



(12) 发明专利

(10) 授权公告号 CN 102279856 B

(45) 授权公告日 2013. 10. 02

(21) 申请号 201010201129. 0

CN 101529865 A, 2009. 09. 09,

(22) 申请日 2010. 06. 09

US 2009150343 A1, 2009. 06. 11,

(73) 专利权人 阿里巴巴集团控股有限公司

US 2009282013 A1, 2009. 11. 12,

地址 英属开曼群岛大开曼岛资本大厦一座  
四层 847 号邮箱

US 2010082640 A1, 2010. 04. 01,

JP 2007233443 A, 2007. 09. 13,

审查员 张文

(72) 发明人 廖剑 林锋 张寿松 张勤

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

代理人 逯长明 王宝筠

(51) Int. Cl.

G06F 17/30 (2006. 01)

(56) 对比文件

US 2002138525 A1, 2002. 09. 26,

US 2005267902 A1, 2005. 12. 01,

US 2003126235 A1, 2003. 07. 03,

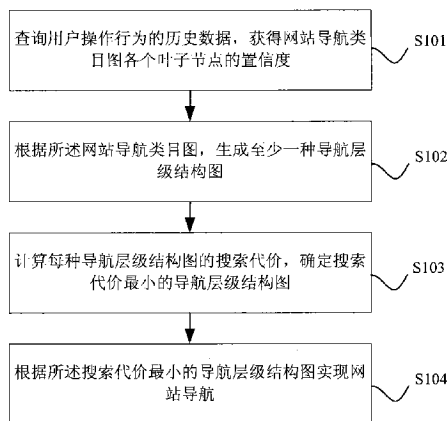
权利要求书3页 说明书9页 附图3页

(54) 发明名称

一种网站导航实现方法及系统

(57) 摘要

本申请公开了一种网站导航实现方法及系统。一种网站导航实现方法包括：查询用户操作行为的历史数据，获得网站导航类目图各个叶子节点的置信度，所述网站导航类目图对应于网站导航类目的初始静态层级关系；根据所述网站导航类目图，生成至少一种导航层级结构图，其中，每种导航层级结构图具有与所述网站导航类目图相同的叶子节点；计算每种导航层级结构图的搜索代价，确定搜索代价最小的导航层级结构图；根据所述搜索代价最小的导航层级结构图实现网站导航。应用上述方案进行网站导航，可以有效减少导航时用户的点击操作，从而减轻网站服务器的负担，节省网络带宽资源。从用户的角度来看，也能够更快捷、更方便地找到自己期望的结果。



1. 一种网站导航实现方法,其特征在于,包括:

查询用户操作行为的历史数据,获得网站导航类目图各个叶子节点的置信度,所述网站导航类目图对应于网站导航类目的初始静态层级关系,所述网站导航类目图为树形结构,所述叶子节点为所述树形结构的最末端节点;

根据所述网站导航类目图,生成至少一种导航层级结构图,其中,每种导航层级结构图具有与所述网站导航类目图相同的叶子节点;

计算每种导航层级结构图的搜索代价,确定搜索代价最小的导航层级结构图;

根据所述搜索代价最小的导航层级结构图实现网站导航;

其中,导航层级结构图搜索代价的计算方法包括:

根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的分支位置,计算该导航层级结构图的观察代价;

根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的层级位置,计算该导航层级结构图的点击代价;

根据预设的观察代价权重和点击代价权重,对所述观察代价和点击代价进行加权处理,获得该导航层级结构图的搜索代价。

2. 根据权利要求 1 所述的方法,其特征在于,在所生成的导航层级结构图中,各个叶子节点按照置信度从大到小进行排列;其中,叶子节点的排列顺序对应于该节点在网页中的观察顺序。

3. 根据权利要求 1 所述的方法,其特征在于,根据以下约束条件生成导航层级结构图:

置信度大于预设阈值的叶子节点位于导航层级结构图的第一级、

和 / 或

导航层级结构图的第一级节点个数,不大于网页所允许的显示数量、

和 / 或

导航层级结构图的同一层级中,不同时出现某个节点与其在所述初始静态层级关系中的祖先节点。

4. 根据权利要求 1 所述的方法,其特征在于,所述根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的分支位置,计算该导航层级结构图的观察代价,包括:

预先为位于导航层级结构图第一层级不同位置的节点设置观察代价,其中,位置靠前节点的观察代价小于位置靠后节点的观察代价;

对导航层级结构图中所有第一层级节点的观察代价和置信度的乘积求和,得到所述导航层级结构图的观察代价,其中,第一层级节点的置信度为该第一层级节点所有子孙叶子节点的置信度之和。

5. 根据权利要求 4 所述的方法,其特征在于,利用以下公式计算导航层级结构图的观察代价  $w\_penalty_{ij}$ :

$$w\_penalty_{ij} = w\_penalty(query_i, tr_j) = \sum_{k=1}^K w_k * h(query_i, v\_cat_k)$$

其中:

$query_i$  表示搜索请求  $i$ ;  $tr_j$  表示第  $j$  种导航层级结构图;

$v\_cat_k$  表示导航层级结构图第一层级的第  $k$  个节点;

$w_k$  表示  $v\_cat_k$  的观察代价；

$h(query_i, v\_cat_k)$  表示  $v\_cat_k$  的置信度,  $h(query_i, v\_cat_k)$  等于  $v\_cat_k$  所有子孙叶子节点的置信度之和；

$K$  为导航层级结构图的第一级的节点总数。

6. 根据权利要求 5 所述的方法, 其特征在于  $w_k=k$ 。

7. 根据权利要求 1 所述的方法, 其特征在于, 所述根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的层级位置, 计算该导航层级结构图的点击代价, 包括:

