对象都是在受限路网中运动,必然受到路网的限制,不能在空间中随意运动。

学者对受限路网中移动对象的轨迹预测提出一些方法。文献[4]详细介绍了受限路网的建模方法。文献[5]提出了在给定起始点和目的地的情况下预测移动对象轨迹的方法。文献[6,7]通过移动对象的位置更新策略来预测移动对象将来的位置。文献[8]通过建立移动对象位置和速度的统计模型来挖掘移动对象的轨迹模式。文献[9]提出了一种在历史轨迹不确定的情况下预测未来路径的方法。文献[10]提出一种基于路径频繁度的聚类方法,即挖掘出热点路径进行轨迹预测。文献[11]先建立移动对象的转弯模型,利用基于最大相似度的贪婪算法进行路径预测。

到稿日期:2011-09-06 返修日期:2011-11-25 本文受国家自然科学基金项目(60970030),教育部留学回国人员科研启动基金项目(外教司[2006J331])资助。

徐怀野(1987-),男,硕士生,主要研究方向为数据库与知识库系统、移动计算,E-mail:huaiye@nfs.iscas.ac.cn;丁治明(1966-),男,博士,研究员,主要研究方向为数据库与知识库系统、移动计算、信息检索;刘奎恩(1983-),男,博士,助理研究员,主要研究方向为时空数据管理、移动对象数据库及数据挖掘。

在受限路网中,现有的移动对象轨迹预测方法,是建立在已知这个移动对象大量历史轨迹基础上的,先挖掘出这个对象的移动模式,再利用挖掘出的移动模式来预测这个移动对象的未来运动路径。但是,这类方法需要移动对象自己的大量轨迹作为基础,因此有很大的局限性。现实中,往往是在仅知道移动对象当前所在路径和一条近期轨迹的情况下,预测其未来运动路径。在这种情形下,上述轨迹预测方法无法进行有效的预测,这是轨迹预测技术面临的新挑战。

交通路网是一种典型的受限路网。通过观察发现,受限路网中各种移动对象的轨迹与受限路网联系紧密。单个移动对象在受限路网中的运动是随机的、不确定的,而大量移动对象在路网中却呈现出一定的运动特点,即在交叉路口进行道路选择时表现出一定的规律性。

本文提出一种轨迹预测方法,即以受限路网固有性质为基础,通过全局学习机制,充分利用已有大量移动对象的历史轨迹构建 LPP-tree,并进行统计分析。利用统计出的信息结合移动对象本次的轨迹,对其未来运动路径进行预测。这为受限路网中移动对象的轨迹预测带来了极大的灵活性和可行性,在智能交通管理方面有着重要的意义,应用前景广阔。

本文主要贡献点:

- 1)提出了利用全局学习机制对所有轨迹进行统计分析,从而指导移动对象轨迹预测的思想。
- 2)提出了一种快速索引结构 LPP-tree 及其构建方法。
- 3)提出了一种基于 LPP-tree 结构的移动对象轨迹预测方法。

本文第 2 节介绍问题定义;第 3 节介绍轨迹预测的思路和步骤,并进行分析;第 4 节进行实验,并给出实验结果;最后总结全文。

## 2 定义

**定义 1(原子路径)** 受限路网表示为有向图  $G=(V,E)$ ,其中  $V$  是顶点集合,每个顶点表示一个交叉路口; $E$  是有向边集合,每条边表示一个最小的路径单元,即原子路径。 $E=\{e_1,e_2,\dots,e_n\}$ ,其中  $e_i$  为原子路径, $1\leq i\leq n$ 。

**定义 2(移动轨迹)** 移动轨迹  $T$  表示移动对象在受限路网中相继走过的原子路径序列, $T=\langle e_1,\dots,e_j\rangle$ ,其中  $e_k\in E$ , $1\leq k\leq n$ ,序列中项的个数称为该移动序列的长度,记为  $|T|$ 。

**定义 3(后缀)** 给定移动轨迹  $T=\langle e_1,e_2,\dots,e_n\rangle$ , $T_1=\langle a_1,a_2,\dots,a_m\rangle$ , $m\leq n$ ,轨迹  $T_1$  为  $T$  的后缀,当且仅当对  $\forall i$ , $1\leq i\leq m$ , $e_{i+n-m}=a_i$  成立。

