

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/00 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510077624.4

[43] 公开日 2006年3月22日

[11] 公开号 CN 1749988A

[22] 申请日 2005.6.17

[21] 申请号 200510077624.4

[30] 优先权

[32] 2004.9.14 [33] US [31] 10/940,075

[71] 申请人 安捷伦科技有限公司

地址 美国加利福尼亚州

[72] 发明人 约翰·M·海曼 李 强

[74] 专利代理机构 北京东方亿思知识产权代理有限
责任公司
代理人 王 怡

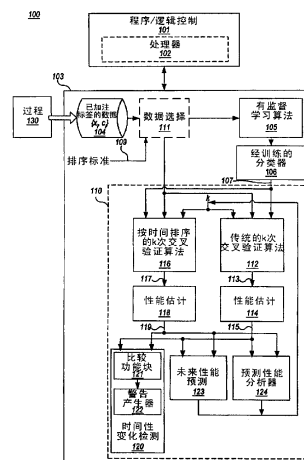
权利要求书 2 页 说明书 20 页 附图 12 页

[54] 发明名称

用于管理及预测自动分类器性能的方法和装置

[57] 摘要

本发明公开了用于检测时间性过程变化的技术，以及通过使用基于样本的时间排序而获得的性能估计来管理和预测被应用于这种过程的自动分类器的性能的技术。



1. 一种用于预测改变训练数据集合大小对分类器性能影响的方法，所述包括如下步骤：

5 从已加注标签的训练数据中选择不同大小的多个训练子集以及相应的测试子集；

在所述训练子集上训练多个分类器；

使用相应的分类器来分类所述测试子集的成员；以及

将分配给所述测试子集的成员的分类与已加注标签的训练数据中相应

10 成员的相应真实分类进行比较，以生成作为训练集合大小的函数的性能估计。

2. 如权利要求 1 所述的方法，还包括如下步骤：

对性能估计进行内插或外插直至所期望的训练集合大小。

3. 一种有形地包含程序指令的计算机可读存储介质，所述程序指令实现一种用于预测改变训练数据集合大小对分类器性能影响的方法，所述方法包括如下步骤：

从已加注标签的训练数据中选择不同大小的多个训练子集以及相应的测试子集；

在所述训练子集上训练多个分类器；

20 使用相应的分类器来分类所述测试子集的成员；以及

将分配给所述测试子集的成员的分类与已加注标签的训练数据中相应成员的相应真实分类进行比较，以生成作为训练集合大小的函数的性能估计。

4. 如权利要求 3 所述的计算机可读存储介质，其中所述方法还包括如下步骤：

对性能估计进行内插或外插直至所期望的训练集合大小。

5. 一种用于预测改变训练数据集合大小对分类器性能影响的系统，所述系统包括：

数据选择功能块，其从已加注标签的训练数据中选择不同大小的多个

训练子集以及相应的测试子集；

在各自的多个训练子集上被训练的多个相应分类器，所述多个相应分类器使用相应的分类器来分类相应测试子集的成员；和

5 比较功能块，其将分配给所述测试子集的成员的分类与已加注标签的训练数据中相应成员的相应真实分类进行比较，以生成作为训练集合大小的函数的性能估计。

6. 如权利要求 5 所述的系统，还包括：

统计分析器，其对性能估计进行内插和/或外插直至所期望的训练集合大小。

10 7. 一种用于预测改变训练数据集合大小对分类器性能影响的方法，所述方法包括如下步骤：

在训练数据上用不同的 k 执行按时间排序的 k 次交叉验证；以及对得到的性能估计进行内插或外插直至所期望的训练集合大小。

15 8. 一种有形地包含程序指令的计算机可读存储介质，所述程序指令实现一种用于预测改变训练数据集合大小对分类器性能影响的方法，所述方法包括如下步骤：

在训练数据上用不同的 k 执行按时间排序的 k 次交叉验证；以及对得到的性能估计进行内插或外插直至所期望的训练集合大小。

20 9. 一种用于预测改变训练数据集合大小对分类器性能影响的系统，所述系统包括：

按时间排序的 k 次交叉验证功能块，其在训练数据上用不同的 k 执行按时间排序的 k 次交叉验证；和

统计分析器，其对得到的性能估计进行内插和/或外插直至所期望的训练集合大小。

用于管理及预测自动分类器性能的方法和装置

5 技术领域

本发明涉及用于检测时间性过程变化的技术，以及用于管理和预测自动分类器性能的方法和装置。

背景技术

10 很多依赖于物体的模式识别和/或分类的工业应用（诸如自动制造检查或分类系统）都在利用有监督学习技术。如图 1 所代表的有监督学习系统是一种基于已加注标签的训练数据 2 的代表性输入集合，利用有监督学习算法 4 来创建经训练的分类器 6 的系统。训练数据 2 的集合中的每个成员都包含特征向量 x_i 以及指示出具体成员所属唯一类别的标签 c_i 。给定特征

15 向量 x ，经训练的分类器 f 将返回相应的类别标签 $f(x) = \hat{c}$ 。有监督学习系统 4 的目标是使分类器 6 的准确性或相关的量度不仅对于训练数据 2 最大化，而且对于类似获得的、学习算法 4 无法获得的测试数据的集合也最大化。如果用于具体应用的类别标签的集合只包含两个条目，则将该应用称为二元（或两类）分类问题。二元分类问题在自动检查中常见，例如其目

20 标通常是判断制成品是好还是坏。例如在将项目分类成一个或多个子类的过程（例如按品种对鱼分类、按速度对计算机存储器分类等等）中，也会遇到多类的问题。在统计模式识别中，已经广泛研究了有监督学习，并且基于不可见的测试数据来训练分类器并且预测经训练的分类器的性能的多种学习算法和方法是公知的。

25 再次参考图 1，给定已加注标签的训练数据集合 2 ($D = \{x_i, c_i\}$)，有监督学习算法 4 可被用于产生经训练的分类器 6 ($f(x) = \hat{c}$)。风险或成本 α_{ij} 可能与以下情况相关联，即当样本所属的真正类别为 j 时将其错误地分类为属于类别 i 。传统上，正确的分类被指定为零成本 $\alpha_{jj} = 0$ 。典型的目标是估计预期损失（即预计分类器 6 在从相同过程中得到的新样本上引起的

成本的加权平均)，并使其最小化。损失的概念非常普通。在 i 和 j 不同时设置 $\alpha_{ij} = 1$ ，并且在 i 和 j 相同时设置 $\alpha_{ij} = 0$ （所谓的 0/1 损失），这种情况下等同于将所有错误看作是相同的，并且致使总的错误分类率最小。更典型地，不同类型的错误将具有不同的相关成本。更复杂的损失公式也是可能的。例如，损失 α_{ij} 可以是函数，而不是常数。但是，在每种情况下，都定义了预测分类器性能的某种量度，并且目标是使性能最大化，或者等同地，使损失最小化。

目前存在若干用于预测分类器性能的现有技术。一种这样的技术是使用独立的训练和测试数据集合。使用训练数据来构造经训练的分类器，然后基于独立的测试数据来评估经训练的分类器的性能。但是在很多应用中，收集已加注标签的数据困难而又昂贵，因此希望在训练期间使用所有可获得的数据，从而使所产生分类器的准确性最大化。

另一种被称为“传统 k 次交叉验证”（或简称为“ k 次交叉验证”）的用于预测分类器性能的现有技术不需要单独的测试数据，因此允许将所有可获得的数据用于训练。如图 2A 和 2B 所示，在 k 次交叉验证中，训练数据 $\{x_i, c_i\}$ 被随机地分割成 k 个子集 D_i ($1 \leq i \leq k$)，这 k 个子集具有大致相等的大小（图 2B，步骤 11）。对于迭代 $i=1$ 到 k （步骤 12-17），通过使用除 D_i 之外的所有可获得的数据而使用有监督学习算法来训练分类器（步骤 14）。然后，这个经训练的分类器被用于分类子集 D_i 中的所有样本（步骤 15），并且存储分类结果（步骤 16）。在很多情况下，也可以在步骤 16 处只保存摘要统计量，而不保存个别的分类。例如在恒定损失的情况下，保存各种类型的错误的总数就足够了。在 k 次迭代之后，对于全部数据集合，就已经知道了真实 (c_i) 和估计 (\hat{c}_i) 的类别标签（或相应的足够统计量）。然后可以计算诸如错误分类率、工作特性曲线或预期损失之类的性能估计（步骤 18）。如果样本总数为 n ，那么每个样本的预期损失例如可以被估计为 $\sum \alpha_{\hat{c}_i, c_i} / n$ 。当 $k = n - 1$ 时， k 次交叉验证也被称为“留一法交叉验证”。在某些应用中，被称为“广义交叉验证”的、在计算上更有效的变体方案可能是优选的。这里，我们将这些和类似的现有技术称为“传统交叉验证”，并且不对它们进行区分。

在 k 次交叉验证中，数据样本只在它们不对分类器的训练起作用时，

才被用来估计性能，从而得到合理的性能估计。另外，对于足够大的 k ，在上述每次迭代期间的训练集合大小（大致为 $\frac{(k-1)}{k} \cdot n$ ，其中 n 是已加注标签的训练数据样本的数目）只略小于全部数据集合的大小，从而导致对性能只是稍有悲观的估计。

5 很多有监督学习算法致使具有一个或多个可调整参数的分类器控制工作点。为了简单起见，这里将讨论局限于二元分类问题，其中 c_i 是两个不同类别中的一个或另一个类别中的成员。但是，将会意识到，这里所论述的原理可以被扩展到多类分类问题。在二元分类中，假肯定（false positive）被定义为：在样本实际上属于负（或好）类别时，将其错误地分类为属于正（或缺陷）类别。相似地，真肯定被定义为将样本正确地分类为属于正类别。然后，假肯定率（也被称为假报警率）被定义为假肯定的数目除以负类别的成员数目。相似地，敏感度被定义为真肯定的数目除以正类别的成员数目。根据这些定义，具有可调整工作点的二元分类器的性能可以由图 3 所示的工作特性曲线（有时被称为接收方工作特性(ROC)曲线）来概括。分类器工作点的变化等同于选择 ROC 曲线上的点。在每个工作点处，发生任一类型错误分类的比率估计是已知的。如果相关成本 α_{ij} 也已知，则可以计算出任一工作点的预期损失。对于单调的工作特性，可以选择出唯一的工作点以使预期损失最小化。如前所述， k 次交叉验证提供了构造二元分类器的估计 ROC 曲线所需的信息。

10 除了有效地利用所有可获得的数据之外， k 次交叉验证还具有允许估计预测性能的可靠性的额外优点。对于将数据划分到 k 个子集中的不同伪随机划分，可以重复 k 次交叉验证算法。这种方法例如不仅可用于计算预期损失，而且可用于计算该估计的标准方差。相似地，可以执行非参数假设测试（例如， k 次交叉验证可被用于回答诸如“损失超过估计值两倍的可能性有多大？”之类的问题）。

25 用于预测分类器性能的现有技术的方法假设训练数据的集合具有代表性。如果不是这样，并且具体而言，如果产生训练数据样本的过程受时间性变化的影响（例如过程随时间偏移或改变），那么经训练的分类器的性能可能要比预测差得多。这样的性能差异或改变可以在其发生时被用于检