对导航层级结构图中所有叶子节点的置信度与其所处层级位置的乘积求和, 得到所述导航层级结构图的点击代价。

8. 根据权利要求 7 所述的方法, 其特征在于, 利用以下公式计算导航层级结构图的点击代价  $c\_penalty_{ij}$ :

$$c\_penalty_{ij} = c\_penalty(query_i, tr_j) = \sum_{k=1}^K level_k * h(query_i, leaf\_cat_k)$$

其中:

$query_i$  表示搜索请求  $i$ ;  $tr_j$  表示第  $j$  种导航层级结构图;

$leaf\_cat_k$  表示导航层级结构图中的第  $k$  个叶子节点;

$level_k$  表示  $leaf\_cat_k$  在导航层级结构图中所处的层级位置;

$h(query_i, leaf\_cat_k)$  表示  $leaf\_cat_k$  的置信度;

$K$  为导航层级结构图的叶子节点总数。

9. 一种网站导航系统, 其特征在于, 包括:

历史数据查询模块, 用于查询用户操作行为的历史数据, 获得网站导航类目图各个叶子节点的置信度, 所述网站导航类目图对应于网站导航类目的初始静态层级关系, 所述网站导航类目图为树形结构, 所述叶子节点为所述树形结构的最末端节点;

导航层级结构图生成模块, 用于根据所述网站导航类目图, 生成至少一种导航层级结构图, 其中, 每种导航层级结构图具有与所述网站导航类目图相同的叶子节点;

搜索代价计算模块, 用于计算每种导航层级结构图的搜索代价;

导航实现模块, 用于根据搜索代价最小的导航层级结构图实现网站导航;

其中, 所述搜索代价计算模块包括:

观察代价计算子模块, 用于根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的分支位置, 计算该导航层级结构图的观察代价;

点击代价计算子模块, 用于根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的层级位置, 计算该导航层级结构图的点击代价;

搜索代价计算子模块, 用于根据预设的观察代价权重和点击代价权重, 对所述观察代价和点击代价进行加权处理, 获得该导航层级结构图的搜索代价。

10. 根据权利要求 9 所述的系统, 其特征在于, 在所述导航层级结构图生成模块生成的导航层级结构图中, 各个叶子节点按照置信度从大到小进行排列; 其中, 叶子节点的排列顺序对应于该节点在网页中的观察顺序。

11. 根据权利要求 9 所述的系统, 其特征在于, 所述导航层级结构图生成模块根据以下约束条件生成导航层级结构图:

置信度大于预设阈值的叶子节点位于导航层级结构图的第一级、  
和 / 或

导航层级结构图的第一级节点个数, 不大于网页所允许的显示数量、  
和 / 或

导航层级结构图的同一层级中, 不同时出现某个节点与其在所述初始静态层级关系中的祖先节点。

12. 根据权利要求 9 所述的系统, 其特征在于, 所述观察代价计算子模块预先为位于导航层级结构图第一层级不同位置的节点设置观察代价, 其中, 位置靠前节点的观察代价小于位置靠后节点的观察代价; 对导航层级结构图中所有第一层级节点的观察代价和置信度的乘积求和, 得到所述导航层级结构图的第一层级节点的观察代价, 其中, 第一层级节点的置信度为该第一层级节点所有子孙叶子节点的置信度之和。

13. 根据权利要求 12 所述的系统, 其特征在于, 所述观察代价计算子模块利用以下公式计算导航层级结构图的第一层级节点观察代价  $w\_penalty_{ij}$ :

$$w\_penalty_{ij} = w\_penalty(query_i, tr_j) = \sum_{k=1}^K w_k * h(query_i, v\_cat_k)$$

其中:

$query_i$  表示搜索请求  $i$ ;  $tr_j$  表示第  $j$  种导航层级结构图;

$v\_cat_k$  表示导航层级结构图第一层级的第  $k$  个节点;

$w_k$  表示  $v\_cat_k$  的观察代价;

$h(query_i, v\_cat_k)$  表示  $v\_cat_k$  的置信度,  $h(query_i, v\_cat_k)$  等于  $v\_cat_k$  所有子孙叶子节点的置信度之和;

$K$  为导航层级结构图的第一级的节点总数。

14. 根据权利要求 9 所述的系统, 其特征在于, 所述点击代价计算子模块对导航层级结构图中所有叶子节点的置信度与其所处层级位置的乘积求和, 得到所述导航层级结构图的点击代价。

15. 根据权利要求 14 所述的系统, 其特征在于, 所述点击代价计算子模块利用以下公式计算导航层级结构图的叶子节点点击代价  $c\_penalty_{ij}$ :

$$c\_penalty_{ij} = c\_penalty(query_i, tr_j) = \sum_{k=1}^K level_k * h(query_i, leaf\_cat_k)$$

其中:

$query_i$  表示搜索请求  $i$ ;  $tr_j$  表示第  $j$  种导航层级结构图;

$leaf\_cat_k$  表示导航层级结构图中的第  $k$  个叶子节点;

$level_k$  表示  $leaf\_cat_k$  在导航层级结构图中所处的层级位置;

$h(query_i, leaf\_cat_k)$  表示  $leaf\_cat_k$  的置信度;

$K$  为导航层级结构图的叶子节点总数。

## 一种网站导航实现方法及系统

### 技术领域

[0001] 本申请涉及计算机应用技术领域,特别是涉及一种网站导航实现方法及系统。

### 背景技术

[0002] 随着互联网的迅速发展,网站所包含的内容越来越丰富,为了令用户能够迅速找到自己感兴趣的内容,很多网站都提供了导航功能。目前,较为常用的一种网站导航方式是基于分层数据驱动的导航方法,即形成图结构的层次化的类目导航体系;对于用户查询,通过某些方法匹配到某个导航状态节点,推出该节点下一级的导航状态;同时,用户可以遍历导航结构的各个节点,进行动态导航。

