Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

Astronomical Catalogues - Simultaneous Querying and Matching

Hans-Martin Adorf, Gerard Lemson, Wolfgang Voges

Max-Planck-Institut für extraterrestrische Physik, Garching, Germany

Harry Enke, Matthias Steinmetz

Astrophysikalisches Institut Potsdam, Germany

Abstract. We report on our experience gained by executing multiple simple cone searches on a number of published astronomical catalogues. Individual search results are fed into a catalogue cross-matcher developed by GAVO. The matcher is designed to perform a probabilistic "fuzzy outer join" based on sky-positions and their uncertainties. We describe current features of the GAVO architecture that support such simultaneous queries, and outline some requirements for future versions.

1. Introduction

The German Astrophysical Virtual Observatory (GAVO)[1] is setting up an infrastructure that will allow (1) exercising the existing simple cone search (SCS) services; (2) searching for exotic objects like isolated neutron stars, brown and white dwarfs; and (3) constructing a multi-band spectral energy distribution (SED) from various catalogues, useful e.g. for source identification and classification purposes.

To this end GAVO is developing a multi-catalogue multi-cone (MCMC) search service feeding a probabilistic cross-matcher. The overall architecture of the search and matching service is depicted in Fig. 1.

2. The MCMCS Download Manager

The MCMCS application is similar to the IVOA "VODownload" manager[2]. Using a SOAP/WSDL-based Web-service, it queries the Virtual Observatory Registry Prototype[3] at Johns-Hopkins University in order to retrieve the base URLs of available simple cone searches. The MCMCS download manager is an event-based, multi-threaded Java application, designed to minimize the latency

[1] `http://www.g-vo.org`

[2] `http://skyservice.pha.jhu.edu/develop/vo/ivoa/default.aspx`

[3] `http://skyservice.pha.jhu.edu/devel/registry`

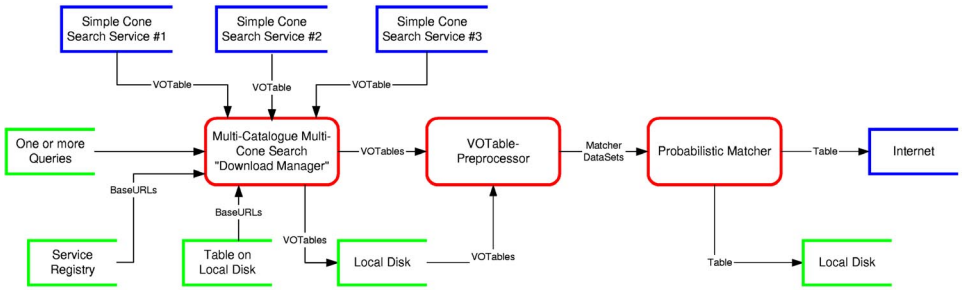

Figure 1. Dataflow through GAVO's search and matching service: the MCMCS application queries a registry of available cone search services. The MCMCS application takes one or more queries, executes multiple simple cone searches, and retrieves catalogue subsets in VOTable-format. Each dataset is pre-processed and forwarded to the probabilistic matcher.

between query submission and retrieval of the last result. It passes the incoming VOTables (Ochsenbein et al. 2002, Williams et al. 2002) to one or more registered "result handlers" for further processing. The default result handler stores the VOTables on disk in different directories.

GAVO intends to offer the MCMCS-functionality within its Web-services. In addition, GAVO plans to make this application generally available as a stand-alone tool, and/or as a plug-in component usable by other software systems.

3. The VOTable Processor

We are experimenting with different approaches for processing the VOTables, in order to extract the data needed by the matcher: XSL translation into tabular formats (e.g. comma-separated value "CSV" files), and XML-parsing using a JAXB parser compiled from the VOTable schema. XSLT-processing is rather robust, but requires the handling of table files. The alternative approach, JAXB-based VOTable parsing, while elegant, is hampered by the fact that many VOTables received do not (yet) validate, thus causing parsing errors. Other VOTable parsers will be evaluated in the near future.

4. The Probabilistic Cross-Matcher

GAVO's cross-matcher is based on positional information. It aims at performing a probabilistic match of the sources found in datasets – equivalent to a "fuzzy outer join" in database terminology. We are experimenting with a symmetric, recursive algorithm. Match candidates are selected from a starting pair of datasets; the result may be matched with further primary datasets or with other intermediate datasets, in order to obtain higher-order match candidates.

In our work we are pursuing goals similar to those of the SkyNode / SkyQuery project (Malik et al. 2002, Thakar et al. 2003). We differ, however, in several aspects: firstly, we try to use individual positional uncertainties on

a per-object basis; secondly we try to take into account the full information on the astrometric uncertainty including correlations between RA and Dec (as displayed by some scanning sky-survey instruments). Whether this additional complexity pays off in the end is still an open question. Finally, our matcher does not run in a distributed fashion, but locally, which simplifies the processing in some respect.

