Notes on Search Document

Note: I've not yet had time to read the new stuff from the maths guy, so will leave any comment on the new math out of this doc.

Making Use of Dimensionality

Since we have a multidimensional dataset, we can/should make use of some of the characteristics of this:

- Decorrelation to reduce the dimensionality of the data, maximise variance, etc for analysis (Principle Components Analysis, for example)
- Exploit basis functions – we'd need to think these through carefully in terms of 'what does this basis function actually mean).

Since we have a time dimension, we have the notion of a 'path' in (coordinate) space-time. We can therefore do things like:

- predict trajectories (which equates to predicting 'next user action') using Kalman filters
- multi-dimensional clustering, either by split-merge algorithms (e.g. isoclass) or by guided methods (e.g. Bayesian)
- We can do path analysis; anomalous paths mean odd behaviour which we might want to look at; Spam for example would be easily detectable by this means (the paper I sent you some time back touches on this idea in a different domain). There are concepts known as 'chain analysis' and 'link analysis' that I have used in the intelligence world for this type of task; we could probably usefully lift some of these.
- Actions taken by users to direct content down a path are directly analogous to Google's 'page rank' concept – the more times that any given action is taken, the higher the 'path rank'.
- We ought to look at Fourier based methods too, and maybe also the joint time-frequency type methods (e.g. Wigner-Ville transform).

All in all, we have access to some really novel analysis techniques, which should be able to produce some interesting results that are totally unique to v3.

Weighting Importance and Relevance

- This one is definitely a good candidate for Bayesian methods – setting the 'slider' value gives us a (user selected) weighting. Presumably, we can pick up at least one more from the system calculated weighting for the same object.

- Similar actions by a group of users are likely to have more importance than a single user's actions. Remember, signal to noise ratio will go up as the square root of the number of users, which is how we weight collective actions (this gets rid of your question about 'how do we decide if an action by 10 users is not 10 times more important than a single user', by the way).

- We're probably much better off looking for a probabilistic method that gives us an acceptable local solution to the importance and relevance problems, rather than trying to solve the whole thing by brute force.

General Comments on the Math

Estimation

The problem of estimating user (and content) importance is perhaps best treated using Bayesian probabilistic methods:

- Would allow us to use a priori weighting techniques – using the original weights perhaps.
- We could manipulate the weights and weighting algorithm to 'tune' the importance algorithm to any given problem domain.
- We could conceivably use a posteriori weighting too, not sure what this would buy us (yet, cos I haven't had time to think about it).

This is a pretty standard problem that crops up all over the place – pattern recognition and image classification are the areas where I have worked with it most often.

Solving the resulting linear systems

The system of linear equations that we have to solve is going to be large, sparse and ill-conditioned. This is going to affect how we do the numerical computation. For example, consider a 1000 person company.

To do the overall importance ranking, we essentially have to compute the importance of each person relative to every other person. This gives us a linear system to solve of 1000 equations in 1000 unknowns – in other words, a million elements in matrix form.

Issue two: on any given day, any given person will likely communicate with only a (small) subset of the total number of people. Thus, most of the matrix elements will be zero and the matrix will be sparse.

Issue three: since the non-zero elements will be randomly distributed, the matrix will likely be ill-conditioned (and conceivably singular). This will throw a lot of numerical algorithms, and play merry hell with convergence and precision. (Gauss-Seidel for example will likely not converge in this environment – better to go for something like successive over-relaxation, or even something more sophisticated than this).

Solving this system of equations is not a trivial task, speaking from the standpoint of numerical computing. Certainly, the classic/standard solution techniques will either not work, or will possibly/probably give wrong answers (local minima versus global ones, for example).

We'd have to resort to something much more heavy duty – offhand, I don't know what exactly, but I'd probably start thinking around Cholesky or Householder transform methods. I'd also try and reduce the sparse matrix to some sort of banded

dense one if possible, and also try and partition the problem so that we don't have to solve the whole thing at one go.

Practicality issues and Observations

- Assuming we do double precision math (64 bit), then this matrix requires 8 MB to hold. We will have at least an original dataset, plus a result, plus at least one temporary matrix (more if we have to do something like rotation). Thus, we're going to soak up 24MB for the matrices alone. Add some working space and this will go up, let's say to 30MB. We can therefore see that each user requires 30kB of storage for a 1000 user system.