测时间性变化，但是优选的做法是在训练阶段的过程中检测时间性变化。有监督的学习通常不能解决这个问题。

两种明确地对过程中的时间性变化进行预测的技术是时间序列分析和统计过程控制。时间序列分析试图理解和建模数据集合中的时间性变化，通常这种方法的目标是以一定时期内的行为来预测未来的行为，或者校正季节性或其他变化。统计过程控制（SPC）提供了使过程操作保持在可接受的限度内的技术，并且用于在无法这样做时发出警报。理想地，统计过程控制能够被用来使过程保持在其最佳工作点上或最佳工作点附近，几乎消除了由于基础过程中的时间性变化而引起的恶劣分类器性能。在实际中，由于时间、成本以及相关困难的影响，因此这种理想情况难以达到。因此，即使在被很好控制的过程中，时间性变化也可能存在于预定限度之内，并且这种变化可能足以干扰使用有监督学习所创建的分类器的性能。在出现时间性过程变化时，时间序列分析和统计过程控制都无法提供可直接用于分析和管理的这种分类器的工具。

当 a) 生成训练数据集合的基础过程不具有严重的时间性变化，或者 b) 时间性变化出现时，可以应用预测分类器性能的现有技术方法，但是所述基础过程是平稳的并且是各态历经的，并且样本在一段足够长的时间段内被收集，因此它们具有代表性。在很多情况下，在所述基础过程中存在显式或隐式的时间性变化，在这些情况下，认为训练数据集合代表了基础过程的假设并不合理，并且 k 次交叉验证对性能的估计会明显过高。例如考虑图 4A、4B 和 4C 中所示的过程。在这些图形中的“状态”仅仅出于图示的目的。实际状态将具有很高的，通常未知的大小，并且其自身难以获知。如图 4A 所示的过程不具有时间性变化。如图 4B 所示的过程是具有随机的、各态历经的波动的平稳过程。如图 4C 所示的过程示出了稳定的偏移，其在局部均值周围伴随有随机波动。如果给定足够的训练数据，传统的 k 次交叉验证则将准确地预测出对于图 4A 所示的过程的分类器性能。对于图 4B 所示的过程，如果数据集合是在一段足够长的时间内被收集，从而大致以均衡分布对状态采样，那么也将获得正确的结果。如果没有这样做，则通常将对性能做出过高估计。对于图 4C 所示的过程，实际的性

能可能在最初与预测性能相匹配，但是实际性能将随着对未来工作点的进一步采样而下降。这一系列采样过程仅仅出于图示的目的，决不是穷举性的。

判断训练数据的集合是否代表过程的步骤通常需要收集附加的已加注
5 标签的训练数据，这可以是相当昂贵的。作为示例，考虑对复杂印制电路
配件的制造。使用 SPC，可以以高可靠性（例如以百万分之 100（100
ppm）数量级上的缺陷率）来构造这种印制电路配件上的单独的焊点。因
此，有缺陷的焊点非常少。但是，大的印制电路配件可以有超过 50000 个
10 的焊点，因此如果没有自动检测需要修复的焊点的能力，那么缺陷的经济
影响将是巨大的。有监督的学习通常被用于为这种应用构造分类器。几千
个缺陷需要训练，但是由于好的焊点以 10000 比 1 的比例在数量上超过坏
的焊点，因此必须对几百万个好的焊点进行检查，以便获得足够的缺陷样
本以训练分类器。这会给负责分配真实类别标签的分析者（通常是人类专
15 家）带来沉重负担，因此对训练数据的收集非常耗时、昂贵，而且易于出
现错误。另外，对训练数据的收集大于所需，这减慢了训练过程，并且不
会提高性能。因此，希望尽可能使用最小的训练数据集合来获得所需的性
能。

由于上述原因，因此希望能够从训练数据自身的指示中检测在过程中
时间性变化的出现或可能出现。还希望即使在基础过程中出现时间性变化
20 的情况下，也能够预测预期的未来分类器性能。最后，如果能从附加训练
数据的收集中获得性能增益，并且为附加训练数据的使用提供不同的选项
（例如用于回答如下问题：是简单地增加现有训练数据的做法更好，还是
基于训练数据样本的滑动窗而周期性地重新训练分类器的做法更好）将是
非常有用的。

25

发明内容

本发明提供了用于检测时间性过程变化的技术，以及通过使用基于样
本的时间排序而得到的性能估计来管理和预测被应用于这种过程的自动分
类器的性能的技术。具体而言，本发明详述了在已加注标签的训练数据上

检测在过程中出现或可能出现的时间性变化的方法，用于在这种时间性变化出现时预测使用有监督学习算法来训练的分类器的性能的方法，以及用于研究包含了附加训练数据的收集和最优利用过程的情形的方法。所描述的技术还可以被扩展到处理多个源的时间性变化。

- 5 本发明的第一方面包括根据所产生的过程样本中的指示来检测过程中的时间性变化，所述过程样本被用作已加注标签的训练数据，其用于利用有监督学习来训练分类器。根据本发明的第一方面，该方法包括以下步骤：根据一个或多个第一标准选出已加注标签的训练数据的一个或多个第一教导子集，并且根据一个或多个第二标准选出已加注标签的训练数据的
- 10 相应第一测试子集，其中所述一个或多个第一标准和一个或多个第二标准中的至少一个标准至少部分地基于时间顺序；分别使用相应的一个或多个第一教导子集来训练一个或多个第一分类器；分别使用相应的一个或多个第一分类器来分类一个或多个第一测试子集的成员；将分配给一个或多个第一测试子集的成员的分类与已加注标签的训练数据中的相应成员的相应
- 15 真实分类相比较，从而基于比较结果而产生一个或多个第一性能估计；根据一个或多个第三标准选出已加注标签的训练数据的一个或多个第二教导子集，并且根据一个或多个第四标准选出已加注标签的训练数据的相应第二测试子集，其中所述第三标准中的至少一个标准至少部分不同于所述第一标准并且/或者所述第四标准中的至少一个至少部分不同于所述第二标准；
- 20 分别使用相应的一个或多个第二教导子集来训练一个或多个第二分类器；分别使用相应的一个或多个第二分类器来分类一个或多个第二测试子集的成员；将分配给一个或多个第二测试子集的成员的分类与已加注标签的训练数据中的相应成员的相应真实分类相比较，从而基于比较结果而产生一个或多个第二性能估计；以及分析所述一个或多个第一性能估计和一个或多个第二性能估计，从而检测出时间性变化的证据。
- 25

还可以根据如下步骤来执行对过程中时间性变化的检测：对训练数据的一个或多个第一子集执行按时间排序的 k 次交叉验证，以产生一个或多个第一性能估计；对训练数据的一个或多个第二子集执行按时间排序的 k 次交叉验证，以产生一个或多个第二性能估计；以及分析所述一个或多个

第一性能估计和一个或多个第二性能估计，从而检测出时间性变化的证据。

本发明的第二方面包括预测在一组已加注标签的训练数据上被训练的分类器的性能。根据本发明的这个第二方面，该方法包括以下步骤：根据
5 一个或多个第一标准选出已加注标签的训练数据的一个或多个第一教导子集，并且根据一个或多个第二标准选出已加注标签的训练数据的相应第一测试子集，其中所述一个或多个第一标准以及一个或多个第二标准中的至少一个标准至少部分地基于时间顺序；分别使用相应一个或多个第一教导子集来训练一个或多个第一分类器；分别使用相应一个或多个第一分类器来分类一个或多个第一测试子集的成员；将分配给一个或多个第一测试子集的成员的分类与已加注标签的训练数据中的相应成员的相应真实分类相比较，从而基于比较结果而产生一个或多个第一性能估计；根据一个或多个第三标准而选出已加注标签的训练数据的一个或多个第二教导子集，并且根据一个或多个第四标准而选出已加注标签的训练数据的相应第二测试子集，其中所述第三标准中的至少一个标准至少部分不同于所述第一标准并且/或者所述第四标准中的至少一个标准至少部分不同于所述第二标准；分别使用相应一个或多个第二教导子集来训练一个或多个第二分类器；分别使用相应一个或多个第二分类器来分类一个或多个第二测试子集的成员；将分配给一个或多个第二测试子集的成员的分类与已加注标签的训练数据中的相应成员的相应真实分类相比较，从而基于比较结果而产生一个或多个第二性能估计；以及基于所述第一性能估计和第二性能估计的统计分析来预测所述分类器的性能。
10
15
20

分类器性能预测还可以根据如下步骤来执行：对训练数据的一个或多个第一子集执行按时间排序的 k 次交叉验证，以产生一个或多个第一性能估计；对训练数据的一个或多个第二子集执行按时间排序的 k 次交叉验证，以产生一个或多个第二性能估计；以及对所述一个或多个第一性能估计和一个或多个第二性能估计执行统计分析，以预测所述分类器的性能。
25

可替换地，分类器性能预测还可以根据如下步骤来执行：根据一个或多个第一标准选出已加注标签的训练数据的一个或多个教导子集，并且根

据一个或多个第二标准选出已加注标签的训练数据的相应测试子集，其中所述一个或多个第一标准和一个或多个第二标准中的至少一个标准至少部分地基于时间顺序；分别使用相应一个或多个教导子集来训练一个或多个第一分类器；分别使用相应一个或多个第一分类器来分类一个或多个测试子集的成员；将分配给一个或多个测试子集的成员的分类与已加注标签的训练数据中的相应成员的相应真实分类相比较，从而基于比较结果而产生一个或多个性能估计；以及基于所述一个或多个性能估计的统计分析来预测分类器的性能。

本发明的第三方面包括预测由于训练数据集合大小的变化对分类器性能所产生的影响。根据本发明的这个第三方面，该方法包括以下步骤：从已加注标签的训练数据中选出多个具有变化大小的训练子集以及相应的测试子集；在训练子集上训练多个分类器；使用相应分类器来分类测试子集的成员；以及将分配给测试子集成员的分类与已加注标签的训练数据中的相应成员的相应真实分类相比较，从而产生作为训练集合大小的函数的性能估计。

由于改变训练数据集合大小而进行的分类器性能预测还可以根据如下步骤来执行：在训练数据上执行改变 k 值的按时间排序的 k 次交叉验证；并且内插或外插所产生的性能估计，以得到所期望的训练集合大小。

本发明的第四方面包括预测通过将滑动窗用于训练数据集合而被训练的分类器的性能。根据本发明的这个第四方面，该方法包括以下步骤：根据一个或多个至少部分基于时间顺序的第一标准将训练数据集合排序成经排序的训练数据集合；选择一个或多个具有大致相等的第一预定大小的教导子集以及相应一个或多个具有大致相等的第二预定大小的测试子集，所述教导子集包括所述经排序的训练数据集合中的第一相邻成员，并且所述测试子集包括来自所述经排序的训练数据集合的至少一个成员，该成员在时间上紧随其相应的一个或多个教导子集的所有成员之后；使用所述一个或多个教导子集来训练相应一个或多个分类器；使用所述相应一个或多个分类器来分类所述相应一个或多个测试子集的成员；将分配给所述相应一个或多个测试子集的成员的分类与分配给已加注标签的训练数

据中的相应成员的相应真实分类进行比较，以产生一个或多个性能估计；并且基于所述一个或多个性能估计的统计分析来预测所述分类器的性能，所述分类器是通过将滑动窗用于大致具有所述第一预定大小的训练数据而被训练的。

- 5 由于将滑动窗用于训练而进行的分类器性能预测还可以根据如下步骤来执行：根据至少部分基于时间顺序的一个或多个第一标准来选择训练数据集的一个或多个组，所述一个或多个组具有大致相等的大小；从所述一个或多个组中的每个组中，根据至少部分基于时间顺序的一个或多个第二标准而选出具有大致相等的第一预定大小的一个或多个教导子集，并且
- 10 根据至少部分基于时间顺序的一个或多个第三标准而选出相应的具有大致相等的第一预定大小的测试子集；使用来自所述一个或多个组中每个组的一个或多个教导子集来训练相应的一个或多个分类器；使用所述相应的一个或多个分类器来分类所述相应的一个或多个测试子集的成员；将分配给
- 15 所述相应的一个或多个测试子集的成员的分类与分配给已加注标签的训练数据中的相应成员的相应真实分类进行比较，以产生与每个组相关联的一个或多个性能估计；并且基于与每个组相关联的一个或多个性能估计的统计分析来预测所述分类器的性能，所述分类器是通过将具有大致相等的第一预定大小的滑动窗用于训练数据而被训练的。