For each sky-position, we are assuming a multivariate Gaussian probability distribution. The inspected catalogues specify the astrometric uncertainties in different ways, and so far we have identified four types:

- *Type 0:* no error information is specified in the catalog/dataset;
- *Type 1:* a single error column specifies an *isotropic* astrometric error;
- *Type 2:* two error columns specify two *uncorrelated* errors, one in the direction of the right ascension and the other in the direction of the declination;
- *Type 3:* three error columns specifying a general error ellipse through its major and minor axis, and a position angle.

We assume that the error for the right ascension always specifies the uncertainty in form of an arc-distance in the direction of the right ascension, implying a correction with cos(Dec). However, it is unclear whether this assumption can be relied upon (see Ortiz 2004). The difference would be most notable near the poles.

For each candidate match the matcher computes an estimated position for the hypothetical astrophysical object, along with an estimate of the uncertainty of this position.

Different statistical measures are conceivable for assessing the "plausibility" of a candidate match. We are exploring the use of the Mahalanobis distance. Inferior match candidates are discriminated against by applying a threshold.

5. Observations and Issues

Overall we found most of the advertised SCS services operational, with a failure rate as low as 5%. However, the results returned vary syntactically and semantically to a degree that currently prevents a fully automated, unassisted search and matching service.

Here is a preliminary list of our findings: (1) Many VOTables received do not validate. (2) The service name is not unique (e.g. 2MASS-PSC is used by both Vizier and Irsatest). (3) There is no established standard for determining which columns are returned at a given verbosity level. (4) When no source is found, some services return an error, others return an empty VOTable. (5) Some VOTables have more than one field with a `POS_EQ_RA_MAIN` (or `POS_EQ_DEC_MAIN`) Unified Content Descriptor (UCD). (6) It is difficult to automatically detect the type of the positional error information (see above). Likewise, even if the type were known, it is not easily possible to automatically find the columns containing the uncertainty information. (7) The positional uncertainty information may not be available at SCS verbosity level 1. Thus different verbosity levels have to be tried, or one has to resort to always using verbosity level 3. (8) It seems to be unclear whether the `ID` or the `NAME` attribute should contain the "official" name of a data column. Some VOTables use both attributes. (9) The angular units are not homogeneously specified; sometimes "degrees" was found. The

positional uncertainties are usually not given in units *deg*, but *arcsec*, so a unit conversion is required.

6. Suggestions

We should like to make some suggestions for improving the format and content of VOTables, so that a fully automated search and match process will be possible in the future: (1) Use unique service names in the registry, and include them in the VOTable (e.g. 2MASS-PSC@Vizier). (2) Replicate the SCS query in the VOTable. (3) Standardize on a mechanism that allows retrieving just the field descriptions, e.g. by issuing a SCS with a zero or negative search radius. (4) Always return the positional error information along with the positions at the same SCS verbosity level. (5) Specify and implement a unique mechanism that allows an automatic identification of the position and uncertainty fields. (6) Support groupings of VOTable fields. (7) Indicate the type of the astrometric uncertainty specification (0 to 3 error columns). (8) Standardize on how angular units are specified. Perhaps, always use decimal degrees, also for the positional uncertainties. (9) As a stop-gap measure, include extensive comments in the field descriptions (following Vizier's practice), so that at least humans can find out what the fields are.

7. Conclusion

It is certainly an impressive accomplishment of the VO community that, with rather modest effort, it is possible to invoke a simultaneous cone search on 60+ catalogue services on the Internet. It is likewise impressive that the resulting datasets are available in "almost" the same data format. However, in order to enable a fully automated search and matcher service, the VO community needs to spend some further work on straightening out different interpretations of the existing standards, as well as on augmenting these.

Acknowledgments. This work was carried out as part of the GAVO project, funded by the Bundesministerium für Bildung und Forschung (BMBF). The MCMCS download manager was kindly made available to GAVO by Julius E. Adorf.

References

Malik, T., et al. 2002, SkyQuery – A distributed Web-based Query Service for Astronomy. The Johns Hopkins University: Baltimore

Ochsenbein, F., et al. 2002, VOTable: Tabular Data for Virtual Observatory.

Ortiz, P. 2004, "Merging data from a collection of sources", this volume, 173

Thakar, A.R., et al. 2003, SkyQuery – A Prototype Distributed Query and Cross-Matching Web Service for the Virtual Observatory. in AAS 201^{st} Meeting, 606-607

Williams, R., et al. 2002, VOTable: A Proposed XML Format for Astronomical Tables. CDS: Strasbourg. 28