

Анализ связанных объектов и визуализация результатов

Попов И.В.
ООО “Твинком бизнес”
popoff@rubricon.com

Фролкина Н.А.
ООО “Твинком бизнес”
nfrolkina@rubricon.com

Предлагается механизм для анализа и визуализация связанных между собой объектов (Linked objects mining). Для автоматизированного выделения групп взаимосвязанных между собой объектов из их большого количества (несколько десятков тысяч) одновременно использованы методы из теории графов: поиск в глубину, поиск в ширину, поиск кратчайшего пути, выделение сильно связных компонент, операции над множествами: объединение, пересечения и разность а также методы кластерного анализа. На конечном этапе полученные результаты могут быть уточнены при помощи методов ручного увеличения и уменьшения количества объектов в группе, при этом результат группировки и степень влияния каждого объекта оценивается при помощи параметров качества. Все объекты и связи между ними визуализируются как точки и дуги на сферической поверхности, при этом достигается простота и наглядность представления результатов анализа.

Создан инструментарий для разработчика (SDK) и на его основе демонстрационная программа для анализа и визуализация связанных между собой объектов, информация о которых и результаты анализа представлены в формате XML.

Предлагаемый инструмент может быть полезен для расширения результатов поиска поисковых машин, для визуализации и анализа тезаурусов, в задачах анализа индексов цитирования, в системах документооборота, при анализе текстов, в криминалистике, в патентном поиске, в экономическом анализе и в других аналитических исследованиях, где присутствуют связанные объекты.

Постановка задачи

Современные компьютерные технологии позволяют накопить за короткий промежуток времени гигантский объемы информации. Для обработки этой информации в последнее время появились решения, известные под общим названием “Data mining”. Эти решения позволяют обнаруживать в сырых численных данных при помощи математических методов (моделирование, прогнозирование, кластеризация, классификация и т.д.) ранее неизвестные нетривиальные практически полезные и доступные для интерпретаций новые знания. Как правило, методы Data mining оперируют однородными данными нескольких

типов (покупатели – товары, история торгов - цены на акции и т.д.). С появлением интернета и систем электронного документооборота все большее количество информации стало храниться в текстовом виде, Это привело к появлению решений для обработки текстовой информации – Text mining. Такие системы осуществляют при помощи лингвистических методов следующие действия: тематический поиск в текстах, классификация документов, ответ на запросы, тематическое индексирование документов, поиск по ключевым словам, выявление объектов и связей между ними, реферирование и т.д. В качестве примера таких решений можно привести продукты компании «Гарант-Парк-Интернет» (RCO) или систему PolyAnalyst фирмы Megaputer Intelligence (<http://www.megaputer.ru>). Крупные СУБД, такие как ORACL или MS SQL Server имеют свои встроенные инструменты для Data mining и Text mining .

Однако существует класс данных, который по своей природе располагается “между” данными для Data mining и данными для Text mining. Речь идет об разнородных объектах, связанных между собой связями различного происхождения и различной силы. Эти данные могут существовать как данность (библиографические ссылки в научных статьях, тезаурусы, ссылки между сайтами, log-файлы сотовых операторов или операторов электронной почты и т.д.), а могут являться результатом деятельности инструментов Text mining (выявление объектов и связей между ними из документов). Методы обработки таких данных используют математический аппарат, отличный от средств Data mining, поскольку он работает с множественными совокупностями объектов, а не с единой массой объектов. Наиболее подходящими средствами для такого рода задач можно назвать методы из теории графов и теории множеств.

Задачу обработки таких данных можно сформулировать следующим образом: выявление неизвестных нетривиальных практически полезных и доступных для интерпретаций групп объектов из общей совокупности разнородных объектов, связанных между собой связями разной силы и разных типов а также выявление аналогичных связей между этими группами, а также между группами и отдельными объектами. При этом обработку данных должны проводить эксперты в предметной области, а не специалисты в математике или вычислительной технике.

В дальнейшем методы для решения этой задачи будем называть Linked Objects mining или кратко LOMining.

Ниже приведено описание попытки создания инструмента для LOMining. Так как строение реальной системы, предназначенной для операций с данными, сильно зависит от структуры и объема данных, то перед началом описания системы необходимо уточнить эти параметры.