- 20 上述（多种）方法优选地使用实现所述功能的计算机硬件系统和/或包括程序指令的软件来执行，所述程序指令有形地实施上述（多种）方法。

附图说明

- 结合附图参考以下详细描述，本发明的更完整评价及其很多附加优点将变得更加明显，并且更容易理解，在附图中，相同的标号指示出相同或
- 25 相似的组件，其中：

图 1 是传统的有监督学习系统的框图；

图 2A 示出了传统的 k 次交叉验证的数据流图；

图 2B 示出了传统的 k 次交叉验证算法的流程图；

图 3 示出了接收方工作特性（ROC）曲线的示例的图；

图 4A 示出了不具有时间性变化的按时间绘制的示例性过程的图；

图 4B 示出了具有随机的、各态历经波动的按时间绘制的示例性平稳过程的图；

图 4C 示出了具有稳定偏移的按时间绘制的示例性过程的图，其中所述稳定偏移在均值周围伴随有随机波动；

图 5A 示出了按时间排序的 k 次交叉验证的数据流图；

图 5B 示出了根据本发明实现的按时间排序的 k 次交叉验证算法的流程图；

图 6 示出了基于用于训练分类器的训练数据而预测过程中的时间性变化的创造性技术的流程图；

图 7 是根据本发明实现的用于实现时间性变化管理器的系统的框图；

图 8 示出了用于预测分类器的未来性能的操作方法的流程图；

图 9 示出了用于判断将滑动窗用于训练数据是否将提高分类器性能的操作方法的流程图；

图 10 根据图 9 的方法，示出了在训练分类器时，使用训练数据样本的滑动窗的数据流图；

图 11 示出了用于判断在训练分类器时，训练数据样本的滑动窗的使用是否将提高分类器性能的另一操作方法的流程图；以及

图 12 根据图 11 的方法，示出了在训练分类器时，使用训练数据样本的滑动窗的数据流图。

具体实施方式

本发明提供了一种技术，该技术从利用有监督学习来训练分类器的训练数据的指示中检测出在过程中出现或可能出现的时间性变化。本发明还提供了用于在基础过程中出现时间性变化时预测预期的分类器未来性能的技术，以及如果以及在收集附加的训练数据时，用于研究各种使附加的已加注标签的训练数据的使用最优化的选项的技术。本发明采用了被称为“按时间排序的 k 次交叉验证”的新颖技术，并且将使用传统的 k 次交叉验证获得的性能估计与使用按时间排序的 k 次交叉验证获得的性能估计进

行比较，从而检测出基础过程中的时间性变化的可能指示。

如图 5A 和 5B 中附图所代表的按时间排序的 k 次交叉验证与传统的 k 次交叉验证的不同之处在于将已加注标签的训练数据 ($D = \{x_i, c_i\}$) 的集合划分成 k 个子集的过程不是随机完成的。相反，首先根据一个或多个相关标准（例如到达时间、检查时间、制造时间等等）按时间的递增顺序对训练数据进行排序（图 5B 中的步骤 31）。然后将经排序的训练数据的集合 (D_{SORTED}) 划分成（保持按时间排序的顺序）k 个子集 D_1, D_2, \dots, D_k ，这 k 个子集（大致）具有相同数目的样本（步骤 32）。

该过程的其余步骤都与传统的 k 次交叉验证相同。对于每个 $i=1, \dots, k$ ，基于去除掉 D_i 的训练数据来训练分类器，并且所产生的分类器用于为 D_i 的成员生成估计的类别标签 \hat{c}_i （步骤 33-38）。最后，根据真实的和估计的类别标签，或者相应的摘要统计来计算预测性能 $PE_{TIME_ORDERED}(k)$ 。如前所述，可以计算出性能的一个或多个标准量度，例如预期损失、错误分类率和工作特性曲线。与传统的 k 次交叉验证一样，数据集合中的所有样本都被用于训练和测试。

通常已经注意到，在性能的传统预测和按时间排序的预测有所不同的过程中，按时间排序的性能估计 $PE_{TIME_ORDERED}(k)$ 通常提供比传统 k 次交叉验证的性能估计 $PE(k)$ 好得多的未来分类器性能预测。根据本发明的一个方面，用于检测基础过程中可能出现的时间性变化的方法通过比较在传统和按时间排序的 k 次交叉验证中获得的性能估计而利用了这一事实。更具体而言，本发明采用了诸如图 6 所示的方法 50 之类的方法，该方法在已加注标签的训练数据上执行传统的 k 次交叉验证（步骤 51）和按时间排序的 k 次交叉验证（步骤 52）两者。在步骤 53 中，对根据两种技术产生的性能估计进行比较。如果由按时间排序的 k 次交叉验证估计出的性能与由传统的 k 次交叉验证估计出的性能相比不足够坏，则将传统的 k 次交叉验证用作对分类器未来性能的准确预测器（步骤 54），并且没有找到时间性变化的证据，即，在收集训练样本的时间范围内没有出现时间性变化，或者即便出现时间性变化，那么由于训练样本是在一段足够长的时间段内收集的，它们具有代表性，因此所述过程表现出平稳性和各态历经

性。

但是，如果基于按时间排序的 k 次交叉验证估计出的性能足够坏（步骤 55），则选择性地生成警告（步骤 56），该警告指出在基础过程中可能出现时间性变化，并且保证做进一步分析。另外在这些条件下，按时间
5 排序的 k 次交叉验证的性能估计与传统的 k 次交叉验证的性能估计相比，提供了更好的对未来分类器性能的短期预测器。

在本发明的另一方面中，当检测到时间性变化时，自动地或者在人工用户控制下执行进一步分析，从而预测通过收集附加的训练数据，可能获得性能的怎样提高。具体而言，构造训练集合大小和预测性能之间的曲线
10 图。另外，执行分析来判断是通过将新获得的训练数据和之前收集的训练数据组合起来将产生更好的性能，还是通过对正在进行的训练数据获取过程使用具有给定大小的滑动窗将产生更好的性能。

图 7 是根据本发明实现的系统 100 的框图。系统 100 检测用于产生已加注标签的训练数据集合 104 的过程 130 中可能出现的时间性变化，并且
15 预测在数据集合 104 上使用有监督学习算法 105 而训练的分类器的未来性能。另外，系统 100 对于是否收集附加的训练数据给出建议，并且如果收集，则对如何利用它给出建议。系统 100 一般包括用于执行存储器 103 中所存储的实现本发明功能的代码（例如多个程序指令）的程序和/或逻辑控制 101（例如处理器 102）。具体而言，存储器 103 优选地包括实现有
20 监督学习算法 105、分类器 106、时间性变化管理器 110 和数据选择模块 111 的代码。

有监督学习算法 105 使用由数据选择模块 111 选出的训练数据 104 中的一些或全部来构造经训练的分类器 106。数据选择模块 111 在程序的控制下，能够根据指定标准 109 对数据进行排序，还能够以确定性的或者伪
25 随机的方式选择经排序数据或者原始数据的子集。按时间排序的 k 次交叉验证算法和传统的 k 次交叉验证算法分别由模块 116 和 112 来实现。如图所示，模块 118 和 114 所产生的性能估计分别与将由图 5B 和图 2B 中的算法所产生的性能估计相同，并且因此模块 116 和 112 可以被认为在逻辑上截然不同。但是，在优选实施例中，所有的排序、子集选择和划分实际上

都由数据选择模块 111 来执行，因此模块 116 和 112 实际上被实现为单个共享的 k 次交叉验证模块，该模块预计数据已被预先划分为 k 个子集。如图 5B 和 2B 所示，交叉验证模块使用学习算法 105 来构造经训练的分类器 106，接着经训练的分类器 106 被用于为每个输入向量 x_i 生成估计分类 \hat{c}_i 。然后，通过比较真实的和预期的分类集合 $\{c_i\}$ 和 $\{\hat{c}_i\}$ 或相应的摘要统计来获得按时间排序的和传统性能估计 118 和 114。在优选实施例中，预期损失被用作公共的性能估计。时间性变化管理器 110 根据从按时间排序的和传统的 k 次交叉验证中获得的摘要统计来构造 ROC 曲线，并且选择使预期的每样本损失最小化的工作点。

10 时间性变化管理器 110 还包括时间性变化检测功能块 120，并且优选地包括未来性能预测功能块 123 和预测性能分析器 124。

时间性变化管理器 110 的时间性变化检测功能块 120 包括比较功能块 121，该功能块将传统 k 次交叉验证的性能估计 113 和按时间排序的 k 次交叉验证的性能估计 117 进行比较，从而确定在基础过程中可能出现的时间性变化。在优选实施例中，比较功能块 120 在使各个预期的每样本损失最小化的各条 ROC 曲线的各个工作点处，对分别从传统 k 次交叉验证性能估计 113 和按时间排序的 k 次交叉验证性能估计 117 中计算出的预期损失 115 和 119 进行比较。因此，在优选实施例中，比较功能块 120 判断使用按时间排序的 k 次交叉验证计算出的预期的每样本损失 119 与使用普通传统的 k 次交叉验证计算出的预期的每样本损失 115 相比是否足够大（在合理的误差边界内）。（对于非二元的情况，产生更高维平面而不是 ROC 曲线；但是，最优工作点以及相关的预期损失仍旧存在，它们可以被计算和比较。）

25 如果按时间排序的 k 次交叉验证性能估计 117 与传统 k 次交叉验证性能估计 113 具有可比性，或者比其更好，则不存在未受控的时间性变化的证据，并且传统的 k 次交叉验证提供了适当的性能预测 123。另一方面，如果由按时间排序的 k 次交叉验证预测的性能与由传统的 k 次交叉验证预测的性能相比足够坏，则暗示存在时间性变化，并且因此传统的 k 次交叉验证方法过高地估计了使用当前可获得的训练数据 104 中的全部数据进行

训练的分类器的性能。在这种情况下，警告产生 122 优选地产生警告，该警告指出在基础过程中可能存在时间性变化。该警告可以以很多不同的方式产生，包括在指定的寄存器或存储器位置中设置一个比特或值、由处理器 102 产生中断、从过程调用中返回参数、调用产生警告的方法或过程 5（例如在图形用户界面中，或者作为外部信号），或者任何用于以信令来通报状态的已知计算机化方法。另外，在这种情况下，预测性能 123 将基于由按时间排序的交叉验证所估计出的每样本预测损失。

一种用于判断由按时间排序的 k 次交叉验证预测出的性能是否比由传统的 k 次交叉验证预测出的性能“足够坏”的方法如下：由于按时间排序的分组是唯一的，因此按时间排序的分组无法被重新采样，从而按照通常由普通的交叉验证所使用的方式来估计所述性能估计的可变性。但是，由于传统的 k 次交叉验证分组是随机选择的，因此可以测试如下虚假设：按时间排序的估计和传统估计之间的差别是由于传统 k 次交叉验证估计中的随机变化。如果在传统 k 次交叉验证的重复应用中，估计性能有 $p\%$ 的时间比按时间排序的 k 次交叉验证所获得的估计性能坏，那么如果所获得的显著性水平 p 小的话，则所述差别可能相当大。 15