  Problem is, for a million user system, this goes up substantially – we now have 10**12 matrix elements, or 8TB of matrix. Oops. Each user is now taking up 30MB of storage.

  Such a system is not practicably soluble in core.

- Compute times will be horrible. This is likely to remain an offline or separated task. Not something that we can dare put anywhere near our transactional modules.

- We're probably much better off looking for a probabilistic method that gives us an acceptable local solution to the importance problem, rather than trying to solve the whole thing by brute force.

Odds and Ends

- Bottom of p7 on 'future submessages' has a repeated typo where the word 'history' is substituted for 'future'.
- p8: 'all definitions of history and future are this relative to a point in time' should have the words *'and space'* added to the sentence. It is possible for any given mail (say B|A) to be simultaneously acted on by more than one recipient; hence, we need to include the extra data dimension to ensure uniqueness. Will be needed for chain type analysis anyway. Where is as important as when here.
- p11: 'users interact with relevance' – the system also interacts with relevance too. A couple of obvious examples are prioritisation of mail sending (e.g. QoS) and prediction / detection of some types of compliance event.
- p12: 'other criteria are unique to BlueSpace' – we can add another unique criteria, 'state in the objects lifecycle'. Also, the security minded types (e.g. military) will want to see security classification / security level here too so it ought to go on the list.
- p12: 'if a user is looking for a word file…' – the situation is a bit more complex, because we might have to consider related file types e.g. PDF, SXI, TXT, or whatever.
- Section 5.2.3 spawned an interesting thought – we could implement a concept that I have called 'predictive remailing'. If a user – say B – always forwards the mails she receives from A to users C and D, then we could learn this behaviour over a short period of time, and do it automatically – or at least, put up a popup user

window that says 'do you want to do this now, press this one button'. Might be some interesting applications of this.

- Section 5.3.2 – we need to be careful to separate a documents 'relevance' from its urgency. It's likely that users will (want to) look at urgent documents more quickly, irrespective of their relevance rating.
- Section 5.4: any given document has an 'intrinsic importance' that is a function of the documents content and is independent of the sender and recipient(s). A document with a content of 'our key competitor has just gone bust' has the same importance whether it originates with the CEO or the janitor (of course, the actions taken are likely to be rather different, but that's a different issue).
- Section 5.4 again: calculation of an overall importance of a document (or person) to an entire enterprise is likely to be impossible; we're combining apples with pears and bananas and calling the result a peach. Yes, we can compute a number, but it's not likely to have any real meaning. Note that such values would also only be relative (A is > B) not relating to any sensible absolute scale. They'll also change with time – potentially rapidly – and will be sensitive to external events. For example, the competitor going bust scenario above is will change the relevance of all related docs immediately. I suspect – though cannot prove at this point – that the way in which this sort of change happens will be governed by catastrophe theory math, not by conventional stats.
- Section 5.4 (III) – bottom: in all cases, we can generalise the importance of individuals versus roles by:
  $wt_{total} = r * wt_{role} + s * wt_{individual}$ where $(r + s) = 1.00$
  This will make the whole thing consistent and allow us to handle both cases without changing the code or functionality. Also allows intermediate cases where both have relevance.
- Section 7.4.1: we could use views of search results that might be termed a 'relevance view' as well as link analysis and time based views. All of these are probably graphical rather than textual presentations.
- Section 8.1 – para on outliers: we need to think in terms of probability distribution functions rather than absolutes, which will get rid of the outlier type problems. Presumably most of the distributions around actions etc will be normal though I'd suspect the importance/relevance weightings to be Gaussian.

By the way, I do search for relevant Dilbert cartoons in my archive!

Random Thoughts

Finally, a random thought – what topics from game theory are relevant to this set of problems?????

Could well feed in to autonomous agent type functionality for searching. Needs thought.

EXHIBIT 2052

metadata slip sheet

BegDoc	REV0000096
BegAttach	REV0000096
File Name	Some Thoughts on the TransMail Relavance.doc
LastCreateOn	5/17/2005 3:39 AM
LastModifiedDate	5/17/2005 8:27 PM
Custodian	Marston, Justin
Author	Marston
SortDate	5/17/2005 8:27 PM
Office-Revision Number	5
Last Saved By	Marston
Exhibit No	