This document is the property of Mohawk Software and is proprietary and confidential

Computer Generated Recommendations

User's Guide

Version 1.0.5

Mohawk Software
80 Clapp Street
Milton, MA 02186
www.mohawksoft.com
info@mohawksoft.com

Introduction

The Mohawk Software CGR system is a statistical analysis program and server intended to provide connection based recommendations for e-commerce web sites and retail stores

Components

The CGR system consists of several components:

CGR Server – A simple text or XML based server which calculates recommendations based on a set of input items.

CGR Calculator – A program designed to calculate the statistical relevance and interconnections of items in a database.

PLUGINS – A number of C++ software modules which can replace or augment the default behavior of of the CGR server and the CGR calculator without altering the main program.

Platforms

The CGR system can be configured to work on Microsoft Windows, FreeBSD, Linux, MacOS/X. Other platforms may be available upon request.

Requirements

The CGR system integrates with your existing e-commerce system in the form of a remote application server. Depending on the size of your customer database, the size of your product catalog, and the amount of activity on your site, a standard single or dual processor x86 system should be sufficient. If your site is extremely active, the CGR server can scale up to much more powerful hardware.

For very high loads and redundancy, multiple CGR servers can be run on different machines using the same information.

Concepts

In the simplest terms, the CGR system learns, from your customer base, the connections between items normally purchased.

Suppose you own a gourmet wine and cheese store where you sell fine wine, cheese, and breads. You notice over time that the customer that buys a single bottle of Chianti also buys a loaf of Italian bread. Seeing this, you start suggesting bread to people buying only the single bottle of Chianti. In doing so, you notice that 50% of these people ended up taking your recommendation and buying the bread

They simply didn't remember to buy it, or, even though you sell bread, it didn't “click” with them that they should buy bread from you. This is a common phenomenon in retail. Small store owners often remind a customer that they may have wanted additional products.

That's how CGR works, it is an age old, successful, technique of generating increased sales by using your knowledge of your customers. The only difference is that CGR can analyze your whole customer base and their history for these connections and make a recommendation in milliseconds.

Information

The CGR system requires only a list of customer purchases or behaviors defined in the terms of “owner” and “item.” For example, items could be products from your catalog, and the owners could be customers. Actually, they can be almost anything, as long as the owners can be used to declare connections between the items.

If an owner buys four items, then it can be said there is a connection between those items. Depending on various tuning parameters, the size and diversity of your customer base, and the number of products in your catalog, it may take many customers for statistically significant trends to become apparent.

CGR Calculator

The CGR calculator is the program which finds the connections and trends of various items in your catalog (like Chianti and Italian bread). It analyzes your customer purchasing habits and finds all the various connections between items.

CGR Server

During operation, the CGR server runs on a computer and provides live recommendations calculated specifically for each customer. Your client application calls the CGR server with a list of items. Based on this list, the CGR server returns a list of recommended items the customer is likely to want.

Getting Started

The CGR system is fairly trivial to setup and run. The technical tasks are getting the data in a usable form and tuning the system for your business.

Database Extract

The first thing that needs to be done is to acquire data from your system in a form the CGR calculator can use. This is the database extract file[1]. It has a very simple layout, multiple lines of text in the format of: "owner|item," each on its own line. For instance:

```
175|202355359
175|203410820
175|204482429
175|202306428
175|201147644
175|202367684
175|202082687
```

Items are ascii text representations of positive 32 bit integers. If your database is not in this format, a simple mapping table can be created.

This format was chosen because most database systems can produce this file easily. Typically, a database extract can be created from a SQL database with a query similar to this:

```
select customer, item, from customer_history order by customer
```

In the above SQL query, the customer (owner) field followed by the item number is returned in a number of rows. Note, the "order by" clause, it is very important that all a customer's items must be contiguous in the file as CGR calculator does not store or track the "owner" field.

Configuration File

The CGR System requires a configuration file to define its operation. Within this file, the user specifies any number of tuning options, location of the database extract, server parameters, etc. The configuration file is explained later on in this document.