也可以使用其它用于估计性能估计的变化性，并且判断它们是不是非常不同的方法。例如，在不重复传统的 k 次交叉验证的情况下，可以完成传统性能估计和按时间排序的性能估计之间的比较。对于传统的 k 次交叉验证和按时间排序的 k 次交叉验证两者，都可以在 k 个评估子集中的每一个或其组合上单独计算性能估计。然后，这些估计在每种类型的交叉验证内的变化性（例如标准方差或范围）可被用作相应的全面性能估计的可信度量度。然后，可以应用传统的统计测试来判断所述估计是否相当不同。 20

由于收集附加训练数据的步骤可能非常昂贵，因此希望在实际收集之前预测可能预期对分类器性能产生什么影响。时间性变化管理器 110 优选地包括预测性能分析器 124，该预测性能分析器 124 与其它功能块一起预测增大已加注标签的训练数据集的大小所产生的影响。通过估计可能产生的性能增益，可以用获取数据的成本换得收益。图 8 示出了操作 60 的优选方法，在操作 60 中，预测性能分析器 124 执行以上功能。如其所 25

述，未来性能预测方法 60 重复地执行按时间排序的 k 次交叉验证，同时改变 k 并且将所产生的性能估计（优选地，在最优工作点处的预期损失）存储为有效的训练集合大小的函数。如果发现预测性能随着训练集合大小的增大而提高，那么结果可以被外插，从而预期训练集合大小的给定增长可能会导致性能收益。相反，如果没有发现性能随着训练集合大小的增大而提高或者提高很小，则说明附加的训练数据未必有用。注意，在该示例中，我们考虑获取附加的训练数据，并且将它们简单地添加到之前的数据中。诸如移动窗之类的附加选项将在下面进行描述。

更详细地讨论方法 60，首先按照时间的递增顺序对可获得的已加注标签的训练数据进行排序（步骤 61），并且将它们划分成具有大致相同大小的 $k = k_1$ 个子集，同时保持排序的顺序不变。如上所述，排序和划分功能由数据选择模块 111 来执行。执行按时间排序的 k 次交叉验证 116，并且存储所产生的性能估计 118 以及有效的训练集合大小 $\frac{(k-1)}{k} \cdot n$ 。然后，增大子集数目 k ，并且重复该过程，直到 k 超过选定的上限 $k > K_2$ 。

当对于 k 次迭代的每个值的性能估计都已被收集时，可以对性能估计（或其摘要数据）进行分析，并且可以计算未来分类器性能的预测。由于训练集合的大小大致随 $\frac{(k-1)}{k} \cdot n$ 而变，因此 k 的较大值接近较大训练集合的效果，当然易于发生统计偏差。然后通过外插，可以利用各种数量的附加训练数据而估计出预期的分类器性能。当然，外插总是带有风险，因此必须利用实际的性能结果来核实这样的预测。但是，即使不经过外插，这样的曲线图也将指示出性能是否仍旧随着训练集合大小而快速改变。预测性能随着训练集合大小的快速提高清晰地指示出训练数据不代表基础过程，并且强烈指出需要收集附加的已加注标签的训练数据。利用内插或外插，还可以使用这样的曲线图来校正从具有不同大小的数据集合（例如两个分别包含 $N1$ 和 $N2$ 个点的数据集合）得出的预测，以回到公共的比较点（例如将对于包含 $N2$ 个点的数据集合的预测性能校正为与对于包含 $N1$ 个点的数据集合的预测性能具有可比性）。这种排序的校正增大了如下可能性：性能上的其余差别是由于数据的实际变化所引起的，而不是由于对

样本大小的简单人为改变所引起。

如果判断出将要收集附加的已加注标签的训练数据，则预测性能分析器 124 优选地判断怎样才能最好地利用已收集的附加的已加注标签的训练数据。例如，可以将附加的已加注标签的训练数据与已加注标签的训练数据 104 的初始集合组合起来，并在单个训练期间被用于训练分类器。可替换地，附加的已加注标签的训练数据可以被用于根据滑动窗方案而使用组合数据的子集来周期性地训练分类器。为了判断如何最好地利用附加的已加注标签的训练数据，预测性能分析器 124 可以利用滑动窗方案来仿真训练，并且可以将所产生的性能估计与使用所有可获得的训练数据得到的性能估计进行比较。这样的分析可以在收集附加的训练数据之前或者之后执行。

图 9 示出了示例性的方法 70，该方法用于判断与使用整个训练集合的情况相比，将滑动窗用于已加注标签的训练数据是否将提高分类器的性能。出于这个目的，按相关时间的递增顺序对训练数据 D 进行排序（步骤 71），然后将经排序的已加注标签的训练数据 D_{SORTED} 划分为 M 个子集 D_1, D_2, \dots, D_M ，优选地，这些子集具有大致相等的大小（步骤 72）。这些操作由数据选择模块 111 来执行。从概念上讲，然后对仿真大小为 n/M 的滑动窗的 D_1, \dots, D_M 中的每一个单独执行按时间排序的 k 次交叉验证，并且将所产生的性能估计与从使用整个数据集 D_{SORTED} 的 k 次交叉验证中得到的结果进行比较。如上所述，在优选的实施例中，排序和划分操作都在数据选择模块 111 中执行，而不是由交叉验证模块来执行。为了对 D_{SORTED} 执行按时间排序的 k 次交叉验证，例如，数据选择模块 111 将 D_{SORTED} 确定性地划分为 k 个子集 $D_{SORTED_1} \dots D_{SORTED_k}$ ，同时保持排序顺序不变。然后这些子集被传递到普通交叉验证模块 116/112，所述普通交叉验证模块 116/112 无需执行任何附加的排序或划分，就可以计算出性能估计。相似地， D_1, \dots, D_M 中的每一个都被单独地划分为 k 个子集，以用于交叉验证模块的处理。

将所产生的性能估计分别标注为 $PE_1 \dots PE_M$ 以及 PE_{SORTED} ，对这些性能估计进行比较（步骤 74）。可能产生若干结果。如果 $PE_1 \dots PE_M$ 在大范围

内变化，那么窗大小 n/M 可能太小，应该被增大。假设这些估计基本一致。在这种情况下，如果 $PE_1 \dots PE_M$ 与 PE_{SORTED} 具有可比性，则意味着将滑动窗用于训练数据的做法将不会提高性能。相反，如果 $PE_1 \dots PE_M$ 好于 PE_{SORTED} ，则指出需要使用滑动窗。利用变化的窗大小（即改变 M ）的进一步分析将被用于选择最优的窗大小。最后，如果 $PE_1 \dots PE_M$ 与 PE_{SORTED} 相比足够坏，那么滑动窗大小可能太小。在这种情况下，可以降低 M 并且重复分析，或者在步骤前进之前收集附加的训练数据。

根据第四种情况，当对于各个子集 D_1, D_2, \dots, D_M 中的每一个子集的性能估计 PE_1, PE_2, \dots, PE_M 彼此之间变化很大时，在产生训练数据样本的基础过程中可能存在时间性变化。在这种情况下，具有不同数据集合大小的滑动训练窗的使用可以提高分类器的性能。因此，可以以各种不同的数据集合大小来重复过程 70，从而判断是否可以提高分类器性能，并且如果可以，则优选地，还使用导致最优分类器性能的数据集合大小。

图 10 示意性地示出了用于训练分类器的滑动窗的概念。在所示实施例中，按时间排序的已加注标签的训练数据 D_{SORTED} 被划分为四个相互独立的子集 D_1 、 D_2 、 D_3 和 D_4 ，这四个子集具有大致相等的大小（即在任意子集中，都没有成员属于任何其他子集）。理想地，应该以大致恒定的采样频率来收集训练数据，从而使相等的样本大小对应于大致相等的持续时间。所述大小代表滑动窗在训练数据中的样本长度。这样，分类器将在子集 D_1 上被训练，稍后在子集 D_2 上被训练，以此类推。滑动窗的最优大小依赖于反映基础过程中的时间性变化的需求和代表性样本数目的需求之间的权衡。

当然，本领域技术人员将会意识到，子集的数目 M 可以根据具体应用而变化，并且还可以将子集构造为彼此重叠，从而使一个或多个子集包括来自于在时间上紧接给定子集之前或者紧随给定子集之后的那个子集的数据样本。按时间排序的 k 次交叉验证提供了用于选择这种滑动窗大小以使性能最优化的机制。

图 11 示出了另一示例性方法 80，该方法用于判断与使用整个训练集合的情况相比，将滑动窗用于已加注标签的训练数据是否将提高分类器的

性能。在该方法中，按相关时间的递增顺序对训练数据 D 进行排序（步骤 81）。 M 个具有大致相等大小的子集 D_1, D_2, \dots, D_M 是从经排序的已加注标签的训练数据 D_{SORTED} 中选出的，同时保持时间上的顺序（步骤 82）。从 M 个子集中选出训练数据子集和相应的测试数据子集对（步骤 83）。测试数据子集被优选地选为在时间上跟随（将数据集合看作循环的）并靠近其相应的训练数据子集。同样，这些操作优选地由数据选择模块 111 来执行。然后，将每个选出的训练数据子集用于训练相应的分类器（步骤 84），然后将相应的分类器用于分类其相应测试数据子集的成员（步骤 85）。利用大小为 n/M 的有效滑动窗，将所分配的分类与已知的真实分类进行比较，从而产生性能估计（步骤 86）。对这些性能估计 $PE_1 \dots PE_M$ 进行比较（步骤 87）。

如果 $PE_1 \dots PE_M$ 基本上具有可比性，那么它们的平均值（或者其他统计摘要）预测出使用大小为 n/M 的滑动窗将会获得的性能（步骤 88）。为了判断滑动窗是否将提高性能，需要将利用大小为 n/M 的滑动窗估计出的性能与使用完整的数据集合估计出的性能进行比较。这样，然后将基本上具有可比性的性能估计 $PE_1 \dots PE_M$ 或者它们的总体摘要（例如它们的平均值）与性能估计 PE_{SORTED} 进行比较（步骤 89），如以上所描述并如图 9 所示出的，性能估计 PE_{SORTED} 是由在总的按时间排序的训练数据集合 D_{SORTED} 上被训练的分类器所产生的。如果来自步骤 89 的比较指示出性能估计 $PE_1 \dots PE_M$ 或其统计摘要足够好于性能估计 PE_{SORTED} ，那么使用大小为 n/M 的滑动窗来训练分类器的做法应该引起分类器性能的提高（步骤 90）。可以以各种不同的数据集合大小（ n/M ）来重复过程 80，从而利用窗大小进行实验以找到能够产生最好的预期性能结果的大小。

但是，如果比较过程（来自步骤 89）指示出性能估计 $PE_1 \dots PE_M$ 或其统计摘要没有足够好于性能估计 PE_{SORTED} ，则不存在证据表明大小为 n/M 的滑动窗将会提高分类器性能（步骤 91）。为了找到能够提高性能的窗大小，可以用各种不同的数据集合大小（ n/M ）来重复过程 80，从而利用窗大小进行实验。

相反，如果发现（在步骤 87 中）性能估计 $PE_1 \dots PE_M$ 差别相当大，则

无法得出明确的结论（步骤 92）（除非性能估计 $PE_1 \dots PE_M$ 的总体或其他统计摘要相当不同于 PE_{SORTED} ）。这样的结果可能是由于窗大小 n/M 太小，而使用更大的窗大小进行训练会得到更加具有可比性的性能估计 $PE_1 \dots PE_M$ 。因此，可以以各种不同的数据集合大小（ n/M ）来重复过程
5 80，以判断是否能够获得分类器性能的提高，如果可以，还优选地使用能够产生最优分类器性能的窗大小 n/M 。

