Recommending Geocaches

Andrew Trotman *Timothy Jones* *Chris Handley*
Department of Computer Science
University of Otago
Dunedin, New Zealand
{andrew, tljones, chandley}@cs.otago.ac.nz

Abstract *Players downloading GPS coordinates from the internet, hiking to the given spot, and hunting for a hidden box – this is the new sport of geocaching. Today there are nearly 200,000 such boxes in over 200 countries. With so many to find, a recommender is needed, one that takes into account not only the boxes, but also the geospatial and temporal nature of the sport.*

A database of geocaches in the South Island of New Zealand is made by trawling a prominent geocaching web site. This is then used to estimate the home-coordinates (geospatial playing centre) of players. Predictions are verified against a set of correct coordinates solicited from players.

Several geocache recommenders are discussed and compared. The precision, computed using mean of mean reciprocal rank (MMRR), of each is measured. The best method tried is a collaborative filter using intersection over mean to find similar players and a voting scheme to recommend geocaches. This method is proposed as a replacement for the currently used distance from home-coordinate; doing so will increase the precision of existing systems such as geocaching.com.

Keywords Information Retrieval.

1. Introduction

When the US president Bill Clinton announced the descrambling of the GPS satellite navigation system on the 1st of May 2000, he unwittingly also invented a new outdoor individual sport today known as geocaching.

The selective availability scrambling was removed on the 2nd of May 2000 and the next day Dave Ulmer hid a bucket of miscellaneous items (including a log book) in a forest outside Portland, Oregon. He published the coordinates on USENET and within a day the bucket had been found [9]. Within a month there were similar geocaches hidden in not only other US states, but also in other countries (including Australia and New Zealand). Today there are 196,250 caches in 217 countries [3].

Proceedings of the 10th Australasian Document Computing Symposium, Sydney, Australia, December 12, 2005.

This new sport is like orienteering; however, unlike orienteering, it is an individual sport. Players download longitude/latitude coordinates from a website (such as geocaching.com), go to the given location and then search for a hidden box. On finding it they open the box, log the find in the log book, then put the box back. Later they return to the website and log their find electronically. The sport can be played any time of the day or night, by anyone with a GPS receiver – there is no setup, no cleanup, and there are no teams.

Each new player expands the sport by hiding geocaches in places they enjoy visiting. Some players use the sport to swap walking tracks – they might, for example, hide a geocache on the ridge of a mountain. Other players might prefer obscure locations in big cities. Lunch-time players hide them in easy to get to places that make a good location for a lunch break.

This user preference brings both diversity and confusion to the sport. When geocaching in a new city (perhaps on holiday) a player is faced with several hundred geocaches from which to choose the few they might enjoy finding.

In this investigation we ask the question – is it possible to build a recommender for geocaching? But first we ask – is it possible to determine from behaviour where (geographically) players are playing?

We build a list of geocaches in the South Island of New Zealand by trawling geocaching.com.au[1]. Gaussian filtering is shown to be effective in home-coordinate estimate. Voting by similar players is an effective recommender; similar players are those with a high ratio of finds in common.

2. Recommender Systems

In a traditional collaborative filter recommender system such as MovieLens [13] an object's rating is predicted using a statistical analysis. For a given user the nearest other users are computed (perhaps using a *k*-nearest neighbour algorithm) and from that a weighted average of those users' ratings is used to rate the object. In effect, the rating a user will give the object is estimated using a weighted average of the ratings that similar users already gave it.

In a supermarket recommender [10], recommendations are made based on objects the user has pur-

[1] geocaching.com forbids trawling.

chased. For example, users who buy cheese and grapes are likely to also need crackers and wine. The recommendations are made by mining the shopping lists of customers. The purchases of all customers are collected together and data mining techniques used to find objects that are usually purchased together.

The domain of the recommender must be taken into account when choosing algorithms. Using a collaborative filter in a supermarket might tell us that a user would like a given brand of milk, but milk is milk regardless of brand and the user already knows this.

Equally, telling a user that if they enjoyed the first movie in a trilogy they should watch the others is futile – they already know this.

Recommending geocaches is quite unlike recommending supermarket purchases or movies for many reasons:

Players rarely return to the same geocache twice. This is quite unlike a supermarket where the same people usually buy essentially the same objects each visit.

Players cannot rate geocaches so it is not possible to predict a “movie rating” as there is no notion of rating.