Create Intelligence File

Once you have the database extract and the configuration file, you are ready to run the CGR calculator. Depending on the size and complexity of your customer data, this can take several minutes or several hours.

[1] The CGR plugin (a stand-alone software module) "sqlcgr" can be used to query a database directly, see "CGR Plugins" later in this document.

The CGR calculator is run as:

```
cgrcalc –f configuration.cfg
```

Run the Server

Once the intelligence file is created, you are ready to run the server. The server will read the intelligence file into memory and listen on a TCP/IP port for requests from your client application.

On-going Maintenance

Update the Intelligence File

Once setup, the CGR system is easy to maintain. Periodically run the CGR calculator to update the intelligence file with more current customer trends. Once the new intelligence file has been created, simply restart the CGR server program and it will use the new data.

Tuning

As the comfort level with the system increases, it is common for the administrator to tune the system so it provides “better” recommendations. The CGR system can be tuned by changing various settings in the configuration file, using different ranking strategies, or by coding a new ranking system in a plugin.

Software Updates

As with all software, the CGR system will be updated and improved over time. Maintaining your software license ensures you will have the updates available to you.

CGR Client Interface

The CGR system is easy to use. Most programming environments are capable of calling the system without any difficulties. There are two primary interfaces by which the CGR system can be incorporated into your environment.

Raw Text Over TCP/IP

The simplest interfacing to the system is through raw text. This is the default and most efficient interface. A client application opens a TCP/IP socket to the server, sends a list of items in the format of text number followed by newline for each item, then a single line for the strategy. Using the word "recommend" gets you the default ranking strategy. There is a list of CGR strategies at the end of this document.

example:

```
1002565

1002577

1002456

recommend
```

Recommendation Line

The raw text results will be a number of recommendation lines, followed by an end of line character. Each recommendation line will look something like this:

```
201354074,920797,35.80,59.07,33,118170
```

The above line can be broken into 6 data items:
(a)201354074,(b)920797,(c)35.80,(d)59.07,(e)33,(f)118170

a) This is the item number

b) This is the "rank[2]" of the recommendation.

c) This is the average connection ratio this item has with the items presented.

d) This is the best connection ratio this item has to the items presented.

e) This is the number of entered items to which this item connects.

f) This is the sum of all the connection ratios

End Line

The last line of the the result looks like this:

```
done: 35,34,1160,100,10
```

[2] A recommendation "rank" is an arbitrary number based on combining the result statistics. Its properties can be altered by using different ranking strategies or by developing a plugin.

The above line can be broken up into 5 data items:

(a)done :(b)35,(c)34,(d)1160,(e)100,(f)10

a) This keyword is used by your application to find the end of the result set.

b) This is the number of items entered.

c) This is the number of items which were entered that had connections.

d) This is the total number of possible results.

e) This is the actual number of results returned, this can be limited by the server.

f) This is the amount of time (milliseconds) that the system took to generate the results

HTTP Protocol and XML

Some programming environments may not be capable of making raw TCP/IP calls or some programmers may not be comfortable using raw TCP/IP sockets and dealing with the protocol issues involved. For these cases, there is an output plugin (xmlcgr) which returns the recommendations in XML format, and an input plugin (httpcgr) which takes HTTP requests. These two plugins make the CGR server a simple HTTP server that accepts a standard HTTP “GET” request and returns a properly formatted XML (“text/xml”) stream.

(For more information, see “CGR Plugins” later in this document.)