例如,给定串  $\langle e_1,e_2,e_3,e_4\rangle$ ,则它的后缀为  $\langle e_1,e_2,e_3,e_4\rangle$ , $\langle e_2,e_3,e_4\rangle$ , $\langle e_3,e_4\rangle$ , $\langle e_4\rangle$ 。那么给定轨迹序列  $T=\langle e_1,\dots,e_j\rangle$ ,则  $T$  的后缀(Suffix)记为  $TS_k$ , $1\leq k\leq |T|$ ,表示轨迹序列  $T$  以第  $k$  项为首的后缀。

**定义 4(LPP-tree)** 表示移动轨迹的后缀树,每个结点表示一个原子路径,从根结点到任意叶结点所对应的原子路径序列表示一个轨迹的后缀。

LPP-tree 有以下特点:

1. 树中存在一个根结点 Root,根结点有两个域  $size$  和  $total$ 。 $size$  表示路网中原子路径总数, $total$  表示建成该树所用历史轨迹的条数。

2. 当有大量历史轨迹时,树的规模只与  $size$  有关,因此可

以用其来统计海量历史轨迹。

3. 树中除根结点外,每个结点有 3 个域  $atom$ 、 $sup$ 、 $flag$ 。 $atom$  表示所代表的原子路径, $sup$  表示经过该原子路径的轨迹个数, $flag$  表示该原子路径是否频繁。

4. 树中除根结点外,每个结点的  $sup$  值大于等于其子结点的  $sup$  值。

## 3 轨迹预测的思路与步骤

### 3.1 LPP-tree 的建立

根据已有历史轨迹建立一棵 LPP-tree。依次插入每条轨迹的所有后缀,同时记录轨迹的统计信息,如  $sup$  值,即可形成一棵 LPP-tree。因为只需要对所有轨迹扫描一次,所以效率很高。如图 1 所示,由两条轨迹  $\langle A,B,C,D\rangle$  和  $\langle B,C,E,F\rangle$  建立一棵 LPP-tree。对第一条轨迹  $\langle A,B,C,D\rangle$ ,将其所有后缀  $\langle A,B,C,D\rangle$ 、 $\langle B,C,D\rangle$ 、 $\langle C,D\rangle$ 、 $\langle D\rangle$  依次插入树中,再将第二条轨迹  $\langle B,C,E,F\rangle$  的所有后缀  $\langle B,C,E,F\rangle$ 、 $\langle C,E,F\rangle$ 、 $\langle E,F\rangle$ 、 $\langle F\rangle$  依次插入树中。

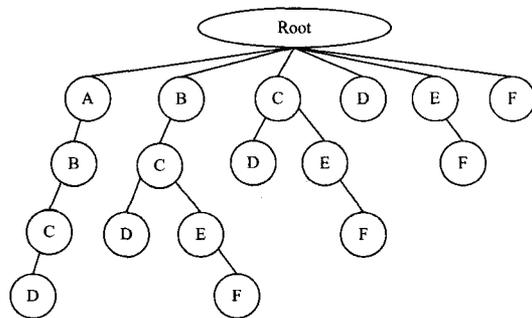


图 1 LPP-tree 结构

#### 算法 1 建立 LPP-tree

输入:轨迹集合  $M=\{T_1,T_2,\dots,T_n\}$

输出:由这些轨迹建立的 LPP-tree:Root

1. for each Trajectory  $\in$  轨迹集合  $M$  do
2.   for each Trajectory suffix do
3.     current=Root;
4.     for each  $e_k\in$  Trajectory suffix do
5.       if current 的子结点中存在  $e_k$
6.         更新 current 子结点的  $sup$
7.       else//InitNode 表示生成新结点
8.         InitNode(newNode);
9.         current.child=newNode;
10.       current=current.child;
11.     end
12.   end
13. end

第 1,2 行,对每一条轨迹的所有后缀执行以下操作;第 3 行,将每个后缀的初始插入位置设定在根 Root;第 4-11 行,依次将后缀插入到树中。 $min\_sup$  即路径支持度阈值,表示至少有多少条轨迹经过此路径,此路径才是频繁的。提供一个路径支持度阈值,以提高轨迹预测的准确率。在 LPP-tree 建成之后遍历该树,树中每个结点根据  $sup$  值来设定该结点的  $flag$ ,以标识此结点是否频繁。当结点的  $sup$  大于等于  $min\_sup$  时,置结点的  $flag$  为 true;否则置结点的  $flag$  为 false。

