Информационные процессы, Том 6, № 4, 2006, стр. 289–309.

ПЕРЕДАЧА ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ

Субъективная кластеризация. Эволюционные процессы на множестве объектов

А.В.Гащенко

[*] *Институт проблем передачи информации, Российская академия наук, Москва, Россия*
[**] *Институт проблем управления им. В.А.Трапезникова, Российская академия наук, Москва, Россия*
Поступила в редколлегию 20.10.2006

Аннотация—Статья содержит базовые идеи метода субъективной кластеризации, т.е. такой кластеризации, где близость объектов оценивается не формально, а определяется человеком. При этом признаки, по которым производится кластеризация, как правило, являются субъективными. В статье описан общий подход к задаче, но, так как работа находится на начальной стадии, особое внимание уделено процессу накопления субъективных знаний для кластеризации.

1. ВВЕДЕНИЕ

Кластеризация – довольно общее понятие, означающее выделение из множества объектов относительно однородных групп. Его часто используют, когда хотят сказать об организационном разбиении некоторой структуры на части, например, кластеризация компьютерной сети или кластеризация предприятия. В прикладном же смысле под ним понимают некоторые статистические методы, позволяющие в пространстве свойств выделять группы близких объектов. Этих методов существует уже несколько десятков и еще больше их модификаций. Это связано в первую очередь с разнообразием задач: сегментация изображений, маркетинг, борьба с мошенничеством, прогнозирование, анализ текстов и многое другое. В статистике кластеризация часто выступает первым шагом при анализе данных. После выделения схожих групп применяются другие методы, для каждой группы строится отдельная модель.

Таким образом, кластеризация – один из фундаментальных методов современного анализа данных. Её идея – проста, понятна и стара; еще в древние времена среди воинов, например, стали выделять лучников и мечников, среди растений съедобные, ядовитые, красивые и колючие, а среди людей - быстрых/медленных, сильных/слабых, глупых/умных. Но результат этого “выделения групп” не очень-то похож на результаты алгоритма кластеризации, да и исходные данные разные. Алгоритмы используют для анализа вектора признаков объектов. Признаки могут быть числовые или категориальные и они задают координаты в некотором пространстве. В этом же пространстве задается некоторая формальная мера близости и на её основе разными способами ищутся “близкие” объекты. В качестве меры близости для числовых признаков часто используется Евклидова метрика, для категориальных Хеммингова. Что будет значить расстояние, если по одной оси отложен возраст, по второй вес, а по третьей время время завтрака – не понятно. Ещё более непонятно, получится ли выделить зависимость, явно следующую из веса и роста, если кроме этих двух признаков анализируются еще 2000. Какую реальную пользу удастся извлечь из такой оценки близости – сложно сказать. Но, конечно, это не единственная проблема подобных алгоритмов. Есть еще, например, проблема масштабируемости – это проблема создания такого алгоритма, который смог бы работать на гигантских базах данных. Подходы к решению этой проблемы понятнее, и раз не ясно, что делать с мерами – можно позаниматься, например, масштабируемостью.

Но всё же, было бы полезно получить возможность кластеризации объектов, признаки которых затруднительно складывать и вычитать. Возможность выделять осмысленные группы симптомов и диагнозов в медицине, товаров на рынке, акций на бирже, музыки, фильмов и софта на файл-сервере. Для этого нужен инструмент, предоставляющий возможность произведения субъективной кластеризации, т.е. такой кластеризации, которая вместо абстрактных формул, предлагающих некоторую подгонку под желаемое, использовала бы субъективные оценки непосредственно пользователя.

Какого рода оценки юзер может предоставить? Есть измеряемые данные, и с ними работают стандартные пакеты. Но существует и огромное число неизмеряемых приборами данных, таких, как "болезненность цвета лица пациента", "кредитоспособность заёмщика" или "напористость ритма". Человек сравнительно легко может оценивать эти свойства и они были бы очень полезны при анализе. Возможно, именно среди них имеются ключевые, ведь раз их научилось выделилять наше сознание, значит они, как минимум, имеют право на существование. Но они зачастую не измеряются автоматически и нет удобной возможности их учитывать и работать с ними. Инструменты позволяющие фиксировать подобного рода данные сейчас есть, например, у врачей в хороших клиниках, но их возможности бывают ограничены стандартным вопросником. Там пользователь отвечает на стандартные вопросы, которые задаются ему в предварительно установленном порядке, зависящем от уже представленных ответов. Таким образом, человек может являться "измерителем" некоторых параметров. Вообще говоря, в таком случае, используются его знания о том, что такое "болезненный цвет лица" и умение описать, например, форму опухоли с помощью принятых параметров. Необычные же случаи требуют перестройки вопросника и, в лучшем случае, описываются в коментариях и в последствии это как-то будет учтено.

Такую систему можно сделать гораздо более гибкой, если использовать не только стандартные вопросы, но дать возможность свободным образом описывать каждый объект новыми характеристиками. Новое слово или словосочетание добавляет к множеству характеристик новую. Такой метод описания объектов помогает сохранять знания человека об объекте, со временем накапливать их, выделять среди них значимые, различным образом анализировать накопленное, собирать группы похожих. Собственно здесь и проявляется эволюционность – внимание человека уделяется ограниченному числу характеристик, и потому используемыми остаются только самые значимые. Постепенно на основе старых или же со стороны появляются новые характеристики вытесняющие накопленные ранее. Однако старые при этом остаются в системе и ничего не мешает продолжать их использовать.

Все эти действия требуют много внимания человека и, по сути, практически не содержат никакой автоматики, они лишь помогают по новому взглянуть на материал, и не дают пропасть результатам ознакомления пользователя с материалом. Получается своеобразное хранилище знаний человека о множестве объектов с которыми работает парограмма. При этом своеобразная кластеризация выполняется автоматически – все объекты описанные одним словом образуют кластер. Т.к. работа программы напрямую связана с тем, что мы назвали "человеческими знаниями", стоит уточнить, о чем идет речь.

1.1. О человеческих знаниях

Наши знания – это продукт работы нашей нервной системы, это отображение получаенного опыта, это то, в чем мы уверены, то, что мы умеем делать. С точки зрения кластеризации, знание – это как раз принадлежность объекта к группе, или же группа со всеми её объектами (это и есть кластер, собственно). Можно ли назвать кластеризацией на множестве возможных действий человека поиск предпочтительных в некоторой ситуации? Почему бы и нет. Таким

образом, продукт кластеризации связан со знаниями, а набор вариантов, которыми обладает человек в некоторой ситуации, вполне можно назвать субъективным кластером.

Что происходит с нашими знаниями в течение жизни? Что угодно. Мы можем забывать, узнавать что-то, что считаем более правильным, хранить знания и стараться передавать их, или просто использовать их и никогда об этом не задумываться, как, например, не задумывается о своих движениях водитель. Так или иначе, большая часть человеческих знаний уходит вместе с человеком. Умение кататься на горных лыжах не передается словами или произведениями искусства, этому можно научиться только на практике, хотя многое может помочь . Написаны тысячи книг, обучающих людей общаться друг с другом, но они не способны передать знания автора читателям. Обычно, максимум, что удается – описать новую для читателя ситуацию и натолкнуть его на размышления. Это и не удивительно. Знание автора включает в себя широчайший спектр различных ситуаций, основывается на жизненном опыте, подпитывается эмоциями. А что мы получаем, читая книгу? Даже об интонациях, с которыми проносились в голове записываемые фразы, можно только догадываться.

Необходимость работы со знаниями заключается в том, что люди хотят знать то, что будет помогать им жить лучше. Но находить эти знания и принимать их – остается проблемой. Даже если современными средствами удается найти что-то по интересующей теме, то совсем не факт, что материал поможет достичь желаемого результата.

Кроме того, современный мир переполняется информацией, очень лёгкой для восприятия, но несущей крайне мало полезного: например, различной рекламой. Сколько рекламы можно увидеть, если час гулять по городу? Я насчитал около сотни только знакомых имён. Что видят люди смотрящие телевизор каждый вечер? Что слышат люди у которых целый день в машине и на работе играет радио? Чем наполняется их лексикон? Современным миром производится огромное количество информации, которая течет на нас из СМИ, от знакомых болтунов, из книг, газет и Интернета. Когда человек едет в метро, ему даже не на что смотреть, кроме рекламы. Какая её доля полезна для людей? Кто может отличить информацию полезную для дела и здоровья (как физического, так и душевного), от вредной? Кто, вообще, об этом задумывается?

Если активный запас знаний у человека ограничен, то что приносит в нашу жизнь постоянный повторяющийся рекламный фон? И что уносит?