Изначально предполагается, что количество объектов, подлежащих обработке – несколько десятков тысяч, а количество связей для каждого объекта – несколько десятков, а для отдельных - и несколько сотен. Для меньшего количества объектов все методы, конечно, также работоспособны. Но для значительно большего количества объектов необходимы специальные действия, которые позволят производить операции за приемлемое время. Любые два объекта могут быть связаны связями нескольких типов. Каждая связь характеризуется силой действия. Каждый тип связи характеризуется коэффициентом изменения силы связи. Каждая связь может иметь или не иметь направление действия. Направление действия характеризуется своим коэффициентом изменения силы связи. Ядро инструмента для LOMining должна иметь минимальную зависимость от операционной системы и от типа СУБД, в которой хранятся данные об объектах.

Представление и хранение данных

Все объекты в обязательном порядке должны иметь следующие атрибуты:

- уникальный идентификатор
- название объекта
- вес, позволяющий сравнивать объекты между собой, т.е. позволяющий выполнять операции $>$ (“больше”) и $<$ (“меньше”) и операцию сложения.
- список связей с другими объектами

Кроме того, объекты могут иметь необязательные атрибуты:

- идентификатор типа объекта
- дополнительная информация об объекте В списке связей с другими объектами содержатся следующие параметры:
- идентификатор объекта, расположенного на другом конце связи
- идентификатор типа связи
- сила связи

Кроме того, необходимы описания типов связей и описание направлений связей, включающих в себя соответствующие идентификаторы и коэффициенты изменения силы связи.

Результаты работы должны храниться в виде списков групп объектов, состоящих из параметров:

- уникальный идентификатор группы
- название группы
- список идентификаторов объектов

Кроме того, необходимо сохранять параметры системы, при которых получены результаты, такие как настройка фильтров, значения коэффициентов и т.д.

Для хранения данных выбран формат XML. Этот формат не зависит от выбора платформы, а инструмент для преобразования данных в этот формат имеют все современные СУБД.

Обработка данных

Выполнение процесса LOMining можно представить в виде последовательного выполнения следующих действий:

1. Загрузка данных
2. Настройка параметров системы
3. Определение центральных объектов групп (если этого требует алгоритм из п.5)
4. Автоматическое выделение шаблонов групп
5. Формирование групп из числа шаблонов
6. Уточнения состава групп
7. Операции над группами
8. При необходимости повторение пунктов 2-7
9. Сохранение результатов

Загрузка данных

В начале обработки все исходные данные для облегчения манипуляций с ними загружаются в структуры специального формата – контейнеры. В качестве контейнеров используются контейнеры из библиотеки Standard Template Library (STL). STL реализована для большинства современных ОС и хорошо отвечает требованию независимости от выбора платформы.

Настройка параметров системы

Входное множество объектов и связей между ними составлено с максимально полной информацией о модели изучаемой сущности. Однако, при решении реальных задач такая полнота может сильно исказить желаемый результат. Так, хорошо известна теория о том, что двух случайно взятых людей отделяет друг от друга только шесть рукопожатий. И действительно, если учитывать все связи, которые могут связывать людей, то выяснится, что каждый житель Земли связан с Папой Римским достаточно тесно. Однако, если учитывать только родственные связи, то количество людей, близко связанных с Папой, резко уменьшится. Поэтому при решении задачи при помощи LOMining первым делом нужно произвести выбор типов объектов и типов связей, необходимых для ее решения. Для этой цели используются фильтры объектов и связей.

Автоматическое выделение шаблонов групп

Под шаблоном группы понимается совокупность объектов, близких друг другу по формальному признаку, предназначенную для дальнейшей обработки. Объекты могут группироваться как вокруг выбранного заранее объекта (центрального объекта группы), так и без него. В зависимости от природы изучаемой сущности и решаемой задачи шаблонами групп могут быть:

- Результаты поиска в ширину на заданное число ребер графа или на заданную длину пути
- Множества подобных объектов
- Компоненты (сильно связные, двухсвязные)
- Кластеры
- Базы (вершинные базы, антибазы и сильные базы)

Наиболее естественным способом создать шаблон группы – это дополнить центральный объект группы объектами, лежащими на заданном пользователем или меньшем расстоянии. Под расстоянием мы понимаем длину кратчайшего пути (shortest path). Для расчета расстояния и числа ребер от центрального объекта группы до всех объектов (особенно, если граф разреженный) целесообразно использовать модифицированный алгоритм Дейкстры с использованием очереди с приоритетами, реализованной в виде двоичной кучи или кучи Фибоначчи.