### 3.2 LPP-tree 的分建与合并

在现代交通管理系统中,交通流和时间有着紧密的联系。

交通流随着时间的推移而不断变化,即同一个路网在不同的时间段,交通流表现出不同的特征。在轨迹预测时,结合时间这个因素,才能更好地进行预测。因此,我们可以首先根据需要对不同的时间段,然后根据不同的时间段建立相应的 LPP-tree。同时给树附加一个时间属性,用来标记这棵树所对应的时间段,那么此时就会存在多棵 LPP-tree,每棵树对应一个时间段。

这时,每条历史轨迹都包含一个时间域,用来标记轨迹所在时间段,利用相应时间段的轨迹建立 LPP-tree。当按时间段建立相应的树后,进行预测时,待预测的移动对象就需要指明时间域。首先,根据这个时间域确定相符合的 LPP-tree,然后在这个选定的树上进行轨迹预测。这样预测更精确,更有时效性。

为了使预测更加精确,充分体现交通流的时效性,可以根据时间建立多棵 LPP-tree。但为了从整体上了解路网情况,以便从宏观上把握整个路网,也可以将多棵 LPP-tree 合并为一棵,时间域则为合并前多棵树时间域的并集。

#### 算法 2 合并两棵 LPP-tree

输入:两棵待合并的 LPP-tree:Root1,Root2

输出:合并成一棵树 Root1

1. for each SNode $\in$ Root2
2. if Root1 中存在与 SNode 对应的结点 TNode
3. TNode.sup=TNode.sup+SNode.sup;
5. else //复制 SNode 结点
6. TNode=copy(SNode);
7. 将 TNode 插入 Root1 中对应位置;
8. end

如果将 Root1 和 Root2 合并为一棵树 Root1,只需遍历 Root2,对 Root2 中每一个结点,在 Root1 中找到与之相对应的结点,同时修改 Root1 中该结点 sup 值;如果 Root1 中没有与之相对应的结点,那么生成一个新结点并插入到树中相应位置。

### 3.3 LPP-tree 增量更新算法

随着时间的推移,不断有新轨迹需要加入,旧轨迹需要删除。这就需要及时更新 LPP-tree,以反映受限路网最新动态,利于轨迹预测。增量更新算法提供添加新轨迹和删除旧轨迹两种更新操作。

那么,更新缓冲区中既有添加新轨迹的请求,也有删除旧轨迹的请求。增量更新算法在处理更新请求时采用了“懒操作”方式,即设置一个更新缓冲区,同时为缓冲区设置两个域:Timeout 和 Count。其中 Timeout 表示更新等待时间,Timeout 在第一个更新请求到达开始计时,Count 表示更新请求的总个数。当有添加或删除轨迹的请求到来时,将这些更新请求放置在缓冲区中,当 Timeout $\geq$ max\_time 或者 Count $\geq$ max\_num 时(max\_time 表示最长等待时间,max\_num 表示最大请求个数),先利用缓冲区中所有轨迹建立一棵更新树,然后将这棵更新树合并到原树中。这就避免了频繁的更新操作,大大提高了更新效率。

#### 算法 3 增量更新

输入:更新缓冲区中的轨迹集合 M={T<sub>1</sub>,T<sub>2</sub>,...,T<sub>n</sub>}

输出:更新后的 LPP-tree

1. 根据轨迹集合建立更新树 Update-tree
2. 将更新树 Update-tree 合并到原树 LPP-tree 中

首先建立更新树,混合了添加和删除轨迹的建树过程在处理删除操作时,只需要将相应节点 sup 值减 1 即可;如果相应结点不存在,则生成相应结点,将 sup 值置为-1。将更新树和原树合并时,遍历更新树,对于更新树中的每一个结点,在原树中找到相对应的结点,同时修改原树相应结点 sup 值。只有当原树中某个结点 sup 值等于 0 时,才将这个结点释放。

