Multi-domain Predictive AI

Correlated Cross-Occurrence
with
Apache Mahout and GPUs

Pat Ferrel

ActionML, Chief Consultant

Apache Mahout, PMC & Committer

Apache PredictionIO, PMC & Committer

pat@apache.org

pat@actionml.com

ActionML

What is the Goal for Predictive AI?

Use all we can record about users to predict their preference for anything

ActionML

What is the Goal for Predictive AI?

Use all we can record about users to predict their preference for anything

- Recommenders
- Behavioral Search
- Personalized Apps

ActionML

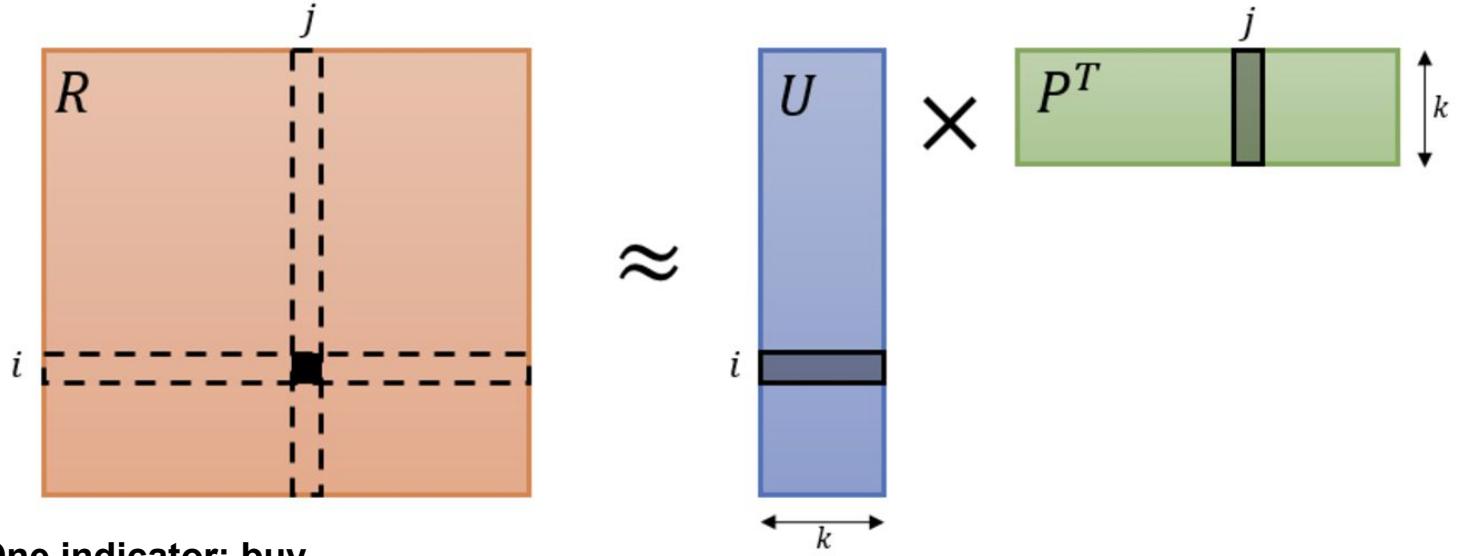
What Problem Does this Solve?

- Multi-domain, multi-modal, multi-action, multi-behavior, multi-indicator data means we know more about a user
- Coverage is greatly increased if we can use multi-indicator data
- Carefully correlating behavior means much better predictions if only because we have new data sources
- Being able to target any type of prediction from the same dataset allows us to predict new things (caveats apply)

ActionML

Matrix Factorization ALS-style

Users by Items, “buy”



One indicator: buy

ActionML

Problems with ALS

- Only one indicator of behavior
  - Buy: can bring good results but limits user and item coverage to past buyers
  - Ratings: mostly useless
  - Others: yes but only one at a time

ActionML

For the same E-Commerce Example: Multi-modal, multi-domain behavior

What if we could use:

- Buying behavior indicator (user-id, buy, item-id)
- Viewing behavior indicator (user-id, view, item-id)
- Category-preference behavior indicator (user-id, cat-pref, item-id)
- Sharing behavior indicator (user-id, share, item-id)
- Search behavior indicator (user-id, search, keyword)

to make better:

- buy recommendations or
- augment search indexes or
- understand a user's category preferences, or ...

