

УДК 681.5.033:656.2

ОЦЕНКА СТЕПЕНИ АДЕКВАТНОСТИ МОДЕЛИ ПРОТОТИПА ПРИ ПОМОЩИ АЛГОРИТМА КОРА

*Бутенко В.М., к.т.н., доцент (УкрГАЗТ)
Меркулов В.С. (УкрГАЗТ)*

Постановка задачи. Все больше и больше технологических операций железнодорожного транспорта реализуется компьютерами [1]. Расчетные среднесуточные объемы погрузочно-разгрузочных работ (ПРР) устанавливаются в результате статистического анализа данных по погрузке-выгрузке станций за период, для которого разрабатывается технология местной работы. Для анализа используются плановые отчетные и учетные формы по прибытию и отправлению.

Анализ последних исследований. В работах [2]–[4] рассматривалась функция автоматизации ПРР. Основой для разработки месячных технических норм являются сбалансированные плановые загруженные и пустые вагонопотоки, задаваемые в виде соответствующей корреспонденции, которая представляет симметричную матрицу, получившую название кривой таблицы или "шахматки".

В таблицу исходной корреспонденции заданных плановых вагонопотоков вводятся итоговые графы, которые разрешают разделить ввоз-вывоз и местные соединения региона. Установленные таким образом объемы выгрузки и погрузки распределяются между станциями с сохранением характерных зависимостей и учетом направления движения. Такой подход к определению расчетных вагонопотоков допускает возможное уменьшение или увеличение, что в дальнейшем может вызвать необъективные технологические решения.

Изложение основного материала исследования. Для оценки полученного плана погрузки-выгрузки в предлагаемом АРМ инженера-технолога отдела станций предусмотрен специальный режим, иницируемый, когда имеются сомнения в качестве полученного плана.

Моделирующий алгоритм, который базируется на методе минимизации эмпирического риска, носит название алгоритм Кора. Рассмотрим конструктивные идеи алгоритма Кора. Задана

обучающая последовательность (эталон) – векторы \bar{x} и \tilde{x} для 1-го и 2-го классов соответственно.

Задано множество характеристических функций $\varphi(x, \tau)$ – признаков. Из него выделим достаточные признаки: для 1-го класса $\varphi(x, \tau^*)$ – на всех векторах 2-го класса равные нулю, а на некоторых векторах 1-го класса равные единице. Аналогично для 2-го класса.

Выбираем t достаточных признаков 1-го класса и t достаточных признаков 2-го класса, чтобы для любого вектора обучающей последовательности существовало несколько достаточных признаков, равных единице (признаки должны "накрывать" все множество \bar{x} или \tilde{x}).

Опознание вектора, не участвующего в обучении, заключается в следующем: вектор относится к тому классу, для которого число достаточных признаков, равных единице, больше.

Особенностью алгоритма является то, что при анализе бинарного пространства X , в качестве множества признаков $\varphi(x, \tau)$ берутся все возможные конъюнкции 3-х переменных. Причем отбор производится следующим образом:

1) из всех возможных отбираются достаточные, которые упорядочиваются по правилу: $\varphi(x, \tau_1)$ лучше $\varphi(x, \tau_2)$, если число векторов \bar{x} или \tilde{x} , обладающих этим признаком, для которых $\varphi(x, \tau_1) = 1$, больше числа векторов, обладающих признаком $\varphi(x, \tau_2)$;

2) из множества достаточных признаков исключаются "подчиненные", для которых множество обучающей последовательности $\{x: \varphi(x, \tau_1) = 1\}$ включает в себя множество $\{x: \varphi(x, \tau_2) = 1\}$;

3) из оставшихся признаков производится окончательный отбор по принципу: в окончательный набор должны входить признаки, накрывающие все множество обучающей последовательности; примеры, по возможности должны обладать одинаковым количеством признаков.

Рассмотрим действия подсистемы, реализующей алгоритм Кора.

1. Установить номер текущего исходного элемента i -номер последнего элемента списка.
2. Взять i -й элемент (исходный) с r признаками.
3. Установить номер текущего сравниваемого элемента ii ($ii=i-1$).

4. Начиная с i найти элемент (сравниваемый), у которого число признаков $\geq r+1$ и который является ближайшими к i -му (исходному). Если таких элементов нет, то п.14, иначе п.5.

5. Поочередно сравнить исходные признаки со сравниваемыми.

6. Если в сравниваемом элементе r признаков сравнения не совпадали со всеми r исходными признаками, то берем следующий сравниваемый элемент $i=i-1$ и п.4.

7. Если в сравниваемом элементе r признаков совпали со всеми r исходными признаками, то выбираем $k=r+1 \cdot n$ -й сравниваемый признак из не совпавших.

8. Просматриваем список, начиная с j -го ($j=i+1$) элемента (просматриваемого), за исключением i -го (исходного) элемента.

9. Перебираем все признаки j -го элемента (просматриваемого) поочередно. Если закончен список признаков j -го просматриваемого элемента, то $j=j-1$ и п.8, иначе п.10.

10. Если у просматриваемого элемента нет признаков, совпадающих с признаком $k=r+1$ сравнения, и просматриваемый список закончен, то исключаем данный признак в сравнении; увеличив номер признака $k=k+1$ и п.13.

11. Если у просматриваемого есть признак, совпадающий с признаком $k=k+1$ сравнения, то $k=k+1$ п. 13.

12. Если закончен список просматриваемых элементов то п.13, иначе то $j=j-1$ и п.9.

13. Если закончились признаки у сравнения, то новое сравнение элемента $i=i-1$ и п.4, иначе п.7.

14. Конец.

