Инструкция по использованию программы LAGMA

Институт Космических Исследований РАН

2022-04-14

Содержание

1 Что такое LAGMA.

Программа LAGMA предназначена для осуществления классификации типов земных покровов по их спектрально-отражательным характеристикам с использованием одноименного локально-адаптивного подхода LAGMA. Программа представляет собой приложение для командной строки, управление ходом выполнения программы можно осуществлять через задание опций программы в командной строке. После запуска программа не требует вмешательства пользователя, процесс выполнения полностью определен заданными на старте в командной строке опциями.

2 Общее описание алгоритма LAGMA.

Для осуществления классификации LAGMA сначала делит всю область классификации на клетки равного размера, образуя регулярную сетку. Размер клетки в сетке можно определить с помощью опции -Step. В начале обработки LAGMA собирает данных о выборке в каждой клетке (сигнатуры) и сохраняет их в хранилище 1-ых сигнатур (опция -fsig1).

Затем для каждой клетки осуществляется агрегация выборки их соседних клеток. Агрегация происходит индивидуально для каждого класса. Программа находит наименьший радиус, при котором суммарно во всех клетках в этом радиусе лежит не менее определенного количества точек выборки для данного класса (пороговое значение задается опцией -MS). Расстояние между клетками понимается как минимальное из покоординатных расстояний, т.е. все клетки равноудаленные от данной клетки образуют квадрат с центром в данной клетке. При этом на радиус агрегации можно наложить ограничения сверху (опция -Rmax) и снизу (опция -Rmin), т.е. радиус агрегации всегда не меньше определенного значения (даже если при меньшем значении достигнуто пороговое значение) и не может быть больше определенного предела. Если не удалось найти такой радиус агрегации, при котором достигнуто пороговое значение на количество точек выборки данного класса (т.е. достигнуто ограничение сверху или обработано все изображение), то данный класс исключается из выборки в данной клетке. Результат агрегации записывается в хранилище 2-ых сигнатур (опция -fsig2). Программа также предоставляет возможности по выводу значений радиусов для каждой клетки и каждого класса в файлы растровых изображений с помощью опции -rout.

После агрегации программа в каждой клетке по локальной выборке создает локальный классификатор и с помощью него классифицирует каждый пиксел в этой клетке, относя его к одному из классов.

3 Ход выполнения.

Ход выполнения программы LAGMA можно разделить на несколько этапов:

1. 1-ый проход по выборке. На этом этапе происходит составление представления о локальной для каждой клетки выборке, сохранение их в промежуточные файлы.

2. 2-ой проход по выборке. На этом этапе происходит объединения обучающих выборок соседних клеток и также запись результатов в промежуточные файлы.

3. Классификация. На этом этапе строятся локальные классификаторы для каждой клетки, и по ним осуществляется классификация всех пикселов в клетке.

4. Постобработка. Подсчет и вывод различной вспомогательной информации о результатах классификации, выборке, использованных классификаторов.

Каждый из этапов может быть отключен (опция -Mode), но их порядок не может быть изменен. Конкретные действия, осуществляемые на каждом этапе, также зависят от опций, заданных пользователем.

4 Управление ходом выполнения программы.

Программа представляет собой приложение для командной строки, которое не требует пользовательского вмешательства во время выполнения, все настройки (опции) программы задаются в момент ее запуска. Опции задаются в виде аргументов командной строки в формате название_опции значение (через пробел, в значении не может быть пробелов, все пробелы должны быть выделены кавычками). Порядок опций при таком задании неважен. Повторение одной и той же опции приведет к использованию последнего значения из списка.

Также в списке опций есть пять обязательных позиционных параметра, значения которых обязательно должны идти в начале программы. Эти аргументы (по порядку) - файл обучающей выборки, файл признаков, хранилище 1-ых сигнатур, хранилище 2-ых сигнатур, файл результата классификации. В программе существует метод альтернативного задания этих пяти аргументов командной строки: они могут быть заданы с помощью опций -fsamp, -fchan, -fsig1, -fsig2, -fout, как в командной строке так и в конфигурационном файле. Вне зависимости от задания этих опций в командной строке или конфигурационном файле, первые пять опций все равно должны присутствовать в начале командной строки (возможно с фиктивными значениями).

Кроме пяти обязательных опций пользователь также должен задать число классов при классификации (опция -NC), причем в изображении с обучающей выборкой не должно быть пикселей со значениями больше заданного, это может привести к ошибке.

Опции могут быть заданы в файле конфигурации, описание формата файла приведено в опции -conf и в примерах. Значения, полученные из конфигурационного файла, имеют приоритет над значениями в командной строке, и именно они будут использованы в случае, если опция присутствует и там и там. Также программа позволяет вывести все заданные значения опций с помощью опции -optprt.

Также у каждой опции есть альтернативное длинное имя, более удобное для понимания. Длинное имя дано для каждой опции в общем списке опций в конце файла, и они могут быть использованы вместо коротких имен в файле конфигурации (но не в командной строке).

При возникновении любой ошибки в ходе выполнения программы будет выведено сообщение о возникшей проблеме и работа программы будет прекращена. В программе отсутствует возможность восстановления после ошибки, пользователь должен изменить значения поданных на вход опций и запустить программу повторно или связаться с разработчиками, если причину ошибки невозможно устранить изменением опций.

5 Входные данные.

На вход программе LAGMA подаются растровые изображения с положительными целочисленными значениями (рациональные числа необходимо преобразовать в целочисленные, например `[0,1]->[0,10000]` с 10000 градаций между 0 и 1). Для работы с изображениями используется библиотека GDAL, в связи с этим LAGMA поддерживает изображения в любом из форматов, приведенных на web-странице: `http://gdal.org/formats_list.html`. Для задания изображения в опциях программы достаточно указать имя файла, содержащего изображение в нужном формате.

Также LAGMA поддерживает свой собственный бинарный формат, изображения в нем состоят из двух файлов - заголовочного файла с разрешением ".ihdr" и бинарного файла с произвольным разрешением. Заголовочный файл является простым текстовым файлом следующего формата (после # указаны комментарии, которые должны отсутствовать в реальном заголовочном файле):

`ihdr_version=1.1` # обязательное поле

`cols=9` # число столбцов в изображении (ширина в пикселах)

`Projection=SIN` # нерабочее поле

`uly=3800000` # нерабочее поле

`ulx=-5700000` # нерабочее поле

`dy=230` # нерабочее поле

`dx=230` # нерабочее поле

`NFList=apri1.raw` # имя бинарного файла

`work=81` # число пикселей в файле

`bits=8` # размер пиксела файла в битах

`signed=0` # знаковость - поддерживаются только беззнаковые

Бинарный файл должен содержать последовательно все пиксели изображения, перечисленные сначала по строкам, затем по столбцам. В бинарном файле отсутствует заголовок, значения пикселов идут с начала файла. Для задания бинарного файла в данном формате в опциях программы нужно указать имя заголовочного файла.

Также LAGMA определяет формат стека изображений “.istk” - текстовый файл, в котором на каждой строке указано имя одного из файлов стека. Для опций программы, принимающих стек изображений, будет использовать все изображений из этого стека в том порядке, в котором они перечислены в файле.

В LAGMA есть обязательные для задания входные изображения - обучающая выборка и набор признаков. Обучающая выборка представляет из себя единственный растровый файл (первый аргумент командной строки или опция -fsamp). Набор признаков представляет из себя стек растровых изображений (второй аргумент командной строки или опция -fchan).

Кроме растровых файлов программа может принимать файлы в другом (обычно текстовом) формате. Такие данные не требуются для работы программы и используются в ряде опций. Описание данных и их формата представлено в описании соответствующих опций.

5.1 Тайловые изображения

LAGMA поддерживает возможность работы с тайловыми изображениями. Есть 2 варианта тайловых изображений:

1 Простое сшивание Предполагается, что все изображения тайлов одного размера, и лежат на регулярной сетке. Тайлы виртуально собираются в одно изображение, размер итогового изображения – сумма размеров всех тайлов. Заголовочный файл с расширением ".tile"в формате:

`ignore_missing_tiles` - заполнять пропущенные тайлы определенным значением или выходить с ошибкой

`default_vals` - значения для заполнения пропущенных тайлов

`source_images` - перечисление всех тайлов (начиная со следующей строки)

Y,X=изображение - формат задания изображений тайлов

...