图 12 示意性图示出图 11 的滑动窗方法。在所示实施例中，按时间排序的已加注标签的训练数据 D_{SORTED} 被划分为具有大致相等大小的四个相互独立的子集 D_1 、 D_2 、 D_3 和 D_4 。每个子集 D_1 、 D_2 、 D_3 和 D_4 被用于训练相应的分类器，并且每个相应的分类器被用于分类每个时间上连续的子集
10 （在所示实施例中，以覆盖线表示） D_4 、 D_1 、 D_2 、 D_3 的成员。分类结果被用于产生性能估计 $PE_1 \dots PE_4$ 。（注意，如果假设按时间排序的已加注标签的训练数据 D_{SORTED} 是周期性的，那么可以将其看作循环的，并且因此在时间上随子集 D_4 之后的子集将是 D_1 。如果不假设按时间排序的已加注标签的训练数据 D_{SORTED} 是周期性的，那么将从分析中省略掉相应于训练/测试子集对 D_4/D_1 的性能估计 PE_4 。）

如前所述，训练数据应该以大致恒定的采样频率被收集，因此相等的样本大小对应于大致相等的持续时间。当然，本领域技术人员将会意识到，子集数目 M 可以根据具体应用而变化，并且还可以将子集构造为彼此重叠，从而使一个或多个子集包括来自于在时间上紧接给定子集之前或者紧随给定子集之后的那个子集的数据样本。
20

之前的论述已经假设单个时间足够刻画所考虑过程的时间性变化。这种假设并不总是有效的。可以引入时间性变化的多个源，并且每个源可能需要其自身用于特征化的时间戳。按时间排序的 k 次交叉验证可以被容易地扩展到处理多个时间。继续上述制造示例，假设制造和测量过程中的变化都很重要，并且以样本的制造时间和检查或测量时间两者来标注每个样本。因此现在每个样本具有两个相关时间 t_1 和 t_2 ，这两个时间分别对应于制造和测量时间。可以将它们看作欧几里得空间中的正交维度。因此在本示例中的样本（训练数据）点可以被看作位于两维图形（例如沿着 x 轴的
25

t_1 和沿着 y 轴的 t_2) 上。假设 t_1 的变化比 t_2 的变化具有更大的影响。(可以随机地断开联系。) 通过沿着选择 t_1 轴选择断点而将样本划分为具有大致相等大小的 k_1 个集合。然后, 再通过沿着 t_2 轴选择断点而将这 k_1 个集合中的每一个进一步划分为具有大致相等大小的 k_2 个集合。这就得到了 $k = k_1 k_2$ 5 个方形区域, 其中每个区域包含的样本点数目大致相等。与一维的情况相同, 这些区域可以在训练期间被保持, 从而产生按时间排序的 $k_1 \times k_2$ 次交叉验证。可以将相同的过程方便地扩展到处理更多维数的情况。

注意, 这种按时间排序的分组是一种可能在传统的 k 次交叉验证的过程中产生的有效样本, 虽然其概率较低。如前所述, 可以将由传统的和按 10 时间排序的 k 次交叉验证预测出的性能进行比较, 以检测时间性变化的证据, 从而判断附加训练数据的收集是否适当, 并且判断如何最好地利用这样的附加训练数据。

总之, 本发明在有监督学习和自动分类系统的环境中, 利用传统的和按时间排序的 k 次交叉验证两者来检测和管理时间性变化的某些有问题的 15 情况。它还提供了用于预测在这种情况下构造的分类器的性能的工具。最后, 本发明可被用于提出如下方法, 该方法用于管理训练数据库以及正在进行训练的分类器, 从而在这种时间性改变的情况下使性能最大化。虽然以上内容是根据随时间变化的过程而设计和描述的, 但是应该意识到, 根据其他变量(例如温度、位置等等)的变化也可以用上述方式来应对。

20 虽然已经出于举例说明的目的而公开了本发明的优选实施例, 但是本领域的技术人员将会意识到, 无需脱离所附权利要求中所公开的本发明的范围和精神, 可以进行各种修改、添加和替换。当前公开的发明的其他益处或用途也可能将随着时间而变得更加明显。

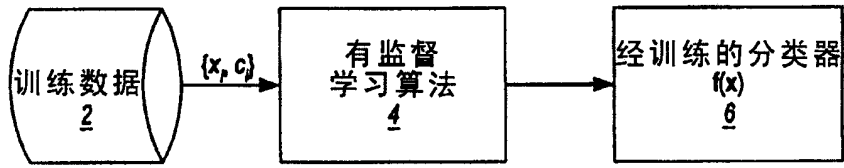


图1
(现有技术)

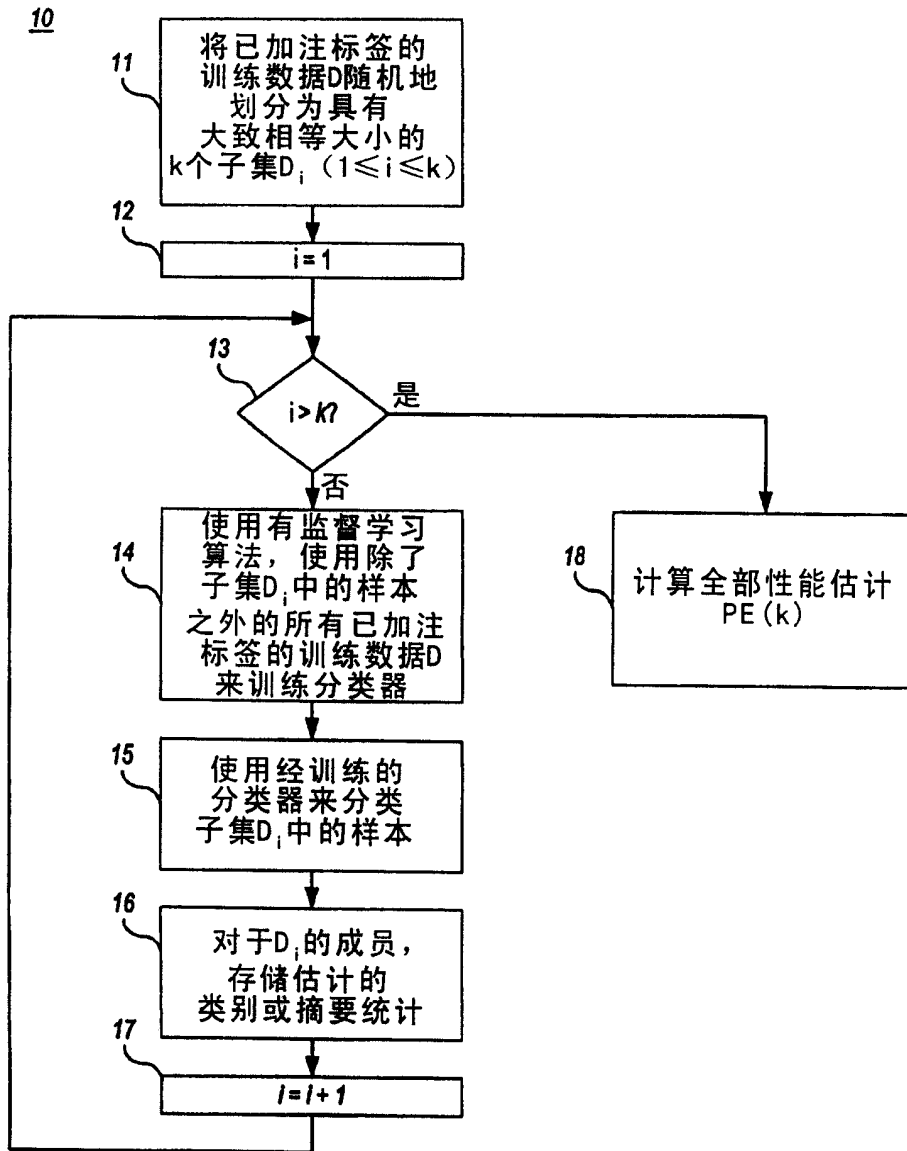


图2B
(现有技术)

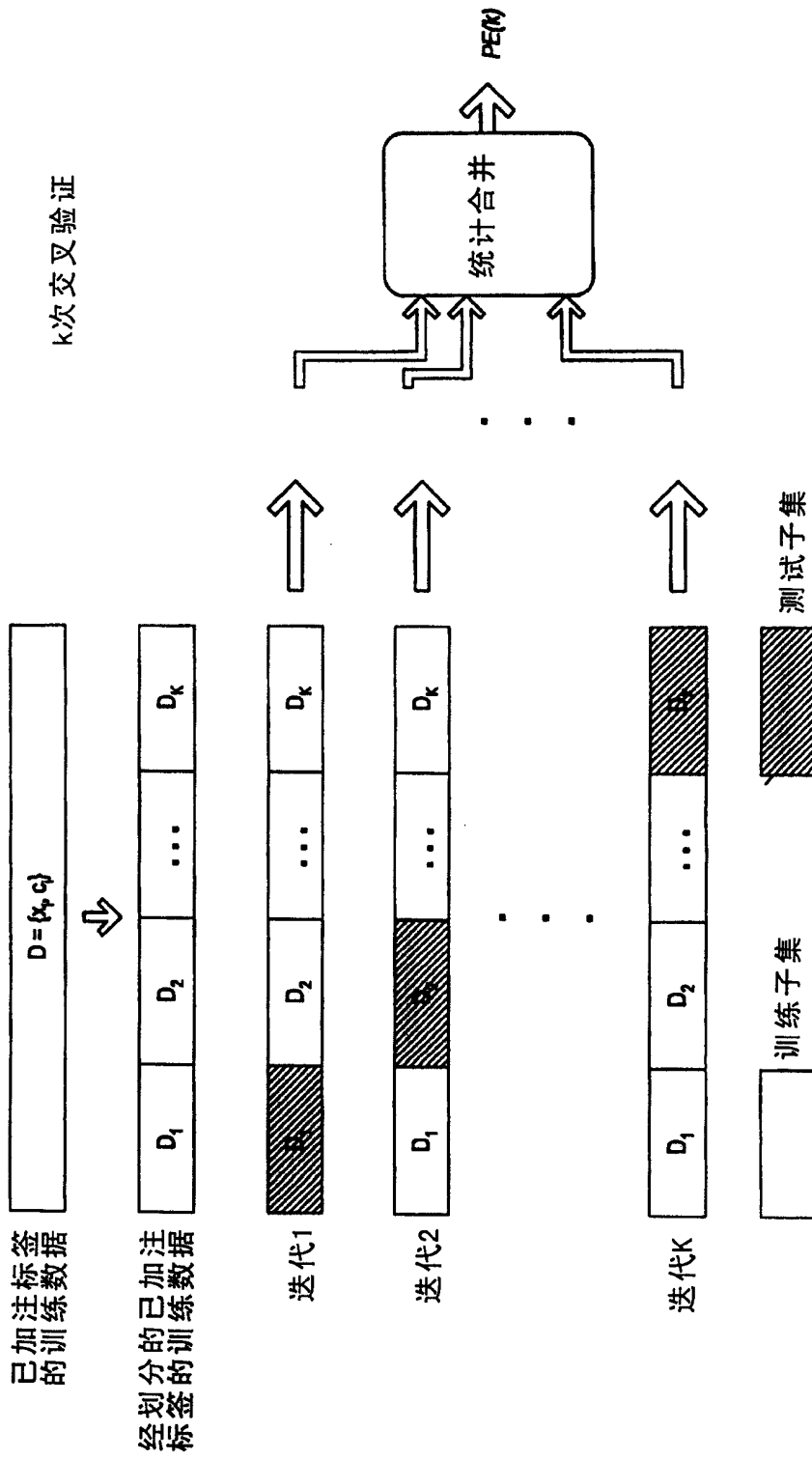


图2A
(现有技术)

