Информационные процессы, Том 17, № 1, 2017, стр. 14–18

МАТЕМАТИЧЕСКИЕ МОДЕЛИ, ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

Решение задачи классификации для построения прогнозных моделей пассажиропотока в среде MATLAB[1]

В. М. Антонова[*,a], **Д. О. Волков**[**,b], **Н. А. Кузнецов**[***,c], **А. М. Старостенко**[**]

[*] *Московский государственный технический университет им. Н.Э. Баумана, Москва*
[**] *Московский физико-технический институт (государственный университет), Долгопрудный*
[***] *Институт радиотехники и электроники им. В.А. Котельникова РАН, Москва*
e-mail: [a] *xarti@mail.ru,* [b] *danwolf@mail.ru,* [c] *kuznetsov@cplire.ru*

Поступила в редколлегию 18.01.2017

Аннотация—В данной работе рассмотрено решение задачи классификации и описаны способы построения прогнозных моделей для входящего на станцию пассажиропотока на основании набора исходных данных с помощью двух известных методов: наивного байесовского классификатора и нейронных сетей.

КЛЮЧЕВЫЕ СЛОВА: пассажиропоток, наивный байесовский классификатор, нейронная сеть

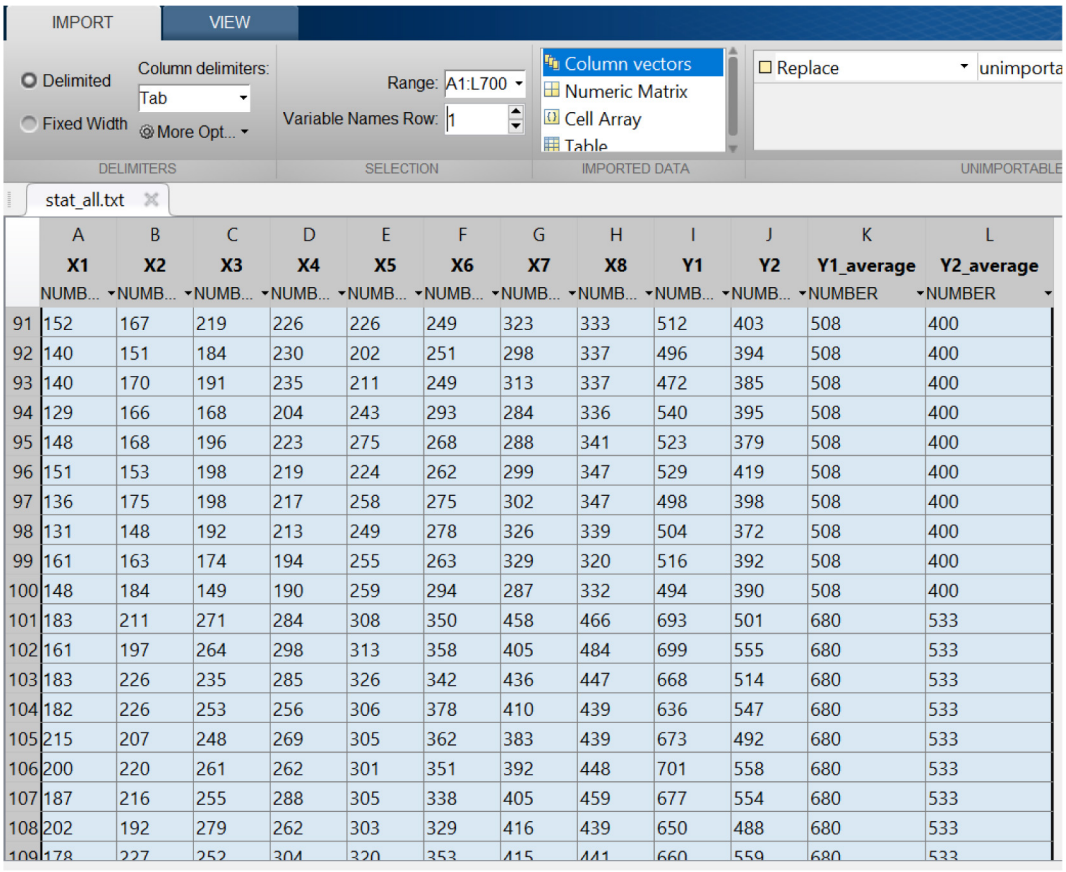
Задачи классификации являются наиболее популярными в современной индустрии. В области транспортных задач с их помощью можно регулировать: потребности в перевозках пассажиропотока, расписание движения транспорта, а также учитывать предоставление дополнительных услуг. Сформулированная в данной работе задача относится к проблеме построения модели классификации. В качестве исходных данных был взят каталог, содержащий в себе численные значения входящих на станцию пассажиров в различные отрезки времени, сгенерированный при помощи имитационного моделирования, в качестве которого используется двугорбое распределение, получаемое путем сложения трех нормальных распределений с пиками утром, днем, вечером [1]. Утром и вечером пики высокие и явные, так как в эти часы интенсивность пассажиропотока очень высока, в дневные часы – пик меньше и шире, это происходит из-за снижения интенсивности пассажиропотока.

