

## ОБУЧЕНИЕ КОМПЬЮТЕРА РАСПОЗНАВАНИЮ ХАРАКТЕРИСТИК ЖЕЛЕЗНЫХ РУД

ЛЕОНОВ Р. Е.

*Рассмотрено обучение компьютера распознаванию участка месторождения железной руды, из которой был получен концентрат, по параметрам концентрата: крупности; содержанию железа; магнитной восприимчивости; намагниченности; коэрцитивной силе. Участки месторождения, из которых взята руда, условно обозначены как «классы». Классам присвоены условные номера. Для обучения и последующего распознавания класса на контрольной выборке данных использован метод ближайших соседей. Особенностью рассматриваемой задачи является очень малая обучающая выборка. По необходимости выборка была дополнительно уменьшена, так как из ее состава случайным отбором была сформирована контрольная выборка, в которую входили представители каждого класса. Общий окончательный объем обучающей выборки составил 46 значений, объем контрольной выборки – 13 значений. Количество различных классов – 13. Несмотря на ограниченный набор данных для обучения, результаты правильного распознавания (точность распознавания) составили более 46 %, что значительно превосходит случайное значение на независимой выборке данных. Исследовано влияние на точность распознавания количества «ближайших соседей». Показана необходимость нормирования данных и исследованы различные способы нормирования, в том числе и с помощью препроцессорных модулей, входящих в состав пакетов программы Python. Приведены документированная программа на языке Python, версия 3.6.0, и результаты распознавания. Решение задачи выполнено в IPython, Spyder.*

**Ключевые слова:** машинное обучение; железные руды; распознавание; программа в IPython.

В настоящее время все большее внимание уделяется использованию методов машинного обучения в различных прикладных областях. Большинство работ относятся к непроизводственным и гуманитарным областям деятельности. Опыт использования машинного обучения в промышленности небольшой. Не является исключением и горная промышленность. Между тем представляется, что и в сфере горного производства некоторые задачи, связанные с машинным обучением, в частности задачи распознавания производственных ситуаций, могут найти применение.

В предлагаемой статье на примере распознавания участка месторождения, из которого была взята руда для получения концентрата, показана возможность применения методов и средств машинного обучения. Несмотря на то что решалась частная задача, приводимая методика и программа, по мнению автора, могут найти более широкое применение. Программа может быть использована при решении любых задач, где по набору некоторого числа количественных признаков (факторов) изучаемый объект должен быть отнесен к одному из нескольких возможных классов.

Числовые данные для обучения и контроля взяты из анализа влияния класса крупности концентрата на его магнитные свойства [1]. Учитывались следующие параметры концентрата (факторы): класс крупности – пять фракций от 0,1–0 до 2,0–0 мм; содержание железа в концентрате, %; коэрцитивная сила, Э; намагниченность при напряженности магнитного поля (от 36 до 1450 Э, всего пять значений).

Все перечисленные величины измерялись на концентрате, полученном из руд различных участков Соколовского и Сарбайского месторождений (всего 13 участков). Задача состояла в проверке возможности обучения компьютера распознаванию пробы, из которой получен концентрат, по параметрам концентрата.

Для обучения компьютера была сформирована обучающая выборка, которая состояла из замеренных указанных ранее величин и обозначения соответствующего участка месторождения, из которого была взята проба руды. После обучения компьютеру была предъявлена контрольная выборка, которая состояла из проб концентрата с замеренными перечисленными параметрами (факторами), но без указания участка, с которого была взята проба руды, по которой получен контрольный экземпляр. Контрольная выборка в обучении компьютера не участвовала.

Надо отметить, что вследствие коммерциализации экономики получение представительных данных, аналогичных приведенным, чрезвычайно затруднено. В связи с этим первоначальная выборка для обучения и контроля была значительно меньше, чем обычно используется в машинном обучении. Выборка состояла из 59 строк параметров полученного концентрата и обозначения участка для каждой такой строки. Из этой выборки случайным образом была сформирована контрольная выборка из 13 строк параметров и соответствующего обозначения участка месторождения (класса руды). В итоге обучающая выборка состояла из оставшихся 46 строк параметров и соответствующего им класса, контрольная – из 13 строк параметров и соответствующего им класса. Обозначение класса в контрольной выборке использовано для проверки правильности распознавания компьютером.

Обучение и последующий контроль выполнены на языке Python (версия 3.6.0) в приложении Anaconda 3. Отладка и результаты получены в IPython, Spyder.

Для распознавания использован метод ближайших соседей. В качестве одного из параметров программы предусмотрен выбор числа «ближайших соседей». Программа распознавания класса выбирает обозначение класса, рассматривая близость предъявленных признаков ко всем возможным образцам, использованным при обучении. Окончательный выбор формируется по количеству наиболее близких образцов, установленных при настройке программы. В действительности в общем случае число «ближайших соседей», которое следует учитывать, заранее установить невозможно. Например, пусть предъявленный контрольный экземпляр очень близок к классу руды № 1. При этом программа настроена на двух ближайших соседей. Кроме класса № 1 ближайшими оказались два экземпляра класса № 4, хотя их параметры гораздо менее соответствуют контрольному экземпляру. В этом случае компьютер определит контрольный образец как принадлежащий классу № 4, и распознавание будет неудовлетворительным.