[0003] 以某电子商务网站为例,假如用户希望搜索 apple 品牌的笔记本电脑,输入搜索关键词“apple”后,首先进行一级类目导航,导航结果见图 1a;用户根据实际需求,点击一级类目导航结果中的“Electronics”导航节点,进入二级类目导航,导航结果见图 1b;用于再次点击二级类目导航结果中的“Computer&Accessories”导航节点,进入三级类目导航,导航结果见图 1c;最终在第三级导航结果中找到所期望的类目“Laptops”。

[0004] 在上述导航方式中,用户最终所需要的搜索结果一般位于导航体系的最末一级,也就是导航类目图的叶子节点。但是用户找到叶子节点的过程需要进行逐级的多次点击操作,而针对用户的每次点击操作,网站服务器都要向用户反馈相应的导航结果,这会造成网站服务器负担及网络带宽占用的增加。此外,从用户的角度来看,除了需要多次点击操作之外,还需要多次观察判断,才能正确选择到所期望的下一级节点,造成使用上的不便。

### 发明内容

[0005] 为解决上述技术问题,本申请实施例提供一种网站导航实现方法及系统,以减小网站服务器及网络带宽的占用,同时简化用户的操作,提高导航效率。本申请所提供的技术方案如下:

[0006] 一种网站导航实现方法,包括:

[0007] 查询用户操作行为的历史数据,获得网站导航类目图各个叶子节点的置信度,所述网站导航类目图对应于网站导航类目的初始静态层级关系;

[0008] 根据所述网站导航类目图,生成至少一种导航层级结构图,其中,每种导航层级结构图具有与所述网站导航类目图相同的叶子节点;

[0009] 计算每种导航层级结构图的搜索代价,确定搜索代价最小的导航层级结构图;

[0010] 根据所述搜索代价最小的导航层级结构图实现网站导航;

[0011] 其中,导航层级结构图搜索代价的计算方法包括:

[0012] 根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的分支位置,计算该导航层级结构图的观察代价;

[0013] 根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的层级位置,计算该导航层级结构图的点击代价;

[0014] 根据预设的观察代价权重和点击代价权重,对所述观察代价和点击代价进行加权处理,获得该导航层级结构图的搜索代价。

[0015] 一种网站导航系统,包括:

[0016] 历史数据查询模块,用于查询用户操作行为的历史数据,获得网站导航类目图各个叶子节点的置信度,所述网站导航类目图对应于网站导航类目的初始静态层级关系;

[0017] 导航层级结构图生成模块,用于根据所述网站导航类目图,生成至少一种导航层级结构图,其中,每种导航层级结构图具有与所述网站导航类目图相同的叶子节点;

[0018] 搜索代价计算模块,用于计算每种导航层级结构图的搜索代价;

[0019] 导航实现模块,用于根据搜索代价最小的导航层级结构图实现网站导航;

[0020] 其中,所述搜索代价计算模块包括:

[0021] 观察代价计算子模块,用于根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的分支位置,计算该导航层级结构图的观察代价;

[0022] 点击代价计算子模块,用于根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的层级位置,计算该导航层级结构图的点击代价;

[0023] 搜索代价计算子模块,用于根据预设的观察代价权重和点击代价权重,对所述观察代价和点击代价进行加权处理,获得该导航层级结构图的搜索代价。

[0024] 本申请实施例提供的技术方案,仍然基于图结构的层次化的类目导航体系,首先根据网站导航类目的初始静态层级关系生成多种导航层级结构图,然后依据其中搜索代价最小的导航层级结构图实现网站导航。其中,搜索代价由观察代价和点击代价决定,因此,应用上述方案进行网站导航,可以有效减少导航时用户的点击操作,从而减轻网站服务器的负担,节省网络带宽资源。从用户的角度来看,也能够更快捷、更方便地找到自己期望的结果。

### 附图说明

[0025] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请中记载的一些实施例,对于本领域普通技术人员来讲,还可以根据这些附图获得其他的附图。

[0026] 图 1a- 图 1c 分别为现有技术网站导航方式的三级导航结果示意图;

[0027] 图 2 为本申请实施例网站导航实现方法的流程图;

[0028] 图 3 为本申请实施例提供的一种网站导航类目图;

[0029] 图 4a- 图 4c 分别为本申请实施例提供的三种导航层级结构图;

[0030] 图 5 为本申请实施例网站导航系统的结构示意图;

[0031] 图 6 为本申请实施例搜索代价计算模块的结构示意图。

### 具体实施方式

[0032] 首先对本申请实施例所提供的网站导航实现方法进行说明,该方法包括以下步骤:

[0033] 查询用户操作行为的历史数据,获得网站导航类目图各个叶子节点的置信度,所

述网站导航类目图对应于网站导航类目的初始静态层级关系；

[0034] 根据所述网站导航类目图，生成至少一种导航层级结构图，其中，每种导航层级结构图具有与所述网站导航类目图相同的叶子节点；

[0035] 计算每种导航层级结构图的搜索代价，确定搜索代价最小的导航层级结构图；

[0036] 根据所述搜索代价最小的导航层级结构图实现网站导航。

[0037] 上述方法仍然基于图结构的层次化的类目导航体系，与现有技术的区别在于，上述方法在生成导航信息时，引入了搜索代价作为依据，在所有的导航方式中选择搜索代价最小的方案，因此能够较好地减少用户的搜索成本，也减少了用户与网站服务器的信息交互。并且，导航方式也可以更加多样化，例如可以出现叶子类目和父类目混合推荐的样式。