Люди в целом слабо борются с новыми волнами красиво сделанной, блестящей, бодрящей информации – она приятна, а приятного в простой жизни не так уж и много. Но при этом, она зачастую бесполезна и абсолютно несвоевременна. Какой смысл мужчине везде читать про шампуни для длинных и ломких волос? Или женщине, если у неё короткие неломкие волосы? Полезность такой информации мизерна, но всё же она необходима, как и многая другая. Это как раз одна из областей, где необходима кластеризация: не надо показывать всю рекламу всем, нужно умело выбирать целевые группы и, аналогично, группировать товары. Это могло бы существенно оптимизировать рекламные процессы и сделать мир PR не таким хищным. Теоретически, конечно.

Итак, мы непрерывно получаем новые знания. Многие из них моментально признаются ненужными, и им не уделяется внимание. Какие-то, напротив, обращают на себя внимание. Сам факт обращения внимания, кстати, тоже может явиться новым знанием. Потом появляются новые и новые знания и невостребованные старые теряются среди них. Мы может вспомним, что сказал нам друг, если, словно deja vu, окажемся снова на той же скамейке в тех же сумерках, но не вспомним, когда он будет говорить: “Эй, ну ты помнишь, я до этого тебе про новую машину рассказывал!”.

Известен опыт (Карл Лэшли, http://www.i-u.ru/biblio/archive/godfrua_chto/07.aspx, “Локализация функций памяти”) в котором у мыши последовательно удаляли разные участки мозга,

чтобы выяснить, где хранятся знания необходимые для прохождения лабиринта. Выяснили, что какой бы участок не удалялся – мышь помнит, как выходить из лабиринта, но может совершить больше ошибок. Следовательно, знания распределяются по всему мозгу, и тогда как что-то может быть совсем удалено оттуда – не очень понятно. Скорее в активе не остается ничего, что напоминает нам о забытом.

Я несколько лет активно собираю музыку. В среднем за месяц я получаю около 30-50 CD новой музыки, которую прослушиваю хотя бы по одному разу, и вот что обнаружил:

1) когда у меня хорошее настроение, я не помню, где у меня лежит грустная музыка и наоборот

2) отдельная песня может быть очень в тему ситуации, но я не вспомню про неё, пока не увижу на экране её название. Это маловероятно, т.к. нужно оказаться в папке с этой песней, а папок очень много

3) я часто не помню, как называется песня, но помню, где лежит. Притом даже папки помню, прежде всего, не по названиям, а по их примерному положению на экране.

4) очень обидно понимать, что эффективность использования 300 Гб музыки мизерная – я с трудом нахожу наиболее подходящую под ситуацию музыку. Часто приходится просто водить глазами по папкам в поисках той, что вызовет желание в неё зайти. Часто приходится подготавливать материал заранее.

В итоге, у меня создалось впечатление, что я помню и использую около 300-400 трэков. Остальные вспоминаю, только если увижу их название, и в них было что-то очень ярко запоминающееся. Про многие трэки я могу сказать только, что они мне раньше очень нравились. Что именно мне в них нравилось я объяснить не могу, лишь смутно помню какой-то звук, и то бывает ошибочно. Зато сказать, что “я это уже слышал” могу гораздо чаще. В общем, ситуация очень похожа на известные описания “активного” и “пассивного” словарного запаса (http://www.gramma.ru/RUS/?id=6.35).

Еще один пример: 2.5 года интенсивной работы над большим программным проектом. Притом проект имел такое свойство, что часто важнее было не качество кода, а количество, т.е. постоянно нужно было добавлять что-то новое. К концу второго года я мог сказать, что совершенно не помню, как работают участки кода написанные за год до этого. Могу только догадаться, поглядев на них, но тонкости забыты. Иногда же не удается вспомнить код, написанные две недели назад, например, потому что приходит известие, что старый друг женится и нужно срочно ехать к нему, искать подарки, выкраивать время, откладывать дела. Что-то отвлекло, и все детали вспомнить уже не получается. А детали могут быть очень важными.

Таким образом, складывается потребность в создании некоторого инструмента, позволяющего справляться с наплывом информации. Уже существуют удачные примеры инструментов, выполняющих свои цели:

1.2. Google

Самый известный и обсуждаемый поисковик в интернете, который произвел, ни много ни мало, революцию в сетевом бизнесе. Он анализирует несколько милиардов страниц, для того чтобы предоставить пользователям поисковый сервис.

Компания Google была создана в 1998 году. В конце 2005ого её основатель получил титул человека года от Financial Times:

http://www.utro.ru/articles/2005/12/23/507171.shtml

По данным статьи, к концу 2005ого года капитал компании составил 130 миллиардов долларов. Деньги, заработанные на поиске информации в Интернете, позволили компании в дальнейшем осуществить проекты, недавно казавшиеся абсолютной фантастикой: Google Earth

(http://earth.google.com/) позволяет увидеть на экране Интернет-браузера глобус такого уровня детализации, что на нём возможно найти собственный дом. Это ярчайший показатель того, что работа с информацией востребована.

1.3. Wiki-Pedia

http://www.wikipedia.org/

Этот проект одним своим существованием дарит веру в лучшее. Он уже 5 лет живет и эволюционирует на простейшей идее – это энциклопедия, в которой любую статью может отредактировать любой человек без указания какой-либо информации о себе. Просто зайти и отредактировать. В проекте ведутся множественные войны между несогласными друг с другом группами людей, существует огромная проблема вредительства, но и её удаётся решать. Сейчас энциклопедия wiki – самая большая и динамично развивающаяся. В ней более миллиона статей только на английском языке. Ссылки на материал википедии идут первыми в том же google. Количество статей на русском языке пока только приближается к 70 тысячам.

Энциклопедические статьи полезны, когда ясен смысл слов, которыми они написаны. Но не во всех интересующих областях язык может быть развит достаточно, чтобы хотя бы два эксперта пришли к единому мнению относительно значения многих понятий. Многие наши знания затруднительно описать словами.

1.4. Знания, которые сложно передать

“- С этим двигателем мой корабль будет лететь быстрее скорости света!

- Но это невозможно!

- Возможно всё, что можно себе представить!

- Тогда объясни, как!

- А вот это невозможно”

Футурама

Свои практические знания человек, как правило, не может описать так, чтобы другой человек понял именно то, что имелось в виду. “Вас на самом деле не существует”, - может сказать буддист со всем пониманием дела, но что он имел в виду? С чего он это взял и как его нужно понимать? Можно решить, что он не прав и не вникать в детали. Можно пообщаться с ним на эту тему, и понять, что он вложил в эти слова громадный личный опыт и еще более громадный опыт поколений, несущих некоторое знание, но удастся ли перенять этот опыт?

Я, говоря о музыке, могу сказать про трэк в стиле jungle, что это old school или new school. Если меня спросят, что я имею в виду, я не смогу четко объяснить. Можно сказать, что old school - это некоторый стиль, существовавший до появления нового, современного стиля (new school), можно сказать, что там совсем другое звучание, что используются другие инструменты, но это не поможет стороннему человеку отличать одно от другого. Можно попробовать описать особенности исполнения, и собеседник поймет меня, если мы уже год разговариваем с ним о музыке или читаем одну и ту же музыкальную литературу, но опять же, может и не понять.

А можно показать несколько характерных примеров и дать человеку возможность найти принцип отличия самому. Не факт, что после этого вы будете одинаково различать old и new school, но общение уже будет более объектным, слова обретут смысл. Взгляд на группу объектов, именованную некоторым словом, часто объясняет смысл понятия лучше любых слов. А если предложить сразу и объекты и описание к ним – можно получить вообще превосходный результат. Это и понятно, при передаче знаний часто бывает недостаточно слов, потому во всех учебных заведениях обязательна практика применения полученных знаний.

2. ПОСТАНОВКА ЗАДАЧИ

Целью работы является получение метода работы с субъективными знаниями пользователя о некотором множестве объектов. Метод должен позволять хранить знания, копить их и способствовать их эволюции. Так же он позволяет эффективно обмениваться ими и передавать их. Метод основывается на прирождённом умении человека распознавать объекты, относя их к некоторым именованным группам. Особенность метода в том, что он практически не использует никаких автоматических методов, он лишь позволяет получить новый, гибкий взгляд на данные, а все совершаемые над ними действия прозрачны, понятны и наглядны.

2.1. Отличия автоматических и мануальных методов, принцип усиления интеллекта

Есть два подхода к созданию инструментов: автоматический и мануальный. Автоматика сама выбирает, что и как нужно сделать, и делает. Мануальный метод – позволяет человеку самому рассмотреть ситуацию в нужных подробностях и сделать выбор. В то время, как мануальные методы, по сути, просто предоставляют новую возможность взглянуть на мир и участвуют в принятии решений только косвенно, автоматические методы всё решают сами, т.е. можно сказать о наличии у них некоторого поведения. Если сделать автоматический метод достаточно разумным, то он начнет навязывать свои модели поведения человеку, что, вообще говоря, может привести к тому, что в работу над задачей будет вовлечено больше человеческих ресурсов, чем планировалось изначально.