`source_images_end` - окончание списка тайлов

2 Перепроецирование. Позволяет указать целевое изображение (размер, координаты, проекцию), перепроецирует тайлы в это целевое изображение. Перепроецируется по методу ближайшего соседа, для каждого пиксела целевого изображения находится ближайший к нему пиксел из тайла на входе. Если несколько тайлов попадают на 1 пиксел, то можно считать, что выбирается значение случайного тайла. Заголовочный файл с расширением ".mosaic"в формате:

`ignore_missing_tiles` - заполнять пропущенные тайлы определенным значением или выходить с ошибкой

`default_vals` - значения для заполнения пропущенных тайлов

`target_projection_WKT` - проекция целевого изображения в формате WKT, например PROJCS["Albers Conical Equal Area GEOGCS["WGS 84 DATUM["WGS_1984 SPHEROID["WGS 84 6378137,298.257223563,AUTHORITY["EPSG "7030"]],AUTHORITY["EPSG "6326"]],PRIMEM["Greenw Conic_Equal_Area"],PARAMETER["standard_parallel_1 50],PARAMETER["standard_parallel_- 2 70],PARAMETER["latitude_of_center 56],PARAMETER["longitude_of_center 100],PARAMETER["fals easting 0],PARAMETER["false_northing 0],UNIT["metre 1,AUTHORITY["EPSG "9001"]]]

`target_projection_geog` - проекция целевого изображения из geog (например, WGS84)

`size_x` - размер целевого изображения, число пикселов по X

`size_y` - размер целевого изображения, число пикселов по У

`min_x` - координата первого пиксела по X

`min_y` - координата первого пиксела по У

`step_x` - размер пиксела по X

`step_y` - размер пиксела по У

`source_images` - перечисление всех тайлов (начиная со следующей строки)

Y,X=изображение - формат задания изображений тайлов

...

`source_images_end` - окончание списка тайлов

6 Выходные данные.

Выходные данные, как и входные, представлены преимущественно растровыми изображениями и стеками изображений. Все, сказанное о формате растровых данных и опциях для них остается верным и для выходных данных.

Существенным отличием выходных файлов от входных является возможность их перезаписи. Если файлы выходных изображений, заданные пользователем, уже существуют, то программа может использовать уже существующие файлы, не создавая новые, записывая новые значения пикселов поверх уже имеющихся Уже существующие файлы могут быть удалены и созданы заново при задании опции ovrw(см. далее).

При создании новых файлов LAGMA будет создавать их в формате, указанном в опции -drv (по умолчанию - ERDAS Imagine .img).

Каждый растровый файл на входе и на выходе имеет заданный формат пиксела (число бит, целое или рациональное), его описание приведено в соответствующей каждому файлу опции. LAGMA также позволяет менять формат пиксела растрового изображения и формат изображения. Для этого после имени файла (одного файла или стека) нужно написать специальный разделитель «???» и параметры файла (в виде имя_параметры=значение_параметра;имя=значение, разделитель - ';'). Возможные параметры файлов: drv - имя драйвера для создания файла, возможные значения те же, что и у опции -drv, а также вариант RAW - создаст бинарный файл LAGMA с заголовочным файлом (".ihdr"). bits - число бит в пикселе. signed - знаковый (не равно нулю) и беззнаковый (равно нулю) формат пиксела float - целочисленное (равно нулю) и дробно-рациональное (не равно нулю) значение пиксела ovrw - при значении не равном нулю сотрет файл если он существовал ранее и создаст новый файл, заполненный нулевыми значениями.

Пример использования: filename???drv=RAW;float=1;bits=32;signed=1 (бинарный файл, с 32 битным рациональным числом). Рекомендуется задавать три опции типа вместе, иначе будут использованы их значения по умолчанию, что может привести к некорректным типам (например, беззнаковое рациональное 8 битное число).

Для файлов на входе программы изменение формата файла не приводит к изменениям в файле или программе. Для файла на выходе в случае, если заданный формат пиксела отличается от формата пиксела изображения, то изображение будет удалено и создано заново. Все вышеуказанное верно и для стеков изображений (применяется к каждому изображению в стеке).

Единственным обязательным типом выходных данных являются результат классификации. Результат классификации представляет из себя единственный растровый файл, его имя задается пятым аргументом при запуске программы или опцией -fout.

7 Область обработки, проекции, разрешение.

В LAGMA существует два разных разрешения - разрешение уровня пикселов и разрешение уровня клеток. Все изображения в LAGMA обладают одни или другим разрешением (прописано в описании опции для соответствующего изображения).

Все изображения считаются совпадающими по левому верхнему углу, то есть левый верхний угол всех изображений вне зависимости от разрешения находится в одном и том же месте. При этом информация о проекциях или разрешениях, которая может быть прописана в растровых файлах, игнорируется, также LAGMA не записывает эту информацию и в выходные файлы.

Число пикселей по горизонтали и вертикали в изображениях с попиксельным разрешением определяется по первому изображению из списка признаков, данного на входе. Размеры изображений с поклеточным разрешением определяется по размеру сетки, который в свою очередь определяется по числу пикселов в клетке (опция -Step) и по размеру изображения обучающей выборки на входе. Альтернативно можно определить число клеток в сетке по горизонтали и вертикали (опции -SXsize, -SYsize), что также определяет размер клетки (как отношение размера изображения к -SXsize). Если какое-то из изображений, заданных пользователем, меньше соответствующих размеров, то программа может выдать ошибку об обращении за пределы изображения и прекратить выполнение. Новые изображения создаются всегда с правильным разрешением.

Программа позволяет ограничить область для обработки, для чего существует несколько опций. Группа опций -RXmin, -RXmax, -RYmin, -RYmax ограничивает область классификации по координатам сверху и снизу, пиксели за пределом этой области не будут классифицированы, их значения не будут изменены (при этом данные о выборке и агрегированная выборка за пределами этой области будут собраны). Альтернативно можно указать более сложную структуру области для классификации с помощью опций -AOIv, -AOIr через растровые или векторные файлы изображений.

8 Типы классификаторов.

В LAGMA на данный момент существует 5 типов классификаторов: метод максимального правдоподобия, метод k ближайших соседей, случайные леса, метод опорных векторов и нейронные сети. Тип классификатора в программе меняется с помощью опции -clas. Стоит отметить, что в программе существуют серьезные различия в том, как работают разные классификаторы и какие опции можно использовать для каждого из них.

8.1 Метод максимального правдоподобия.

Метод максимального правдоподобия отличается высокой скоростью вычисления классификатора. Кроме этого данный метод может производить быструю агрегацию по соседним клеткам, суммируя локальные классификаторы соседних клеток (вместо суммирования выборки). Использование данного метода дает существенное ускорение но требует создания временных файлов на диске для сохранения в них классификатора (опция -clstr,всегда включена для метода максимального правдоподобия), что может замедлить программу в отдельных случаях.

Для метода максимального правдоподобия LAGMA может вывести вспомогательные данные о процессе оптимизации: значение наилучшей функции правдоподобия в виде растрового файла (-bestLK) и значения функции правдоподобия для всех классов в виде стека изображений (-LKout). Также программа может вывести данные об обратимости матрицы ковариации в виде растрового изображения (-invout) (если матрица необратима, то функция правдоподобия не может быть построена для класса, он не может быть результатом классификации).

8.2 Метод случайных лесов.

Метод случайных лесов существенно медленнее метода максимального правдоподобия, но при этом он не делает никаких предположений о распределении признаков и в большинстве случаев дает более высокую точность классификации. В программе используются две разные имлементации случайных лесов - из библиотек alglib и opencv2 (выбор происходит через опцию -clas), часть настроек для метода случайных лесов доступна либо для одной либо для другой библиотеки. В целом версия opencv2 предпочтительнее в связи с меньшими затратами памяти и большей производительностью. ПРЕДУПРЕЖДЕНИЕ - метод случайных лесов для части приложений вызывает сильную фрагментацию памяти и вылет программы, проблема пока не решена и в подобных случаях рекомендуется разделить область классификации на несколько частей и выполнить классификацию для них по отдельности.