[0038] 为了使本技术领域的人员更好地理解本申请中的技术方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员所获得的所有其他实施例，都应当属于本申请保护的范围。

[0039] 图 2 所示为本申请实施例所提供的一种网站导航实现方法的流程图，包括以下步骤：

[0040] S101，查询用户操作行为的历史数据，获得网站导航类目图各个叶子节点的置信度；

[0041] 其中，置信度用来描述导航信息相对应用户搜索请求的可信度。例如，对于某个用户搜索请求  $query_i$  以及导航系统推荐类目  $category_j$ ，条件概率  $p(category_j | query_i)$  可以描述对于搜索请求  $query_i$  而言，推荐类目  $category_j$  的可信程度，该条件概率即称为置信度。

[0042] 在现有的类目导航体系中，各个类目及子类之间，构成一种树形结构，如图 3 所示，根节点“-1”表示一个虚拟的“总类”，根节点“-1”具有两个子节点“Indust1”和“Indust2”分别表示“总类”的两个子类。同理，“cat1”是“Indust1”的一个子类，“leaf\_cat1”、“leaf\_cat2”“leaf\_cat3”分别是“cat1”的三个子类，“leaf\_cat4”“leaf\_cat5”分别是“Indust1”的三个子类。

[0043] 在树形结构中，位于最末端的节点称为叶子节点，例如图 3 中的 leaf\_cat1 ~ leaf\_cat5。一般而言，用户最终所期望的搜索结果都位于导航体系的最后一级，即对应于导航类目图的叶子节点，即，用户的每一次搜索至少会对应一次对叶子节点的点击，因此，可以通过查询用户点击行为 (offer) 的历史数据，得到各个叶子节点的置信度  $h_{ij}$ ：

[0044]

$$h_{ij} = p(category_j | query_i) = \frac{\text{count}(category_j, query_i | offer)}{\text{count}(query_i | offer)} \quad (1)$$

[0045] 其中， $\text{count}(category_j, query_i | offer)$  表示用户搜索请求与点击某个叶子节点的行为发生关联的次数， $\text{count}(query_i | offer)$  表示用户搜索请求出现的总次数。

[0046] S102，根据所述网站导航类目图，生成至少一种导航层级结构图，其中，每种导航层级结构图具有与所述网站导航类目图相同的叶子节点；

[0047] 图 3 所示的树形结构图即为现有的网站导航方式所应用的网站导航类目图，其严格对应于网站导航类目的初始静态层级关系。在这种导航方式下，用户要找到最终所期望

的结果（即叶子节点），需要进行逐级的观察以及点击，造成网站服务器的负担增加和用户使用上的不便。

[0048] 对于网站而言，其网站导航类目的初始静态层级关系是固定的，但是在网页的实际展示过程中，并不一定需要严格按照这种关系。在本申请实施例中，可以根据初始的网站导航类目图，生成多种导航层级结构图，每一种导航层级结构图对应一种网页导航的展现方式。导航层级结构图的基本要求是：与网站导航类目图具有相同的叶子节点，即：与原导航体系具有相同的最小子类。图 4a- 图 4c 分别示出了三种根据图 3 所示的网站导航类目图所生成的导航层级结构图。

[0049] 图 4a 所示的网站导航层级结构图实际是与初始静态层级关系完全一致的；图 4b 去除了原图中的 Indust1 节点；图 4c 在图 4b 的基础上，将叶子节点 leaf\_cat1 移动到了第一层级。

[0050] 可见，在本申请实施例所生成的导航层级结构图中，可以不受网站导航类目的初始静态层级关系的限制，提供多种导航方式，例如，与原有导航方式相比，应用图 4b 的导航方式，用户不需观察 Indust1，并且要达到叶子节点 leaf\_cat1 ~ leaf\_cat3，可以减少一次点击的次数；应用图 4b 的导航方式，则可以直接观察并点击到叶子节点 leaf\_cat1。

[0051] 此外，在所生成的导航层级结构图中，各个叶子节点可以按照置信度从大到小进行排列；叶子节点的排列顺序对应于该节点在网页中的观察顺序，常见的观察顺序包括从左到右、从上到下等等，这样排列的作用是为了让用户能够更先看到置信度较高的节点，这也符合一般搜索引擎的搜索结果反馈原则。

[0052] S103，计算每种导航层级结构图的搜索代价，确定搜索代价最小的导航层级结构图；

[0053] 在 S102 中，实际上提供了多种导航的方案，然而，最终应用的方案只能有一种，在本步骤中，根据每种导航层级结构图的搜索代价来确定最终所选择导航方案。导航层级结构图的搜索代价是由观察代价和点击代价来确定的，以下分别进行介绍：

[0054] a) 观察代价  $w\_penalty$ ：

[0055] 对于某个搜索请求  $query_i$ ，用户需要观察导航层级结构图中的一级导航节点，以决定是否进行点击展开， $w\_penalty$  用以描述这种观察的代价；一般而言，位置靠后的节点被观察的成本较高（因为需要首先对前面的类目进行比较判断），因此可以将  $w\_penalty$  定义为关于查询  $query_i$ 、位置  $pos$  等的函数；例如，一种方法是通过查询用户点击行为的历史数据，统计类目  $cat$  处于位置  $pos$  被点击的概率  $p$ ，并将  $p$  作为类目  $cat$  在该位置被观察的权重，因此  $w\_penalty$  是一个关于查询  $query_i$ ，位置  $pos$ ，类目  $cat$ ，权重  $p$  等的函数：