Принцип усиления интеллекта заключается в стремлении не избежать интеллектуальной работы, а сделать её более эффективной. Это важный принцип позволяющий избежать многих ошибок, рождающихся от надежды сделать что-то такое, что позволит нажатием одной клавиши решить все проблемы.

3. ПОДХОД К РЕШЕНИЮ

Для решения поставленной задачи пишется программа, позволяющая собирать, хранить и различным образом демонстрировать и анализировать накопленное. В её основе лежит несколько основных идей: специальная “файловая система” для хранения данных, “демоны Селфриджа” (Selfrige O., Pandemonium: A paradigmfor learning // Sympoziumon the mechanizationof thought processes. London, 1959), использование цветов, зонирование в папках.

3.1. “Файловая система”

В этой идее нет ничего оригинального, но интересно, что такое название она получила не сразу. Изначально интерфейс программы планировался, как дающий возможность объединять объекты по единому признаку простым перемещением объектов по экрану, притом признак тоже считался объектом и мог описываться. Но довольно быстро стало понятно, что это то же самое, что и считать признак – “папкой”, а объект “файлом” в некоторой файловой системе. Но, поскольку у объекта может быть несколько характеристик, он может лежать в нескольких папках сразу. Такая файловая система, как я слышал, планируется в Windows Longhorn. Поскольку, структура данных так похожа на файловую, естественно и интерфейс по работе с ней приблизить к привычным файловым интерфейсам, таким как Windows Explorer.

3.2. “Демоны Селфриджа”

Довольно быстро складывается ситуация, что в некоторых папках накапливается сотня и больше файлов. В такой ситуации найти нужное сложно. Но, допустим, у нас есть некоторый

способ объяснить программе, какие именно объекты нас интересуют. Тогда подходящие объекты мы можем увеличить в размерах, а неподходящие – уменьшить. Папки тоже оцениваются по их содержимому. Такой режим можно включать/выключать, менять силу его эффекта (пока в основном, чтобы избежать проблемы с автоматическим нормированием), кроме того, можно прятать все объекты меньше какого-то заданного размера. Что тоже очень помогает избавиться от лишнего.

3.3. Цвета

Поскольку работа идет с системой сильно взаимосвязанных данных, необходимо иметь возможность единовременно наблюдать наиболее полную картину. Смотреть папку с композициями с "женским вокалом" и не видеть никакой дополнительной информации об этих файлах – малополезно. Потому было решено каждой характеристике привязать цвет. Во-первых, это сильно добавляет узнаваемости характеристикам (особенно если цвет подобран удачно), во-вторых, позволяет компактно отображать принадлежность объекта характеристике. В результате было решено выбирать набор ключевых характеристик и указывать цветами на иконке файлов принадлежность к этим характеристикам (Ring Info). Была еще идея задействовать форму иконок, как не менее информативный элемент, но пока это привело к ряду технических проблем (интенсивная работа с изображениями на Java – проблема). Еще планируется добавить к цветам текстуры.

3.4. Зонирование в папках

Обычной табличной раскладки файлов в папках явно не хватает для полноценно творческого подхода к именованию данных, поэтому экземплярам файлов в папках было решено приписывать координаты, и пользователю дана возможность раскладывать файлы так, как ему удобнее. Это было сделано еще и с другой целью – чтобы дать возможность выделить в папках некоторые зоны, которым можно было придать отдельную семантическую нагрузку. По сути, это означает, что характеристика задается не только принадлежностью к папке, но и положением в ней.

Первый, простейший вид зонирования – выделение на фоне папок цветовых зон, принадлежность к которым означает характеристику. Например, папка может называться "опухоли" и разделяться на "злокачественные" и "доброкачественные". Но, как показала практика, это не слишком удачная идея, т.к. практически всегда находятся исключения из правил, которые хочется положить и туда и сюда, а это становится невозможно. В таких случаях отдельно создаются папки "злокачественные" и "доброкачественные" и всё, что нужно, кладется в них. Однако возможность оставлена для построения обучающих баз, т.к. очень просто и понятно вникать в терминологию, когда одно наглядно отделено от другого.

Второй вид зонирования – непрерывный. Когда указывается некоторая ось (например, ось красивости) и координата файла может проектироваться на эту ось (указывая на сколько файл "красивый").

Еще введение координат объектов в папках позволяет делать гибкие автоматические расстановки объектов. Например, легко применим принцип самоорганизующихся карт, по которому объекты могут быть автоматически разложены внутри папки по похожести.

3.5. Хранение информации

Общих подходов к хранению получилось два. Первый – хранить всю информацию, порождаемую программой, в её собственном документе, в одном месте. И в этом случае всё понятно, он стандартный.

Второй вариант заключается в том, что характеристики файлов, если они реально присутствуют на диске, хранятся в специальном файле в той же папке, что и сам файл. Этот подход отличается большим неудобством в реализации, но зато он позволяет легко копировать папки с файлами, перенося и их характеристики тоже. При появлении удобного инструмента для добавления чужих характеристик в собственную коллекцию это может очень помочь обмену знаниями о коллекциях.

В результате, первый подход существенно удобнее при работе программы, а второй гораздо лучше масштабируется, переносится и вообще отличается большой гибкостью. Судя по всему в результате появится некоторый гибрид этих подходов. Например, по некоторому запросу будет выбираться и формироваться единый документ с которым уже программа и будет работать. В данный момент, в одном из экспериментов, в программе хранится 4.5 тысяч объектов и им задано уже более 15 тысяч характеристик. С этим количеством большая часть функций программы на современном компьютере справляется легко.

3.6. Поиск

Поиск данных в программе можно осуществлять множеством способов. Во-первых, очевидно, что можно искать объекты непосредственно по характеристикам. Можно строить логические утверждения из характеристик и находить все объекты, удовлетворяющие условию.

Со временем, в программе накапливается довольно много знаний об отдельных объектах и это позволяет сравнивать объекты и находить похожие.

Кроме того, предусмотрен механизм “фокусировки” на некоторых свойствах и наборах свойств, что позволяет быстрее и эффективнее выделять объекты, обладающие интересующими свойствами.

3.7. Поддержка большого числа пользователей

В программе предусмотрена поддержка работы нескольких пользователей сразу. Осуществляется это за счет того, что происходит запоминание, какой именно пользователь дал характеристику объекту. Соответственно есть возможность переключать активного пользователя и есть возможность смотреть заполнение папок другим пользователем.

4. ПРИНЦИПЫ РАБОТЫ С ПРОГРАММОЙ

4.1. Общая технология процесса накопления

Накопление информации в программе сейчас может происходить в разных режимах. По сути, всё в программе направлено на то, чтобы любые знания пользователя об объектах запоминались и повсеместно использовались. Потому есть разные способы вносить новые знания в программу. Самый примитивный – простое копирование объектов из папки в папку, аналогичное копированию файлов в обычном Windows Explorer. Но это, так скажем, самый низкоуровневый, примитивный и неудобный способ. Им не обработать несколько сотен объектов – слишком много лишних действий нужно произвести. Поэтому для обычного описания объектов используются специальные инструменты.

Простейший из них позволяет быстро описывать характеризуемый объект. В процессе ввода названий характеристик, он автоматически сужает выбор имеющихся до тех, что соответствуют вводу, и потом характеристика присваивается одним нажатием клавиши Enter. Таким образом, чтобы присвоить десяток описаний, нужно только набирать уникальные части их названия и давить Enter – никаких лишних действий.

Эксперементальные данные:

В результате непринужденного анализа новых музыкальных поступлений в течение двух месяцев накопилась база из 4478 объектов, 376 из них являются характеристиками. Общее число назначенных характеристик ("файлов" в системе) 17.500. Используемость характеристик:

Кол-во объектов в характеристике	Кол-во характеристик
200<=n	6
100<=n<200	36
50<=n<100	41
20<=n<50	96
10<=n<20	63
n<10	134

Если считать объекты рекурсивно, получим следующую таблицу:

Кол-во объектов в характеристике	Кол-во характеристик
2000<=n	11
1000<=n<2000	9
500<=n<1000	22
200<=n<500	42
10<=n<20	46
50<=n<100	55
20<=n<50	84
n<20	107

При этом, 44 объекта имеют более 30 собственных характеристик, 122 от 20 до 30, 411 от 10 до 20, 2739 вообще не имеют собственных характеристик, однако могут иметь унаследованные.

Далее возникают интересные моменты. Во-первых, это набор характеристик, накопившихся в системе. Что он из себя представляет и как изменяется? Это, пожалуй, самый интересный вопрос. Во-вторых, как увязать характеристики между собой, чтобы тратить минимум времени на присвоение взаимосвязанных характеристик и, чтобы основная задача программы удовлетворялась наилучшим образом?