ActionML

Correlated Cross-Occurrence

Apache Mahout + Apache PredictionIO + AML code

=

The Universal Recommender

ActionML

ANATOMY OF A RECOMMENDATION: Simple Cooccurrence Algorithm

r = recommendations

h_a = a user's history of some primary action (purchase for instance)

A = the history of all users' primary action rows are users, columns are items

$[A^tA]$ = compares column to column using log-likelihood based correlation test

$$r = [A^tA]h_a$$

ActionML

The Theory Doesn't End There

- Virtually all existing collaborative filtering type recommenders use only one indicator of preference

$$r = [A^t A]h_a$$

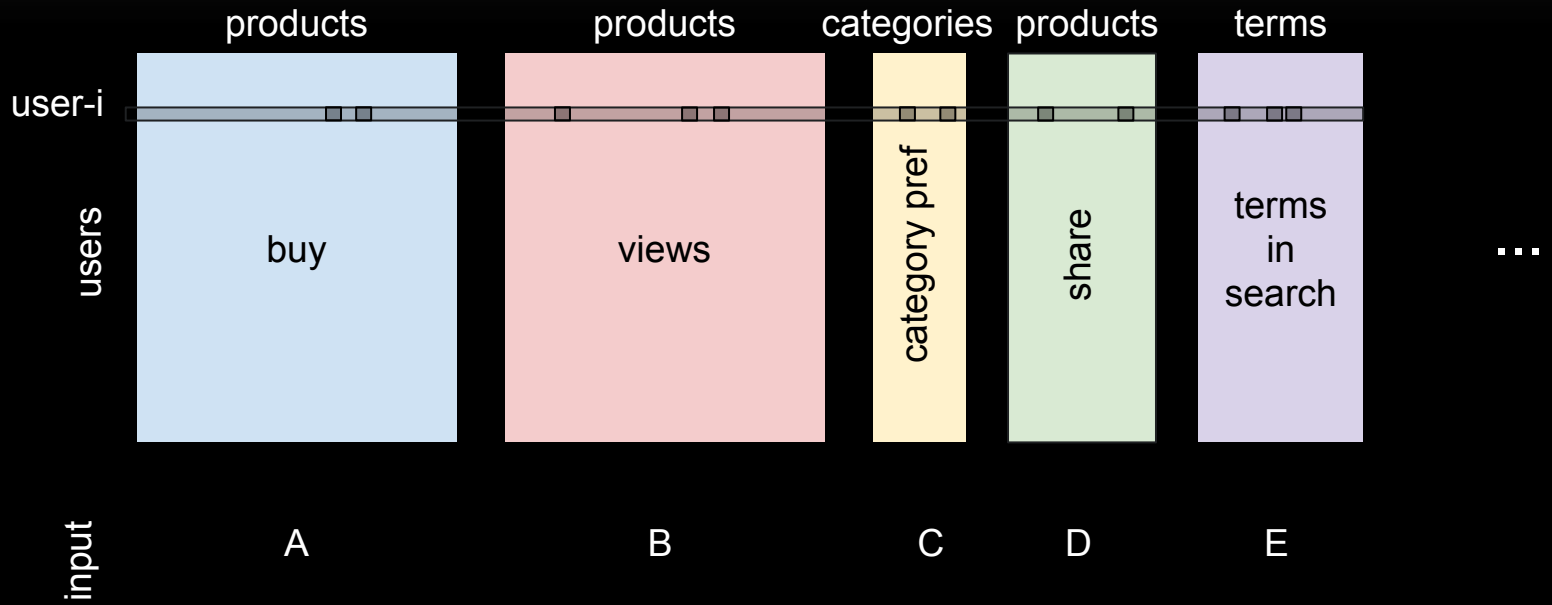
- But the theory doesn't stop there, we can find correlation between different behavior (CCO)

$$r = [A^t A]h_a + [A^t B]h_b + [A^t C]h_c + \ldots$$

- Virtually anything we know about the user can be used to improve recommendations—purchase, view, category-preference, location-preference, device-preference…