При больших обучающих выборках метод ближайших соседей работает хорошо, конечно, если признаки достаточно информативны и хорошо представляют классы. По-видимому, для производственных задач с их ограниченными наборами данных количество «ближайших соседей» должно быть невелико. Это подтверждают и приведенные далее результаты работы программы.

В методе ближайших соседей близость контрольного экземпляра к экземплярам обучающей выборки определяется евклидовым расстоянием (или любой другой метрикой, основанной на разности аналогичных признаков контрольного и

обучающего экземпляров). В этих условиях координаты должны быть приведены к соизмеримым единицам.

Действительно, если крупность контрольного и анализируемого обучающего образца составляет по порядку величин миллиметры, а намагниченность – сотни единиц, то в итоговую оценку близости контрольного и обучающего образца намагниченность внесет сотни единиц, а крупность – доли единицы. При этом независимо от действительной информативности признака влияние крупности на процесс распознавания ничтожно.

Чтобы избавиться от такого положения, все значения признаков должны быть нормированы – приведены к соизмеримым единицам. В рассматриваемой программе использованы несколько методов нормирования. Проверялась нормировка

$$X_n = (X_f - X_c) / s_x, \quad (1)$$

где  $X_n$  – нормированное значение фактора;  $X_f$  – фактическое значение фактора;  $s_x$  – стандартное отклонение фактора.

Возможность такого нормирования предусмотрена в стандартном пакете Scipy Python.

Лучшие результаты получены, однако, при предварительной нормировке признаков в виде

$$X_n = (X_f - X_{\min}) / (X_{\max} - X_{\min}), \quad (2)$$

где  $X_{\min}$  и  $X_{\max}$  – минимальное и максимальное значения соответствующего фактора. Такая нормировка приводит все факторы к диапазону от 0 до 1.

Известно [2], что большое значение имеет качественный состав признаков, определяющий класс. Признаки должны быть информативными и действительно существенно количественно различаться для разных классов.

В связи с этим выполнена еще одна серия обучения и контроля для решения этой же задачи с несколько другими признаками, характеризующими классы. Из факторов (признаков) были исключены данные по намагниченности для напряженности поля 90, 360, 730, 1270 Э и дополнительно в число признаков включены данные по магнитной восприимчивости для напряженности 36, 180, 550, 1000, 1450 Э.

Распознавание класса в соответствии с приведенной программой показало, что точность распознавания существенно зависит от способа нормирования признаков. В лучшем случае при нормировании признаков в соответствии с (1) точность составила как для первоначального набора признаков, так и для набора признаков с магнитной восприимчивостью 38,46 %.

При нормировании по выражению (2) точность составила 46,15 %. Это небольшие величины. Тем не менее это значительно больше, чем просто случайное угадывание класса руды. Можно констатировать, что даже при таком небольшом количестве обучающих данных распознавание возможно. Использование программы показало также, что эффективность распознавания сильно зависит от числа «близких соседей». При практическом использовании метода близких соседей целесообразно заранее сформировать независимую контрольную выборку и на ней определить для конкретной решаемой задачи способ нормализации и оптимальное число «близких соседей».

В результате исследования обучения компьютера распознаванию класса руды по конечному продукту, полученному из нее, можно констатировать следующее.

Несмотря на небольшой объем обучающей выборки, допустимо использование распознавания производственной ситуации по числу «ближайших соседей» в некоторых задачах производственной деятельности горных предприятий.

С появлением пакетов программ решения задач распознавания открывается перспектива их применения для решения производственных задач. Учитывая наличие готовых пакетов компьютерных программ, основное внимание, по-видимому, следует уделить правильной формулировке производственных задач, для которых могут быть использованы программы машинного обучения.

Далее рассмотрена программа, использованная автором для обучения и распознавания. Приведены вариант для набора признаков с магнитной восприимчивостью с двумя «ближайшими соседями» и результаты работы программы.