In a supermarket all the objects on the shelf are available for purchase. In a movie recommender like Amazon.com all movies are also available for purchase. But this property does not hold for geocaching – just because a database is aware of a given geocache it does not mean the player can get to it. With a movie recommender like MovieLens, some very obscure items may be recommended, but no longer available for purchase [11], however with geocaching these objects are not obscure, they are geographically separated from the player. Recommending an Australian geocache to a New Zealand player is of little value – they cannot get to it!

New geospatial collaborative filtering algorithms are needed for this sport – algorithms that take into account the player’s habits and recommend only accessible geocaches. We are aware of no such pre-existing algorithms and focus on such algorithms (both content based and user based) in this investigation.

3. Home-Coordinates

At present, geocaching.com recommends based on distance from a player’s registered home-coordinate. This coordinate is the location that the player gives as the centre of their geocaching activity. Players are believed to use either their true residential home location or their work location (although this is anecdotal). Knowing the player’s current location is essential for recommending any geospatially dispersed objects – without it, it is not possible to recommend close objects. These home coordinates are protected by geocaching.com and are not on geocaching.com.au (and neither site is ours) so estimates are needed.

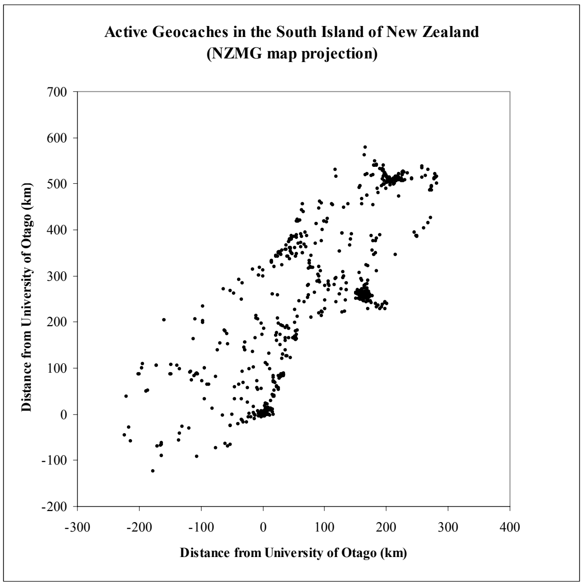

Figure 1: Geocaches located in the South Island of New Zealand (151,215 km^2)

Experiments were conducted to determine if it is possible to calculate a home-coordinate from the geocaches logged as found by a player.

Actual home-coordinates were solicited from players by posting a message on the geocaching discussion board gps.org.nz. This resulted in 12 replies from South Island players (7.5% of the active players). This set, although small, was used to compute the error in home-coordinate estimate.

3.1. Methods

The geocaches in the South Island of New Zealand were trawled from geocaching.com.au (on the 20th of May 2005). This produced a set of 806 geocaches (Figure 1) with 9,271 logs from 390 players who had found those geocaches.

This dataset contained geocaches that, although found by some players, have subsequently been removed from the sport (one of those placed by an author was washed away during a flood). Geocaches marked as unavailable at the time of the trawl were never recommended, but were used to compute user / user similarities and player habits.

Players were divided into two groups, inactive and active.

An inactive player was any player who had not found a single geocache during 2005 (in nearly 5 months), or who had found fewer than 5 in total. The first category includes those who have stopped playing, the second those who have not embraced the sport. Both do not enjoy the sport so building a recommender for them is futile.

After unavailable geocaches and inactive players were removed, there remained 741 geocaches, 160 players, and 8,299 logged finds.

The home-coordinate of each of the 12 players was estimated using five methods.

The geographic mean of all players' known home coordinates is used as a baseline in method *naïve*.

In method *geomean*, the home-coordinate was taken as the geographic mean of the geocaches a player had found.

In method *geomean2sd*, the home-coordinate was computed as the geographic mean of the finds, then those finds outside 2 standard deviations of the mean are removed and the geographic mean recomputed from the remainder set. This method was expected to outperform *geomean* as many players are known to play when on holiday at locations outside their home territory. As these holiday finds are likely to be only a small subset of the total finds of a player, they are likely to fall outside 2 standard deviations of the mean and will, therefore, be filtered out using this method – *geomean2sd* computes the mean from only the remaining finds.

Method *geomean1sd* was computed in the same manner as *geomean2sd*, except those finds outside one standard deviation of the mean were removed before the mean was recomputed.