Для другого способа выделения шаблона группы введено понятие подобия объектов. Под подобием объектов здесь понимается схожесть топологии их связей с другими объектами. Для количественной оценки такой схожести предлагается два метода.

В первом методе необходимо вычислить матрицу расстояний графа $D(G) = (a_{ij})$. Затем для каждого объекта (с индексом m) вычисляется сумма рассогласований R_m элементов столбцов и строк с одноименными элементами строки и столбца центрального объекта

группы (с индексом c): $R_m = \sum |a_{ic} - a_{im}| + \sum |a_{ej} - a_{mj}|$. Сумма рассогласований R_m и будет оценкой подобия объектов. Другой метод заключается в использовании только тех объектов (назовем их объектами-кандидатами), которые имеют общие концевые вершины с центральным объектом группы. При этом рассматриваются не суммарные связи, а их

составляющие. Сумма рассогласований

$$R_m = \frac{\sum (F * k_{type})}{\sum F_{coin} + k_{precis} * \sum F_{uncoin}}$$

Здесь $\sum (F * k_{type})$ - сумма сил совпадающих связей между центральным объектом группы и его концевыми вершинами и объектами-кандидатами и их концевыми

вершинами с учетом понижающего коэффициента k_{type} . $k_{type} = 1$ при совпадении типов связей и $0 \leq k_{type} \leq 1$ при их несовпадении. $\sum F_{coin} + k_{precis} * \sum F_{uncoin}$ суммы сил совпадающих и несовпадающих связей с учетом коэффициента соотношения

полноты/точности. Если нам более важно совпадение связей, то $k_{precis} < 1$, в противном

случае $k_{precis} > 1$. Недостатком этого метода является малое количество объектов для формирования группы при малом количестве связей центрального объекта группы, а к достоинствам – малая величина затрат на быстроедействие, пропорциональное объединению связей центрального объекта группы и объектов-кандидатов.

В качестве усиления обоих методов можно использовать дополнительный учет совпадения или несовпадения типов объектов.

Для выделения шаблонов групп, особенно на первом шаге итерации – когда число объектов достаточно велико, можно использовать сильно связанные и двухсвязные компоненты. При этом для выделения не требуется наличия центрального объекта группы.

Алгоритмы кластеризации, использующиеся в LOMining, можно разбить на два больших класса, различающиеся по методике получения результата: иерархические методы и итеративные методы группировки [2 с.363].

Суть иерархических методов заключается в поиске и объединении наиболее близких в каком-то смысле объектов. В случае с LOMining наиболее естественной мерой близости является длина связи между объектами. В зависимости от способов выявления расстояний между кластерами, можно различить следующие методы [2]:

- Одиночная связь (метод ближайшего соседа)
- Полная связь (метод наиболее удаленных соседей).
- Невзвешенное попарное среднее (unweighted pair-group method using arithmetic averages)
- Взвешенное попарное среднее (weighted pair-group method using arithmetic averages).
- Невзвешенный центроидный метод (unweighted pair-group method using the centroid average).
- Взвешенный центроидный метод (weighted pair-group method using the centroid average).
- Метод Варда.

Для выделения кластеров после окончания процесса кластеризации нужно задать некоторый пороговый уровень сходства, на котором выделяется число кластеров. большее 1.

Метод иерархического кластерного анализа хорош для сравнительно малого числа объектов. Его преимущество в том, что можно проследить историю попадания в кластер

для каждого объекта. В отличие от иерархических методов, класс итеративных методов группировки в результате своей деятельности выдает не иерархическую, а плоскую совокупность кластеров. Основным методом этого класса является метод К-средних [2]. Для метода К-средних нет формального доказательства его сходимости, однако на практике он обычно сходится довольно быстро и поэтому может быть использован для начального кластерного анализа совокупности большого количества объектов.

Нахождение базы, антибазы и сильной базы. Базой (base of directed graph) является такое множество V вершин орграфа G , которое удовлетворяет следующим двум условиям [5]:

- каждая вершина графа G достижима хотя бы из одной вершины множества V
- в V нет вершины, которая достижима из другой вершины множества V

Определения антибазы (contrabasis) и сильной базы (power-basis), а также способы их получения можно найти в [5].