Метод случайных лесов предоставляет пользователю набор настроек - число деревьев (-treenum), число переменных в узле дерева (-rfrvars), часть выборки для обучения дерева (-setpart), глубина дерева (-trdepth), число элементов в конечнои узле дерева (-leaflim), точность

поиска разбиения (-dfnslc). Также программа может вывести процент деревьев, проголосовавших за каждый класс (-cprob, -caprob) и ошибку классификации на обучающей выборке (-crele, -corele). Стоит отметить, что если в выборке число элементов каждого класса неодинаково, то случайные леса имеют тенденцию к предпочтению более многочисленных классов при классификации. Для того, чтобы исправить такое поведение, выборку можно проредить (-setequal).

С подробным описанием метода и его опций (в варианте библиотеки opencv) можно ознакомиться по ссылке `https://docs.opencv.org/2.4/modules/ml/doc/random_trees.html`

8.3 K ближайших соседей.

Метод K ближайших соседей медленнее метода максимального правдоподобия (для каждой клетки полностью загружает обучающую выборку), но быстрее любых других непараметрических методов. Для использования данного классификатора необходимо задать опцию -clas = KNN. Перед выполнением классификации метод автоматически нормализует входные данные посредством линейного преобразования до распределения со средним ноль и стандартным отклонением равным 1. Метод K ближайших соседей требует задания только одного параметра - числа соседей (опция -knn).

С подробным описанием метода и его опций можно ознакомиться по ссылке `https://docs.opencv.org/2.4/modules/ml/doc/k_nearest_neighbors.html`

8.4 Метод опорных векторов.

В теории данный метод отличается высокой точность классификации, а также длительным временем обучения. На практике я так и не смог дождаться, пока метод обучится даже для одной клетки при нелинейном ядре (линейной ядро работает быстро). Метод делит пространство признаков кривыми, задаваемыми ядром (опция -svmk). Также метод опорных векторов позволяет задавать тип классификатора - как именно вводится ограничение при нечетком разделении классов - -svmt. При обучении метод ищет оптимальные значения параметров ядра и параметров разделения классов (C или NU). При оптимизации точность классификатора оценивается с помощью кросс-валидации (10 частей). Максимальное количество итераций этой оптимизации задается опцией -it, терминальное значение ошибки задается опцией -eps.

С подробным описанием метода и его опций можно ознакомиться по ссылке `https://docs.opencv.org/2.4/modules/ml/doc/support_vector_machines.html`

8.5 Нейронные сети.

Классическая реализация классификации методом нейронных сетей из библиотеки opencv, на основе многослойных перцептронов. Вид функции активации каждого нейрона устанавливается опцией -nnf. Количество скрытых слоев и нейронов в слое задается опцией -nnl. Нейронная сеть всегда на входе иммет число нейронов, равное числу каналов, а на выходе число нейронов равно числу классов. Достаточно чувствителен к процессу настройки, веса в каждом нейроне определяются итеративно с помощью метода обратного распространения ошибки. Число итераций метода определяется опцией -it, терминальное значение ошибки задается опцией -eps.

С подробным описанием метода и его опций можно ознакомиться по ссылке `https://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html`

9 Возможности LAGMA.

Данная глава посвящена описанию различных вспомогательных возможностей LAGMA по настройке классификации, выводу данных и.т.д., которые еще не были описаны в прошлых главах.

9.1 Настройки производительности и стабильности

Загрузка данных из файлов может потребовать большого количества времени, в связи с чем LAGMA старается загружать изображения большими частями и хранить их в оперативной памяти. Но часто объем имеющихся изображений значительно превосходит количество оперативной памяти, в связи с чем LAGMA вынуждена регулярно выгружать из памяти ненужные части изображений и загружать новые. Ввиду того, что LAGMA регулярно обращается к одним и тем же частям изображения, то это может привести к замедлению работы программы. Ввиду критичности этого аспекта LAGMA позволяет настраивать ее механизм работы с памятью.

Опция -ML определяет всю доступную LAGMA память для загрузки изображений. -BS определяет размер блока изображения, который загружается за одни раз в память (минимальную единицу хранения). Если лимит памяти настолько мал, что LAGMA не может загрузить даже по одному блоку каждого изображения, то это может привести к замедлению работы программы. В этом случае нужно увеличить значение -ML или уменьшить значение -BS. Для 32-разрядной версии программы существует ограничение на лимит памяти, составляющее порядка 1,5 Гб.

Программа LAGMA поддерживает многопоточность на всех этапах работы, кроме постобработки. Ввиду легкости распараллеливания задачи классификации, прирост производительности может быть почти кратен числу потоков программы. Число потоков можно установить по опции -MTnum. Рекомендуется использовать значение равное числу ядер системы. Стоит отметить, что потребление памяти в многопоточном режиме также может существенно возрасти, при этом если объем имеющейся памяти недостаточен даже для обработки одного пиксела (т.е. для загрузки блоков всех изображений для этого пиксела), то многопоточный режим не может быть выполнен, программа завершится ошибкой.

Дополнительно LAGMA позволяет ускорить работу за счет кэширования данных о выборке и классификаторе в предыдущем пикселе (клетке) в режиме классификации (2-ом проходе) при активации опций -S2cache, -datacache, .

Кроме этого в программе можно выставить проверку типов пикселов в файлах изображений на соответствие их внутренним представлениям в программе (не происходит ли конвертации рациональных в целочисленные и наоборот) с помощью опции -typechk. Также можно вывести информацию о времени выполнения каждого этапа программы в файл с помощью опции -Tfile.

9.2 Постобработка.

Программа предоставляет возможности вывода ряда данных о классах и их локальных выборках. В частности LAGMA может вывести в растровые файлы данные о числе элементов выборки в каждой клетке (опция -numout), средних значениях каждого признака в каждой клетке (опция -avgout) и корреляциях между каждой парой признаков в каждой клетке (опция -covout), причем как для всех классов так и только для одного (опция -clsout). Вывод может

быть осуществлен как для исходной выборки, так и для выборки после агрегации (опция -pstsig). Кроме этого программа может вывести данные о расстоянии между классами в каждой клетке с помощью опций -sepout, -sepdist, -sepch

9.3 Хранение выборки

LAGMA хранит информацию о выборке в ряде промежуточных файлов, их расположение и формат может быть определении пользователем. Данные для неагрегированной и агрегированной выборки хранятся отдельно и обычно называются первые и вторые сигнатуры соответственно. Место хранения данных о выборке можно задать с помощью опций -fsig1, -fsig2 (3-ий и 4-ый аргумент командной строки). Опции -S1create, -S2create устанавливают, нужно ли перезаписать имеющиеся файлы выборки, перезапись осуществляется только при их изменении (т.е. на 1-ом этапе для 1-ых сигнатур, на 2-ом для вторых). Опция -SBig меняет конфигурацию файлов, так, что она может состоять или из одного большого файла или из множества малых. Множество малых файлов может ускорить работы на 2-ом этапе, но для остальных этапов одни большой файл обычно предпочтительнее. Опции -sigzero, -sigzeroX, -sigzeroY, -sigzerocl, -sigzeroch позволяют обнулить определенную часть сигнатур для определенных классов и каналов.

Также LAGMA поддерживает возможность прореживания выборки с помощью опции -smplrndmax. Уменьшение объема выборки позволяет ускорить построение непараметрических классификаторов и уменьшить требуемый для них объем оперативной памяти, но при этом может уменьшить точность классификатора.

LAGMA может создать базу данных для промежуточного хранения обучающей выборки с помощью опции -sDBf. База данных может требовать значительный объем места на диске для хранения промежуточной обучающей выборки, но при этом позволяет значительно ускорить сбор обучающей выборки для напараметрических классификаторов.

9.4 Априорные вероятности

Также LAGMA может принять на вход значения априорных вероятностей классов (-aap), представляющие из себя стек изображений с разрешением на уровне пикселов, каждое изображение дает априорную вероятность появления соответствующего класса (номер изображения в стеке) в пикселе. Отсчет номеров в стеке начинается с нуля, поэтому необходимо задать априорные вероятности для фиктивного нулевого класса (не будут использованы при классификации). Значение априорной вероятности в пикселе для класса A позволит модифицировать функцию правдоподобия для данного класса, умножив ее на коэффициент exp(A*coef), где coef задается с помощью опции -aapmult.