In method *gaussian*, the smallest north/south-aligned bounding box containing a player's found geocaches was constructed. The bounding box was divided into axis-aligned 1km by 1km squares and at each vertex a Gaussian filter was applied according to equation (1)

$$g(d) = \sum_{c \in C} \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{d_c^2}{2\sigma^2}\right)} \qquad (1)$$

where d_c is the Euclidean distance (computed using the NZMG map datum [15]) between the vertex and the geocache, c, (from the set of player found geocaches, C), and σ was set to 50km (a "reasonable" player roaming radius). The vertex with the highest score was considered to be the home-coordinate. This method finds an approximation of the centre of the largest cluster in which the player has found geocaches – it is reasonable to believe this is their home-coordinate.

Each of the four methods was tested for the 12 players for which the home-coordinate was known. The error was computed as the mean Euclidian distance between the predicted coordinate and the player's supplied coordinate. Although this may be subject to over-fitting, the sample is too small to divide into training and evaluation sets.

3.2. Results

Figure 2 shows the error for each of the 12 players, and is summarised in Table 1. The two best predictors were *geomean1sd* and *gaussian*.

Geomean estimates using those geocaches the player has found, but which are outside their usual playing area. The number of these finds is reduced when those outside 2 standard deviations are discarded. More are discarded using *geomean1sd*. The exception to this is player 2 who is known to have a very large daily roaming area (due to work commitments) and for whom an estimate of a home-coordinate is therefore difficult. Player 8 is known to live in a low population area and to regularly travel to high cache-density areas to play.

Method	Mean
naïve	152.29
geomean	58.27
geomean2sd	51.70
geomean1sd	39.65
gaussian	35.83

Table 1: Mean error in the home-coordinate using each of the estimators

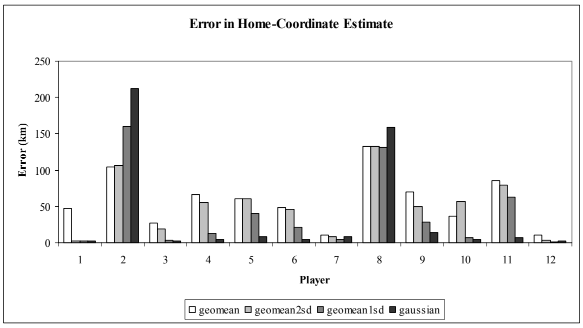

Figure 2: Error in home-coordinate estimates using each of the given estimators. The two best methods are *geomean1sd* and *gaussian*

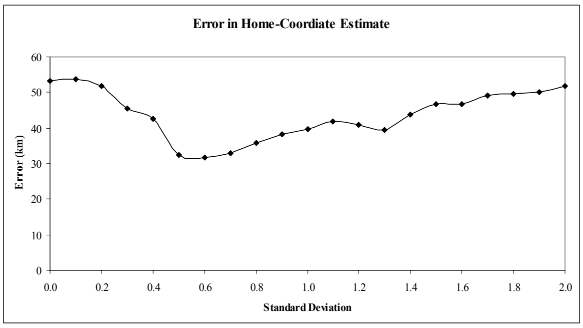

Figure 3: Error in home-coordinate estimate as a function of geocaches disregarded

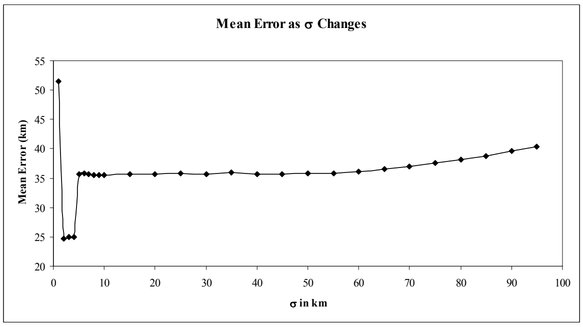

Figure 4: Error in home-coordinate estimate as a function of the sharpness of the Gaussian filter

As the error reduces as more finds are discarded, it is pertinent to ask exactly how many standard deviations from the mean should be used as the cut-off. Figure 3 shows the effect of varying this from 0 (using only the closest point to the mean) to 2. The figure shows that as more outliers are discarded the error decreases. Eventually those points on the edge of the player's true centre are discarded and the error increases. The least error occurred when finds outside 0.6 standard deviations were discarded (giving an error of 31.60km).