Например, если граф описывает структуру некой организации со связями, означающими подчинение одних членов другими, то базу V можно интерпретировать как “коалицию”, включающую наименьшее количество лиц, обладающих властью над каждым членом организации [5].

Формирование групп из числа шаблонов

После получения шаблонов групп пользователь выбирает из них те множества объектов, которые считает необходимым подвергнуть дальнейшей обработке. К таким множествам можно отнести множества объектов, которые интересуют пользователя для дальнейших исследований. И наоборот, в группу можно отнести объекты, которые мало интересуют пользователя. Такие группы можно свернуть в один объект, тем самым улучшив наблюдаемость оставшихся объектов. Начальный состав этих групп формируется из объектов, полученных на предыдущем этапе. В дополнение к созданным автоматически группам или можно добавить пустые группы, созданные вручную.

Уточнения состава групп

После создания списка групп пользователь может обрабатывать каждую группу в ручном и автоматизированном режимах.

Каждая группа имеет критерий качества инкапсуляции k_{inc} характеризующий “связность” объектов в группе. Его можно определять по формуле:

$$k_{inc} = \frac{\sum F_{int}}{N_{int} * \sum F_{extrn}} \text{ где } F_{int} - \text{силы связей между объектами внутри группы; } F_{extrn} -$$

силы связи между объектами внутри группы и объектами вне группы; N_{int} - количество объектов внутри группы. Для определения “важности” объектов введено понятие главного объекта группы. Главный объект группы – это такой объект, до которого определяются расстояния всех других объектов при определении их “важности”. Главный объект группы назначается пользователем, исходя из его задач. Для помощи в выборе главного объекта автоматически определяется ряд объектов в качестве кандидатов и ряд дополнительных параметров: диаметр и радиус графа, центр графа, центральная вершина графа, средняя вершина графа, центр тяжести графа. Для их определения будем рассматривать группу, как связный граф в качестве вершин которого используются объекты группы, а в качестве дуг – связи между ними.

Для каждой из вершин определяется эксцентриситет вершины (*eccentricity of a vertex*) - максимальное из расстояний до других вершин. Радиус графа – это минимальный эксцентриситет, а диаметр графа – максимальный. Множество вершин с эксцентриситетом, равным радиусу, называется центром графа, а сами такие вершины – центральными [3 с.35].

Для каждой вершины a можно также определить понятие среднего отклонения вершин

(mean deviation) m : $m(a) = \frac{\sum d(a,x)}{N}$ где $d(a,x)$ расстояние от вершины a до вершины x ; N – количество вершин. Вершина, для которой среднее отклонение вершин минимально, называется средней вершиной, а соответствующее значение – средним отклонением вершин графа. Аналогично, можно рассматривать сумму

$n(a) = \frac{\sum d^2(a,x)}{N}$. Вершина a с такой минимальной суммой называется центром тяжести графа [7, с.44]. При необходимости последнее выражение можно

модифицировать, учтя влияние весов $P(a)$ объектов, $n(a) = \frac{\sum P(a) * d^2(a,x)}{N}$ или

количество связей $L(a)$ у объектов $n(a) = \frac{\sum L(a) * d^2(a,x)}{N}$. Можно также учесть оба

фактора $n(a) = \frac{\sum P(a) * L(a) * d^2(a,x)}{N}$

Пользователь может выбрать главный объект группы из любой из множества центральных вершин, из центра тяжести графа (с вариациями), из средней вершины или назначить главным объектом любой объект группы. В последнем случае расстояние главного объекта до стандартных центров может служить показателем сбалансированности группы.

“Важность” T объекта a для группы можно определить по формуле:

$T(a) = \frac{\sum F_{int}}{D^n * \sum F_{extrn}}$ где F_{int} - силы связей объекта a с объектами внутри группы;

F_{extrn} - силы связи между объектом a и объектами вне группы; D – расстояние от объекта a до главного объекта группы; n – показатель важности учета расстояния. Если главный объект не выбран, то $n=1$.