В общем случае задача классификации – формализованная задача, в которой имеется множество возможных ситуаций [2], разделённых некоторым образом на классы. Заданное множество ситуаций должно быть конечным (выборка). Классовая принадлежность остальных объектов неизвестна. Для ее определения требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Первый используемый в работе классификатор “Наивный байесовский классификатор” – простой вероятностный классификатор, основанный на применении Теоремы Байеса [3], которая позволяет определить вероятность какого-либо события при условии, что произошло другое статистически взаимозависимое с ним событие, со строгими (наивными) предположениями о независимости. Достоинством наивного байесовского классификатора является малое количество данных для обучения, необходимых для оценки параметров, требуемых для классификации. В применении к рассматриваемой задаче, разработанная программа использует только 10% общих данных для обучения классификатора. Недостатком является малое количество совпадений между прогнозируемыми и реальными величинами, что означает недостаточную точность классификации, то есть данный классификатор относительно прост и примитивен.

[1] Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 16-29-09497 офи-м).

С помощью Excel и промежуточного *.txt файла производится импорт таблицы с характеристиками входящего на станцию пассажиропотока в MATLAB. На рис. 1 представлена входящая матрица пассажиропотока, по вертикале отложены различные дни сбора данных с шагом по горизонтали 10 минут.



	A	B	C	D	E	F	G	H	I	J	K	L
	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2	Y1_average	Y2_average
	NUMB...	NUMB...	NUMB...	NUMB...	NUMB...	NUMB...	NUMB...	NUMB...	NUMB...	NUMB...	NUMBER	NUMBER
91	152	167	219	226	226	249	323	333	512	403	508	400
92	140	151	184	230	202	251	298	337	496	394	508	400
93	140	170	191	235	211	249	313	337	472	385	508	400
94	129	166	168	204	243	293	284	336	540	395	508	400
95	148	168	196	223	275	268	288	341	523	379	508	400
96	151	153	198	219	224	262	299	347	529	419	508	400
97	136	175	198	217	258	275	302	347	498	398	508	400
98	131	148	192	213	249	278	326	339	504	372	508	400
99	161	163	174	194	255	263	329	320	516	392	508	400
100	148	184	149	190	259	294	287	332	494	390	508	400
101	183	211	271	284	308	350	458	466	693	501	680	533
102	161	197	264	298	313	358	405	484	699	555	680	533
103	183	226	235	285	326	342	436	447	668	514	680	533
104	182	226	253	256	306	378	410	439	636	547	680	533
105	215	207	248	269	305	362	383	439	673	492	680	533
106	200	220	261	262	301	351	392	448	701	558	680	533
107	187	216	255	288	305	338	405	459	677	554	680	533
108	202	192	279	262	303	329	416	439	650	488	680	533
109	178	227	252	304	320	353	415	441	660	559	680	533

Рис. 1. Входные данные пассажиропотока на станцию и импорт данных в MATLAB.

Для того, чтобы классифицировать входящий поток пассажиров по потребности в перевозке, сначала определим, сколько всего независимых видов входящих потоков мы имеем. Для этого используем функции length() (возвращает размер или длину) и unique() (производит сравнение) – length(unique(_)) (возвращает количество уникальных элементов в массиве). В результате получаем 14 независимых классов. Далее создается выборка, которая разбивается на обучающую и тестовую. Для этого используется функция cvpartition() с отложенным тестом holdout. Программа, в отдельные переменные извлекает то, что в задаче относится к обучающему подмножеству и к тестовому.

В данной работе, для получения обучающей выборки, использован метод fit и метод predict (прогноз) для получения тестовой выборки. Эти методы используются для сравнения истинных значений с прогнозными, полученными в результате работы классификатора. После суммирования все совпадения, получено количество неверных ответов. Качество работы классификатора оценено с помощью визуализации полученной матрицы.