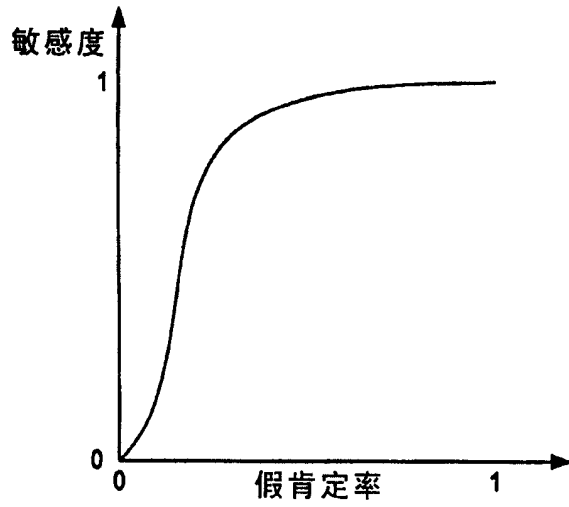


图3

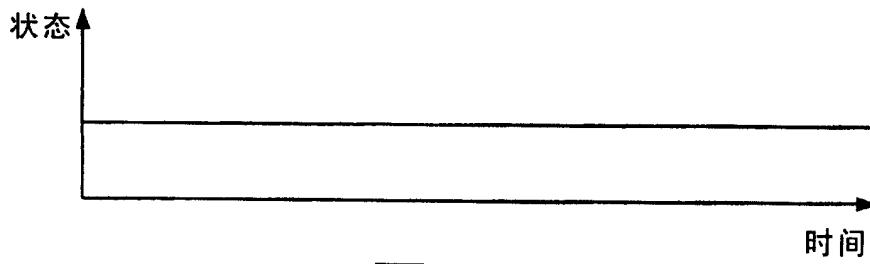


图4A

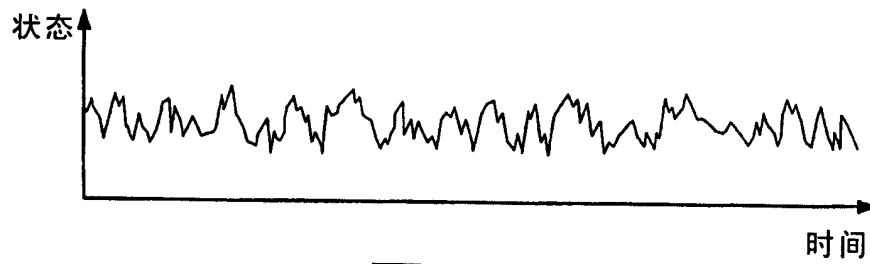


图4B

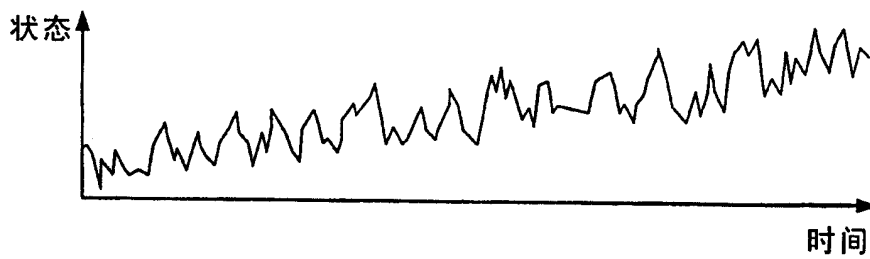


图4C