Процесс ручной обработки

После того, как подсчитаны параметры группы и входящих в нее объектов, можно попытаться вручную улучшить качество группы. Для этого нужно добавлять объекты, улучшающие качество и удалять объекты, ее ухудшающие. Пользователь может добавлять и удалять объекты по одному или сериями. При работе с сериями из N объектов можно выделять объекты-кандидаты на добавление (удаление) по следующим критериям:

- N объектов, сильнее всего улучшающих инкапсуляцию группы
- N объектов, имеющих наибольшую (для удаления – наименьшую) “важность” для группы

- N объектов, имеющих наибольшее (для удаления – наименьшее) подобие с объектами в группе
- N объектов, наиболее (для удаления – наименее) близких к объектам в группе

Операцию добавления и удаления объектов можно выполнять многократно до достижения удовлетворяющего пользователя результата. Интересный эффект можно наблюдать при многократном попеременном добавлении и удалении объектов в группе без назначенного главного объекта. В определенных условиях все время могут добавляться одни объекты, а удаляться другие т.е. группа будет перемещаться по множеству объектов. Если же главный объект будет назначен, то группа будет изменять конфигурацию, оставаясь на месте.

Операции над группами

Операции над группами позволяют создавать новые группы из уже имеющихся групп. Операции над группами реализованы как операции над множествами объектов. Можно выполнять следующие типы операций

- объединение (union)
- пересечение (intersection)
- разность (difference)

В результате этих операций создается новая группа, содержащая элементы результирующего множества.

Для расширения возможностей для двух последних операций дополнительно может быть реализована возможность нечеткой реализации. Эта реализация состоит в задании пользователем порога чувствительности P операции, после превышении которого в результирующее множество добавляются элементы. Например, в операции пересечения в результирующее множество попадают элементы, имеющиеся в $P\%$ или более группах; в операции разности в результирующее множество попадают элементы, отсутствующие в $P\%$ или более группах

Сохранение данных

Результаты обработки также сохраняются в формате XML. Информация о группах может быть записана в двух видах:

- В виде списка объектов-членов групп и условий их получения (данные о настройке фильтров и параметров связей). Для каждого входного файла с данными таких файлов может быть несколько. Эта форма хранения результатов позволяет добавлять сохраненные результаты к имеющимся группам в процессе обработки текущего множества объектов и имеет смысл только вместе с исходным входным файлом данных.
- В формате входного файла с данными. Такой вид записи предназначен для дальнейшей обработки группы, как самостоятельного множества исходных данных.

Визуализация данных

Известно много способов отображения связанных между собой данных. Они могут отображаться в виде двумерных и трехмерных графов, в виде самоорганизующихся карт Кохонена (технология визуализации WebSOM), в виде различных геометрических построений, цвет и форма которых отражали бы характеристики отображаемых данных. Для выбора метода визуализации в LOMining необходимо учитывать следующие требования:

- Полнота изображения
- Информативность изображения
- Возможность навигации
- Возможность управления изображением

Для обеспечения этих требований весьма удобным оказывается отображение информации в виде графа. Для обеспечения требований к изображению можно использовать следующие критерии при визуализации графов [4, с.337]:

- Минимизация пересечений.
- Минимизация сгибов.
- Минимизация области размещения.
- Минимизация общей длины ребер и длины каждого ребра.
- Унификация длин ребер.
- Минимизация сгибов на ребре.
- Унификация сгибов
- Максимальная симметричность.
- Минимизация коэффициентов сторон.

Граф может изображаться на плоскости или в трехмерном пространстве. Двухмерное изображение достаточно просто для реализации и восприятия. Однако, при ее использовании имеется ярко выраженный центр изображения графа и периферийные области, при этом может возникнуть искажение восприятия при анализе взаимодействия объектов, лежащих в разных периферийных зонах.

Трехмерное изображение имеет ряд преимуществ перед двухмерным. Оно позволяет более гибко размещать элементы графа и вообще обойтись без пересечений ребер. С другой стороны, результирующие изображения становятся более сложными и становятся трудными для восприятия. Этот недостаток в какой-то мере компенсируется возможностью смены точки зрения пользователя путем поворота изображения. Но при большом количестве вершин и ребер достаточно сложно найти такой ракурс, чтобы в поле зрения находились все интересующие объекты.

В описываемой реализации предлагается компромиссный метод визуализации. Он заключается в построении двухмерного изображения графа на трехмерном изображении сферической поверхности. Такой подход позволяет построить изображение без периферийных зон.