[0056]  $w\_penalty = f(query_i, cat, pos, p)$ 。

[0057] 为了更容易实现，本实施例还提供一种简化的算法：由于一般用户总是在观察了当前节点之后再观察后一个节点，即：节点位置越靠前，其观察代价越小。设排在第一位的节点观察代价为  $w_1$ ，排在第二位的节点观察代价为  $w_2$ ……，那么将有  $w_1 < w_2 < w_3$ ……。而导航层级结构图的整体观察代价定义为用户对第一级节点进行观察，找到自己所期望的叶子节点所用观察代价的数学期望：

[0058]



$$w\_penalty_{ij} = w\_penalty(query_i, tr_j) = \sum_{k=1}^K w_k * h(query_i, v\_cat_k) \quad (2)$$

[0059] 其中：

[0060]  $query_i$  表示搜索请求  $i$ ； $tr_j$  表示第  $j$  种导航层级结构图；

[0061]  $v\_cat_k$  表示导航层级结构图第一层级的第  $k$  个节点；

[0062]  $w_k$  表示  $v\_cat_k$  的观察代价；

[0063]  $h(query_i, v\_cat_k)$  表示  $v\_cat_k$  的置信度， $h(query_i, v\_cat_k)$  等于  $v\_cat_k$  所有子孙叶子节点的置信度之和；

[0064] 由公式 (2) 可以看出，导航层级结构图的观察代价可以根据其叶子节点的置信度以及叶子节点在导航层级结构图中所处的分支位置来确定，下面分别以图 4a- 图 4c 为例进行说明，假设选取  $w_k = k$ ，叶子节点  $leaf\_cat1 \sim leaf\_cat5$  的置信度分别为  $h_{11}$ 、 $h_{12}$ 、 $h_{13}$ 、 $h_{14}$ 、 $h_{15}$ 。

[0065] 对于图 4a 所示的导航层级结构图  $tr_1$ ，第一级节点为  $Indust1$  和  $Indust2$ ，观察代价分别为 1 和 2，则  $tr_1$  的整体观察代价为：

$$w\_penalty_{11} = 1 \times (h_{11} + h_{12} + h_{13}) + 2 \times (h_{14} + h_{15})$$

[0067] 对于图 4b 所示的导航层级结构图  $tr_2$ ，第一级节点为  $cat1$  和  $Indust2$ ，观察代价分别为 1 和 2，则  $tr_2$  的整体观察代价为：

$$w\_penalty_{11} = 1 \times (h_{11} + h_{12} + h_{13}) + 2 \times (h_{14} + h_{15})$$

[0069] 可见， $tr_2$  和  $tr_1$  的整体观察代价在数值上是相同的，这是因为两张图同样具有两个一级节点，并且在两张图中，叶子节点所处的分支位置也是相同的。

[0070] 对于图 4c 所示的导航层级结构图  $tr_3$ ，第一级节点为  $leaf\_cat1$ 、 $cat1$  和  $Indust2$ ，观察代价分别为 1、2 和 3，则  $tr_3$  的整体观察代价为：

$$w\_penalty_{13} = 1 \times h_{11} + 2 \times (h_{12} + h_{13}) + 3 \times (h_{14} + h_{15})$$

[0072] b) 点击代价  $c\_penalty$ ：对于某个查询  $query_i$ ，用户需要在导航层级结构图中点击导航节点以找到自己想要的叶子节点， $c\_penalty$  用以描述这种点击的代价；一般而言， $c\_penalty$  与导航层级结构图的深度（层级）有关，导航层级结构图的深度越高，点击代价越大，因此可以将  $c\_penalty$  定义为关于查询  $query_i$ 、导航层级结构图  $tr$ 、导航深度  $level$  等的函数：

$$c\_penalty = g(query, tr, level) ;$$

[0074] 在本申请实施例的一种具体实现方式中，可以将导航层级结构图的点击代价定义为用户找到想要的叶子节点的点击次数的数学期望：

[0075]

$$c\_penalty_{ij} = c\_penalty(query_i, tr_j) = \sum_{k=1}^K level_k * h(query_i, leaf\_cat_k) \quad (3)$$

[0076] 其中：

[0077]  $query_i$  表示搜索请求  $i$ ； $tr_j$  表示第  $j$  种导航层级结构图；

[0078]  $leaf\_cat_k$  表示导航层级结构图中的第  $k$  个叶子节点；

[0079]  $level_k$  表示  $leaf\_cat_k$  在导航层级结构图中所处的层级位置；

[0080]  $h(query_i, leaf\_cat_k)$  表示  $leaf\_cat_k$  的置信度；

[0081] K 为导航层级结构图的叶子节点总数。

[0082] 由公式 (3) 可以看出, 导航层级结构图的观察代价可以根据其叶子节点的置信度以及叶子节点在导航层级结构图中所处的层级位置来确定, 下面分别以图 4a- 图 4c 为例进行说明, 假设选取叶子节点 leaf\_cat1 ~ leaf\_cat5 的置信度分别为  $f_{11}$ 、 $f_{12}$ 、 $f_{13}$ 、 $f_{14}$ 、 $f_{15}$ 。

[0083] 对于图 4a 所示的导航层级结构图  $tr_1$ , 其点击代价为:

$$[0084] \quad c\_penalty_{11} = 3 \times h_{11} + 3 \times h_{12} + 3 \times h_{13} + 2 \times h_{14} + 2 \times h_{15}$$

[0085] 对于图 4b 所示的导航层级结构图  $tr_2$ , 其点击代价为:

$$[0086] \quad c\_penalty_{12} = 2 \times h_{11} + 2 \times h_{12} + 2 \times h_{13} + 2 \times h_{14} + 2 \times h_{15}$$

[0087] 对于图 4c 所示的导航层级结构图  $tr_3$ , 其点击代价为:

$$[0088] \quad c\_penalty_{13} = 1 \times h_{11} + 2 \times h_{12} + 2 \times h_{13} + 2 \times h_{14} + 2 \times h_{15}$$