4.2. Эволюция списка характеристик

Это, пожалуй, самый интересный вопрос в поставленной задаче. Дело в том, что субъективные описания напрямую соответствуют лексикону, используемому человеком для описания объектов. Вместе с эволюцией характеристик в программе, происходит и эволюция лексикона. Теряются ничего не значащие слова, и удобные и несущие смысл, находятся синонимы, обнаруживаются неожиданные следствия. Часто оказывается, что для некоторой обширной и значащей группы объектов нету подходящего слова и его требуется ввести.

На начальной стадии работы список характеристик пуст. Имеющийся опыт показал, что наиболее эффективно для начала описать некоторое кол-во объектов, создавая при этом новые характеристики. Т.е. требуется просто брать и описывать объект за объектом всеми словами, которые приходят в голову. Эти определения будут накапливаться в специально отведенной папке.

Когда характеристик наберется достаточное количество, стоит изучить их. Среди них наверняка окажутся нелепые, ничего не описывающие, их можно будет смело удалить. Наверняка окажется, что разные оттенки одного и того же представлены в виде нескольких характеристик. В этом случае стоит завести обобщающую характеристику и описать ей оттеночные.

Одна оттеночная характеристика может попасть в несколько обобщающих. Например, абстрактное описание “светлый” по отношению к музыке одновременно описывает и настроение и звучание.

Таким образом, из набранных характеристик на отдельном этапе работы выстраивается сеть. Эта сеть даёт серьезные приимущества, по сравнению с простым набором характеристик, поскольку в ней, принадлежность одной характеристике может означать принадлежность сразу еще десятку, и наоборот, может означать, потенциальную принадлежность еще небольшому числу характеристик, которые в дальнейшем будут выводиться на экран в виде подсказки. Это очень облегчает процесс накопления характеристик объектов, при этом не ограничивая пользователя в действиях (подробности и иллюстрации в приложении).

На данный момент, наиболее эффективным способом накопления видится такой: новые объекты описываются адекватными им характеристиками, после чего, время от времени, происходит “фаза сна”, когда стоит перестать заниматься новыми объектами, а нужно разобраться с накопленными характеристиками. Перестроить связи между ними, проверить содержимое новых характеристик – адекватно ли оно первоначальному замыслу.

Потом стоит поискать новые, неопределенные описания, на основе имеющихся. Часто оказывается, что крупная группа объектов одинаково описывается сразу многими характеристиками. Это позволяет создать под них отдельную характеристику и разгрузить папки, в которых объекты лежали ранее. Частично новые характеристики находятся на основе здравого смысла. Понятно, что “гитары”, “скрипки” и “духовые” можно объединить в категорию инструменты, что даст возможность легко выйти на них. Но часто здравый смысл несколько запутывается в перемешанном обилии различных папок и требуется возможность поиска характеристик, которые, возможно, имеет смысл группировать. Для этого уже есть и продолжают разрабатываться методы поиска групп похожих объектов.

В результате работы одного из них была получена следующая картина: было понятно, что некоторые объекты довольно стандартны и относятся к некоторой группе интереса, которая однако не имеет собственной характеристики. С помощью соответствующего метода был получен набор объектов близких к указанной группе:

Из них был построен цветовой коврик и его столбцы были отсортированный по использованности:

Полученная картина сразу прояснила, каким образом можно описать группу интересующих объектов: выяснилось, что все объекты обладают некоторым единым набором свойств, после чего была создана новая характеристика и объекты были добавлены в неё. Это позволило оптимизировать имеющуюся сеть характеристик и получить новую группу. Теперь многие новые объекты достаточно описать единственным словом, чтобы получить довольно полное его описание.

Числовых данных по динамике использования характеристик, к сожалению, на данный момент нет, т.к. не был предусмотрен соответствующий механизм сохранения временных меток. Но в скором времени данные начнут появляться.

4.3. Поиск

Поиск является одной из основных функций программы и работы со знаниями вообще. Он должен быть максимально гибок, разносторонен, присутствовать повсюду и в разных формах. Он служит постоянной подсказкой новых характеристик и похожих объектов. Это уже не отдельный инструмент, это концепция, которая пропитывает каждый элемент интерфейса и программы – везде должна быть возможность быстро найти желаемое: в папках, в списках, в вопроснике.