Также может быть задан аналогичный стек с разрешением сетки (по числу клеток на классифицируемой области), используемый при агрегации выборки соседних клеток (опция -apri). Эти априорные вероятности позволяют пропустить процесс агрегации классов для тех клеток, где априорная вероятность равна нулю. Кроме этого программа предоставляет возможности по построению априорных вероятностей с разрешением сетки по априорным вероятностям с попиксельным разрешением (-apricrt). Дополнительно априорных вероятностей с клеточным разрешением можно задать размер окна (-awin), в котором должны присутствовать априорные вероятности, чтобы в клетке произошла агрегация выборки для данного класса (даже если там нет выборки этого класса).

10 Регрессионный анализ.

LAGMA включает возможность проведения регрессионного анализа в режиме постобработки. Есть 2 варианта проведения регрессионного анализа: на основе линейных регрессий и с помощью случайных лесов. Отличия 2 вариантов связаны с невозможностью сохранения случайных лесов на диск. Далее перечислены общие аспекты построения регрессий в LAGMA. Регрессии всегда строятся между наборами входных каналов. Индексы каналов указываются в файле, задаваемом опцией -regch. Файл должен иметь следующий формат:

"имя_регрессии1";(канал1,канал2,зависимый_канал)

"имя_регрессии2";(канал2,канал3,зависимый_канал)

В файле может быть указано несколько (именованных) регрессий для разных комбинаций каналов, каналы задаются по их номеру в стеке каналов (начиная с 0). Имя регрессии будет использоваться в файле вывода. Линейная регрессия поддерживает только один независимый признак, поэтому для нее нужно указать только 2 канала. Регрессия может быть построена только для определенных классов, используемые комбинации классов считываются из файла -regcl. Файл должен иметь следующий формат:

"имя_классов1";(класс1,класс3)

"имя_классов2";(канал2,класс4)

В соответствии с содержимым этого файла будет построена одна регрессия для каждой комбинации каналов. Если один класс попадает в несколько групп классов, то при выборе регрессии для него выбирается регрессия с наибольшим значением корреляции. Стоит отметить, что оба типа регрессий строится с использованием выборки и радиусов агрегации, собранных на 1 и 2 этапе работы LAGMA.

Для линейных регрессий их применение контролируется опцией -regcalc. Коэффициенты линейных регрессий будут сохранены в папке, указанной опцией -regcalc. Файлы будут иметь формат вида A_suffix.img, B_suffix.img, R2_suffix.img, задающих линейную часть, постоянную часть и корреляцию регрессии, suffix определяется именами групп каналов и классов из файлов -regch и -regcl. После построения регрессий программа восстановит значения зависимого канала на основе значений независимого канала и регрессий и сохранит их в файл, задаваемый опцией -regprod. Построение регрессий и восстановление значений канала может проводиться на основе разных наборов данных, для этого необходимо запустить LAGMA 2 раза, в первый раз создав регрессии (-regcalc = 1), а во второй раз используя их для другого набора данных (-regcalc = 0).

Для случайных лесов отсутствует возможность сохранения регрессий на диск (вес регрессий может достигать сотен Гигабайт), поэтому процесс их построения и восстановления по ним данных изменен. Но регрессии на основе случайных лесов также строятся на основе выборки, собранной в области, определенной на втором этапе построения сигнатур, с использованием классов и каналов, заданных опциями -regcl и -regch (можно задавать несколько независимых каналов). Для включения данного режима необходимо задать файл информации о регрессиях

(опция -rfregfile), этот файл описывает информацию о том, как программа должна восстанавливать значения зависимого канала на основе построенных регрессий. Формат файла имеет вид:

файл_каналов1.istk

файл_выборки1

файл_восст_значений1

файл_каналов2.istk

файл_выборки2

файл_восст_значений2

...

Предполагается, что размер всех изображений, число каналов и идентификаторы классов идентичны значениям из опорной выборки. После построения регрессии программа для каждого пиксела из новой выборки подберет оптимальную для него регрессию, оценит значение зависимого канала на основе нового файла каналов и сохранит результат в файл восстановленных значений.

Дополнительно регрессии на основе случайных лесов позволяют задавать значение зависимого канала, которое будет игнорироваться при построении регрессии (опция -rfregignore). Также случайные леса сохраняют информацию о среднеквадратическом отклонении, рассчитанном на основе out-of-bag оценок, и вычисленных на его основе корреляциях (опция -rfreginfo).

11 Режим объектно-ориентированной обработки.

LAGMA предоставляет возможности объектно-ориентированной классификации. В этом режиме вместо файла каналов (опция -fchan, или 2-ая позиция в командной строке) необходимо задать файл в векторном формате. Для этого файла для каждого объекта должны быть указаны значения признаков, а также значения класса(ов). Имена полей, значения которых интерпретируются как классы или признаки, задаются с помощью файла (опция -vecinfo), имеющего следующий формат:

class поле_класса1

class поле_класса2

channel поле_канала1

channel поле_канала2

channel поле_канала3

Имена полей классов и каналов должны в точности совпадать с именами полей в векторном файле. Опция (опция -vecinfo) также является тригером для включения режима векторной обработки. Порядок каналов определяется порядком их появления в файле. При наличии нескольких полей классов один и тот же объект может появиться в выборке несколько раз для разных классов (аналогичная возможность доступна и в растровой обработке при использовании файла .istk в качестве выборки).

Для работы метода в качестве выборки нужно задать "эталонный"растр. Программа для каждого объекта определит его центроид и соответствующий ему пиксел эталонного растра. Для эталонного растра будет построена ругулярная сетка и каждый объект будет причислен к одной из клеток этой сетки. Дальнейшая процедура построения сигнатур и классификации выполняется аналогично растровой обработке, но обрабатываются не пикселы в клетке, а объекты в клетке. Результаты классификации сохраняются в текстовый файл в виде:

идентификатор_объекта результат_классификации

Расчет центроидов и привязка их к растру может быть трудозатратной операцией, поэтому программа предоставляет возможность их предварительно расчета на этапе предобработки (-Mode 0) и сохранения их в файл, задаваемый опцией -vecindex

12 Список опций LAGMA.

Описание дается в виде: ключ опции для командной строки (длинное имя опции) - описание. Если опция не задана, то будет использовано значение по умолчанию.

12.1 Необходимые

-fsamp (sampling_header_filename) - задает имя растрового файла, содержащего обучающую выборку. Разрешение на уровне пикселов, формат пиксела - беззнаковое целое с разрядностью не более 16 бит. Открывается только для чтения. Значение пиксела означает известный номер класса в данном пикселе, нулевое значение означает, что класс в данном пикселе неизвестен. Нет значения по умолчанию.

-fchan (channels_stack_filename) - задает имя стека растровых файлов, содержащих значения признаков. Разрешение на уровне пикселов, формат пиксела - беззнаковое целое с разрядностью не более 64 бит. Открывается только для чтения. Нулевые значения означают пропуски в данных. Значения по умолчанию нет.

-fsig1 (signature_1_stack_filename) - имя текстового файла, который задает структуру хранения первых сигнатур. На трех первых строчках файла задается базовое имя для трех стеков файлов, в которых хранятся характеристики классов (обычно называются num, sum, cov, тогда стеки будут выглядеть как num.istk, num-1.ihdr и.т.д.). Так как число файлов большое, то рекомендуется записывать их в отдельную директорию (т.е. sig1/num и.т.д.). В зависимости от опции -S может потребоваться добавление 4 стека (обычно обозначется ncov). При использовании опции -SBig вместо текстового файла просто задается имя растрового файла (может быть создан программой, в зависимости от опции -S1create). Открывается для записи на первом проходе и только для чтения далее. Значения по умолчанию нет.

-fsig2 (signature_2_stack_filename) - имя текстового файла, который задает структуру хранения вторых сигнатур. Аналогичен опции -fsig1 все описанное там верно и для этой опции. Стоит отметить, что структура файлов для двух хранилищ аналогична, поэтому рекомендуется по разному называть файлы в опциях -fsig1 и -fsig2, опционально размещать их в разных директориях. Открывается для записи на втором проходе и только для чтения далее. Значения по умолчанию нет.