It is also pertinent to ask how error is affected by a changing standard deviation (σ) in method *gaussian*. This is plotted in Figure 4 for values in the range 1km to 100km. The trend shows a general decrease in error as the filter increases in sharpness. As the filter decreases in breadth, the score tends to a measure of the density in an ever decreasing sized area. Eventually, it will choose between dense areas, identifying the densest as the player's home – this is evident by the sudden drop at 4km.

As σ tends to zero, the score no longer represents a player's ordinary roaming radius, but will find sub-clusters within that radius, eventually identifying the two closest geocaches the player has found – this is evident in the sudden rise at σ=1km.

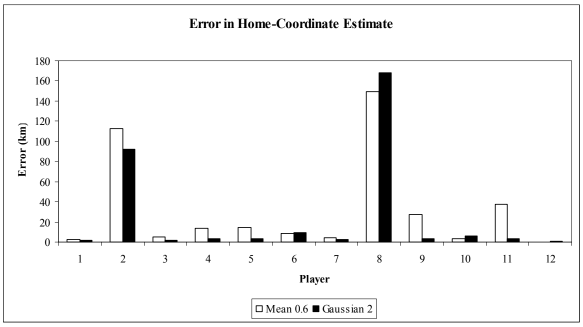

Figure 5: Comparison of best Gaussian and mean estimators. The Gaussian method has a smaller overall error and is better for most players

The best score computed using the Gaussian method occurred at σ=2km (mean error of 24.75km, median of 3.48km) while the best for the mean method occurred at cut-off of 0.6 standard deviations (error of 31.60km, median 11.15km). Figure 5 compares the error using each method for each player. The Gaussian method is better for 8 of the 12 players.

Gaussian with σ=2km is the method we use here on in to estimate the home-coordinate of all players.

4. Recommending Good Geocaches

Knowing a player's home-coordinates makes it possible to compute a list of nearby geocaches, however there remains the problem of recommending only those the player will enjoy.

4.1. Measuring Performance

Before algorithms can be satisfactorily compared a quantitative comparison method is needed. Upon initial inspection geocaching appears to lend itself to several such methods, but in fact does not.

The player's found list is comparable to a set of relevance judgments. For a given player, those geocaches that have been found are considered relevant, all others non-relevant. The task of the recommender is to recommend the geocaches a player has found based on some analysis. In this way each player is comparable to a query in a traditional information retrieval test collection, consequently mean uninterpolated average precision (MAP) [1] might be used to compute performance.

Using MAP does not take into account the temporal nature of the sport. Just because a player has not found a geocache does not mean they will not find it. For example, all those geocaches placed within a day of the trawl will have been found by very few players, however just a few days later they may have been found by many more.

The recommender could try and match the order the player found the geocaches. A metric such as the normalized distance-based performance measure (NDPM), or the half-life utility metric (see Herlocker *et al*. [6] for details of both) would be used to determine how well the recommended order matched the player's chosen order. However, geocaching is a temporal sport – new geocaches are being added, and old ones decommissioned. A metric trying to match a find order would also have to take into account the life cycle of a geocache.

Metrics that predict user ratings (mean error based metrics [6]) are inappropriate because players cannot rate geocaches.

Each metric measures the performance of a system relative to certain assumed user behaviour (a user model). These assumptions should be stated up-front so it is possible to verify the model – and correct it if erroneous.

For the purpose of this investigation it is assumed that a player finding a geocache is a positive vote for it. The converse, however, is not true (there are no irrelevant items in the collection). This assumption is necessary if the recommender is to be effective when a player moves home-coordinate (just because a player has not found a geocache in Sydney, it does not mean they do not want to find them there if visiting).

It is assumed that at any one moment in time the player chooses what they consider the "best" geocache to find next, and do find that geocache next. This assumption is necessary for two reasons. It makes the player choices discrete and deterministic, and it makes it possible to ignore log entries that log events other than finds (such as a did-not-find in the case of a geocache that has been pilfered).

Most specifically we assume that should one single find be removed from the player's found list then the very next geocache they choose to find is that very same geocache.

These assumptions turn the spatiotemporal aspects of geocache recommending into a named entity finding problem. As such, the metric of mean reciprocal rank (MRR) is appropriate. Averaging this over each player (the mean of mean reciprocal ranks, (MMRR)) gives a metric that favours each player equally.