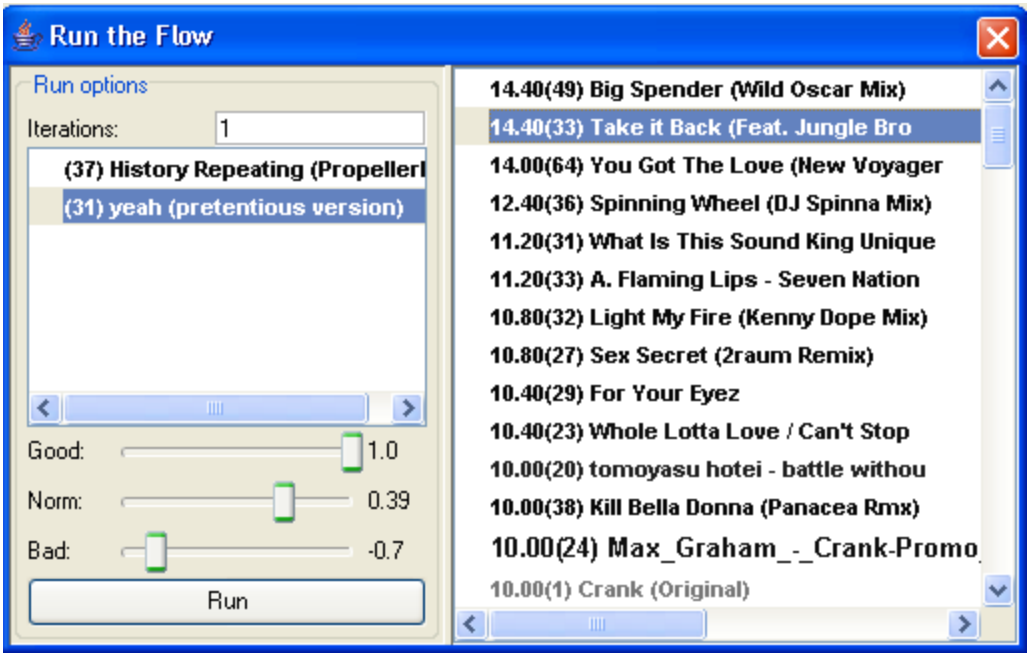

Рис. 1. Результат работы одного из алгоритмов поиска групп

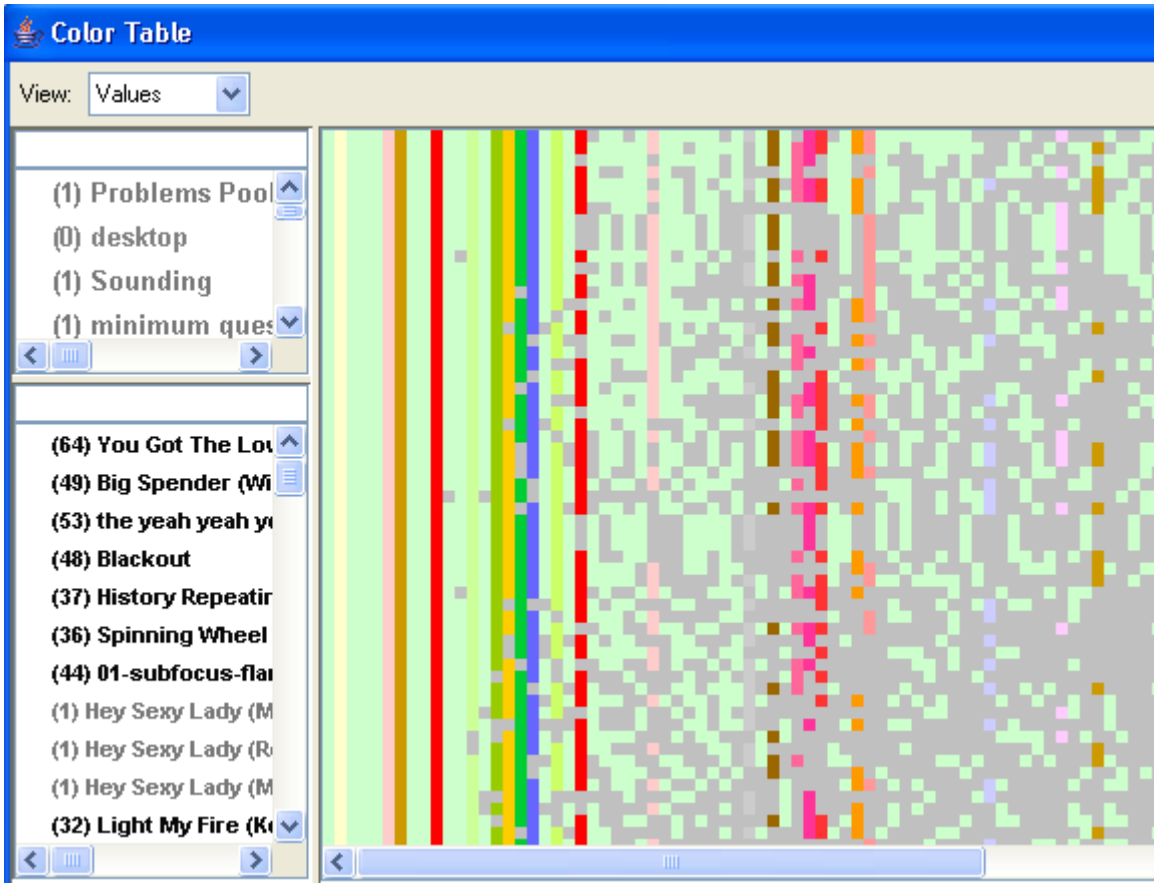

Рис. 2. Цветовой коврик построенный на материале поиска

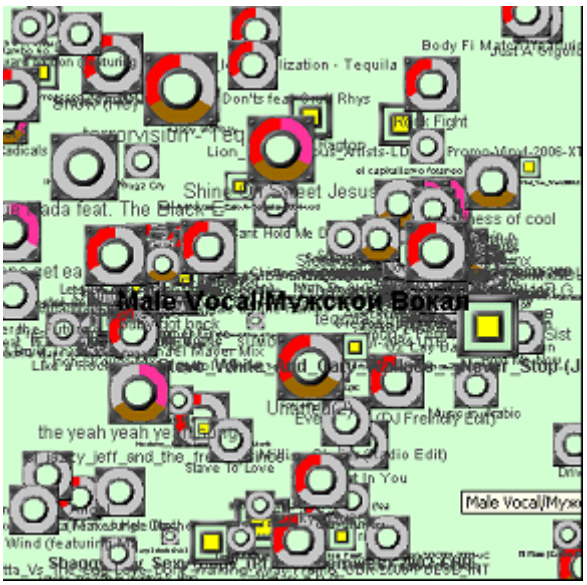

Рис. 3. Содержимое папки с фокусировкой на интересующих характеристиках

Для осуществления поиска интересующих объектов в программе есть несколько механизмов.

1. сама по себе организация хранения данных в программе обеспечивает наиболее быстрый интуитивный поиск нужного объекта.

Раньше на подборку 3-4 часов музыки определенной тематики у меня уходило от часа до трех. Сейчас 15-20 минут.

1. Везде, где присутствуют длинные списки объектов/характеристик, их можно найти простым вводом части названия.
2. Есть набор всевозможных диалогов, позволяющих увидеть полное описание каждого объекта, и быстро перейти в соответствующую папку.
3. Есть общий механизм оценки похожести объектов. В данный момент он действует по хемминговой мере с весами, соответствующими информативности факта принадлежности или непринадлежности папке. С помощью этого механизма можно получать списки объектов, отсортированные по похожести на заданный объект, что позволяет быстро видеть список похожих. Это один из фундаментальных инструментов, т.к. с его помощью легко оценивается качество описаний приписанных объектам. Если похожие по мнению программы объекты не являются похожими в действительности – стоит добавить характеристики описывающие разницу между ними, или же просто более подробно описать имеющимися.
4. Есть возможность быстрого получения списка объектов, позволяющий строить из характеристик логические фильтры.
5. При просмотре содержимого папки, есть возможность сфокусировать внимание на интересующих объектах, указав присущие им характеристики. Объекты обладающие указанными характеристиками увеличиваются в размерах, не обладающие – уменьшаются. Эффект – очевиден, наиболее подходящие объекты сразу становятся видны:

Кроме того, есть механизм, позволяющий вообще спрятать самые маленькие объекты.

1. Во всех случаях, где объекты каким-то образом сравниваются или оцениваются, есть возможность ограничить диапазон используемых характеристик, указав, с какими нужно работать. Аналогичным образом, можно приписывать характеристикам веса.

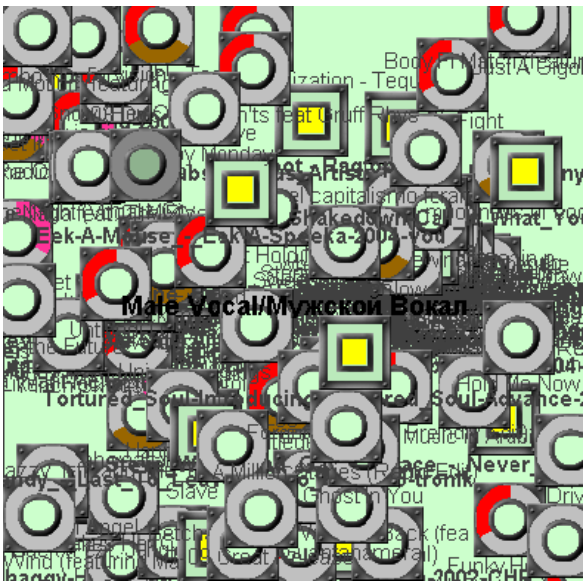

Рис. 4. Содержимое папки без фокусировки на интересующих характеристиках

4.4. Механизм оценки объектов

Поскольку, программа о самой природе объектов ничего не знает, всё что она может использовать – это характеристики объектов. Для какой-либо абсолютной оценки - данных маловато, потому оценка в данном случае сравнительная. Задача оценки ставится, как оценка похожести объекта на другие объекты и удовлетворения им набора заданных критериев. Иначе говоря, критерии для оценки задаются пользователем и строятся эти критерии из сравнений объектов и принадлежности объектов каким-то папкам.

Наиболее простым критерием может быть: “объект должен принадлежать папкам “быстрые” и “сильные””, наиболее сложным на данный момент является “должен принадлежать папкам “быстрые”, “сильные” и быть максимально похож на A и B”. В перспективе критерии могут быть более сложные, например, можно будет задавать в какой именно области объекты должны быть похожи.

Сама оценка также может использоваться разными способами: от самого простого – поиска по критериям, до сложных – поиск совокупностей объектов, удовлетворяющих некоторым абстрактным выражениям (например, можно нарисовать требуемый цветовой коврик и поставить задачу подобрать объекты так, чтобы они давали как можно более похожий узор). Например, при подборе сотрудников в новый отдел, может стоять задача найти людей схожих в каких-то психологических характеристиках и обладающих некоторыми профессиональными навыками. Эта задача несколько сложнее задачи простого профподбора, поскольку критерий выбора зависит не только от отдельно взятого человека, а от всех выбираемых сразу.

В случаях работы с музыкой, например, при создании танцевального сета, желательно отсортировать песни, например по энергетике и скорости, но при этом максимизировать похожесть соседних песен по некоторым характеристикам. Потенциально, программа может решить эту задачу и представить несколько подходящих вариантов на выбор. Возможен и более интересный вариант – программа может не предлагать несколько готовых вариантов, а построить структуру, соответствующую возможному развитию сета. Т.е. после каждой песни предлагать несколько вариантов следующей и в зависимости от выбора показывать следующие подходящие варианты. По сути, такой результат строится на основе имеющегося в программе “файлового” интерфейса.

Что касается самого механизма сравнения объектов – тут есть варианты. Понятно, что для сравнения в первую очередь используется соседство объектов в папках – если два объекта принадлежат одной папке, то они более похожи. В контексте отдельно взятых задач, некоторые папки могут иметь больший вес, некоторые меньший, или вообще не иметь значения. Это должно задаваться.

4.5. Изучение

Изучение множества объектов частично сводится к поиску, частично к анализу. Основные элементы анализа объектов: сравнение и различный поиск групп, категорий, похожих пар объектов/групп объектов/папок. Эта часть работы пока не готова, практически есть возможность только поиска похожих групп объектов. Это одна из самых необходимых на первичном этапе операций, т.к. помогает правильно организовывать данные – вовремя обнаруживать конфликтующие или нечеткие понятия. Напомню, что, пожалуй, самой сложной задачей, стоящей перед пользователем, является подбор подходящих категорий для обозначения папок. Категории должны быть емкие, максимально четкие и значащие и их должно быть как можно меньше, чтобы не совершать лишнюю работу по их заполнению, и чтобы не путались перед глазами.

Основная функция категорий – выделять то, что выделяется, и разделять то, что непохоже. В то время, как выделением пользователь занимается практически постоянно, характеризуя объекты, разделению понятий уделяется меньшее внимание. Потому важно уметь увидеть, какие объекты считаются в системе похожими – это помогает обратить внимание на то, каких категорий в программе не хватает. Довольно удобным инструментом, в этом случае, являются самоорганизующиеся карты. Можно запустить самоорганизацию объектов в папке, и похожие встанут рядом. Если полученная картина соответствует желаемым представлениям – всё нормально. Если нет – значит, внимание при создании характеристик уделялось чему-то не тому и нужно лучше продумать, что стоит выделить, чтобы непохожие объекты раздвинулись подальше друг от друга.