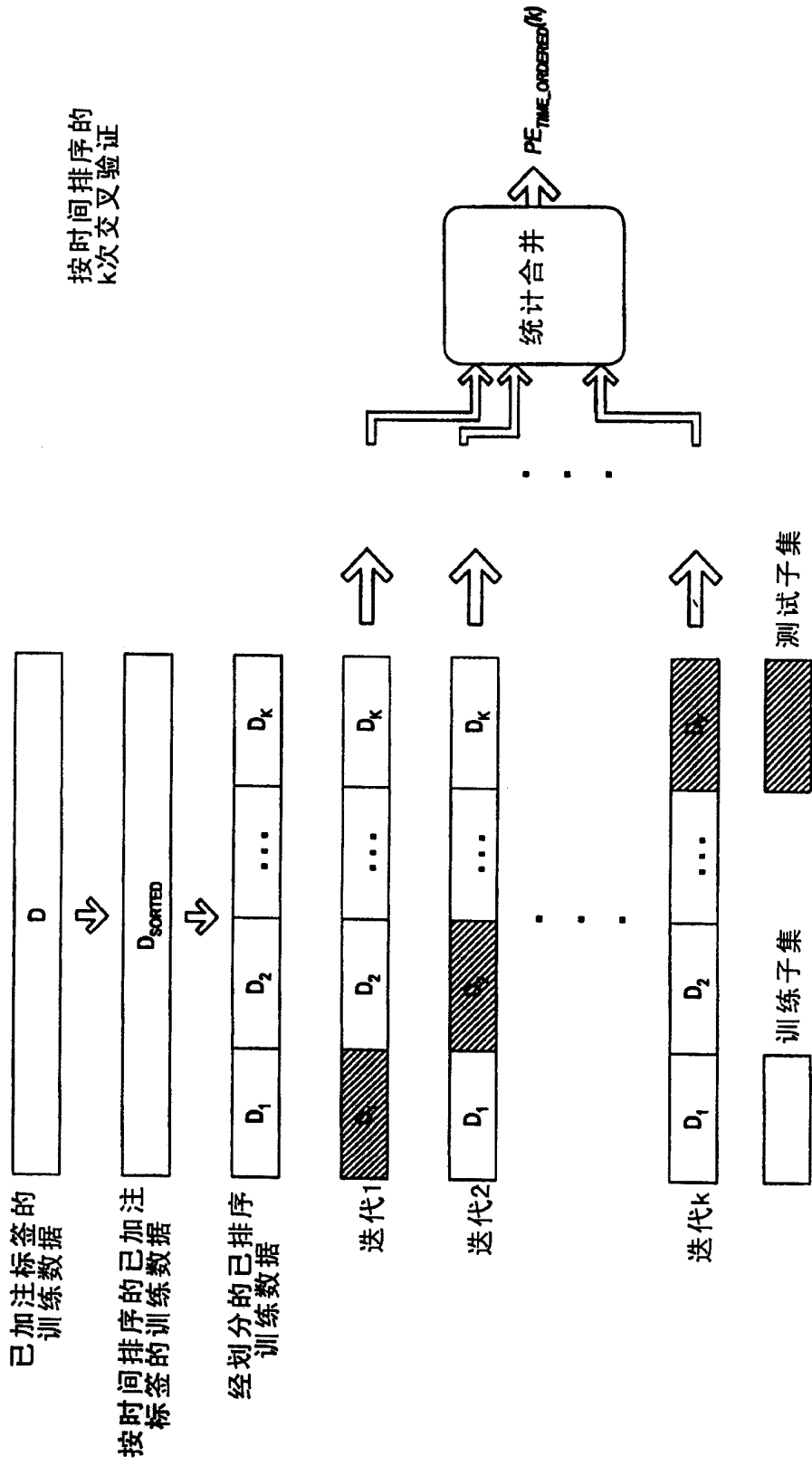


图5A

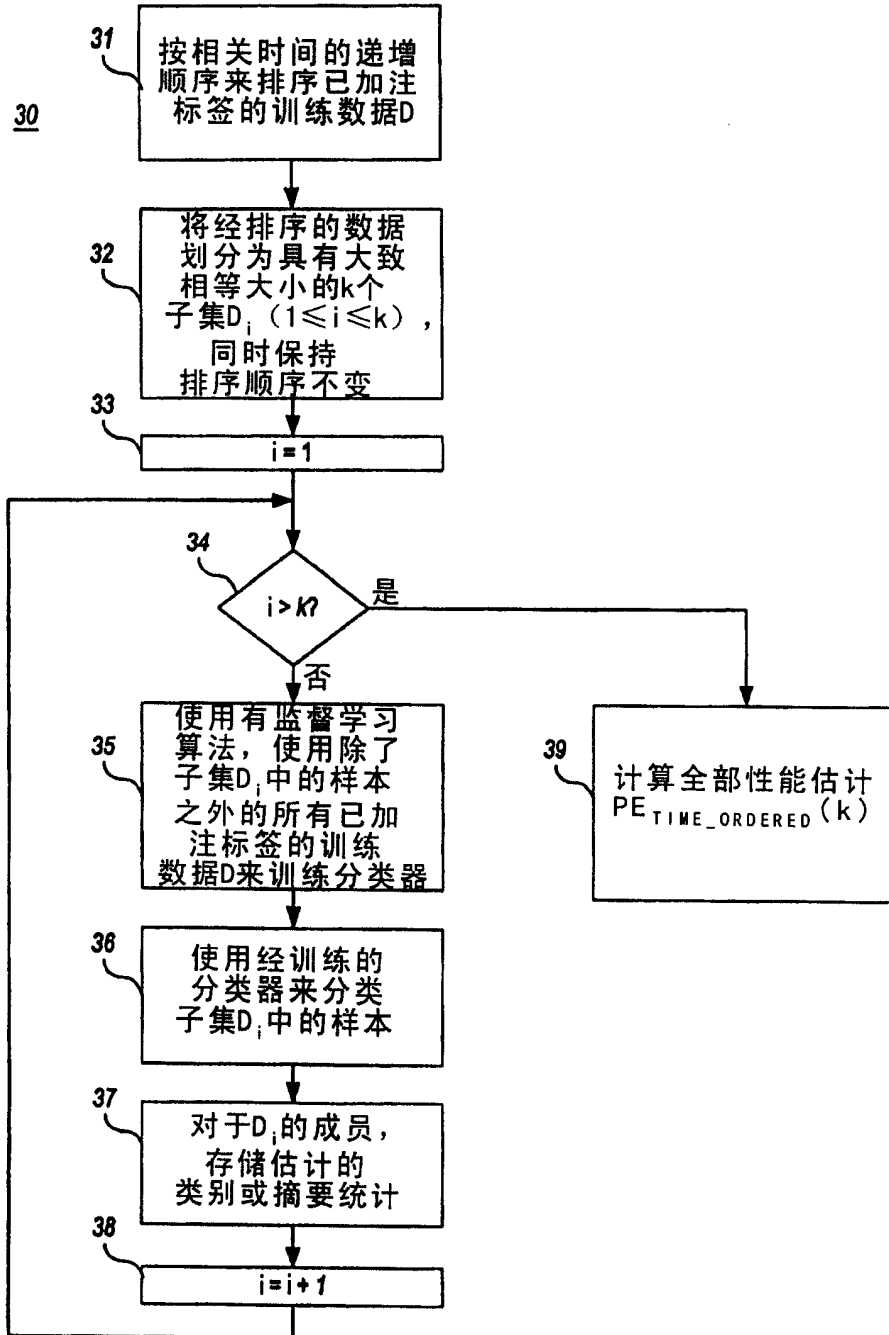


图5B

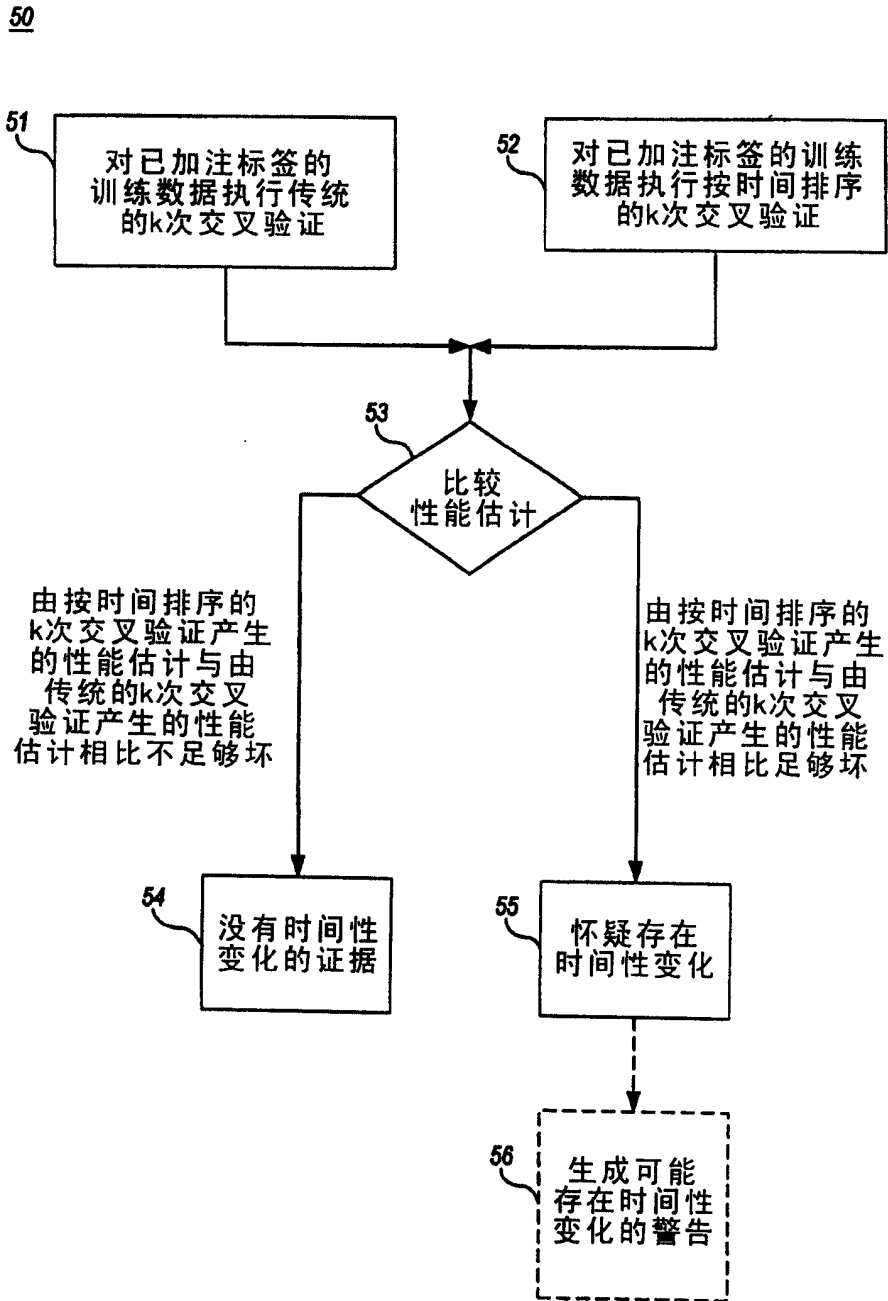


图6

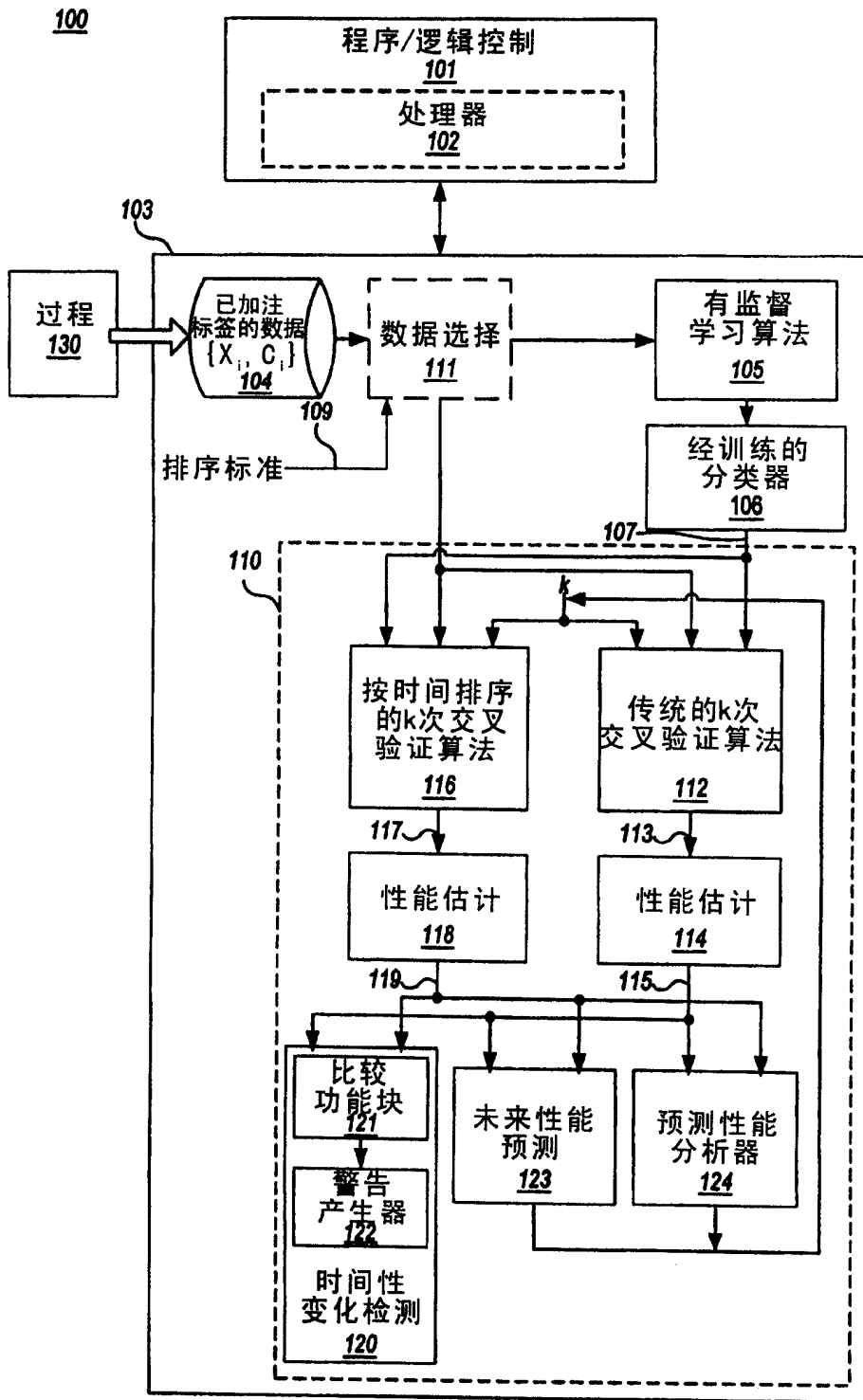


图7

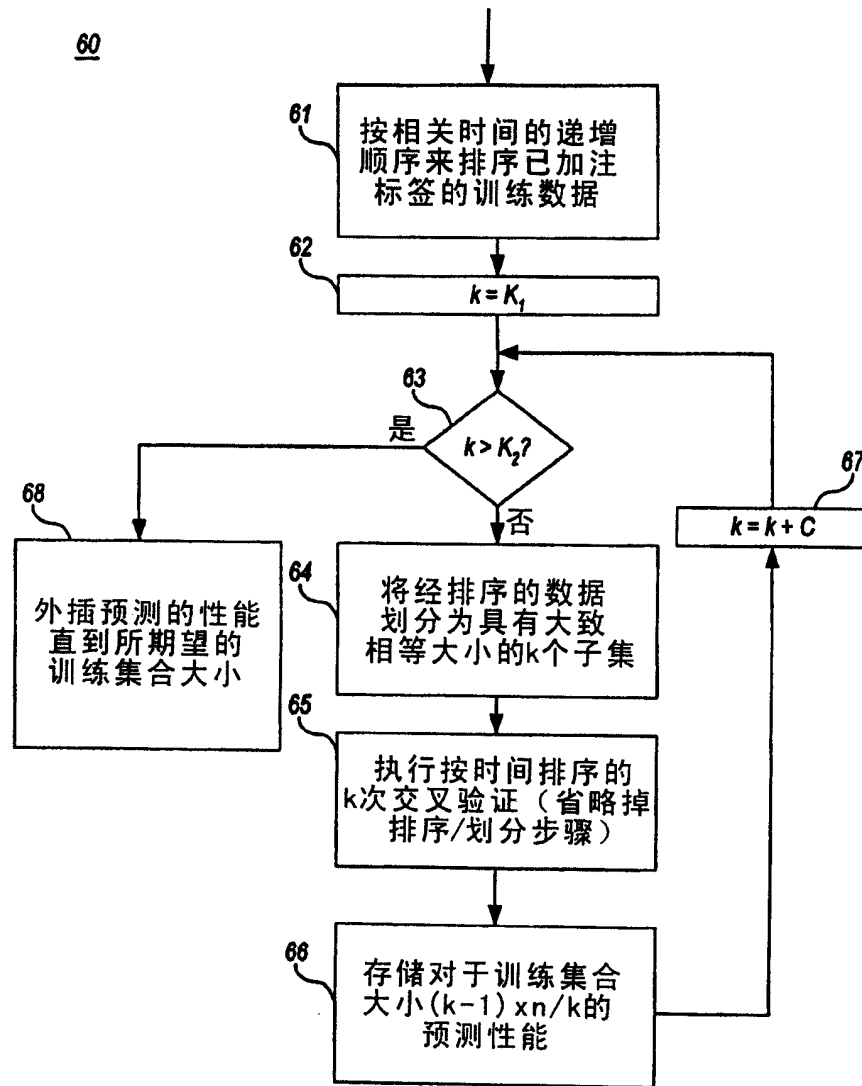