### 3.4 LPP 预测

移动对象在受限路网中运动,必然要受到路网的限制,其中一个重要限制就是:移动对象所在原子路径直接决定了它当前的路径选择。因此,移动对象当前所在原子路径是进行轨迹预测的关键,本次的轨迹起到一个参考作用。所以没有必要考察移动对象本次的全部轨迹,只需要重点考察当前所在的原子路径,同时综合考虑本次轨迹即可。

给定轨迹 T,若 |T| $\leq$ length,直接使用即可;若 |T|>length,轨迹 T 将被截断为 T<sub>0</sub>,且 |T<sub>0</sub>|=length。其中,length 表示预测时所需考察的原子路径个数。例如,给定移动对象本次轨迹为(e<sub>1</sub>,e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub>),其中 e<sub>5</sub> 为移动对象当前所在原子路径。令 length=4,则预测时仅需要(e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub>),其中 e<sub>1</sub> 被舍弃。当 length 一定时,给定移动对象本次轨迹,预测其未来运动路径的时间复杂度为常数且很低,这就为在线轨迹预测奠定了良好的基础。

#### 算法 4 轨迹预测

输入:LPP-tree:Root,待预测移动对象的本次轨迹为(e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub>),

其中 e<sub>5</sub> 为移动对象当前所在原子路径

输出:对该移动对象未来运动路径的预测

1. for each Trajectory suffix do
2. current=Root;
3. for each e<sub>i</sub> $\in$ Trajectory suffix do
4. if current 的子结点中存在 e<sub>k</sub>
5. current=current.child;
6. else
7. break;
8. if current.atom==e<sub>5</sub>  
//即从根结点开始存在一条(e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub>)的后缀
9. path=sup\_max(current);  
//sup\_max()表示选择指定结点的孩子中 sup 值最大的结点;
10. while path.flag==true do
11. path=sup\_max(path);
12. end
13. end
14. end

预测分为两步:

步骤 1 即第 1—9 行,按长度依次考察(e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub>)的所有后缀,首先考察后缀(e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub>),可知 e<sub>5</sub> 是移动对象当前所在的原子路径。如果从根结点开始,存在 e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub> 的结点序列,针对结点 e<sub>5</sub> 执行步骤 2,其他后缀就不需要考察。如果从根结点开始,不存在 e<sub>2</sub>,e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub> 的结点序列,那么就接着考察后缀(e<sub>3</sub>,e<sub>4</sub>,e<sub>5</sub>),依次类推。

步骤 2 即 10—12 行,如果结点 Begin(此处即 e<sub>5</sub>)的所有子结点中 flag 全为 false,即不存在频繁原子路径时,就选择 sup 值最大的一个子结点,用这个结点所表示的原子路径作为对这个移动对象未来运动路径的预测。如果结点 Begin 的所有子结点的 flag 有一个或多个为 true,即存在频繁原子

路径,就选择  $sup$  值最大的一个子结点  $child$ ,对  $child$  执行步骤 2,直到某个结点  $End$  的所有子结点全为  $false$ 。输出从祖先结点  $Begin$  到子孙结点  $End$  这条树中路径所对应的原子路径序列作为预测路径,那么这个预测路径可能是单个原子路径,也可能是原子路径序列。

#### 4 实验分析

为了验证上述算法的性能,我们进行了模拟实验。实验的硬件平台为: Intel Pentium Dual CPU E2200 2.2GHz,内存为 2GB。软件平台为 Linux,实验程序用 C++ 编写,编译工具为 g++ 3.4.3。

鉴于现有研究中,没有依据全局统计分析来指导单个移动对象轨迹预测的方法,因此本文没有比较实验。在 50~300 个原子路径的情况下,将 100000 条轨迹作为输入,利用这些轨迹构建 LPP-tree,然后对 10000 条轨迹进行预测,其中每条轨迹长度的取值范围为 [7,9]。表 1 给出整个实验的参数。