Для построения картинки используем новые переменные и функции:

1) confusionmat() – возвращает совпадения между прогнозируемыми и известными величинами;

2) diag() – диагональ матрицы;
3) imagesc() – обрабатывает изображение;
4) figure() – построение графика;
5) disp() – выводит в командное окно текст;
6) num2str() – преобразует число в строку.

В ходе работы данного классификатора получено 90% правильных ответов в пределах +/- одиного класса. На рис. 2 показана матрица несоответствия. Если классификатор хороший, то на главной диагонали должны быть единицы, что означает, что класс по оси ординат (известные отклики, которые хотели повторить) и класс по оси абсцисс (предикторы) совпадают.

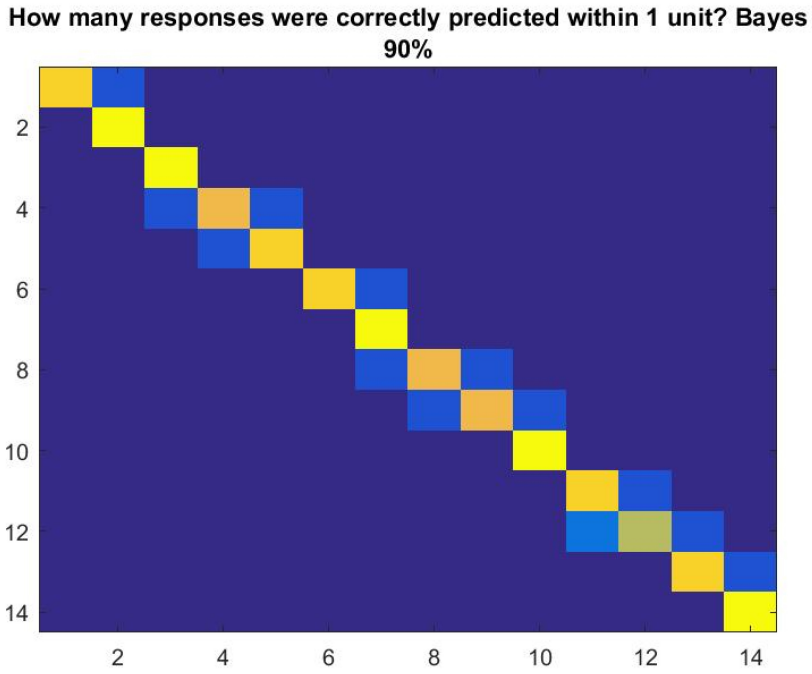


Рис. 2. Количество ответов, полученных с помощью Наивного байесовского классификатора.

Второй рассматриваемый в работе классификатор – “Нейронная сеть” [3], который представляет собой структуру соединенных между собой смоделированных программным образом нейронов. Данная сеть характеризуется: образующими ее нейронами, индивидуальной топологией (архитектурой), а также правилами обучения (тренировки). В классификаторе нейрон выполняет функцию адаптивного сумматора с регулируемыми уровнями входных сигналов, который осуществляет дополнительную линейную обработку вычисленной суммы с целью получения результата. Данный классификатор имеет высокую адаптивность (нейросети могут приспосабливаться к изменению ситуации по средствам своего поведения, и нейронная сеть подходит для моделирования нелинейной зависимости), что является его преимуществом. Однако, его недостатком является то, что качество решения зависит от логики выбора входных данных и способа обучения.

Для реализации данного классификатора в среде MATLAB необходимо произвести предварительную подготовку оценки качества работы нейросети. Для графической оценки создана матрица “*количество классов* x *количество переменных”,* которая будет служить для подстановки в качестве ответа. Для обучения нейросети было использовано 10 нейронов, patternnet(10), сама сеть обучается с помощью функции train().

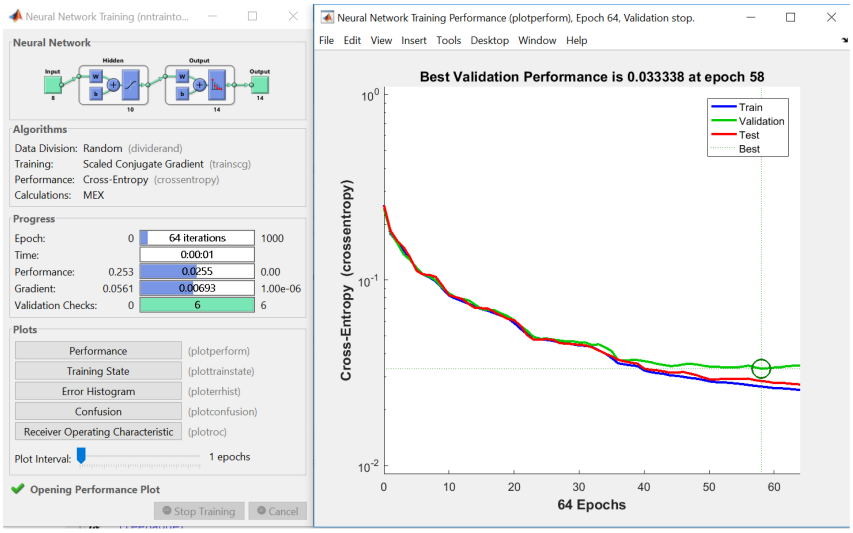


Рис. 3. График качества обучения.