[0089] c) 搜索代价  $s\_penalty$ :

[0090] 搜索代价是观察代价与点击代价的一种综合体现, 即可以描述为  $w\_penalty$  与  $c\_penalty$  的函数:

$$[0091] \quad s\_penalty = f'(\text{query}, tr, w\_penalty, c\_penalty);$$

[0092] 在本申请实施例的一种具体实现方式中, 对于某个查询  $query_i$ , 导航层级结构图  $tr_j$  中的搜索代价可以定义为:

$$[0093] \quad s\_penalty_{ij} = \lambda_1 * w\_penalty_{ij} + \lambda_2 * c\_penalty_{ij} \quad (4)$$

[0094] 其中,  $\lambda_1$  和  $\lambda_2$  分别为观察代价和点击代价的权重, 分别代表观察代价和搜索代价的重要程度, 可以根据实际应用需求预先设置。

[0095] 综上所述, 对于某个导航层级结构图, 首先分别计算其观察代价和点击代价, 然后对所述观察代价和点击代价进行加权处理, 就可以获得该导航层级结构图的搜索代价。

[0096] S104, 根据所述搜索代价最小的导航层级结构图实现网站导航。

[0097] 根据前面的说明, 搜索代价是观察代价与点击代价的一种综合体现, 具有较小搜索代价的导航层级结构图, 可以认为其同时兼顾了较小的观察代价和点击代价, 因此根据搜索代价最小的导航层级结构图实现网站导航, 可以有效减少导航时用户的点击操作, 从而减轻网站服务器的负担, 节省网络带宽资源。从用户的角度来看, 也能够更快捷、更方便地找到自己期望的结果。

[0098] 需要说明的是, 图 4a- 图 4c 仅用于示意性说明, 可以理解的是, 针对图 3 所示的初始网站导航类目图还可以生成更多种类的导航层级结构图。当然, 结合实际的应用需求, 在生成导航层级结构图时, 还可以进一步考虑一些约束条件, 例如:

[0099] a) 置信度大于预设阈值的叶子节点位于导航层级结构图的第一级。

[0100] 该约束条件的意义是: 对于置信度较大的叶子节点, 用户可能更希望在导航过程中较早发现, 例如图 4c 所示的 leaf\_cat1。

[0101] b) 导航层级结构图的第一级节点个数, 不大于网页所允许的显示数量。

[0102] 可以理解的是, 如果第一级节点个数过多, 超过了网页的显示能力, 导致用户还要进行翻页等操作, 这样反而可能导致搜索代价的增加。

[0103] c) 导航层级结构图的同一层级中, 不同时出现某个节点与其在所述初始静态层级关系中的祖先节点。

[0104] 该约束条件可以避免同样的节点在导航层级结构图中重复出现。

[0105] 上述的几种约束条件,在生成导航层级结构图时,可以单独考虑,也可以复合考虑,除了能够令最终的导航方案更符合实际应用需求之外,还可以减少生成导航层级结构图的数量,从而提高处理效率。

[0106] 相应于上面的方法实施例,本申请实施例还提供一种网站导航系统,参见图 5 所示,包括:

[0107] 历史数据查询模块 510,用于查询用户操作行为的历史数据,获得网站导航类目图各个叶子节点的置信度,所述网站导航类目图对应于网站导航类目的初始静态层级关系;

[0108] 导航层级结构图生成模块 520,用于根据所述网站导航类目图,生成至少一种导航层级结构图,其中,每种导航层级结构图具有与所述网站导航类目图相同的叶子节点;

[0109] 搜索代价计算模块 530,用于计算每种导航层级结构图的搜索代价;

[0110] 导航实现模块 540,用于根据搜索代价最小的导航层级结构图实现网站导航;

[0111] 参见图 6 所示,所述搜索代价计算模块 530 具体可以包括:

[0112] 观察代价计算子模块 531,用于根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的分支位置,计算该导航层级结构图的观察代价;

[0113] 点击代价计算子模块 532,用于根据叶子节点的置信度以及叶子节点在导航层级结构图中所处的层级位置,计算该导航层级结构图的点击代价;

[0114] 搜索代价计算子模块 533,用于根据预设的观察代价权重和点击代价权重,对所述观察代价和点击代价进行加权处理,获得该导航层级结构图的搜索代价。

[0115] 其中,在所述导航层级结构图生成模块 520 生成的导航层级结构图中,各个叶子节点按照置信度从大到小进行排列;其中,叶子节点的排列顺序对应于该节点在网页中的观察顺序。

[0116] 此外,所述导航层级结构图生成模块 520 可以根据以下约束条件生成导航层级结构图:

[0117] 置信度大于预设阈值的叶子节点位于导航层级结构图的第一级、

[0118] 和 / 或

[0119] 导航层级结构图的第一级节点个数,不大于网页所允许的显示数量、

[0120] 和 / 或

[0121] 导航层级结构图的同一层级中,不同时出现某个节点与其在所述初始静态层级关系中的祖先节点。

[0122] 所述观察代价计算子模块 531,预先为位于导航层级结构图第一层级不同位置的节点设置观察代价,其中,位置靠前节点的观察代价小于位置靠后节点的观察代价;对导航层级结构图中所有第一层级节点的观察代价和置信度的乘积求和,得到所述导航层级结构图的观察代价,其中,第一层级节点的置信度为该第一层级节点所有子孙叶子节点的置信度之和。

[0123] 具体而言,所述观察代价计算子模块 531 可以利用以下公式计算导航层级结构图的观察代价  $w\_penalty_{ij}$ :

$$[0124] \quad w\_penalty_{ij} = w\_penalty(query_i, tr_j) = \sum_{k=1}^K w_k * h(query_i, v\_cat_k)$$