We note that McLaughlin and Herlocker [11] recommend precision-based metrics for measuring the performance of collaborative filtering algorithms. We find our problem naturally lends itself to doing so.

For the experiments, the performance of the recommenders is computed by iterating over the list of all players and computing the mean of MRR for each player, according to equation (2)

$$MMRR = \frac{\sum_{p \in P} MRR_p}{|P|} \tag{2}$$

where P is the list of players, $|P|$ is the number of players, and MRR_p is the mean reciprocal rank for player p computed according to equation (3)

$$MRR_p = \frac{\sum_{f \in F_p} RR_{fp}}{|F_p|} \tag{3}$$

where F_p is the list of found geocaches, $|F_p|$ is the number of found geocaches, and RR_{fp} is the reciprocal rank of the geocache in the recommended list, computed according to equation (4)

$$RR_{fp} = \frac{1}{r_{fp}} \tag{4}$$

where r_{fp} is the rank of the given geocache in the list recommended by the system.

By the stated assumptions, two recommender systems can be compared quantitatively using MMRR; but this is not quite enough. A very large positive shift in performance with respect to a single player could have a marked effect on the metric. Exactly this problem is seen in information retrieval experiments where it is now common-place to present the significance of a change using the *t*-test or the Wilcoxon test. Sanderson and Zobel [14] compare the reliability of the two tests on TREC [4] data and suggest the *t*-test is more reliable. MMRR along with significance computed using a one-tailed *t*-test is reported here.

4.2. Data Analysis

It seems intuitively obvious that older geocaches have been found more times than newer ones. One would expect a geocache placed in the year 2000 to have been found many more times than one placed last week. To demonstrate this, a plot of age (in 30 day months) against mean number of finds for geocaches of that age is given in Figure 6. There are two points of interest: first, the number of finds is not, in general, a function of age; second, the number of finds is a function of age for some "short time".

The mean and standard deviation of the monthly find rate are 12.88 and 3.75 respectively. Assuming, with reasonable confidence, that any data points above the mean minus one standard deviation are representative of the mean, the intersection of this and the frequency curve will represent the point at which the "short time" ends. This is shown in Figure 7, where the intersection point is between two and three months. For the first three months of the life of a geocache, the number of finds is a function of age, after that, it is not.

As geocaches are geographically dispersed, the chaos around the mean shown in Figure 6 could be caused by geographic isolation of geocaching communities. If this were the case then all geocaches older than three months, ordered by distance from a given point, would show clear peaks in mean find numbers at community centres. In Figure 8 and Figure 9, all geocaches three months and older were ordered by distance from Wellington (North Island) in 10km buckets. Figure 8 shows the number of geocaches in the buckets, while Figure 9 shows the mean number of finds for that bucket. Vertical lines represent (from left to right) Nelson, Christchurch, Timaru, Oamaru, and Dunedin. From visual inspection, there are geocaching centres at large towns, however this has no effect on the mean number of finds of geocaches in the area.

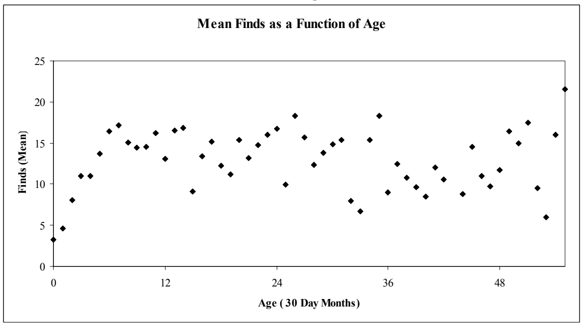

Figure 6: Mean number of finds for geocaches of the given age (in 30 day months)

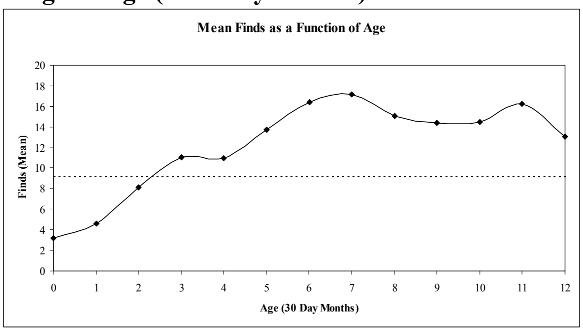

Figure 7: Mean number of finds for geocaches of the up-to one year. The horizontal line is the mean minus one standard deviation