In a web browser or through a web enabled function call in your programing environment, the recommendations system is accessed like a web page:

`http://cgrsystem:8000/recommend?items=1002565,100257,1002456`

This will enter the same data as the example presented for the raw text server (the CGR strategy is the URL path after the server name) but the results will look something like this:

```
<?xml version = "1.0"?>
<!DOCTYPE MCGR SYSTEM "">
<RESULTSET entered="35" resident="34" found="1160" count="100" elapsed="0.300">
<RESULT>
        <ITEM>200577729</ITEM>
        <RANK>1422882</RANK>
        <AVGRATIO>57.15</AVGRATIO>
        <MAXRATIO>89.45</MAXRATIO>
        <SUMRATIO>1886.07</SUMRATIO>
        <COUNT>33</COUNT>
        <CURRENT>0</CURRENT>
</RESULT>
................
```

```
<RESULT>
        <ITEM>201672696</ITEM>
        <RANK>844605</RANK>
        <AVGRATIO>31.89</AVGRATIO>
        <MAXRATIO>55.13</MAXRATIO>
        <SUMRATIO>1052.51</SUMRATIO>
        <COUNT>33</COUNT>
        <CURRENT>99</CURRENT>
</RESULT>
</RESULTSET>
```

RESULTSET

“entered” is the number of items entered.

“resident” is the number of items which were entered that had connections.

“found” is the total number of possible results.

“count” is the actual number of results returned, this can be limited by the server.

“elapsed” is the amount of time (milliseconds) that the system took to generate the results.

RESULT

ITEM is the item number

RANK is the “rank[3]” of the recommendation.

AVGRATIO is the average connection ratio this item has with the items presented.

MAXRATIO is the best connection ratio this item has to the items presented.

SUMRATIO is the sum of all the connection ratios

COUNT is the number of entered items to which this item connects.

CURRENT is the current result number.

[3] A recommendation “rank” is an arbitrary number based on combining the result statistics. Its properties can be altered by using different ranking strategies or by developing a plugin.

CGR Configuration File

The CGR system configuration file is based on a simple parameter=value system.

Parameters

DATAFILE=/home/dba/dataout.txt

The data file is the database extract file produced by your DBA. (see “Getting Started”). It has a simple format that can be generated by virtually any SQL database.

HBA_[n]=255.255.255.0/192.168.1.0

This is a security option that allows you to specify multiple netmast/network pairs that are allowed to access the server. If no HBAs are specified no validation is performed. If HBAs are specified, only those specified will be granted access, all others (including localhost)will be rejected. Multiple HBAs can be specified in the configuration file by incrementing the trailing number, i.e. HBA_0, HBA_1, HBA_2, etc.

MAXCALCMEM=128

The cgrcalc program creates a large two dimensional sparse array matrix of the data from the DATAFILE. Obviously, it could easily exhaust all the memory in the system if not limited. This parameter controls the amount of memory (in megabytes) the calculator will use before swapping data to a hard disk.

MAXCOUNT=1000

An owner may have too many items to be a valuable source of statistical information. This allows you to exclude owners who will tend to generalize the recommendations.

MAXPERCENT=98

Some items may be too popular, everyone has them already. This parameter allows you to specify the upper limit to item's popularity, ranges from 0% to 100%

MAXRATIO=98

This parameter controls the maximum connection ratio between items, ranges from 0% to 100%

MAXRESULT=100

This parameter controls the maximum number of results the system will return.

MINCOUNT=10

An owner may have too few items to really make a statistically relevant contribution to the system. Buy setting this minimum, the administrator can reduce the amount of processing during calculation by eliminating owners.

MINPERCENT=0.5

Some items are so obscure that almost no one has or wants them. This parameter specifies the minimum percentage popularity of an item, ranges from 0% to 100%

MINRATIO=30

This parameter controls the minimum connection ratio between items, range from 0% to 100%

MINTREND=5

This is a "guestimate" parameter. Rather than playing with pure statistics, this allows the administrator to say that a connection between items needs to happen at least this many times before it is considered a trend.

PLUGIN_[n]=xmlcgr.so

The plugin parameter allows you to augment the server behavior with dedicated plugin code modules.

Multiple plugins can be specified in the configuration file by incrementing the trailing number, i.e. PLUGIN_0, PLUGIN_1, PLUGIN_2, etc. This also specifies the order in which the plugins are loaded.