图8

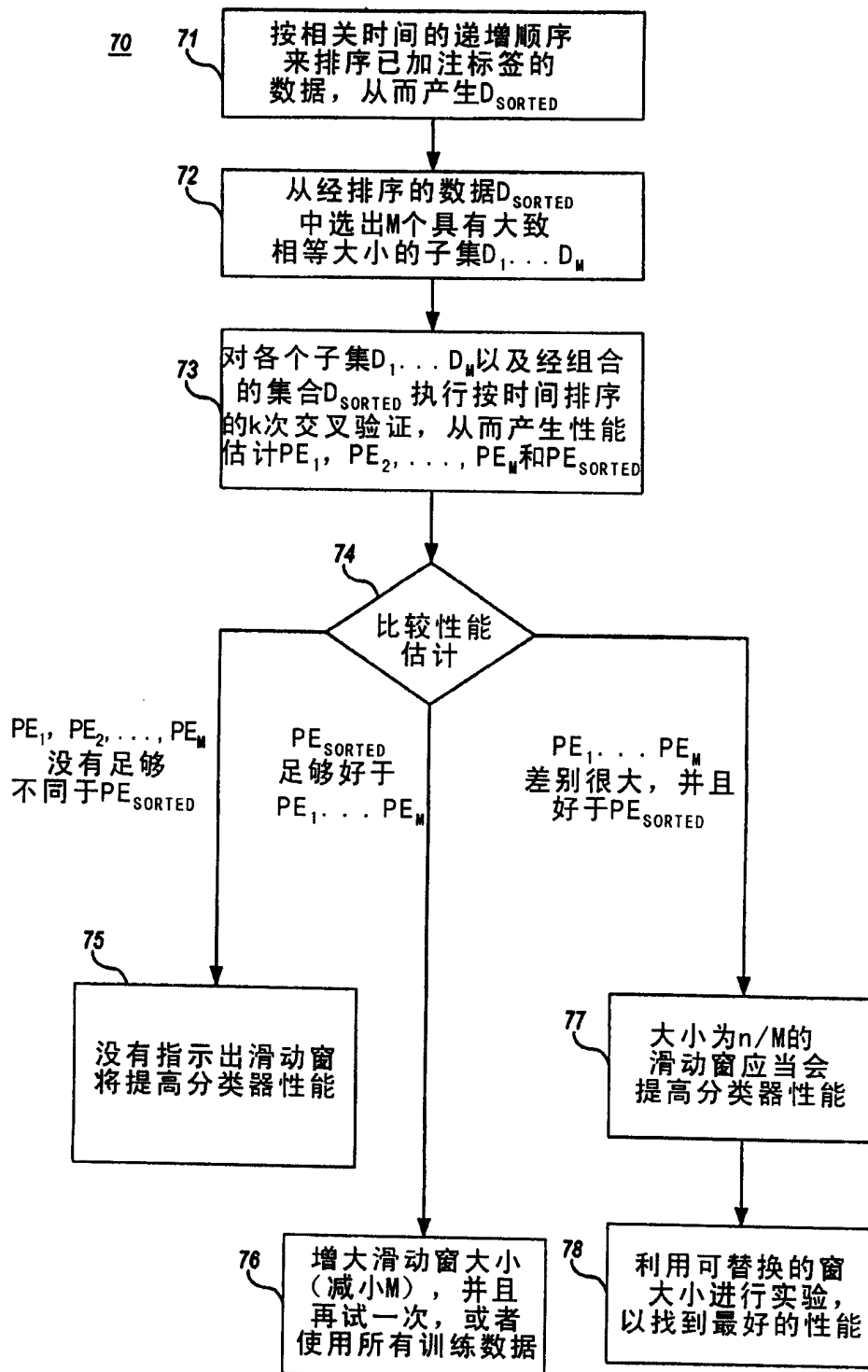


图9

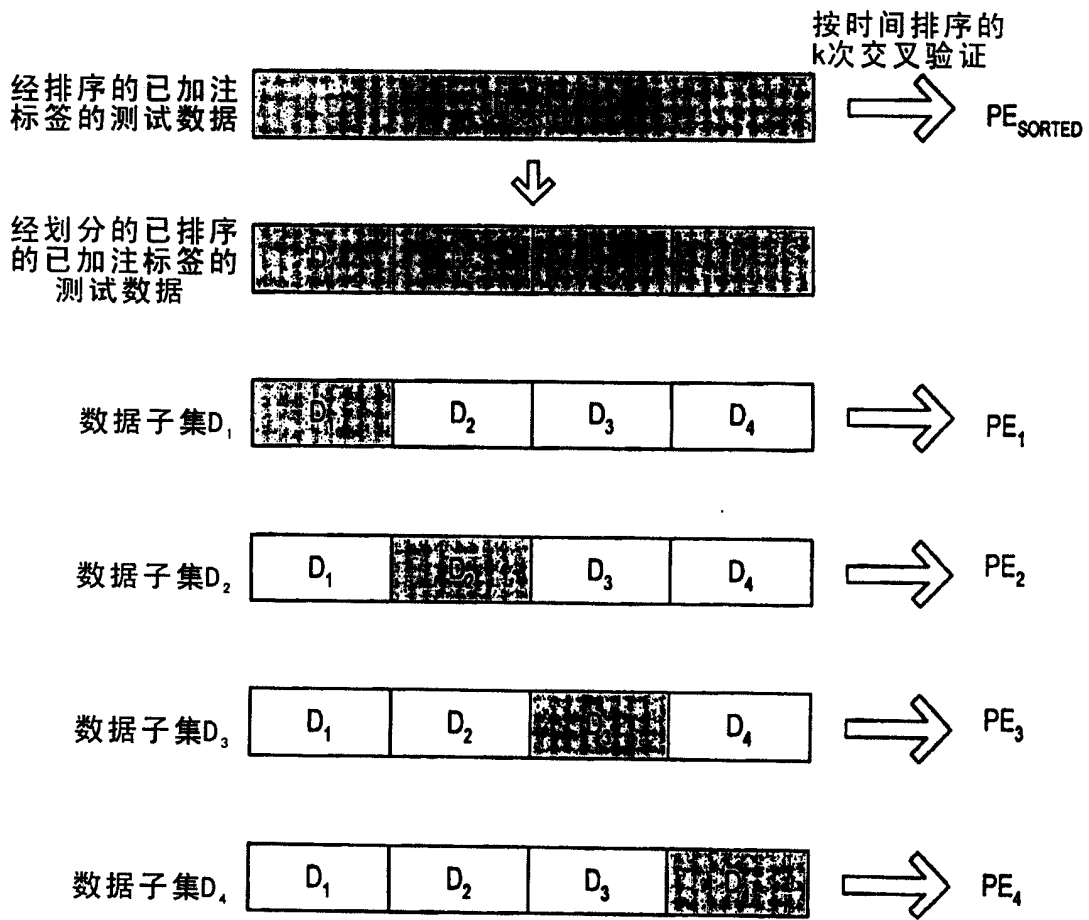


图10

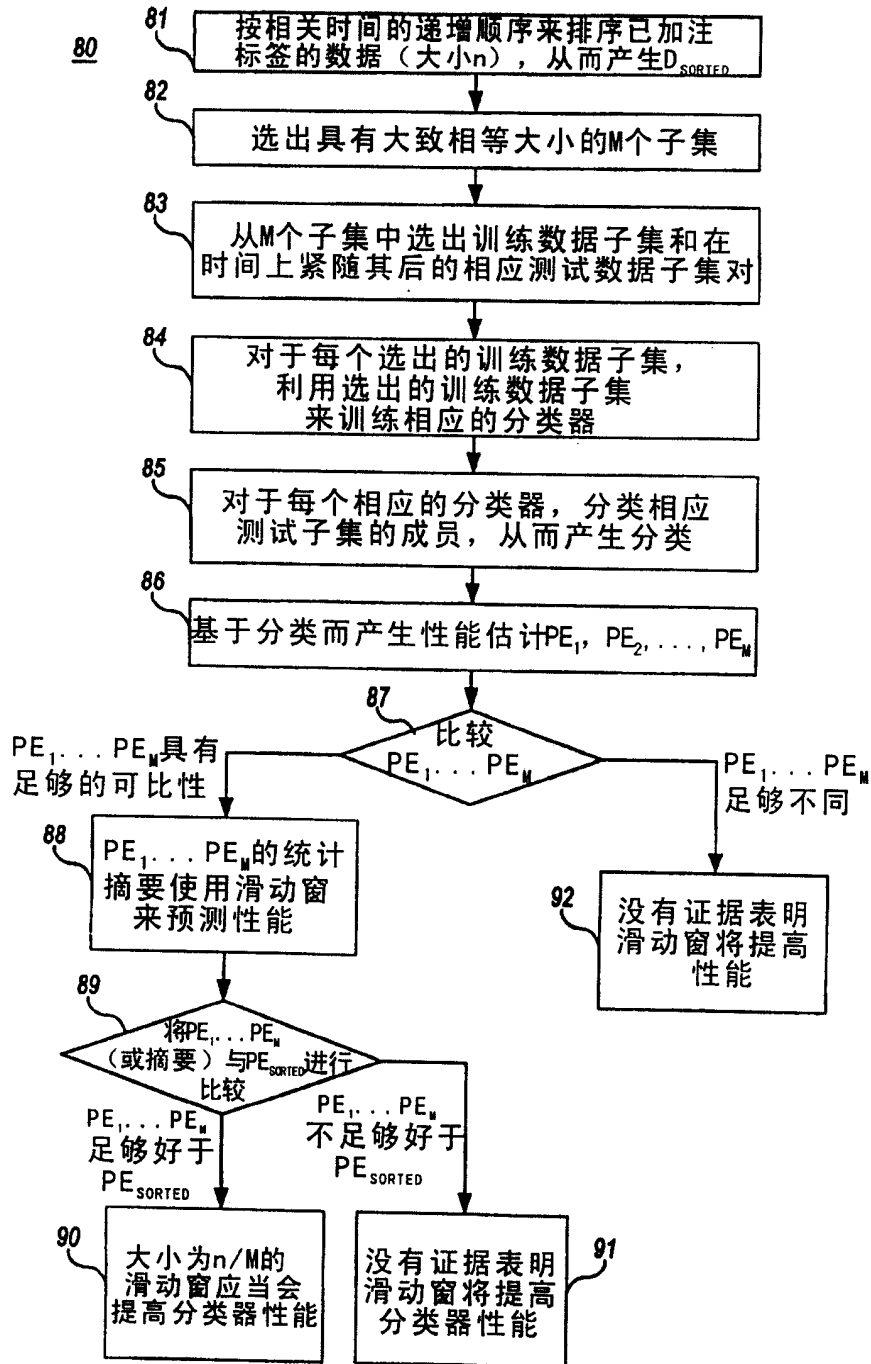


图11

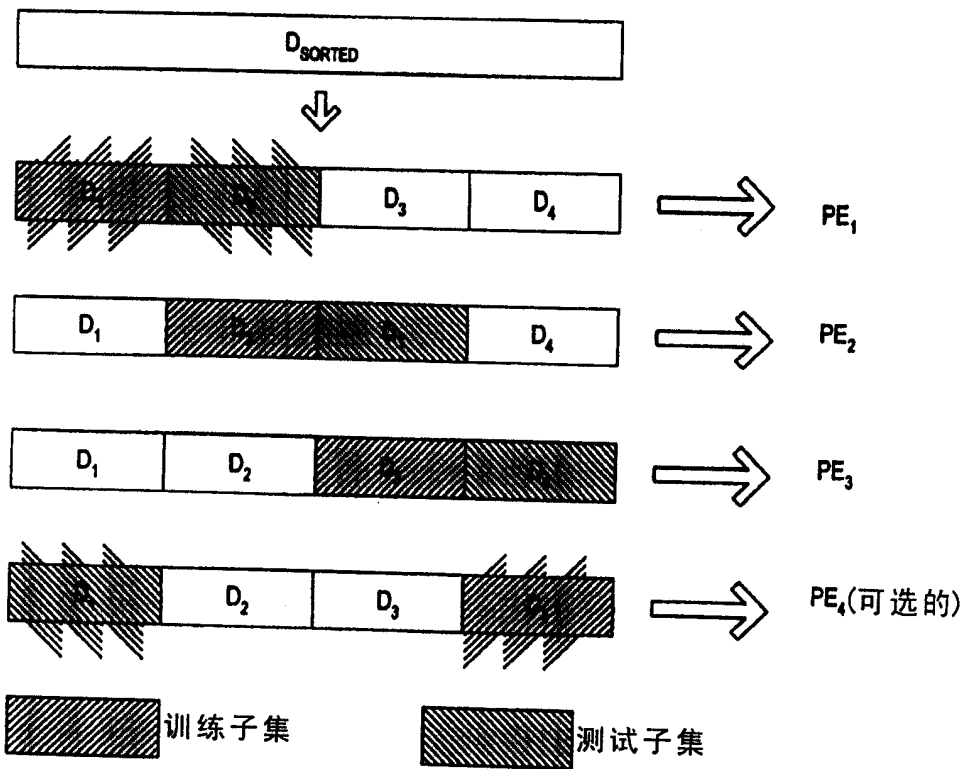


图12