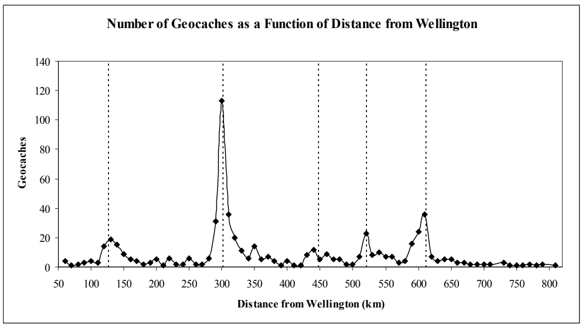

Figure 8: Number of geocaches as a function of distance from Wellington (10km buckets). Dotted lines are (from left to right) Nelson, Christchurch, Timaru, Oamaru, and Dunedin

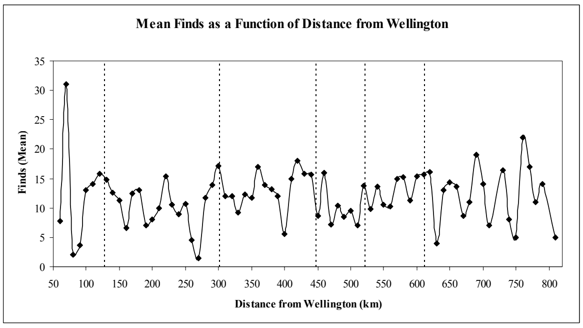

Figure 9: Mean geocache finds as a function of distance from Wellington (10km buckets)

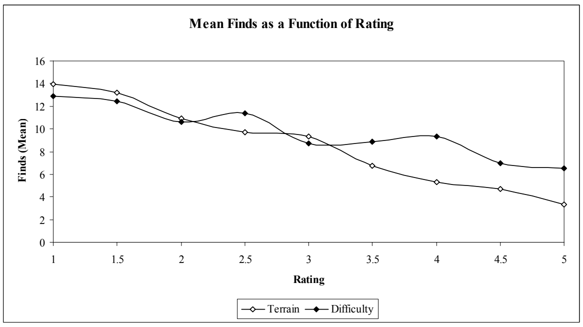

Figure 10: Mean number of finds for geocaches of the given terrain and difficulty ratings

If age cannot be used as a predictor of popularity, then what can?

Hiders rate their geocaches on a 5 point scale (including half points) for each of difficulty and terrain (guidelines exist). The easiest receive a score of 1 whereas the most difficult receive a score of 5. The mean number of finds for caches of the given rating is shown in Figure 10. Both show a near linear correlation, as the rating increases the mean number of finds decreases. Other attributes available for analysis include the type of geocache as well as the physical size – these are presented in respectively Figure 11 and Figure 12.

Although geocaching.com has several additional binary attributes (for example if or not climbing gear is needed), these attributes are not present in our data as they are not available on the site we trawled.

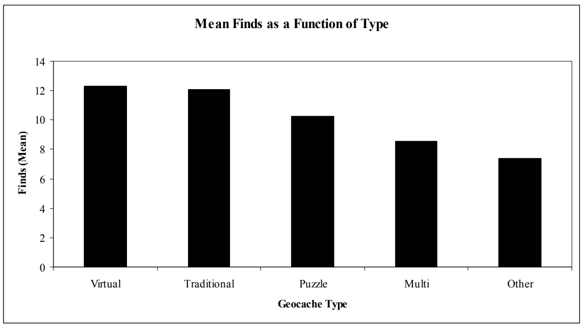

Figure 11: Mean of finds for geocaches types. The Other category includes Earthcaches (2), Event-caches (5), Letterbox caches (3) and Webcam caches (1)

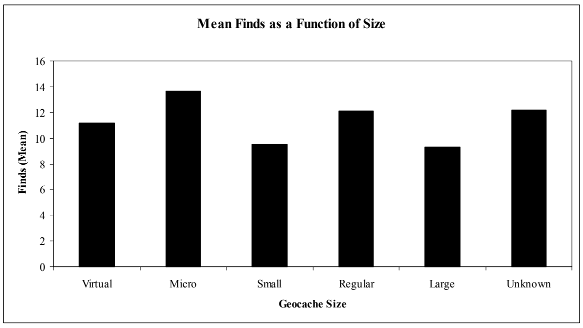

Figure 12: Mean finds for given sized geocaches

4.3. Possible Recommenders

The data analysis suggests terrain, difficulty, size, and type might be used in recommending geocaches. Additionally the number of finds, and the distance from the home-coordinate might be used. Separately, collaborative filtering techniques might be used. Several techniques were tried.