[0125] 其中:

[0126] query<sub>i</sub> 表示搜索请求 i ;tr<sub>j</sub> 表示第 j 种导航层级结构图 ;

[0127] v\_cat<sub>k</sub> 表示导航层级结构图第一层级的第 k 个节点 ;

[0128] w<sub>k</sub> 表示 v\_cat<sub>k</sub> 的观察代价 ;

[0129] h(query<sub>i</sub>, v\_cat<sub>k</sub>) 表示 v\_cat<sub>k</sub> 的置信度, h(query<sub>i</sub>, v\_cat<sub>k</sub>) 等于 v\_cat<sub>k</sub> 所有子孙叶子节点的置信度之和 ;

[0130] K 为导航层级结构图的第一级的节点总数。

[0131] 所述点击代价计算子模块 532 可以通过对导航层级结构图中所有叶子节点的置信度与其所处层级位置的乘积求和, 得到所述导航层级结构图的点击代价。

[0132] 具体而言, 所述点击代价计算子模块 532 可以利用以下公式计算导航层级结构图的点击代价 c\_penalty<sub>ij</sub> :

$$[0133] \quad c\_penalty_{ij} = c\_penalty(query_i, tr_j) = \sum_{k=1}^K level_k * h(query_i, leaf\_cat_k)$$

[0134] 其中 :

[0135] query<sub>i</sub> 表示搜索请求 i ;tr<sub>j</sub> 表示第 j 种导航层级结构图 ;

[0136] leaf\_cat<sub>k</sub> 表示导航层级结构图中的第 k 个叶子节点 ;

[0137] level<sub>k</sub> 表示 leaf\_cat<sub>k</sub> 在导航层级结构图中所处的层级位置 ;

[0138] h(query<sub>i</sub>, leaf\_cat<sub>k</sub>) 表示 leaf\_cat<sub>k</sub> 的置信度 ;

[0139] K 为导航层级结构图的叶子节点总数。

[0140] 为了描述的方便, 描述以上系统时以功能分为各种模块分别描述。当然, 在实施本申请时可以把各模块的功能在同一个或多个软件和 / 或硬件中实现。

[0141] 通过以上的实施方式的描述可知, 本领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件平台的方式来实现。基于这样的理解, 本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来, 该计算机软件产品可以存储在存储介质中, 如 ROM/RAM、磁碟、光盘等, 包括若干指令用以使得一台计算机设备 (可以是个人计算机, 服务器, 或者网络设备等) 执行本申请各个实施例或者实施例的某些部分所述的方法。

[0142] 本说明书中的各个实施例均采用递进的方式描述, 各个实施例之间相同相似的部分互相参见即可, 每个实施例重点说明的都是与其他实施例的不同之处。尤其, 对于系统实施例而言, 由于其基本相似于方法实施例, 所以描述得比较简单, 相关之处参见方法实施例的部分说明即可。以上所描述的系统实施例仅仅是示意性的, 其中所述作为分离部件说明的模块可以是或者也可以不是物理上分开的, 作为模块显示的部件可以是或者也可以不是物理模块, 即可以位于一个地方, 或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下, 即可以理解并实施。

[0143] 本申请可用于众多通用或专用的计算系统环境或配置中。例如 : 个人计算机、服务器计算机、手持设备或便携式设备、平板型设备、多处理器系统、基于微处理器的系统、置顶盒、可编程的消费电子设备、网络 PC、小型计算机、大型计算机、包括以上任何系统或设备的分布式计算环境等等。

[0144] 本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述, 例如程序

模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0145] 以上所述仅是本申请的具体实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本申请的保护范围。

**Department**  
**Any Department**  
Electronics  
Wireless Accessories  
Software  
Home & Garden  
Books  
MP3 Downloads  
Clothing & Accessories  
**+ See All 32 Departments**

图 1a

**Department**  
< Any Department  
**Electronics**  
Computers & Accessories  
Portable Audio & Video  
Accessories & Supplies  
Camera & Photo  
Televisions & Video  
Office Electronics  
Home Audio & Theater  
GPS & Navigation  
Security & Surveillance  
Car Electronics

图 1b

**Department**  
< Any Department  
< Electronics  
**Computers & Accessories**  
Desktops  
Laptops  
Mice & Keyboards  
Networking Products  
Computer Accessories  
Internal Laptop Components  
Monitors  
Video Projectors