POOLSIZE=8192

The CGR server uses a static pool of items to be used as storage for both the source and result item sets. Each thread has its own pool. This number specifies the maximum number of items that can be sent and evaluated for any single request. Each item in the pool takes about 40 bytes.

PORT=8100

This is the TCP/IP port on which the server will listen for requests.

QUEUE=128

On most server operating systems, the OS creates a queue of client TCP/IP connections that have been accepted but which have not yet been handled by the server program. This behavior varies widely over many platforms. This parameter is used to tune this behavior and is intended for experts only.

On systems with good scheduling response, the default of 128 should be sufficient. If you get connection failures, you may want to increase this value.

Windows users: On any windows system less than a "server" version, the backlog *queue is limited to 5. You must Windows (NT,2K, XP) Server or Advanced Server.*

TEMPDIR=/tmp

The cgrcalc program runs through the data contained in DATAFILE a number of times. Once to create a histogram of all the distinct items and owners, the next pass is used to create the network of items, and lastly to merge all the swapped work files into the CGR item intelligence file which is used by the server.

While many programming techniques have been used to reduce the amount of memory and processing required to perform these calculations, depending on the size of your data file and the diversity of your customers and catalog, the system could require a lot of disk space. Remember: the number of distinct items is conceptually squared in a two dimensional matrix during processing.

This parameter is used to direct the system to use this location for its temporary files.

THREADS=4

The number of threads the CGR server uses to answer requests. This number should be one or two more threads than the number of CPUs in the system used as the CGR server.

Alternate CGR Strategies

To understand the recommendation strategies, it is important to understand both the data generated by the CGR calculator program and the data generated by the CGR server in response to a recommendations request. All the statistically relevant items in the intelligence file are networked together by "connections" to one another. When a list of "source" items is sent into the CGR server, the network is searched for items which have "connections" to the source items, these new items are entered into the "result" set.

There are two fundamental concepts: "connectivity" and "connectivity ratio." Connectivity is a sum of all the connections a result item has with all the source items. Connectivity ratio is how often a specific result item is found to have a connection with a specific source item in the network. If two items are always found connected to each other in the network, their ratio is 100% If it is only half the time, then it would be 50%

Each item in the result set must be "ranked" for its position in the set. Some items will be very good matches and others may be less good. Obviously, it is desirable to have the best items come first. The ranking strategy is the method used for deciding which items are best and the order in which they are returned.

Item Data

The items are ranked using the information gathered and calculated during the initial search for connected items. This information is described below:

Connectivity

Connectivity is the number of connections a result item has in the list of source items. For instance, if a result item has a connection to 10 of the source items, its connectivity is 10.

Total

This is the total number of items in the source which were found to have connectivity with items in the network. Some source items may not have any connectivity with the results due to their being rather obscure themselves.

Ratio

This is the sum of an items connection ratios divided by the total number of connections in the source set.

Average Ratio

Similar to "Ratio," this is the sum of an items connection ratios divided by its number of connections with the source items.

Maximum Ratio

This is the best ratio between a source item and a result item.

Strategies

Using the above data, a number of recommendation strategies have been developed which can dramatically alter the result set generated from a consistent intelligence file.

The strategy names are presented in the parentheses. This name is used as either the last line in raw TCP/IP or as the URL path after the host name using the XML plugin.

Ratio Connectivity (ratioconn)

This is the default recommendation strategy, it is the average ratio multiplied by the percentage of the connectivity: avg * ((connectivity*100)/total)

This is a useful strategy for finding the items that most likely have a good connection and good ratio to the source set. The owner of the source set will more than likely know about all the items.

Connectivity (connectivity)

This rank is purely based on the connectivity of an item. Example: (connectivity*100)/total

This is a useful strategy for finding the items that most likely have a good connection to the source set, but maybe not the best ratios. This will have the tendency to find more obscure matches geared to the whole source set.