-fout (output_filename) - имя растрового файла, в который будет записаны результаты классификации. Разрешение на уровне пикселов, формат пиксела - беззнаковое целое с разрядностью не более 64 бит. Открыт для записи на этапе классификации. Значения по умолчанию нет.

12.2 Особые

-conf (config_file) - задает имя config файла, из которого также будут прочитаны опции программы. В этом файле опции должны быть заданы в формате ИМЯ=ЗНАЧЕНИЕ, в качестве имени может выступать как ключ, так и длинное имя опции. Пробелы между знаком равно и именем (значением) также учитываются. Опции из этого файла имеют приоритет над опциями из командной строки (т.е. если одна и та же опция встречается в обоих источниках, то будет использоваться значение из config файла). По умолчанию - выключен.

-optprt (print_options) - указывает, нужно ли печатать текущие значения опций. Если опция равна пустой строке - не печатает ничего. Если значение отлично от пустой строки, то выведет список опций в том же виде, что и в config-файле. Опции будут записаны в файл вида config_-текущая дата.txt в текущей активной директории. Значение опции “key” указывает на то, что при выводе в качестве имен переменных используются ключи командной строки. Значение опции name, означает, что используются длинные имена переменных (которые тоже можно использовать в config-файле) По умолчанию - не выводит список значений опций.

12.3 Общие

-apri (crude_probabilities_stack_filename) - имя стэка, содержащего данные о файлах грубых априорных вероятностей для 2-го прохода. Разрешение на уровне клеток, формат пиксела - беззнаковое целое с разрядностью 32 бит. Открыт всегда только для чтения за исключением случая использования опции -apricrt. В случае если он отсутствует, второй проход по сигнатурам будет осуществляться для всех клеток. По умолчанию не указан.

-aapmult (likelihood_scale_multiplier) - задает коэффициент в формуле, изменяющей значение функции правдоподобия для класса в соответствии с априорными вероятностями. В этой формуле фукнция правдоподобия домножается на exp(A*aapmult), где A - априрорная вероятность, а aapmult - значение данной опции. Значение по умолчанию - 0.04.

-aap (probabilities_stack_filename) - имя стэка, содержащего данные о файлах априорных вероятностей для этапа классификации. Разрешение на уровне пикселов, формат пиксела - беззнаковое целое с разрядностью 32 бит. Открыт всегда только для чтения. В случае если он отсутствует, классификация будет осуществляться для всех пикселей. По умолчанию не указан.

-apricrt (create_apri_probabilities_with_name) - указывает программе, что точные файлы вероятностей должны быть преобразованы в грубые. Значение после ключа указывает на общий вид названия файлов грубых вероятностей (вида значение_ключа_1, значение_ключа_2 и.т.д.). Файл стека грубых вероятностей будет перезаписан и заполнен именами файлов указанного выше вида. По умолчанию - не указан.

-RXmin (minimum_raster_X) -RXmax (maximum_raster_X) -RYmin (minimum_raster_Y) -RYmax (maximum_raster_Y) - описывают размер области, на которой будет происходить классификация (в пикселях, минимальные и максимальные точки по горизонтали и вертикали). Отсчет пикселей начинается с нуля. По умолчанию стоит обработка всех пикселей (от первого и до последнго пиксела, по вертикали и горизонтали) (значение -1)

-S (signature_type) - указывает тип используемых сигнатур. Два возможных значения: simple - обычный подсчет сигнатур (среднее, ковариации), в случае, если в пикселе один из признаков = 0 - игнорируем этот пиксель. Второй вариант - by_characteristics. При нем каждый признак учитывается отдельно, точка учитывается, если хотя бы один признак ненулевой (но только для этого признака). Для классификации в пикселе используются матрица ковариаций и вектор средних с выкинутыми столбцами/строками нулевых признаков. По умолчанию включен обычный вариант сигнатуры (simple).

-SBig (use_one_signature_file) - определяет структуру файлов для хранения сигнатур. При значении отличном от нуля для храниения сигнатур будет использоваться один файл, в который последовательно записаны данные о сигнатурах в каждом классе и каждой клетке. При этом в опциях -fsig1 и -fsig2 вместо специфических файлов формата должен быть указан один растровый файл для хранения сигнатур. Если значени равно нулю, то будет использоваться структура, в соответствии с которой каждй класс и каждый параметр сигнатур хранится в своем изображении. По умолчанию значение равно нулю.

-S2cache (signature_2_window_caching) - ускорение для второго прохода по выборке. Если включено, то при вычислении окна для текущей клетки будет задействованы значения окон для предыдущей клетки, то есть программа лишь только смещает окно агрегации, используя значения клеток в нескольких крайних столбцах, а не пересчитывая все клетки в окне. Может существенно ускорить второй проход по сигнатурам для большого числа классов и редких выборок (большой размер окна). По умолчанию выключено.

-AOIv (AOI_vector_filename) - векторный файл, описывающий область, для которой будет происходить классификация. Формат файла - попарные значения x y для вершин многоугольника данной области, последняя вершина должна совпадать с первой, отделены пробелами и знаками табуляции, десятичный разделитель - точка. Если указан растровый файл AOI (-AOIr) - то будет использоваться он, а не векторный файл. По умолчанию - не указан.

-AOIr (AOI_raster_filename) - header для растрового файла, описывающего область, для которой будет происходить классификация. Разрешение на уровне пикселов, формат пиксела - беззнаковое целое с разрядностью не более 32 бит. Открыт всегда только для чтения. Нулевые значения в файле указывают на область, где не будет осуществляться классификация. Если указан векторный файл с AOI (-AOIv), то будет использован именно растровый файл, векторный файл будет проигнорирован. По умолчанию не указан.

-Mode (active_modes) - указывает, какие именно части программы LAGMA нужно выполнить. Цифра 1 означает первый проход, 2 - второй проход по сигнатурам, 3 - классификацию, 4 - постобработку изображения. Проверяется только наличие конкретных цифр, количество и порядок игнорируется, как и все остальные символы. По умолчанию - выполняется все этапы классификации ("123").

-vecinfo (vector_info_file) - Файл для сохранения восстановленных на основе регрессий значений в одном из каналов. Пустое значение означает, что восстановление не будет проводиться и файл не будет создан.

-vecindex (vector_grid_index_file) - Файл соотношения между объектами векторного файла и пикселами эталонного растра. Пустая строка означает, что такой файл не будет создан.

12.4 Классификация

-MS (MINSAMPLING) - минимальное число точек с известным значением класса в клетке. На втором проходе по выборке окно будет увеличиваться до тех пор, пока число элементов класса в агрегированной выборке не превысит это значение или не достигнут некоторый предельный размер окна (-Rmax). По-умолчанию - 10000

-MS2 (MINSAMPLING2) - используется только при учете точек с частично нулевыми признаками (-S = by_characteristics). Дополнительное ограничение на минимальное число ненулевых точек в окне, если для конкретного признака число этих точек не превысит указанного значения, то данный признак не будет использоваться при классификации в данной точке. По-умолчанию - 10000.

-Rmin (MINR) - минимальный размер окна при втором проходе по выборке. Второй проход всегда будет агрегировать информацию из всех точек, удаленных не далее чем на указанное число от данной, вне зависимости от того, превышено ли минимальное число точек (-MS). По-умолчанию - 3.

-Rmax (MAXR) - максимально возможный размер окна при подсчете вторых сигнатур. Окно не будет увеличиваться дальше, даже если число учтенных в данной клетке точек меньше, чем минимальное (-MS). По-умолчанию - 200.

-awin (probabilities_window_size) - величина окна, с которого агрегируются априорные вероятности при втором проходе по сигнатурам. То есть если в рамках этого окна хотя бы в одной клетке априорная вероятность не ноль, то для текущей клетки будет осуществлен второй проход по сигнатурам. По умолчанию установлено на 1 (то есть учитываются значения априорных вероятностей только для самой точки).

-recalc (recalculate_likelihood_on_overflow) - коррекция аргументов функции правдоподобия в случае, если она выходит за допустимый диапазон значений для чисел с плавающей запятой. На данный момент осуществляется пересчет только степени экспоненты функции правдоподобия (квадратичной формы), аналогичная операция для определителя еще не введена. Логическая переменная, значение по умолчанию - 0 (выключено).