Geocaching.com orders by *distance* so this is used as a comparative baseline. In this method the next geocache a player chooses to find is the nearest unfound geocache to their home-coordinate.

Assuming the distributions discussed in Section 4.2 are probability distributions (each normalized in to range 0 to 1), the probability that a player will choose to find the given geocache is given by the product of the probabilities for each attribute. All five attributes (terrain difficulty, size, type, and popularity) are used exactly in this way in method *unweighted*.

In method *weighted*, the log of each distribution is weighted by a constant (learned using a genetic algorithm (GA) [7]). This is shown in equation (5)

$$P(R \mid g) = \sum_{k \in \{t,d,s,v,p\}} \left(c_k \times \log P(R \mid k) \right) \qquad (5)$$

where g is the geocache, k predictor (t for the terrain, d for the difficulty, s for the size and v for variety (type), and p for popularity).

A global voting scheme is used as a baseline for collaborative filtering techniques. In method *popularity*, the geocaches are ranked by the number of times found with ties broken by distance.

In method *vote* those *m* players with the closest geocaching behaviour to the given player were found. Similarity was measured using the Tanimoto coefficient (intersection over union) [12]. These players then voted for each geocache they had found, and ties were broken on distance.

Several additional methods were tried (for example, weighted without popularity, sum of probabilities, feature-space similarity, etc.), however none performed as well as the best reported herein.

5. Methods

The home-coordinate of each user was computed using the Gaussian method with σ=2km.

In an iteration of the experiment, a single recommender method is tested. A player is chosen and one geocache find is removed from their found list. The collection statistics (e.g. probability at each difficulty level) are then computed without this find (to remove bias). The *n* closest unfound geocaches to the home-coordinate (excluding those placed by the player) are then ordered according to the recommender method. Finally the MMRR score is computed.

For the collaborative filter the number of similar players, *m*, is varied to achieve the optimal value.

Weights, c_k, were learned with a genetic algorithm [7], optimised for *n*=10 nearest geocaches. It was run for 500 generations with a population size of 100, mutation rate of 0.1, single-point crossover rate of 0.6, and reproduction rate of 0.3. Elitism [2] was used with the top 5 individuals carrying over into the next generation (other values were not tried). The experiment was run four times, each had similar results.

The method that works best for the closest few geocaches to the player's home-coordinate might be quite different from the method that works best considering all the geocaches in the whole South Island. To see if such an effect exists, each method was tested on only the closest *n* geocaches to the home-coordinate. Values for *n* varied from 10 to 100 in steps of 10 (representative of pages of results on a web site). Methods that score best at the end of the first page (*n*=10) were considered best as seldom do searchers view past the first page of results [8].

6. Results

The results of the experiment are shown in Figure 13. Examining the baseline (*distance*), a clear upwards trend is shown as the number of geocaches included increases. This is because this method is rank-order preserving (with increasing *n*). It is asymptotic because eventually every find is accounted for. The other methods are not rank-order preserving so precision can decrease with increasing *n*.

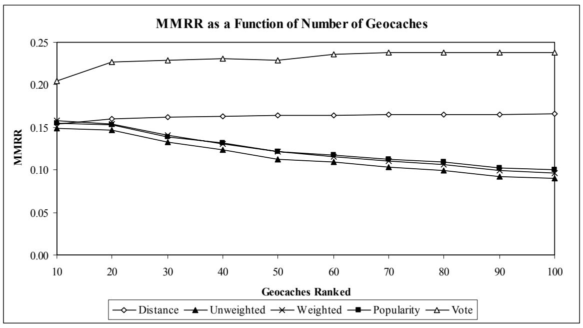

Figure 13: Performance of each recommender

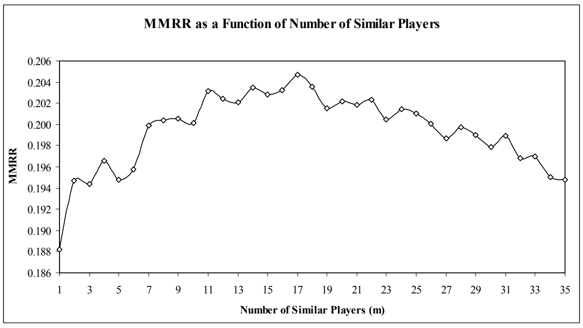

Figure 14: Effect on MMRR of varying the number of similar players (*m*) used in the *vote* method.