Weighted Ratio (weightedratio)

This rank is based on the ratio of the items connectivity with a weighting on the best ratio: (avg+max)/2

This is a useful strategy to finding fairly strong connections to individual source items, connectivity is much less important than the ratios.

Maximum Ratio (maxratio)

This rank is based on the best ratio a result item was with any source item.

This is useful for finding items with very string connections to individual source items with no consideration of the whole set.

Average Ratio (avgratio)

This rank is based on the average ratio a result item has.

This is useful for finding items with strong individual ratios or strong over all ratios with the source set.

Ratio (ratio)

This rank is the the sum divided by the total.

This is useful for finding items with a good ratio within the source set.

Inverse Ratio (inverseratio)

This is the inverse of the ratio. It tends to bring more obscure recommendations to the front of the list.

This is useful for finding really obscure items, almost like a “wildcard.”

CGR Plugins

The CGR system was designed for speed, efficiency, and flexibility. Rather than complicate the system for every conceivable option, it was designed to have virtually infinite options using precompiled code modules called plugins. The standard system comes with three default plugins.

xmlcgr

This is the XML format plugin. It changes the output format from raw text to XML formatted text. As mentioned in “CGR Client Interface,” this is a method by which a prepackaged XML API can be used to communicate with CGR. This plugin is not dependent on the http plugin, i.e. you can use this plugin to return XML while still submitting requests using raw text mode.

httpcgr

This is the HTTP protocol plugin. It changes the input protocol from raw TCP/IP to HTTP GET. As mentioned in “CGR Client Interface” this is a method by which an HTTP based API can be used to communicate with CGR. This plugin is not dependent on the xml plugin, i.e. you can use this plugin to answer HTTP formatted requests while still returning raw text results.

sqlcgr

This is the SQL data source plugin, it allows the CGR calculator to directly access a SQL database instead of the database extract file. In addition to loading this plugin, you will need to specify two additional lines in the configuration file.

SQLCONN=connection parameters

The SQLCONN configuration line is a string that contains a number of parameters that are used to connect to a SQL database. As SQL databases differ on their methodology for connection and authentication, the SQLCONN parameter is greatly dependent on the database used. Within this parameter, there are a number of fields that can be used:

- AUTH: The AUTH: parameter is used by the ODBC SQL object. It is the authorization password for the database connection.
- DB: The DB: parameter is used by the SQL object manager to decide which database technology to use. Currently this is ODBC for an ODBC system DSN, and PGSQL for a PostgreSQL database.
- DS: The DS: parameter is the data source. When using ODBC, this is a system DSN. When using PostgreSQL, this is the database name.
- HOST: The HOST: parameter is used by the PostgreSQL object as the host name for

the database system. If this is omitted, localhost is assumed.

- NAME:This is the user name used to connect to an ODBC database. This parameter combined with “AUTH:” provides the connection credentials.
- PORT: This is the port through which a PostgreSQL connection is made.

Examples:

```
SQLCONN=DB:PGDB PGSQL PORT=5432 DS:cgr HOST:dbserver
```

```
SQLCONN=DB:ODBC NAME:cgr AUTH=foobar123 DS:cgr
```

SQLQUERY=sql query

This is the query used to retrieve data from your database. The salient points are: customer and items must be represented as positive integers within a 32 bit number space (31 bits plus sign), the output must be ordered by customer, and the result row must have “owner” as the first column, and “item” as the second.

Example:

```
SQLQUERY=select customer, item from customer_history order by customer
```

CGR Programs

The CGR system is made up of a number of programs, each with a specific task.

cgrcalc

This is the main calculation program. It is used to create the intelligence file which used by the cgrserver. It is used as:

```
cgrcalc –f config.cfg
```

cgrdump

This is a diagnostic program used to read and output the information in a CGR intelligence file. It is different than the other CGR programs in that it takes the name of he intelligence file, not the configuration file. It is used as:

```
cgrdump filename.cgr
```

cgrserv

This is the main server program. It answers recommendations requests and returns results. It is used as:

```
cgrserv –f config.cfg
```