-rcout (likelihood_correction_filename) - основа для имени файлов, куда будет сохраняться данные о величине коррекции аргументов функции правдоподобия (опция -recalc). Коррекция делится на 2 компоненты: значение, на которое был домножен определитель и значение, которое было прибавлено к степени экспоненты. Для степени экспоненты значения будут писаться в файл с добавкой qform, для множителей определителя - с добавкой det. По умолчанию - не указано (пустая строка) - данные не будут сохраняться.

12.5 Вывод данных

-LKout (likelihood_output_stackname) - имя файла стека, в файлы которого программа будет сохранять данные о значения функций вероятности для каждого класса. Разрешение на уровне пикселов, формат пиксела - рациональное число с разрядностью 64 бит. Открыт для записи на этапе классификации. Порядок файлов в стеке указывает на номер класса (1 - 1класс, 2 - 2 класс и.т.д.). Также в стеке должен присутствовать файл для хранения данных для нулевого класса (на самом первом месте в стек файле). По умолчанию не указан, значения функции правдоподобия не будут сохраняться.

-bestLK (best_likelihood_output) - имя растрового файла, в который программа будет сохранять данные о максимальном значении функции правдоподобия в пикселе среди всех классов. Разрешение на уровне пикселов, формат пиксела - рациональное число с разрядностью 64 бит. Открыт для записи на этапе классификации. По умолчанию не указан, значения функции правдоподобия не будут сохраняться.

-invout (invertible_output_stackname) - имя файла стека, в файлы которого программа будет сохранять информацию об обратимости матриц. Разрешение на уровне пикселов, формат пиксела - беззнаковое целочисленное число с разрядностью 8 бит. Открыт для записи на этапе классификации. Для каждого класса свое отдельное изображение. Значение пиксела равное 1 говорит об обратимости матрицы для данного класса, равное 0 - о необратимости. По умолчанию - пустая строка (не выводить эту информацию).

-numout (output_points_num_base_name) - основа для имен файлов, в которые при постобработке будет выводиться информация о числе точек, на которых были подсчитаны вторые сигнатуры. Разрешение на уровне клеток, формат пиксела - рациональное число с разрядностью 64 бит. Открыт для записи на этапе постобработки. Имена файлов будут иметь вид базовое_имя_class_класс_канал. Будет выводиться для всех классов или только для одного в зависимости от опции -clsout. По умолчанию - пустая строка, вывод не будет осуществляться.

-avgout (output_average_base_name) - основа для имен файлов, в которые будет выводиться информация о средних значениях вторых сигнатур при постобработке. Разрешение на уровне клеток, формат пиксела - рациональное число с разрядностью 64 бит. Открыт для записи на этапе постобработки. Имена файлов будут иметь вид базовое_имя_class_класс_канал. Будет выводиться для всех классов или только для одного в зависимости от опции -clsout. По умолчанию - пустая строка, вывод не будет осуществляться.

-covout (output_covariations_base_name) - основа для имен файлов, в которые будет выводиться информация о ковариациях между каналами вторых сигнатур при постобработке. Разрешение на уровне клеток, формат пиксела - рациональное число с разрядностью 64 бит. Открыт для записи на этапе постобработки. Имена файлов будут иметь вид базовое_имя_class_класс1_канал1-канал2. Будет выводиться для всех классов или только для одного в зависимости от опции -clsout. По умолчанию - пустая строка, вывод не будет осуществляться.

-pstsig (postprocessing_signature_num) - определяет, для каких сигнатур будет происходить подсчет и вывод всех характеристик на этапе постобработки (-Mode=4). Значение равное 1 указывает на первые сигнатуры, 2 - на вторые. Любое другое значение приводит к завершению работы программы. По умолчанию - обрабатываются вторые сигнатуры (равно 2).

-clsout (class_to_print_signature_data) - номер класса, для которого будет осуществляться вывод значений сигнатур (опции). Если ноль - то будет выведено для всех классов. По умолчанию - выводит данные для всех классов (значение = 0)

-sepout (separability_files_base_name) - основа для имен файлов, в которые будет выводиться информация о разделимости классов на этапе постобработки. Разрешение на уровне клеток, формат пиксела - рациональное число с разрядностью 64 бит. Открыт для записи на этапе постобработки. Значения разделимости будут выводиться в растровые файлы попарно для всех классов в виде базовое_имя_класс1-класс2.расширение. Как будет подсчитано значение разделимости, задается опцией -sepdist, какие каналы при этом будут использованы - задается опцией -sepch. Если равно пустой строке - не выводит ничего. По умолчанию равно пустой строке (не выводит данных о разделимости).

-sepdist (separability_distance) - имя вида расстояния, которое будет использовано для подсчета значения разделимости между классами (опция -sepout). Возможные значения: euclidean, divergence, transformed divergence, jeffries-matusita - скопированы из классификатора ERDAS IMAGINE, их подробное описание можно посмотреть в помощи по ERDAS IMAGINE, в файле FieldGuide.pdf, раздел Separabilty. Программа записывает в изображение набор кодов ошибок в случае, если невозможно подсчитать значение разделимости. Список кодов:

Тип расстояния	Название ошибки	Код ошибки
Euclidean	Ошибки отсутствуют	
Divergence Transformed divergence	Матрица или вектор нулевые	2100
	Вектора средних одинаковы	2200
	Матрицы ковариаций одинаковы	2300
	Матрица(ы) необратима(ы)	2400
	Значение равно бесконечности	2500
	Ошибка вычисления	2600
Jeffries-Matusita	Матрица или вектор нулевые	2100
	Вектора средних одинаковы	2200
	Полусумма матриц необратима	2300
	Матрица(ы) необратима(ы)	2400
	Вне допустимого диапазона значений	2500
	Значение равно бесконечности	2600
	Ошибка вычислений	2700

По умолчанию - используется евклидово расстояние (euclidean).

-sepch (separability_channels) - определяет, какие каналы будут использованы при подсчете разделимости между классами (опция -sepout). Каналы задаются в виде строки вида (“01 02 04” - то есть использовать 1 2 и 4 каналы, нумерация с единицы). В матрице ковариаций останутся только строки и столбцы с номерами включенных каналов (в векторе средних - аналогично только строки включенных каналов). Если строка пустая - используются все каналы. По умолчанию используются все каналы - строка пустая.

-rout (pass_2_window_size_output) - указывает стэк файл, куда выводятся размеры окна, на котором были подсчитаны значения вторых сигнатур. Разрешение на уровне клеток, формат пиксела - беззнаковое целое с разрядностью 32 бита. Открыт для записи на этапе 2-го прохода. По умолчанию - пустая строка, значения не выводятся.

-sigzero (signature_set_to_zero) - позволяет осуществлять частичное или полное обнуление сигнатур на этапе постобработки. При значениях, отличных от нуля значения будут обнулены. Какие значения будут обнулены задается опциями -sigzeroX, -sigzeroY, -sigzerocl, -sigzeroch. Какая сигнатура будет обнулена определяет опция -pstsig. По умолчанию значение равно нулю, обнуление не производится.

-sigzeroX (signature_point_to_zero_X) - определяет координату точки, в которой будет происходить обнуление сигнатур по горизонтальной оси (опция -sigzero). Если значение равно -1 - обнуление будет происходить для всей строки. По умолчанию равно -1 (обнуляет все точки в заданной строке).

-sigzeroY (signature_point_to_zero_Y) - определяет координату точки, в которой будет происходить обнуление сигнатур по вертикальной оси (опция -sigzero). Если значение равно -1 - обнуление будет происходить для всего столбца. По умолчанию равно -1 (обнуляет все точки в заданном столбце).

-sigzerocl (signature_class_to_zero) - определяет номер класса, для которого будет происходить обнуление сигнатур (опция -sigzero). Если значение равно -1 - обнуление будет происходить для всех классов. По умолчанию равно -1 (обнуляет сигнатуры для всех классов).

-sigzeroch (signature_channel_to_zero) - определяет номер признака, для которого будет происходить обнуление сигнатур (опция -sigzero). Если значение равно -1 - обнуление будет происходить для всех признаков. Если опция -S="simple", то рекомендуется использовать значение -1 всегда, иначе при повторном подсчете для тех же файлов сигнатур значения могут быть неверными. По умолчанию равно -1 (обнуляет сигнатуры для всех признаков).