The best method tried was *vote*, a collaborative filtering scheme in which *m* similar players voted based on their found list. To find the optimal value of *m* the experiment was re-run with varying values for *m*. The result, shown in Figure 14, suggests that for the first page of results (*n*=10) the optimal value is 17 (that used in Figure 13).

In a collaborative filter the performance is known to increase with a decrease in sparsity [5]. In geocaching the sparsity problem does not occur because there are a relatively low number of geocaches in any one geographical area and a relatively large number of players searching for them.

Popularity	Size	Type	Terrain	Difficulty
.424	.122	.093	.351	.010

Table 2: Weights learned by GA favour mostly popularity and terrain

The best non-collaborative scheme tried was *weighted* in which the final weights are given in Table 2. Popularity and terrain are favoured most, followed by size, type, and lastly difficulty. Popular geocaches that are easy to get to are, in general, preferred over the others.

Significance computed with a one-tailed *t*-test (at *n*=10 geocaches) show that the improvement of *weighted* over *distance* is not significant (*p*=0.33), but *vote* over *distance* is significant (*p*=0.00).

Increasing the number of geocaches, *n*, used in the ranking does have an effect on the performance of the recommender. Only the collaborative filter and ranking on distance maintained their performance as *n* increased. The *vote* method with *m*=17 scoring the highest when n=100 (MMRR= 0.2380; on average at position 4.2 in the ranked list of results).

The analysis and experiments suggest that if the player has a geocaching history, the best recommender is a collaborative filter using the seventeen most similar players. With no geocaching history it is to use either the weighted method, or distance from home-coordinate

7. Conclusions

In the South Island of New Zealand there are 741 active geocaches, mostly located near to high-population centres. A recommender for this sport will help players identify the few geocaches they might enjoy amongst these.

A collection of geocaches, players, and player finds was trawled from the internet. The correct details of player home-coordinate were solicited using an online discussion list.

Several methods of estimating a player's home-coordinate from their logged finds were tested. In the best, computed using a Gaussian filter with σ=2km, the mean error was 24.75km and the median 3.48km. Although we don't know who the players are, we can identify their home-coordinate. There exists an obvious security issue here (especially should we also be able to identify the players).

Several recommenders are discussed and were tried. Each was tested including varying numbers of geocaches close to the player's home-coordinate. The performance of each was measured using mean of mean reciprocal rank.

The best method tested was a collaborative filter that identified the nearest seventeen players, all of which voted for the geocaches they had found. We recommend using such a method – once the player has a geocaching history. Before then we recommend ordering using a weighted probability method, or by distance.

References

[1] Buckley, C., & Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd ACM SIGIR Conference on Information Retrieval*, (pp. 33-40).

[2] De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* Unpublished Ph.D., University of Michigan.

[3] Geocaching.com. (2005). Geocaching - the official global gps cache hunt site. Available: http://www.geocaching.com/ [2005, 1 September].

[4] Harman, D. (1993). Overview of the first TREC conference. In *Proceedings of the 16th ACM SIGIR Conference on Information Retrieval*, (pp. 36-47).

[5] Herlocker, J., Konstan, J., & Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval,* 5(4), 287-310.

[6] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *Transactions on Information Systems,* 22(1), 5-53.

[7] Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.

[8] Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th ACM SIGIR Conference on Information Retrieval*, (pp. 154-161).

[9] Kimbo. (2005). A short history of geocaching: May 2000. Available: http://www.guysnamedkim.com/geocache/geocache_history.html [2005, 1 September].

[10] Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S., & Duri, S. S. (2001). Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery,* 5(1-2), 11-32.

[11] McLaughlin, M. R., & Herlocker, J. L. (2004). A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th ACM SIGIR Conference on Information Retrieval*, (pp. 329-336).

[12] Mild, A., & Reutterer, T. (2003). An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services,* 10, 123-133.

[13] Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on intelligent user interfaces*, (pp. 263-266).

[14] Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of the 28th ACM SIGIR Conference on Information Retrieval*, (pp. 162-169).

[15] Stirling, I. F. (1973). *New zealand map grid* (Technical Circular 1973/32): Department of Lands and Survey, New Zealand.