Процесс распознавания содержит следующие этапы

Формирование последовательности

признаков

Вход : N - число разрядов кода

Выход: P $i=1,3; j=1,T,$

PP $i=1,N; j=1,T$ T - длина

последовательности

**Формирование обучающей
последовательности**

Вход: M - длина последовательности, N
Выход: X_1, X_2, \dots, X_M , $i=1, M; j=1, N$

Выбор достаточных признаков

Вход: X_1, X_2, P
Выход: P_1, P_2, \dots, P_T , $i=0, M; j=1, T$

**Формирование достаточных признаков
(исключение пустых строк)**

Вход: P_1, P_2, PP
Выход: PD_1, PD_2, \dots, PD_M , $i=0, M; j=1, CC(DD)$
 PP_1, PP_2, \dots, PP_N , $i=0, N; j=1, CC(DD)$

**Упорядоченность достаточных признаков
(по убыванию числа распознающих экз-в)**

Вход: PD_1, PD_2, PP_1, PP_2
Выход: PD_1, PD_2, PP_1, PP_2

**Исключения подчиненных достаточных
признаков**

Вход: PD_1, PD_2, PP_1, PP_2
Выход: $PD_{01}, PD_{02}, \dots, PD_{0M}$, $i=0, M; j=1, CD(DC)$
 $j=1, m; i=0, KL_{1+1}(KL_{2+1})$

**Формирование массива совпадений
по каждому экзамену**

Вход: PD_{01}, PD_{02}
Выход: KL_1, KL_2, \dots, KL_M , $i=0, KL_{1+1}(KL_{2+1})$

**Формирование массива совпадений по
каждому экзамену после исключения пустых
строк**

Вход: KL_1, KL_2
Выход: $KLU_1, KLU_2, \dots, KLU_M$, $i=0, M_1(M_2)$, $i=0, KLU_1(KLU_2)$

Упорядочение массива совпадений по убыванию числа признаков

Вход: KLU1, KLU2

Выход: KLU1, KLU2

Формирование последовательности t Достаточных признаков

Вход: KLU1, KLU2, PP01, PP02, PD01, PD02

Выход: PDR1 , PDR2 PPZ1 , PPZ2

$j=1, CD (DC) i=0, N i1=0, M j1=1, C1 (C2)$

Формирование распознающей последовательности

Вход: Z, PPZ1, PPZ2

Выход: Z1 , Z2 ; $i=0, i1 (i2), i1+i2=MR, j=1, N$

Распознавание

Вход: Z, PPZ1, PPZ2

Выход: Z1 ; Z2; $i=0, i1 (i2), i1+i2=MR, j=1, N$

Выводы. В процессе распознавания возможно возникновение следующих ситуаций :

Рабочий вариант: минимальное число признаков на 1 экземпляр по обработке 1 и обработке 2 совпадают, но другие экземпляры число признаков много больше как для 1, так и 2 обработки. Отдельные экземпляры не распознаны для 1, 2 или обоих обработок (неравномерно неполное совпадающее покрытие).

Идеальный вариант: минимальное число признаков на 1 экземпляр по обработке 1 и обработке 2 совпадающих статей; такое же число приходится на остальные экземпляры (равномерное полное совпадающее покрытие). Все экземпляры распознаны.

Запасной вариант: минимальное число признаков на 1 экземпляр по обработке 1 и обработке 2; на другие экземпляры число признаков такое же; не все экземпляры распознаны (равномерное неполное совпадающее покрытие).

Таким образом, наиболее целесообразным применением алгоритма Кора является его внедрение в технологических автоматизированных рабочих местах с целью проверки адекватности модели прототипа.

Следует отметить, что окончательный вывод об адекватности модели прототипу делает технолог, занимающийся планированием., на основе проверки результатов статистическими методами на репрезентивной выборке.

Список литературы

1. Мойсеенко В. И., Бутенко В. М., Кузьменко Д. М. Компьютерная система управления движением поездов //Залізничний транспорт України. – 2000. – № 5 – 6. – С.80 – 82.
2. Меркулов В.С., Самсонкин В.Н. Система оперативного планирования погрузочно-разгрузочных работ //Информационно-управляющие системы на железнодорожном транспорте. – 2001.– № 4. – С. 142 – 143.
3. Використання критерію Кузьміна для кількісної оцінки якості оперативного плану //Информационно-управляющие системы на железнодорожном транспорте. – 2001.– № 4. – С. 142 – 143.
4. Меркулов В.С. Математическая модель автоматизированной системы оперативного планирования погрузки-выгрузки в регионе дороги. //Збірник наукових праць Київського університету економіки і технологій транспорту, Том 6.- 2001.-с.123
5. Меркулов В.С. Методика застосування автоматизованої системи оперативного планування вантаження та розвантаження у регіоні дороги //Інформаційно-керуючі системи на залізничному транспорті. – 2004.– №4.– С. 34 – 39.
- 6.Меркулов В.С. Використання аналітичного критерію для оцінки якості оперативного плану Ж //Інформаційно-керуючі системи на залізничному транспорті. – 2005.– № 4. – С. 25 – 29.

УДК 656.212

*Луханін М.І. к.т.н. (начальник Донецької залізниці)
Чеклов В.Ф., к.т.н. (ДонІЗТ)
Похилко С.П., к.т.н. (ДонІЗТ)*

**ОПТИМІЗАЦІЯ ПРОЦЕСУ РОЗФОРМУВАННЯ СОСТАВІВ
НА СОРТУВАЛЬНИХ СТАНЦІЯХ**

Постановка проблеми. Простій вагонів на технічних станціях є одним з основних якісних показників роботи залізниці. На Донецькій залізниці за період з 1993 по 2000 рік простій вагонів на технічних