12.6 Технические

-MTnum (num_of_threads) - указывает число потоков, используемое на этапе классификации. Каждый поток выполняет обработку для своей группы пикселов и выполнение в несколько потоков может существенно ускорить скорость работы на многоядерных системах. Реальное количество потоков на один больше указанного пользователем, дополнительный поток периодически также потребляет какое-то количество процессорного времени. Индикатор прогресса в многопоточном режиме идет несколько быстрее, чем реальный прогресс классификации. Оптимальное количество потоков в этой опции равно количеству ядер в системе. По умолчанию равно 1.

-NL (NUMLAYERS) - максимально возможное число слоев во всех изображениях (включая файлы сигнатур). По-умолчанию - 25000.

-BS (BLOCKSIZE) - размер блока изображения, который программа считывает из изображения и хранит. По-умолчанию - 512 Кб.

-ML (MEMORYLIMIT) - максимальный размер памяти, который могут занимать все хранимые в памяти программы блоки изображений. Рекомендуется выставить его таким, чтобы программа могла хранить в памяти хотя бы один блок каждого изображения (слоя). По-умолчанию - 8*10E+8 байт (800 Мб)

-S1create (create_first_signature_files) -S2create (create_second_signature_files) - указывает, нужно ли программе заново создавать файлы сигнатур для первого и второго прохода. Для создания файлов сигнатур программе нужно указать число классов (-NC) и число пикселей в клетке сигнатур (-Step). По-умолчанию выключено (значение 0).

-NC (num_of_classes) - число классов в классификации. Нужно только при построении файла сигнатур, в остальных случаях будет определяться по самим файлам сигнатур. По умолчанию не указано.

-Step (gridstep) - размер шага клетки сигнатур (в пикселях). По умолчанию не указано и определяется автоматически, по размеру сигнатур и изображений на входе.

-SXsize (grid_size_X) - размер регялярной сетки (число клеток) в ширину (по координате X). Должна быть задана вместе с опцией -SYsize, иначе игнорируется. При задании данной опции значение опции -Step игнорируется. Именно эта опция используется для подсчета числа пикселов в клетке (ширина_изображения/число_клеток). Неположительные значения означаю, что опция не задана. По умолчанию не задана, значение равно -1.

-SYsize (grid_size_Y) - размер регялярной сетки (число клеток) в высоту (по координате Y). Должна быть задана вместе с опцией -SXsize, иначе игнорируется. При задании данной опции значение опции -Step игнорируется. Неположительные значения означаю, что опция не задана. По умолчанию не задана, значение равно -1.

-datacache (use_signature_data_cache) - указывает, будет ли программа сохранять данные о матрицах ковариации и векторах для последующего использования в рамках той же клетки. Дает существенный прирост (до 2x) для задач с большими размерами матриц, но для более мелких задач (матрицы 1x1 или 2x2) может даже привести к замедлению при очень большом числе классов. По умолчанию выключено (значение 0).

-drv (GDAL_default_driver) - указывает, какой драйвер нужно использовать при создании новых изображений с помощью библиотеки GDAL. Названия драйверов берутся из этого второй колонки этого списка `http://gdal.org/formats_list.html`. По умолчанию используется драйвер “HFA” (Erdas Imagine (.img)).

-typechk (check_types_on_image_IO) - указывает, нужно ли осуществлять проверку типов пикселов изображений на входе и выходе на соответствие их внутреннему представлению в программе. При включении данной опции в случае, если тип пиксела изображения не соответствует его внутреннему представлению (например целочисленное и рациональное числа), то будет выведено предупреждение (но работа программы продолжится). Включение данной опции может сильно уменьшить скорость работы программы. Значения опции отличные от нуля означают, что опция включена. По-умолчанию выключено (значение 0).

12.7 Регрессии

-regch (channels_for_regression) - Файл с информацией о каналах, используемых в регрессии. Подробное описание формата файла и его применение - в разделел регрессий.

-regcl (classes_for_regression) - Файл с информацией о классах, используемых в регрессии. Подробное описание формата файла и его применение - в разделел регрессий.

-regcalc (calculate_regressions_coefs) - При ненулевом значении на этапе постобработки будут расчитаны линейные регрессии (на основе сигнатур, собранных на 1 и 2 этапе).

-regcalc (directory_to_store_regressions) - Папка для сохранения линейных регрессий, полученных на этапе постобработки.

-regprod (restored_channels_base_name) - Файл для сохранения восстановленных на основе регрессий значений в одном из каналов. Пустое значение означает, что восстановление не будет проводиться и файл не будет создан.

-rfregfile (RF_regression_file) - Файл описывающий информацию для построения и восстановления каналов по линейным регрессиям.

-rfregignore (RF_regression_ignored_value) - Значение зависимого канала, игнорируемое при построении регрессий на основе случайных лесов, например 0. Значение по-умолчанию - (-1).

-rfreginfo (RF_regression_info_file) - Базовое имя файла для сохранения вспомогательной информации о регрессиях на основе случайных лесов. Пустая строка указывает, что вспомогательная информация не будет сохранена.

12.8 Разное

-Tfile (time_detalisation_file) - имя файла, куда программа будет сохранять информацию о времени выполнения каждого этапа. По умолчанию - не указан, информация не будет сохраняться.

-smplrndmax (max_random_subsample_size) - опция, задающая возможность ограничения размера выборки до определенного предела. Выборка будет случайным образом прорежена, пока ее размер не станет равен значению опции. Может использовать для ускорения и уменьшения требуемой памяти в непараметрических классификаторах По умолчанию - прореживание не выполняется, значение равно 0

-sDBf (sample_DB_filename) - имя файла базы данных, в который программа на первом этапе запишет обучающую выборку. Доступ к базе данных значительно ускоряет этап классификации (или построения регрессиий) для непараметрических методов, но при этом база данных может занимать много места на диске. Не работает для классификации по методу максимального правдоподобия. По умолчанию - не создается (пустая строка)

-Setone (one_training_storage_file) - указывает, должна ли информация об обучающей выборке храниться в одном файле или в нескольких. Значения не равные еденице указывают на то, что информация должна храниться в одном файле. Значение по умолчанию - 0.

-classtr (classifier_storage_type) - Обозначает тип хранилища файлов классификаторов. В зависимости от него будет меняться физическая структура хранения данных классификатора. Единственное возможное значение на данный момент - "ML_storage файлы для сигнатур метода максимального правдоподобия. Значение по умолчанию - "ML_storage".

-infotype (sample_info_storage_type) - обозначает тип хранилища для файлов с информацией об обучающей выборке. На данный момент единственное возможное значение - "images указывающее на то, что информация будет храниться во множестве файлов изображений. Значение по умолчанию - "images".

-settype (sample_storage_type) - обозначает тип хранилища для обучающей выборки (класс и значения в каналах). На данный момент у опции нет возможных значений, возможность хранения выборки не внедрена. Значение по умолчанию - пустая строка.

-smplinfo (save_sample_info) - указывает, нужно ли сохранять информацию об обучающей выборке, значения отличные от нуля означают, что информацию будет сохранена. Значение по умолчанию - не нулевое.

-clstr (save_classifiers) - указывает, нужно ли сохранять информацию об обучающей выборке, значения отличные от нуля означают, что информацию будет сохранена. Значение по умолчанию - не нулевое (ненулевое для -clas=ML).

-bycell (by_cell_classification) - указывает, должна ли программа выполнять классификацию по пикселям или по клеткам. Выполнение по клеткам позволяет более эффективно работать с классификаторами, по пикселям - эффективнее читать данные из файлов. На данный момент программа оптимизирована под поклеточную обработку. Значения, отличные от нуля означают, что выполнение идет по клеткам. Значение по умолчанию - не равно нулю (по клеткам).

-treenum (decision_forest_number_of_trees) - число деревьев в классификаторе случайные леса. Принимает целочисленные значения отличные от нуля. Значение по умолчанию - 100.

-rfrvars (decision_forest_vars_per_split) - число характеристик (каналов), использующихся в каждом узле дерева для разделения классов. В каждом узле из всех параметров будет случайным образом выбрано указанное количество параметров и по нему будет построено разбиение. Целочисленные положительные значения, обычно используется корень из числа характеристик. Значение по умолчанию - 100.