4.6. Перспективы

Планируется более детальная проработка возможностей анализа накопленной коллекции, улучшение работы с цветовым ковриком и его более глубокая интеграция в интерфейс. А так же проработка механизма сравнения: добавление возможностей задания наиболее интересующих характеристик, их весов, создание единой системы сравнения, на основе которой будут работать все соответствующие части программы.

После этого планируется проработка возможности многопользовательской работы с коллекциями: работа нескольких людей с единым множеством категорий и объектов, анализ их работы, а так же механизм слияния нескольких коллекций.

5. ПРИЛОЖЕНИЕ

5.1. Интерфейс

Интерфейс такой программы решающим образом важен, потому что он дает некоторые принципиально новые возможности, и если интерфейс не окажется простым и ясным – проще будет отказаться от работы с программой, чем пытаться понять, что же она делает. Тем не менее, поскольку программа пишется больше для проверки технологии, чем для коммерческого применения, особенности интерфейса продуманы чисто схематически. Никакой дизайнерской работы проделано не было.

Во-первых, чтобы интерфейс сложной программы был понятен в целом, он должен состоять из похожих элементов. По сути, количество основных рабочих элементов удалось свести всего к нескольким: списки объектов, списки их экземпляров, папка и несколько специальных окон. Практически всё построено на основе этих элементов. Ну и, конечно, стандартных полей ввода текста, различных кнопок и ползунков.

В процессе работы с программой постоянно приходится добавлять объекты в различные папки, листы и списки. Это, можно сказать, основное действие совершаемое в программе. Оно совершается по технологии drag&drop. Добавить объекты в программу проще всего, тоже перенеся в программу файл из системы.

Еще одна особенность интерфейса в том, что одну и ту же информацию мы можем видеть одновременно в разных объектах (принадлежность к папке отображается цветом, например). Если она меняется, то все соответствующие элементы интерфейса должны обновиться. Если изменяется принадлежность объекта характеристике, это может повлиять на половину отображаемых в программе листов сразу. Это привело к необходимости создания системы событий внутри программы и некоторому замедлению работы программы в случае присутствия большого числа объектов на экране. Однако, эти проблемы вторичны и пока игнорируются.

5.2. Списки

Списки в программе сделаны универсально и используются повсеместно. Универсальность обеспечивается в основном несколькими моментами:

1. Добавление фильтров. Изначально список содержит все объекты в системе, но отображаются только нужные. Выбор осуществляется программируемыми фильтрами. Настройки фильтра могут динамично изменяться.
2. Сортировка списка. Можно гибко настраивать сортировку списка.
3. Pop-up меню на элементах списка позволяет делать всё необходимое с каждым объектом.
4. Строка поиска. Наверху списка можно расположить строку поиска по списку.
5. Список поддерживает drag&drop. Из него можно вытащить элемент и в случае необходимости бросить элемент в список.
6. Настраивается отображение элементов списка. Например, стандартная настройка изменяет яркость цвета элемента в зависимости от его веса.

1. Кроме того, в списках экземпляров объектов часто бывает полезным вывод более полной информации:

5.3. Папки

Интерфейс папки предоставляет возможность изучения файлового пространства. Он содержит список объектов, лежащих в текущей папке, показывает поле папки с разложенными в ней объектами. Кроме того, сверху располагается панель с инструментами, позволяющая переключать текущего пользователя, шагать на папку вверх, возвращаться в предыдущую папку, идти в следующую, менять режим скалирования (локальный/общий), менять размер шрифта подписей, увеличивать/уменьшать поле папки, изменять режим отображения объектов, сортировать объекты на поле папки.

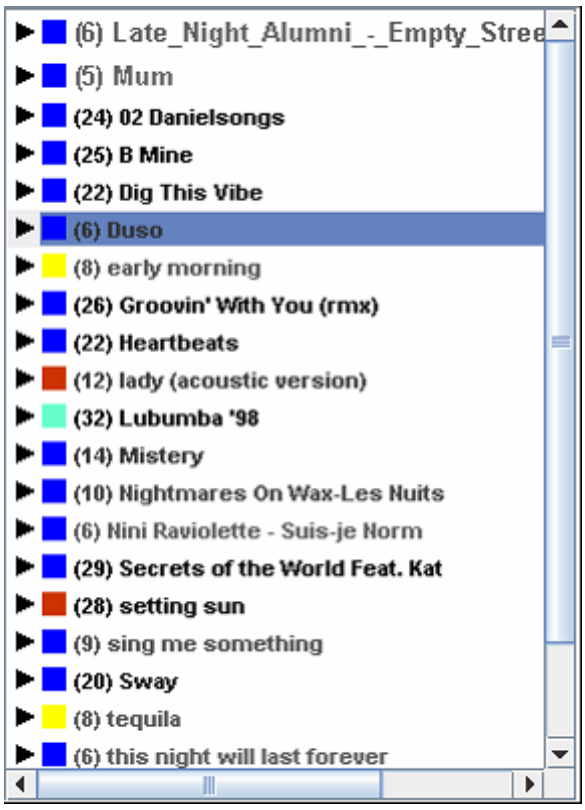

Рис. 5. Пример списка содержимого папки с отображением дополнительной информации

5.4. Окно конфигураций

В программе в режиме “always on top” присутствует окно настроек. Сейчас в нем есть две закладки, одна для настройки скалирования, другая для Ring Info.

Базовая настройка скалирования осуществляется тремя ползунками:

1. общий коэффициент размера иконок
2. коэффициент влияния “демонов Селфриджа”
3. граница исчезновения маленьких объектов

Кроме этого, тут же задаются данные для “демонов Селфриджа” - указываются хорошие и плохие критерии, их веса и настройки.

Настройка Ring Info, это просто список ключевых характеристик с демонстрацией их места на цветовом кольце. Кроме того, здесь есть возможность регулировать ширину этого кольца.

5.5. Окно информации

Это окно тоже всегда в режиме “always on top” и в нем три закладки: информация об объекте под курсором, список открытых окон и список лучших на данный момент объектов по “демонам Сэлфиджа”.

Особое значение имеет информация об объекте под курсором, т.к., во-первых, она отображается для любых объектов, находятся они в папке, в списке, на цветовом коврике или меткой в вопроснике, во-вторых, на этот объект завязаны горячие клавиши, вызывающие различные диалоги. Например, диалог для быстрой установки характеристик.

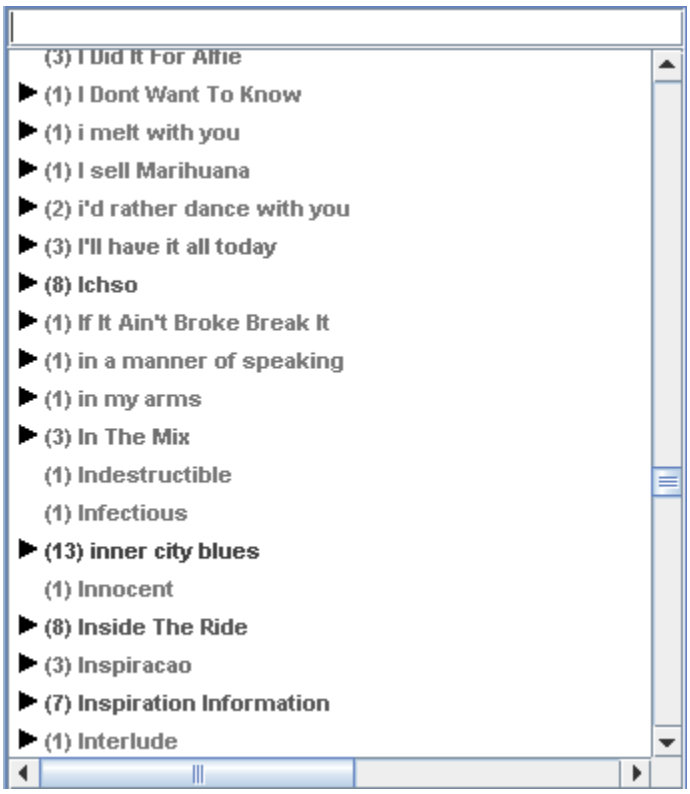

Рис. 6. Пример списка объектов

Рис. 7. Пример списка элементов с информацией о их расположении

5.6. Диалог быстрой установки характеристик

Это одно из ключевых мест программы, т.к. с ним идет большая часть работы. Диалог вызывается для конкретного объекта из всплывающих меню или по нажатию ctrl+B для объекта из информационной панели (объекта под курсором).