#### Программа

```
# -*- coding: utf-8 -*-
"""
Created on Tue Dec 5 21:25:44 2017
@author: Leonov_пользователь
Классификация по "ближайшим соседям"
"""

#Загрузка необходимых пакетов Numpy и Scipy
import numpy as np
import scipy as sp
from sklearn import preprocessing

#Чтение матрицы исходных нормированных данных
x1 = sp.genfromtxt('D:\Leonov\Наука\Классификация\SShi\SShiobn.txt')
X = sp.array(x1[:,0:13])
#Для каждого набора признаков выделение соответствующего класса
#Класс – это номер пробы
y = sp.array(x1[:,13])
#Эти операторы подключить при нормировке из стандартного пакета
#norm_X = preprocessing.normalize(X)
#stand_X = preprocessing.scale(norm_X)
from sklearn import neighbors,metrics
knn = neighbors.KNeighborsClassifier(n_neighbors=2)
#Исправить X на norm_X (stand_X)при использовании
#стандартной нормировки
knn.fit(X,y)
#Печать параметров обучения
print("knn = ",knn)
#Загружаем контрольную выборку
x2 = sp.genfromtxt('D:\Leonov\Наука\Классификация\SShi\SShikn.txt')
X2 = sp.array(x2[:,0:13])
y1 = sp.array(x2[:,13])
#Эти операторы подключить при нормировке из стандартного пакета
#norm_X2 = preprocessing.normalize(X2)
#stand_X2 = preprocessing.scale(norm_X2)
#Распознаем контрольную выборку
#Исправить X2 на norm_X2 (stand_X2)при использовании
#стандартной нормировки
predicted_1 = knn.predict(X2)
#Оцениваем точность классификации m
#Исправить X2 на norm_X2 (stand_X2)при использовании
#стандартной нормировки
m = knn.score(X2,y1)
print('m =',%.2f%(m*100))
#Оцениваем прогноз по каждому классу и повторно среднюю
#точность по всем классам
n = len(y1)
right = 0
for i in range(n):
    print('Предсказано ',predicted_1[i], 'фактически ',y1[i])
    if predicted_1[i]== y1[i]:
        right+=1
tochnost = right/len(y1)*100
print("Точность классификации = ",%.2f%(tochnost))
```

**Результаты работы программы**

```
runfile('F:/Python/Anaconda3/MyFile/untitled2.py', wdir='F:/Python/Anaconda3/MyFile')
knn = KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
    metric_params=None, n_jobs=1, n_neighbors=2, p=2,
    weights='uniform')
```

m = 46.15

Предсказано 1.0 фактически 1.0  
 Предсказано 2.0 фактически 2.0  
 Предсказано 3.0 фактически 3.0  
 Предсказано 4.0 фактически 4.0  
 Предсказано 3.0 фактически 6.0  
 Предсказано 1.0 фактически 8.0  
 Предсказано 10.0 фактически 10.0  
 Предсказано 8.0 фактически 11.0  
 Предсказано 21.0 фактически 21.0  
 Предсказано 11.0 фактически 31.0  
 Предсказано 6.0 фактически 41.0  
 Предсказано 2.0 фактически 61.0  
 Предсказано 1.0 фактически 131.0  
 Точность классификации = 46.15%

**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Дробченко Л. А. Исследование магнитных характеристик железных руд в целях обогащения: приложения к дис. ... канд. техн. наук. Фонды УГГУ, 1966. 40 с.
2. Коэльо Л. П., Ричарт В. Построение систем машинного обучения на языке Python. М.: ДМК Пресс, 2016. 302 с.

Поступила в редакцию 13 декабря 2017 года

**TRAINING A COMPUTER TO RECOGNISE THE FEATURES OF IRON ORE**

**Leonov R. E.** – The Ural State Mining University, Ekaterinburg, the Russian Federation. E-mail: lnprep2011@yandex.ru

Training a computer to recognize the section of deposit of iron ore, from which the concentrate has been obtained, is examined according to the parameters of the concentrate: fineness, iron content, magnetic susceptibility, magnetism, and coercitive force. The sections of deposit the ore has been got from, are nominally denoted as "classes". The classes are awarded with nominal numbers. For training and further recognition of a class at the control sampling of data, the nearest-neighbor method has been used. The peculiarity of the problem under consideration is a small training sample. By necessity the sampling was additionally reduced, because out of its composition, the control sampling was formed with random selection; the control sampling included the representatives of every class. Total and final volume of a training sample constituted 46 values, volume of control sampling – 13 values. The quantity of various classes – 13. Despite the limiting set of data for training, the results of the right recognition (recognition accuracy) constituted more than 46% which significantly exceeds random value of independent data sampling. The influence of "nearest-neighbors" quantity on the accuracy of recognition has been examined. The necessity of data regulation is shown and various methods of regulation are examined, with the help of preprocessor modules, among other, which are part of Python program pack. Documented program in the Python language is introduced, version 3.6.0, and the results of recognition. The problem solution is fulfilled in IPython, Spyder.

**Key words:** machine learning; iron ore; recognition; program in IPython.

**REFERENCES**

1. Drobchenko L. A. *Issledovanie magnitnykh kharakteristik zheleznykh rud v tseliakh obogasheniia: prilozheniia k dis. kand. tekhn. nauk* [Investigation of magnetic characteristics of iron ore with the purpose of concentration: enclosures to cand. of eng. sci. diss.]. UrSMU Funds, 1966. 40 p.
2. Koel'o L. P., Richart V. *Postroenie sistem mashinnogo obuchenii na iazyke Python* [Building the systems of machine learning in the Python language]. Moscow, DMK Press Publ., 2016. 302 p.