表 1 实验中主要参数

参数	值	含义
n	50~300	原子路径总数
length	5	预测所需的原子路径个数
min_sup	50~350	频繁路径的阈值

图 2 为轨迹的平均预测时间,其中  $min\_sup=300$ 。从实验结果可以看出,使用 LPP 算法,轨迹预测所需的时间极短,这就能够保障在线轨迹预测。

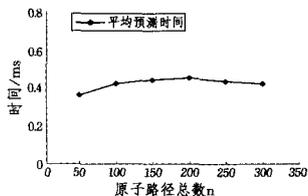


图 2 平均预测时间

图 3 为建树时间和平均预测时间的对比,其中  $min\_sup=300$ 。可以看出,随着原子路径数量的增加,树的规模在不断增长,建树时间也不断增长,建立的 LPP-tree 的规模与原子路径数量紧密相关。从图中可以看出,和建树相比,轨迹预测只需很短的时间。

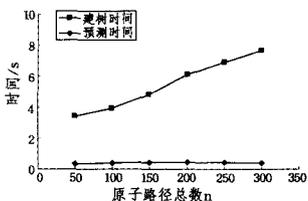


图 3 建树时间和平均预测时间对比

图 4 为支持度阈值与预测准确率的关系,其中  $n=200$ 。随着支持度阈值的增加,预测准确率相应地增加,当支持度阈值达到一定值时,预测准确率趋于平稳。轨迹预测结果为一个序列  $\langle e_i, \dots, e_j \rangle$ ,这个序列由一个或多个原子路径组成。当支持度阈值较小时,预测出的轨迹含有多个原子路径,所含原子路径越多,预测的准确率就越低。当支持度阈值较大时,预测出的轨迹所含原子路径个数较少,预测的准确率较高。

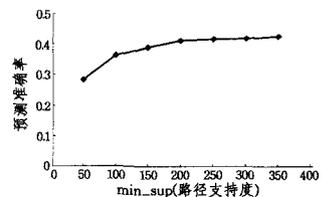


图 4 支持度阈值与预测准确率的关系

**结束语** 在现代交通管理中,针对仅知道移动对象近期轨迹需要预测其未来运动路径的问题,利用受限路网固有性质以及受限路网与轨迹的联系,本文提出一种基于全局学习机制的轨迹预测算法 LPP,并构造访问结构 LPP-tree。利用 LPP-tree 结构和移动对象的近期轨迹,可对未来运动路径进行快速在线预测。

#### 参考文献

- [1] Tao Y, Faloutsos C, Papadias D, et al. Prediction and indexing of moving objects with unknown motion patterns [C] // The 2004 ACM SIGMOD Int'l Conf on Management of Data. New York: ACM, 2004: 611-622
- [2] Peng W-C, Chen M-S. Mining user moving patterns for personal data allocation in a mobile computing system [C] // The 2000 Int'l Conf on Parallel Processing. Los Alamitos, CA: IEEE Computer Society Press, 2000: 573-580
- [3] Peng W-C, Chen M-S. Developing data allocation schemes by incremental mining of user moving patterns in a mobile computing system [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(1): 70-85
- [4] 孟小峰, 丁治明. 移动数据管理: 概念与技术 [M]. 北京: 清华大学出版社, 2009
- [5] Kim S-W, Won J-I, Kim J-D, et al. Path prediction of moving objects on road networks through analyzing past trajectories [C] // The 11th Int'l Conf on Knowledge-Based Intelligent Information and Engineering Systems. Berlin: Springer, 2007: 379-389
- [6] Ding Zhi-ming, Güting R H. Managing moving objects on dynamic transportation networks [C] // The 16th Int'l Conf on Scientific and Statistical Database Management. Washington, IEEE Computer Society, 2004: 287-296
- [7] Ding Zhi-ming, Zhou Xiao-fang. Location update strategies for network constrained moving objects [C] // The 13th Int'l Conf on Database Systems for Advanced Applications. Berlin: Springer, 2008: 644-652
- [8] Stauffer C, Eric W, Grimson L. Learning Patterns of Activity Using Real-time Tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 747-757
- [9] 郭黎敏, 丁治明, 胡泽林, 等. 基于路网的不确定性轨迹预测 [J]. 计算机研究与发展, 2010, 47(1): 104-112
- [10] Li Xiao-lei, Han Jia-wei, Lee, et al. Traffic density-based discovery of hot routes in road networks [C] // The International Symposium on Spatial and Temporal Databases. Boston: Springer, 2007: 441-459
- [11] Jeung H, Yiu M L, Zhou Xiao-fang, et al. Path prediction and predictive range querying in road network databases [J]. The Very Large Database Journal, 2010, 19(4): 585-602