В этом окне сверху отображается название объекта, снизу его характеристики, а в середине находится интерфейс для задания характеристик. В его левой части находится список всех характеристик со строкой поиска, а справа, при выделении характеристики появляются её

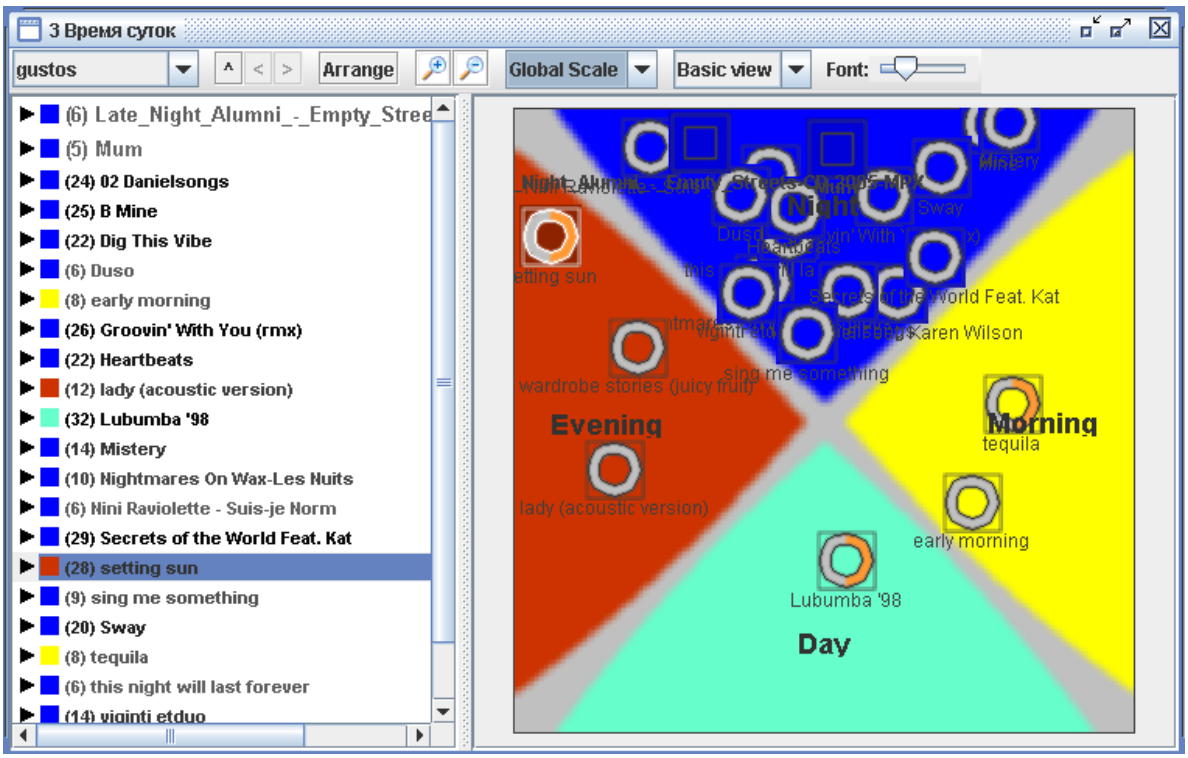

Рис. 8. Интерфейс папки

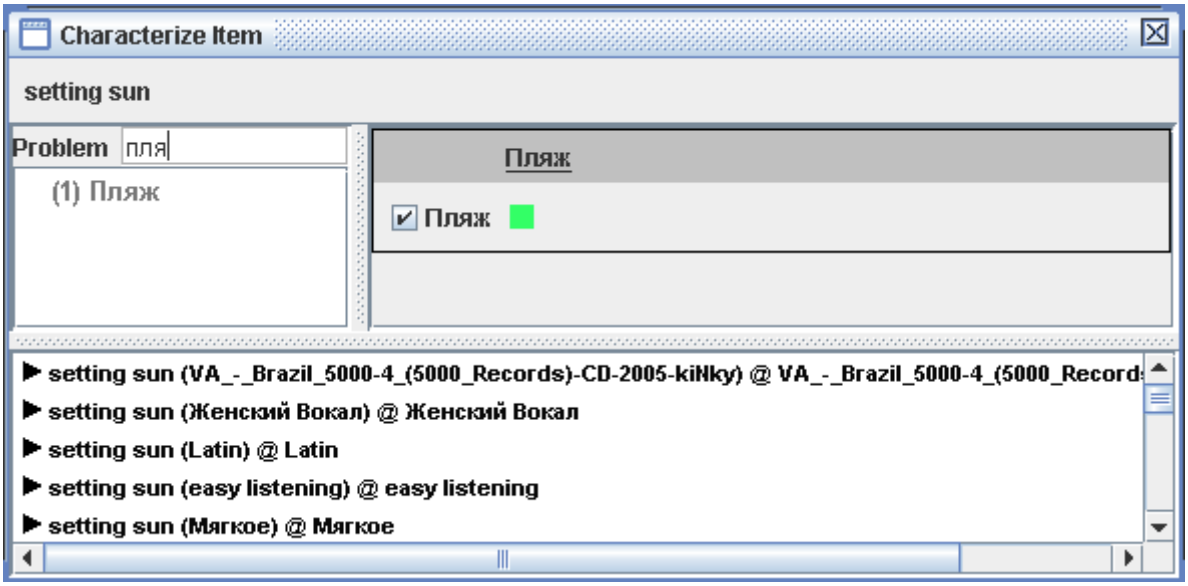

Рис. 9. Диалог назначения характеристик

зоны. Мышкой или нажатием клавиши enter в строке поиска переключается принадлежность объекта характеристике. Таким образом, для того, чтобы дать объекту несколько характеристик, достаточно набирать несколько букв этой характеристики, нажимать enter и начинать набирать новую характеристику. И, наконец, чтобы не утрачивать информацию о иерархии характеристик, в правой части диалога, в случае принадлежности объекта характеристике, раскрываются её подхарактеристики:

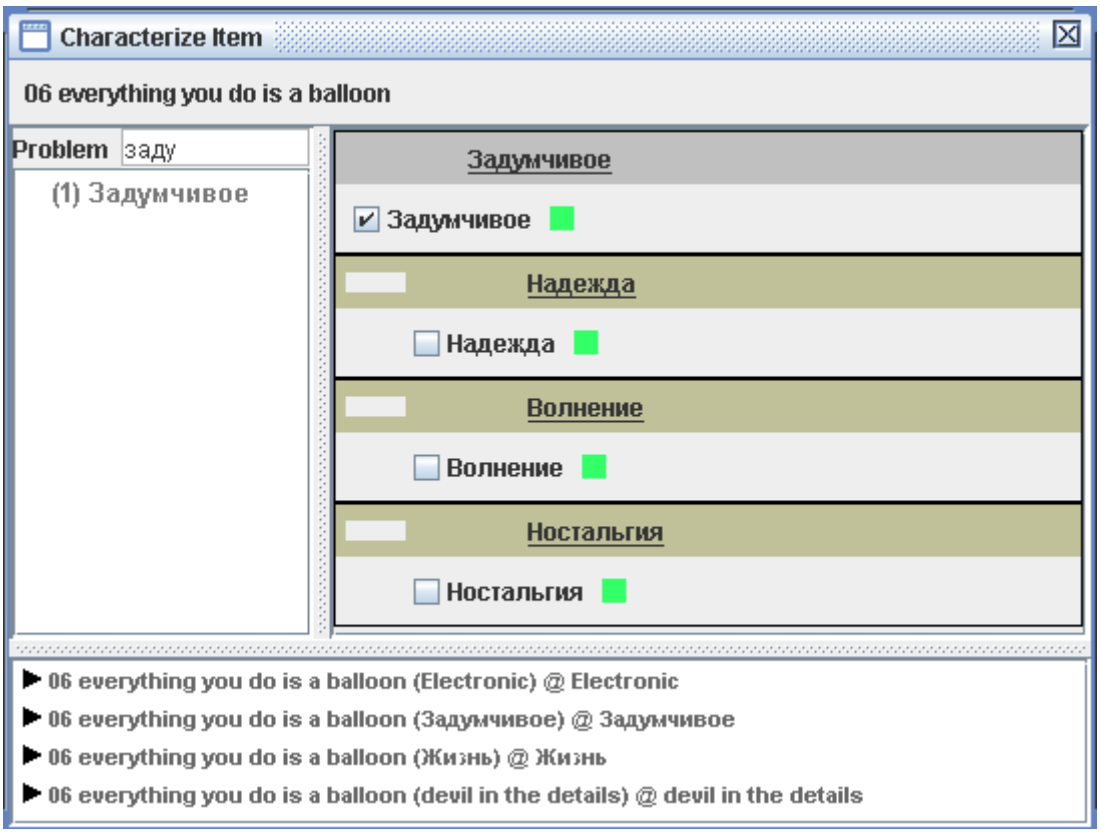

Рис. 10. Диалог назначения характеристик: подхарактеристики

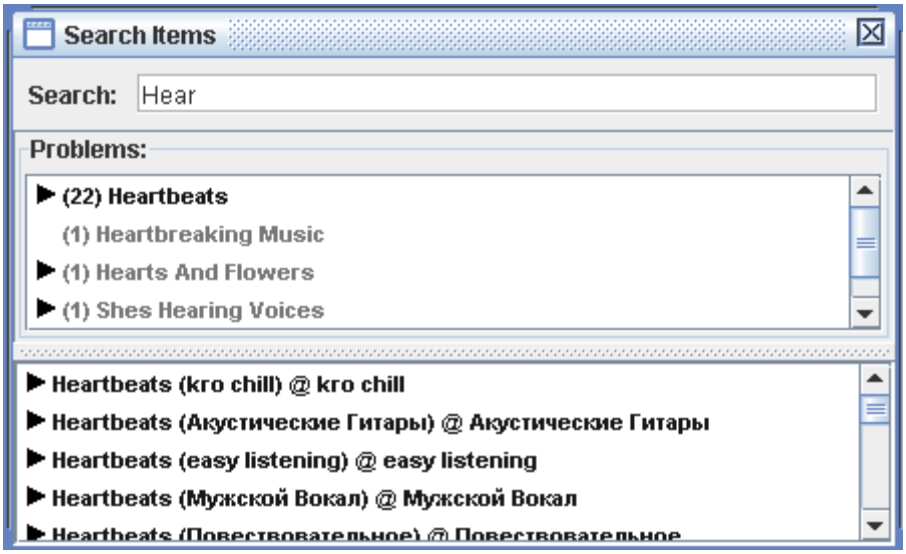

Рис. 11. Диалог поиска

5.7. Диалог поиска

Диалог вызывается комбинацией ctrl+S или же из всплывающих меню объектов. Он позволяет искать объекты, вводя части их имени. При этом в списке остаются только подходящие объекты и для выделенного объекта отображаются его характеристики.

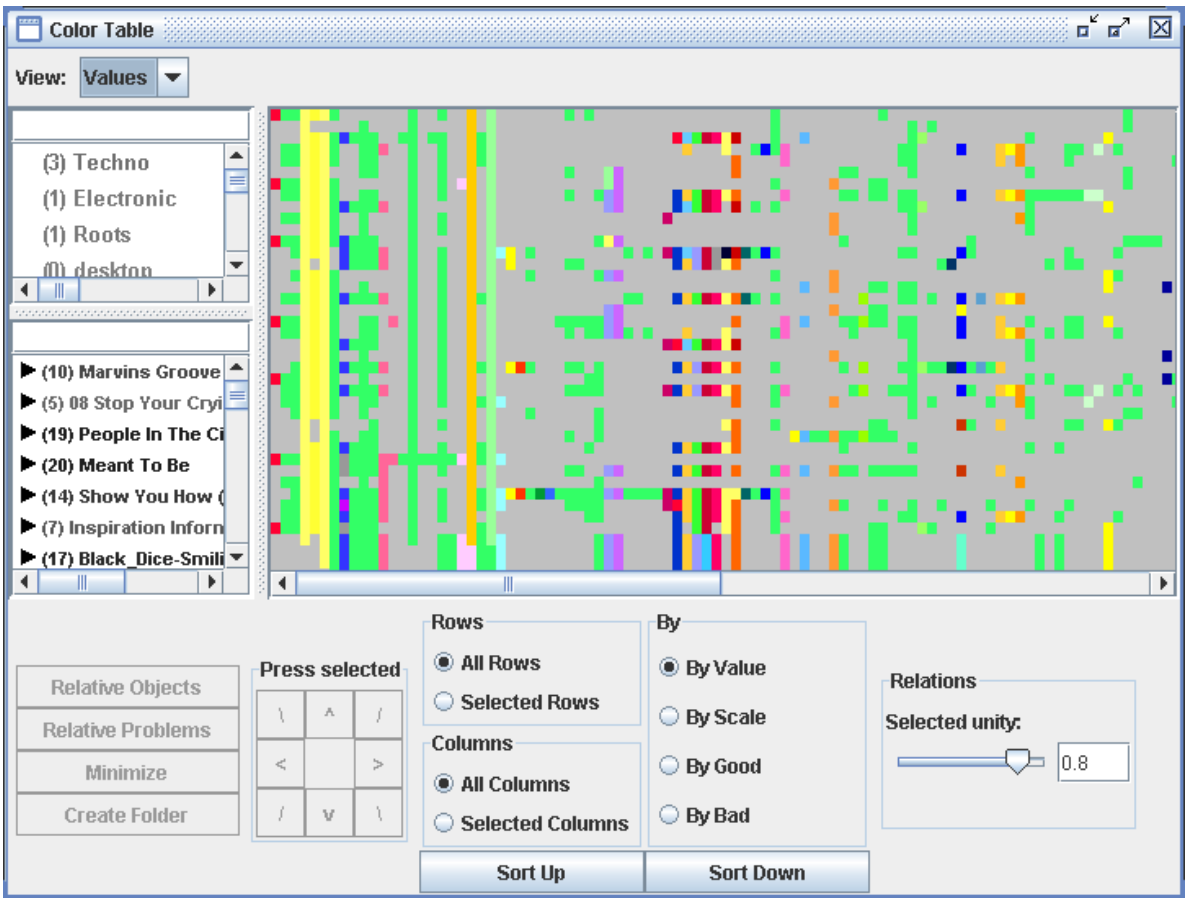

Рис. 12. Интерфейс цветового коврика

Диалог очень удобно использовать, например, для открытия вспомнившейся характеристики. Нажимаем ctrl+S, набираем часть названия характеристики достаточную для идентификации и открываем её. Быстро и просто.

5.8. Цветовой коврик

Цветовой коврик – наиболее насыщенный информацией элемент интерфейса. Это табличка, в которой объекты задают строчки, а столбцы – характеристики. И в ячейках цветом указывается принадлежность характеристикам. Таким образом, вся интересующая информация о наборе объектов оказывается сразу перед глазами, притом в простой и понятной форме – форме таблички.

Но, понятно, что наблюдать коврик, на котором объекты и характеристики разбросаны случайным образом, - довольно бестолково. Интересно иметь возможность сравнить несколько объектов или характеристик, для этого нужно иметь возможность расположить их рядом. Поэтому строчки и столбцы коврика можно свободно перемещать. Кроме того, для удобства предусмотрены операции, позволяющие переместить все выделенные строчки/столбцы в любой край таблицы. С помощью этих операций и гибких возможностей выделения строк/столбцов можно легко группировать интересующие объекты и характеристики вместе, сравнивать их, видеть взаимосвязи и находить группы похожих.

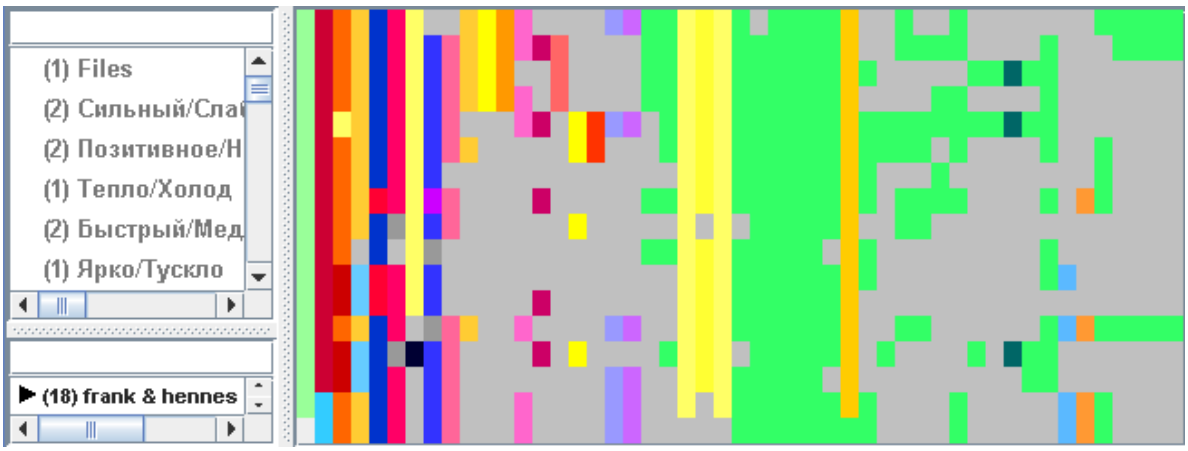

Рис. 13. Похожие элементы на цветовом коврике

Для удобства добавлены еще несколько операций, таких как выделение “похожих” строчек и столбцов (похожесть оценивается линейно и граница отсечения задается) и сортировка объектов по отдельной характеристике.

СПИСОК ЛИТЕРАТУРЫ

1. Raskin J. *The Humane Interface*, ACM Press, 2000.
2. Бонгард М. М. *Проблема Узнавания*. М.: Наука, 1967.
3. Боровиков В. *STATISTICA. Искусство анализа данных на компьтере: Для профессионалов*. СПб.: Питер, 2003, 2-е изд.