图 1c

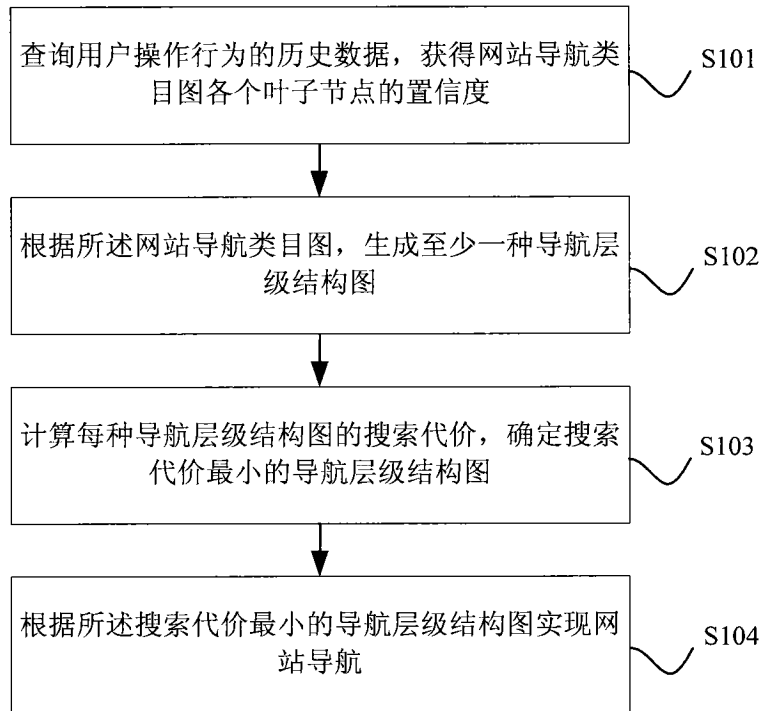


图 2

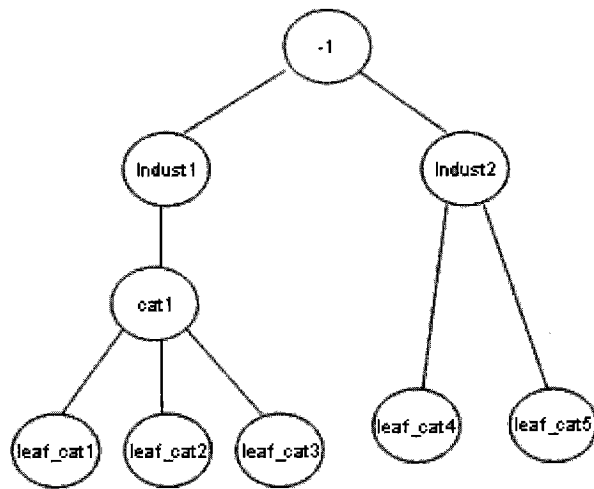


图 3

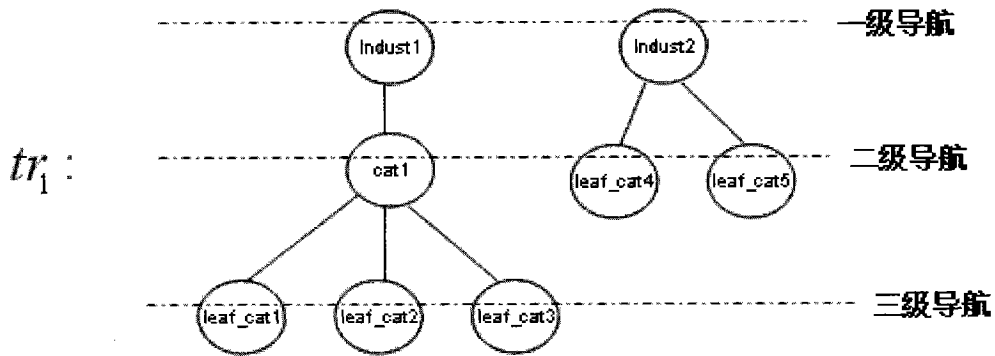


图 4a

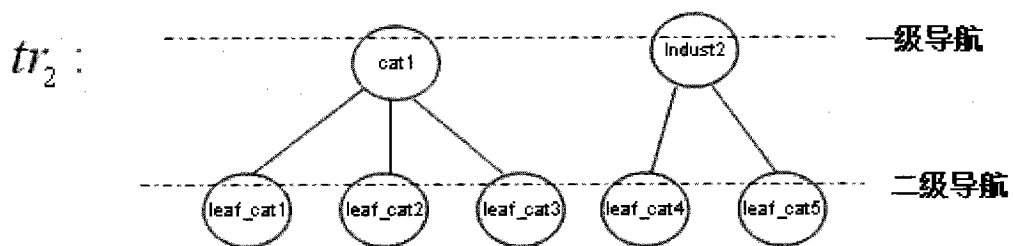


图 4b

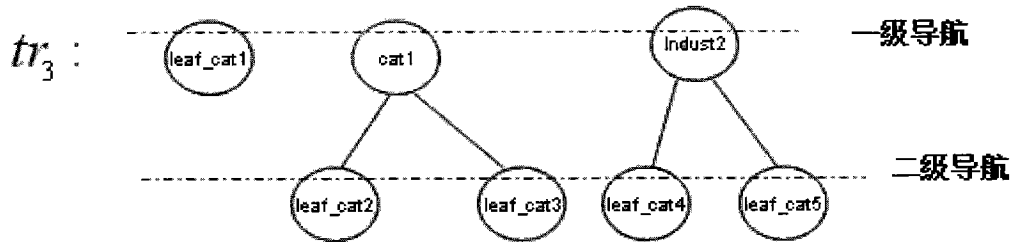


图 4c

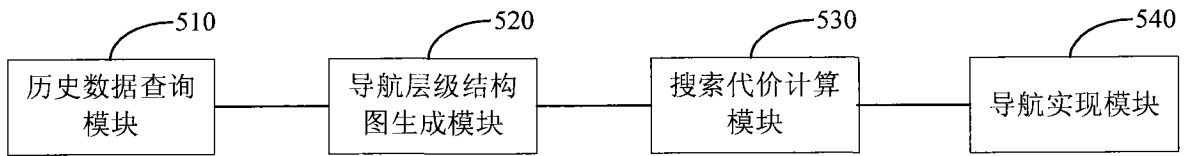


图 5

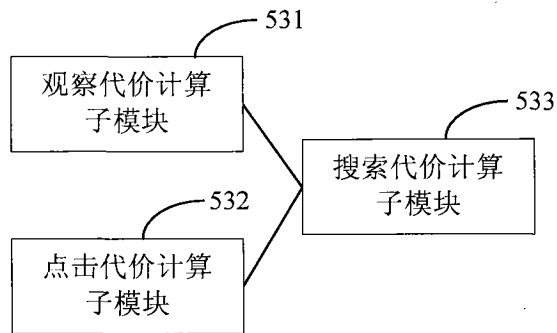


图 6