Joining Data Logs

When assembling a training set, you must sometimes join multiple sources of data.

Types of Logs

You might work with any of the following kinds of input data:

- transactional logs
- attribute data
- aggregate statistics

Transactional logs record a specific event. For example, a transactional log might record an IP address making a query and the date and time at which the query was made. Transactional events correspond to a specific event.

Attribute data contains snapshots of information. For example:

- user demographics
- search history at time of query

Attribute data isn't specific to an event or a moment in time, but can still be useful for making predictions. For prediction tasks not tied to a specific event (for example, predicting user churn, which involves a range of time rather than an individual moment), attribute data might be the only type of data.

Attribute data and transactional logs are related. For example, you can create a type of attribute data by aggregating several transactional logs, creating aggregate statistics. In this case, you can look at many transactional logs to create a single attribute for a user.

Aggregate statistics create an attribute from multiple transactional logs. For example:

- frequency of user queries
- average click rate on a certain ad

Joining Log Sources

Each type of log tends to be in a different location. When collecting data for your machine learning model, you must join together different sources to create your data set. Some examples:

- Leverage the user's ID and timestamp in transactional logs to look up user attributes *at time of event*.
- Use the transaction timestamp to select search history *at time of query*.

itical to use event timestamps when looking up attribute data. If you grab the latest user attributes, your tr
ill contain the values at the time of data collection, which causes training/serving skew. If you forget to dc
arch history, you could leak the true outcome into your training data!

Prediction Data Sources — Online vs. Offline

In the Machine Learning Crash Course (/machine-learning/crash-course/static-vs-dynamic-training/video-lecture) you learned about online vs. offline serving. The choice influences how your system collects data as follows:

- online—Latency is a concern, so your system must generate input quickly.
- offline—You likely have no compute restrictions, so can do similarly complex operations as training data generation.

For example, attribute data frequently needs to be looked up from some other system, which could introduce latency concerns. Similarly, aggregated statistics can be expensive to compute on the fly. If latency is a blocker, one possibility is to precompute these statistics.

Previous
← Size and Quality of a Dataset (/machine-learning/data-prep/construct/collect/data-size-quality)

Next
Label Sources (/machine-learning/data-prep/construct/collect/label-sources) →

Last updated 2019-07-11 UTC.