Из представленного графика качества обучения на рис. 3 видно, что остановка обучения производится в момент, когда график тестовой выборки начинает возрастать, что показывает увеличения количества ошибок (своеобразный механизм защиты от “переобучения” модели). Оценка иллюстрации работы классификатора произведена тем же образом, что и в предыдущей части (рис.4). Здесь доля правильных ответов в пределах +/- одного класса составила 85%, что хуже, чем у Наивного байевского классификатора.

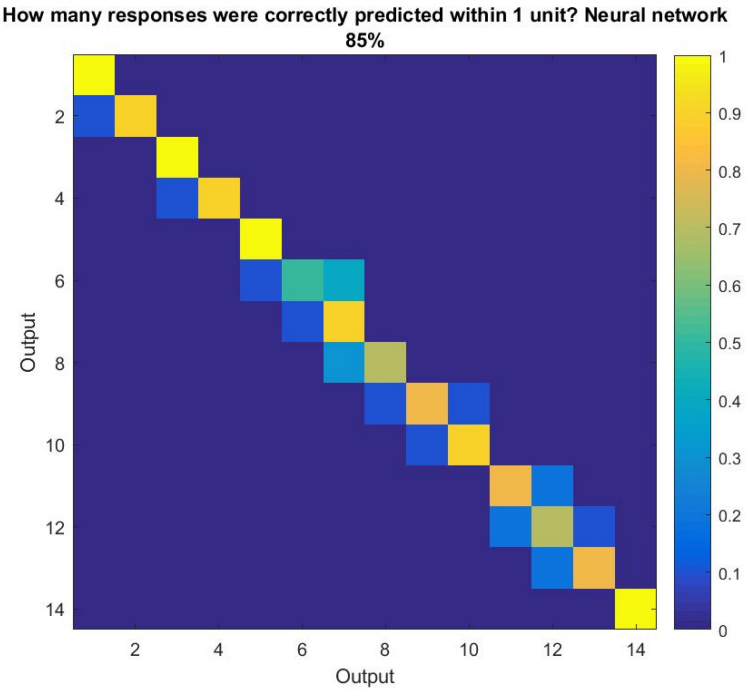


Рис. 4. Количество ответов, полученных с нейронной сети для классификации.

Таким образом, рассмотренное в работе решение задачи классификации в среде MATLAB с помощью двух произвольных моделей позволяет выяснить число групп классификации, на

которые можно разбить исходную матрицу входящего на станцию пассажиропотока. Оценка точности классификации для данной задачи показала, что Наивный байесовский классификатор и нейросети дают примерно одинаковый результат.

СПИСОК ЛИТЕРАТУРЫ

1. Antonova V.M., Kuznetsov N.A., Volkov D.O., Starostenko A. M. Math modeling of passenger traffic in the monorail transport system. 2016 10th International Conference on Application of Information and Communication Technologies. Стр. 90-94.
2. Н.А.Кузнецов, Ф.Ф. Пащенко, Н.Г. Рябых, Е.М.Захарова, И.К.Минашина. Алгоритмы оптимизации в задачах планирования на рельсовом транспорте. Информационные процессы, Том 14, номер 4, 2014, стр. 307–318
3. V.M. Antonova, N.A. Kuznetsov, A.M. Starostenko, D.O. Volkov. Automatic scheduling of monorail transport system. International conference Engineering & Telecommunications - En&T 2016. Стр. 7-9.

Solving the Classification Problem for Building Predicative Models of Passenger Traffic in the MATLAB Environment

V. M. Antonova; N. A. Kuznetsov; A. M. Starostenko; D. O. Volkov

This paper considers a classification problem solution and describes ways of predictive model building for passenger traffic entering the station which is performed on the grounds of a set of source data and by means of two known methods: a Naive Bayes classifier and Neural Networks.

KEYWORDS: passenger traffic, NaiveBayes classifier, Neural Network