**Пакет Neurotree.**

Пакет предназначен для решения задач кластеризации на основе алгоритма растущего нейронного дерева (см. особенности пакета).

Пакет NeuroTree осуществляет кластеризацию с помощью алгоритма растущего нейронного дерева (РНД). (Более подробно с алгоритмом можно ознакомиться в отдельном файле).

**Запуск и настройки**

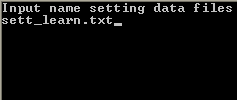
В начале работы пользователь вводит название конфигурационного файла, например, setting.txt. В файле в виде xml-тегов записаны унифицированным образом настройки:

* Схема работы алгоритма – существует два режима работы алгоритма: обучение и тестирование.
* Количество потребляемой памяти в mb.
* Размерность пространства входного множества.
* Способ вычисления нормы (Евклидова (L2) и L1)
* Желаемое количество кластеров.
* Количество итераций.
* Имя файла, где записано обучающее множество.
* Имя файла, куда будет сохраняться (загружаться) веса нейронного дерева.
* Имя файла, куда будет выведен результат работы (принадлежность входного примера к определенному кластеру)
* Имя файла, куда будет выведен список кластеров.

**Пример работы**.

**Обучение**.

Пользователь вводит название конфигурационного файла – sett\_learn.txt



Содержание файла sett\_learn.txt (с комментариями //):

<method>learning</method> //learning или testing. Обучение или тестирование.

<memory>1024</memory> // количество потребляемой памяти в мб

<dimension>5</dimension> // размерность входного пространства

<norma>Evkl</norma> //Выбор нормы. Evkl или L1.

<maxnumber>10</maxnumber> // желаемое количество кластеров

<iteration>100</iteration> //количество итераций

<filedatain>learning.bin</filedatain> //Входной файл с обучающим множеством.

<tree>tree.txt</tree> // Имя файла, куда будет сохраняться веса нейронного дерева

<cluster>NULL</cluster> // Имя файла, куда будет выведен список кластеров.

<result>NULL</result> // Имя файла, куда будет выведена принадлежность входного примера к определенному кластеру.

Выбраны настройки:

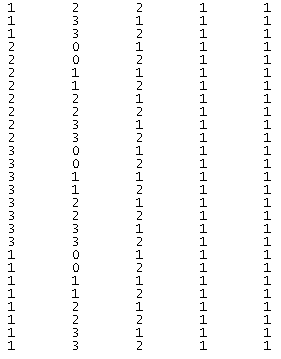
Вход:

Обучение, размерность 5, норма- Evkl, желаемое количество кластеров 10 кластеров, 100 итераций, обучающее множество в файле «learning.bin» (Обучающее множество должно быть записано в бинарном файле с расширением \*.bin, в каждом блоке памяти содержится значение double. Размер памяти, выделяемой для double должен совпадать для машины на которой построен данный бинарный файл и для машины на которой будет выполняться обучение).

Выход:

результат работы сохранится в файл «tree.txt»

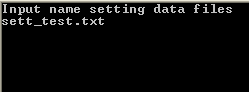
Содержание файла learning.bin:



В результат работы – веса нейронного дерева сохраняется в файл tree.txt, откуда в последствие будет загружен на этапе тестирования.

**Тестирование**.

Пользователь вводит название конфигурационного файла – sett\_test.txt



Содержание файла sett\_test.txt (с комментариями //):

<method>testing</method> //learning или testing. Обучение или тестирование.

<memory>1024</memory> // количество потребляемой памяти в мб

<dimension>NULL</dimension> // размерность входного пространства

<norma>NULL</norma> //Выбор нормы. Evkl или L1.

<maxnumber>NULL</maxnumber> // желаемое количество кластеров

<iteration>NULL</iteration> //количество итераций

<filedatain>testing.bin</filedatain> // Файл с обучающим множеством.

<tree>tree.txt</tree> // Имя файла, откуда будет загружаться веса нейронного дерева

<cluster>cluster.txt</cluster> // Имя файла, куда будет выведен список кластеров.

<result>result.txt</result> // Имя файла, куда будет выведена принадлежность входного примера к определенному кластеру.

Вход:

тестирование, тестируемое множество в файле «testing.bin», дерево находится в файле «tree.txt»

Выход

определяющие кластеры сохранятся cluster.txt

результат принадлежности входного примера к кластерам дерева выводится в файл result.txt