-setpart (sample_part_used_for_training) - процент выборки, используемой для обучения одного дерева из леса (остальная часть используются для оценки ошибки). Принимает рациональные значения от 0.0 до 1.0. Работает только в имплементации alglib (-clas=RF_alglib). Значение по умолчанию - 0.66.

-trdepth (max_depth_of_tree) - указывает максимальное число узлов дерева в глубину (каждое дереве будет "усечено пока не будет достигнут данный лимит). Работает только в имплементации opencv (-clas=RF_opencv). Значение по умолчанию - 100.

-leaflim (leaf_samples_limit) - указывает ограничение на число элементов в конечном узле дерева. Последующее разбиение в узле останавливается, когда число элементов выборки, попавших в узел, меньше данного числа. Работает только в имплементации opencv (-clas=RF_opencv). Значение по умолчанию - 32 (возможно, не очень удачное, можно попробовать также 1, 5).

-dfnslc (num_of_slices) - диапазон значений характеристик (каналов) будет разбито приблизительно на такое число интервалов, чтобы найти оптимальное разбиение. Уменьшение может привести к увеличению скорости работы и уменьшению точности классификатора. Работает только в имплементации opencv (-clas=RF_opencv). Значение по умолчанию - 50.

-clas (type_of_classifier) - тип классификатора, используемый для классификации. Возможные значения:

Идентификатор	Название
"ML"	Максимальное правдоподобие
"RF_alglib"	Случайные леса (имплементация библотеки alglib)
"RF_opencv"	Случайные леса (имплементация библотеки opencv)
"KNN"	Метод K ближайших соседей
"SVN"	Метод опорных векторов
"NN"	Нейронные сети

Значение по умолчанию - "ML".

-cprob (prob) - стек для вывода доли проголосовавших деревьев за каждый класс в методе случайных лесов (-clas=RF_opencv или -clas=RF_alglib). Число элементов в стеке - число классов плюс 1, разрешение на уровне пискелей, формат пиксела - рациональное число с разрядностью 32 бит. Пустая строка означает, что значение выводится не будет. Значение по умолчанию - пустая строка.

-caprob (aprob) - стек для вывода доли проголосовавших деревьев за каждый класс в методе случайных лесов (-clas=RF_opencv или -clas=RF_alglib). Доля проголосовавших деревьев будет выведена с коррекцией на априорные вероятности в пикселе. Число элементов в стеке - число классов плюс 1, разрешение на уровне пискелей, формат пиксела - рациональное число с разрядностью 32 бит. Пустая строка означает, что значение выводится не будет. Значение по умолчанию - пустая строка.

-crele ("rele") - изображение для вывода ошибки в классификации обучающей выборки с помощью построенного по ней классификатора типа случайный лес (-clas=RF_opencv или -clas=RF_alglib). Разрешение на уровне пискелей, формат пиксела - рациональное число с разрядностью 64 бит. Пустая строка означает, что значение выводится не будет. Значение по умолчанию - пустая строка.

-corele ("orele") - изображение для вывода ошибки в классификации обучающей выборки с помощью построенного по ней классификатора типа случайный лес (-clas=RF_opencv или -clas=RF_alglib). Ошибка считается в каждом дереве только по той части выборки, которая не использовалась для построения этого дерева (out-of-bag error). Разрешение на уровне пискелей, формат пиксела - рациональное число с разрядностью 64 бит. Пустая строка означает, что значение выводится не будет. Значение по умолчанию - пустая строка.

-setequal ("use_sample_correction") - указывает, должна ли проводиться коррекция выборки. При значении 1 происходит уменьшение выборки, после коррекции число элементов выборки для каждого класса в выборке становится равным. Для этого находится класс с наименьшим числом элементов в выборке и число элементов для остальных классов уменьшается до этого числа за счет случайного выкидывания элементов данного класса. Значение 2 - расширение выборки. Выбирается один класс с максимальной выборкой, выборка остальных классов раздувается до числа элементов в самом представительном классе. Значение по умолчанию - ноль.

-setcache ("use_sample_cache") - указывает, нужно ли сохранять ранее загруженные выборки для последующего повторного использования. Значения, отличные от нуля означают, что прошлые выборки будут сохраняться, значение по умолчанию - ноль.

-infocache ("use_sample_info_cache") - указывает, нужно ли сохранять ранее загруженную информацию о выборке для последующего повторного использования. Значения, отличные от нуля означают, что прошлые значения будут сохраняться, значение по умолчанию - ноль.

-knn (knn_number_of_neighbors) - Число соседей в методе K ближайших соседей (по-умолчанию - 10).

-svmt (svm_type) - Тип классификатора метода опорных векторов (различаются методом разделения классов). Возможные значения:

Идентификатор	Название
"C-type"	?
"NU-type"	?

Значение по умолчанию - "C-type".

-svmk (svm_kernel) - Тип ядра при классификации методом опорных векторов. Возможные значения:

Идентификатор	Название
"linear"	Линейное ядро (без преобразования переменных)
"poly"	Полиномиальное ядро
"rbf"	Радиальная базисная функция
"sigmoid"	Сигмоидная функция

Значение по умолчанию - "rbf".

-it (training_iterations_number) - Максимальное число итераций при тренировке классификатора (опорные вектора, нейронные сети). Значение по умолчанию - 100.

-eps (training_termination_epsilon) - Минимальная ошибка до прекращения тренировки классификатора (опорные вектора, нейронные сети). Значение по умолчанию - 0.000001.

-nnf (neuron_type) - Тип функции активации для классификатора на основе нейронных сетей. Возможные значения:

Идентификатор	Название
"linear"	Линейная функция
"sigmoid"	Сигмоидная функция
"gauss"	Гауссова функция. В соответствии с документацией opencv внедрена не полностью.

Значение по умолчанию - "sigmoid".

-nnl (mlp_layers) - Количество скрытых слоев и число нейронов в них для метода нейронных сетей. Задается строкой вида "N;K;L в которой перечисляется число нейронов в каждом скрытом слое. Программа автоматически добавит необходимое количество входных и выходных нейронов. Значение по умолчанию - (нет скрытых слоев).

13 Примеры запуска:

LAGMA.exe s_2.ihdr b50.istk sig1.sig sig2.sig out.ihdr
-apri apri/apr.istk //файл грубых априорных вероятностей
-awin 3 //величина окна для грубых вероятностей
-S2cache 1 //ускорение подсчета вторых сигнатур
-Mode 23 //только подсчет вторых сигнатур и классификация
-Tfile D:/Lagma_time.txt //выводить информацию о времени выполнения
-Rmin 2 //минимальный размер окна при втором проходе
-Rmax 100 //максимальный размер окна
-MS 1000 //минимальный размер выборки
-BS 100000 //размер блока памяти
-ML 1000000000 //максимальное количество памяти под изображения
-RXmin 5400 //область для классификации
-RYmin 1400
-RXmax 8100
-RYmax 11100

LAGMA.exe src/set_0106.ihdr src/priznak_sp_0106.istk sig1.sig sig2.sig src/res_sp_0106.ihdr
-apri src/av_low_0106.istk //файл грубых априорных вероятностей
-aap src/av_hi_0106.istk //файл точных априорных вероятностей
-MS 50 //минимальный размер выборки для пикселей
-MS2 10 //минимальный размер выборки для признаков
-W 1 //осреднение окном
-Wx 2 //размер окна по вертикали и горизонтали
-Wy 2
-NC 2 //число классов
-S1create 1 //создание первых и вторых сигнатур
-S2create 1
-Step 101 //шаг клетки для сигнатур
-S2cache 1 //ускорение подсчета вторых сигнатур
-Tfile D:/Lagma_time.txt

LAGMA.exe sampling.ihdr metrics2010.istk sigm1.sig sigm2.sig out.ihdr
-apri apri_new/prob.istk

-aap aap/prob.istk
-S2cache 1
-Mode 3
-Tfile D:/Lagma_time.txt

Пример файла конфигурации:

-RXmax=908
-RYmax=908
gridstep=101
-S2cache=1
num_of_classes=2
-S1create=1
-S2create=1
-SXsize=9
-SYsize=9
time_detalisation_file=D:/Lagma_time.txt