Пользователь может вращать сферу так, чтобы в зоне просмотра оказывался интересующий его участок изображения. Для построения такого изображения использовался метод физических аналогий [4, с.246]. Он заключается в том, чтобы рассматривать граф, как систему тел с взаимодействующими силами. Будем считать вершины графа телами, имеющими электрические заряды одинакового знака, а ребра - пружинами. Сила отталкивания зарядов пропорциональна размеру объектов, а жесткость пружин пропорциональна силе связей. Пружина пытается растянуться или сжаться при ее отклонении от первоначальной длины, причем коэффициенты растяжения и сжатия пружины могут быть различными. В дополнение к этим силам предлагается добавить силу связи, которая притягивает объекты группы друг к другу. Метод пытается минимизировать энергию системы путем перемещения объектов под действием на них сил взаимодействия. Данная модель направлена на удовлетворение следующих требований:

- Силы пружин между соседними вершинами нацелены на то, чтобы расстояние приблизительно равнялись длине пружин.
- Электронные силы должны гарантировать, чтобы вершины не приближались друг к другу.
- Силы гравитации должны способствовать тому, чтобы объекты одной группы находились бы рядом.

Модель позволяет получать результаты, хорошо отвечающие большинству перечисленных выше критериев. Основной трудностью такого метода является его вычислительная сложность, которая пропорциональна квадрату количества объектов. Для ее уменьшения предлагается следующее решение: каждый объект взаимодействует не со всеми объектами, а только с объектами, которые отвечают хотя бы одному из условий:

- Объекты непосредственно связаны с данным объектом (между вершинами есть общее ребро)
- Объекты связаны с данным объектом через объект (между вершинами расположено два ребра и одна вершина)
- Объекты находятся в одной группе с данным объектом
- Объекты находятся с данным объектом на расстоянии, не превышающем заданное. Для вычисления этого условия удобно использовать хеш-таблицы по каждой из трех координат.

Очевидно, что в зависимости от начальных условий система может иметь конечное множество локальных минимумов своего энергетического состояния. Для операций, связанных с размещением объектов на плоскости удобно использовать алгебру кватернионов [1]. Для реализации графических функций использовалась библиотека OpenGL. Она наиболее полно отвечает требованию независимости от выбора платформы.

Реализация алгоритмов

В результате реализации вышеприведенных алгоритмов был создан инструмент для разработчиков (SDK), состоящий из трех частей:

- Набор функций для считывания данных из файлов с форматом XML и сохранение результатов работы в файлы с форматом XML, а также для формирования структур-контейнеров с этими данными.
- Набор функций для обработки данных при помощи описанных выше алгоритмов.
- Набор функций для визуализации результатов обработки.

SDK реализован в виде набора DLL на языке C++ в среде Borland Builder v.6. При помощи SDK написана демонстрационная программа, реализующая основные его возможности.

Возможные области применения и дальнейшее развитие LOMining.

Наиболее вероятными областями применения LOMining можно назвать следующие:

- Использование при вычислении релевантности и уточнение результатов поиска в поисковых машинах
- Дополнение результатов работы Text Mining в части последующей обработки выявленных объектов и связей
- Самостоятельная обработка данных в областях, оперирующих большим количеством разнородных связанных объектов в виде готовых наборов таких данных, а именно – патентоведение, экономический анализ, (физические и юридические лица, различные формы владения и различные формы связи), библиометрия (печатные труды, авторы, ссылочные и справочные материалы, издательства), криминалистика (преступники и преступные сообщества, жертвы, свидетели, орудия и методы преступлений, адреса, телефоны, вещественные доказательства), системы документооборота (авторы, документы, исполнители, руководители, организации).

Дальнейшее развитие LOMining можно осуществлять по направлениям:

1. Реализация прикладных программ LOMining для конкретных задач с уточнением потребностей конечных пользователей в предложенных методах.
2. Разработка новых методов LOMining
3. Интеграция средств LOMining со средствами DataMining, TextMining и другими средствами анализа данных
4. Попытаться ввести в набор методов LOMining временные характеристики объектов и связей (т.е. учитывать при анализе связей начало и окончание их действий) для более тонкого анализа данных.

Литература

1. Бранец В.Н., Шмыглевский И.П. Применение кватернионов в задачах ориентации твердого тела.- М.: Наука, 1973
2. Гайдышев И. Анализ и обработка данных. Специальный справочник.- СПб.: Питер, 2001
3. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов.- М.: Наука, 1990
4. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. - СПб.: БХВ - Петербург, 2003
5. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ.- М.: МЦИМО 1999
6. Кристофидес Н. Теория графов. Алгоритмический подход.- М.: Мир 1978
7. Оре О. Теория графов.- М.:Наука, 1980