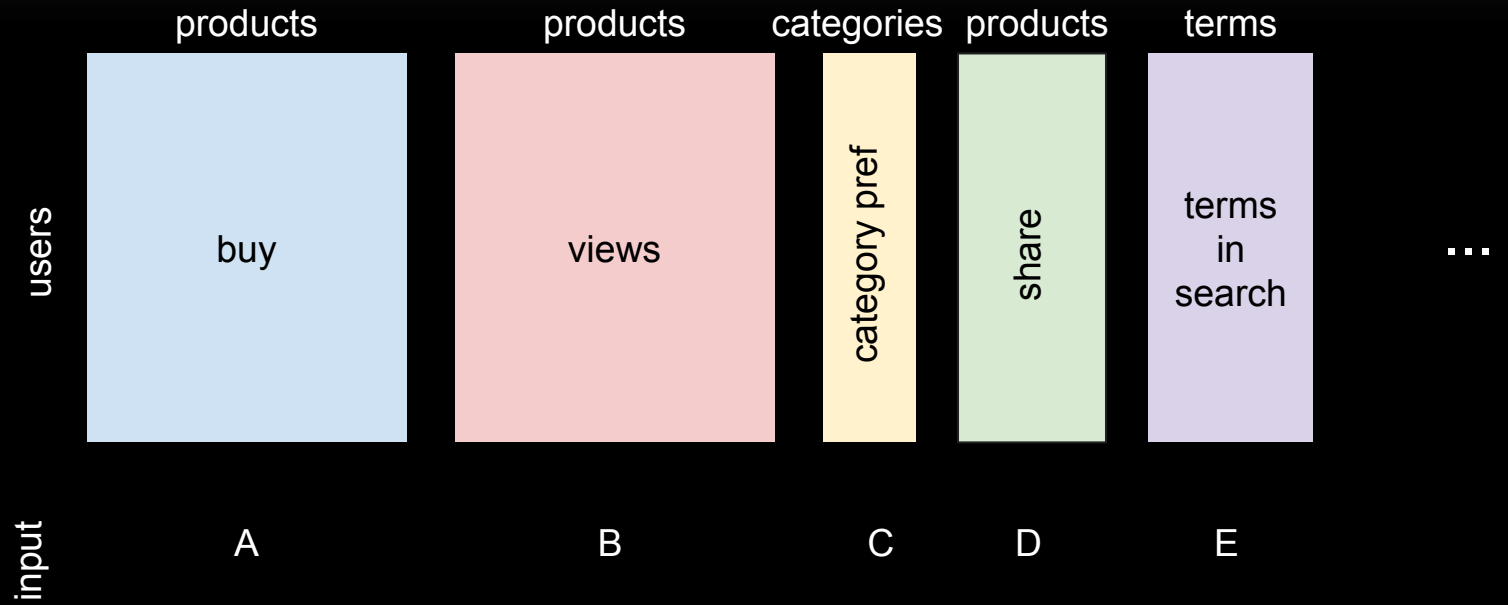
ActionML

Single User History of Multi-modal Behavior



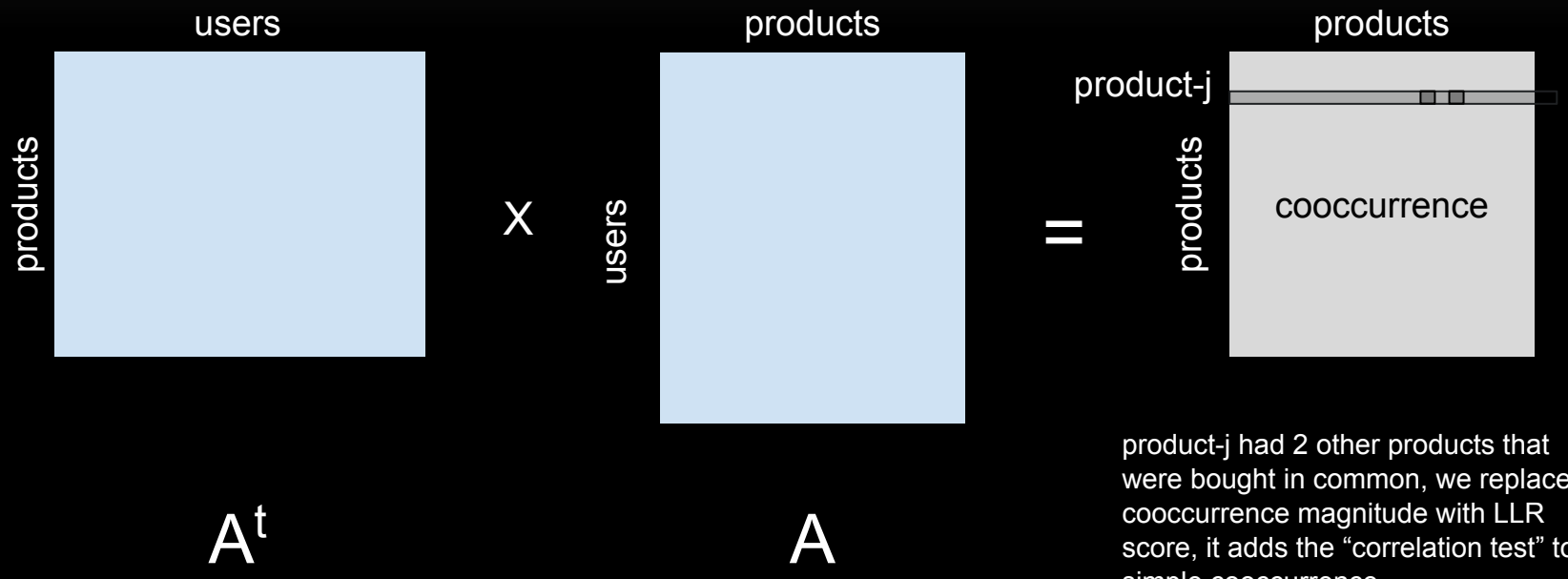
ActionML

All User's Multi-Modal Behavior Indicators: Far More than Conversions



ActionML

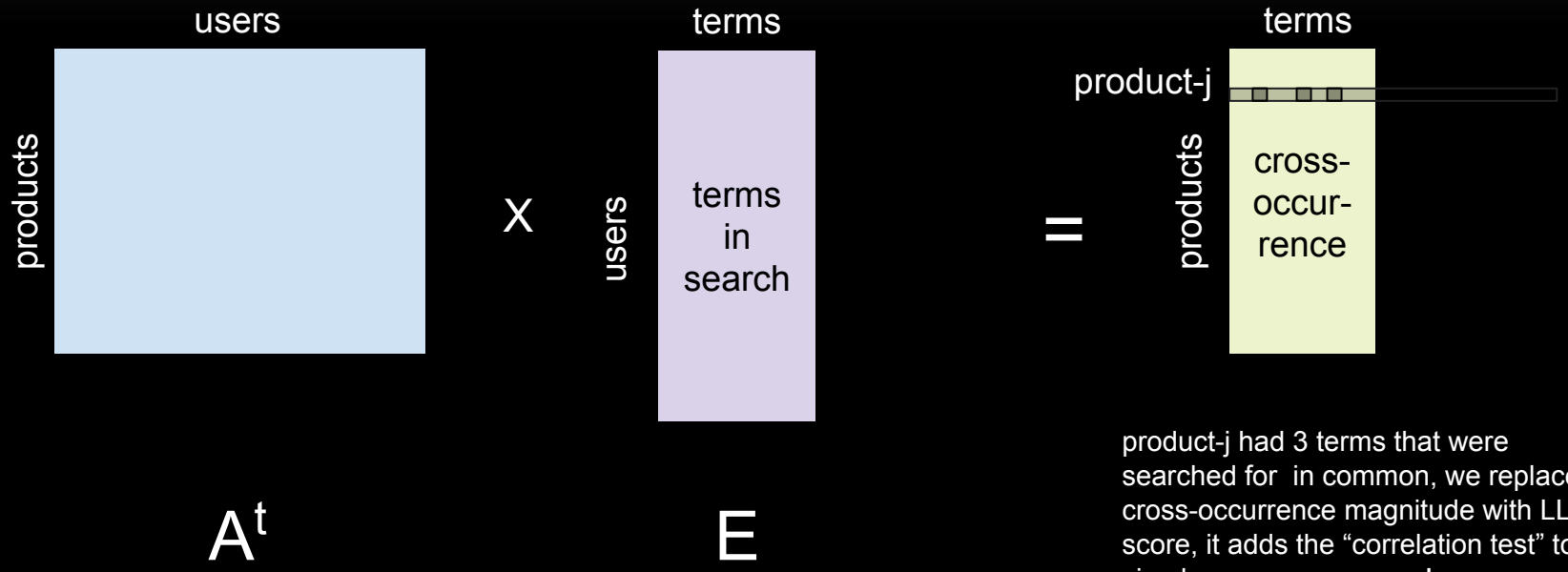
All User's Buys Cooccurrence



product-j had 2 other products that were bought in common, we replace cooccurrence magnitude with LLR score, it adds the "correlation test" to simple cooccurrence

ActionML

All User's Buys Cross-occurrence with Search terms



A^t

E

product-j had 3 terms that were searched for in common, we replace cross-occurrence magnitude with LLR score, it adds the "correlation test" to simple cross-occurrence!

ActionML

CORRELATED CROSS-OCCURRENCE: Apache Mahout-Samsara

$$r = [A^tA]h_a + [A^tB]h_b + [A^tC]h_c + \ldots$$

- Sparse Matrix Multiply, A^tA, A^tB, A^tC …
- Correlation test for non-zero,
  ie co or cross-occurring items
  with the Log-Likelihood Ratio
- All done with Apache Mahout-Samsara
- Why? One of the few libs that does general linear algebra like A^tA and A^tB in a massively scalable way and on GPUs

ActionML

CORRELATED CROSS-OCCURRENCE: The Model

product-j “bought”:

co-occurring “bought” products:	product-1, product-5, …
cross-occurring “viewed” products:	product-1, product-3, product-5, …
cross-occurring “category-preference” categories:	category-9, category-21, category-38, …
cross-occurring “shared” products:	product-50, product-99, product-301, …
cross-occurring “searched” terms:	term-10, term--21, term-49, …

user-i history of all behavior:

bought products:	product-1, product-5, …
viewed products:	product-1, product-3, product-5, …
categories-prefered:	category-9, category-21, category-38, …
shared products:	product-50, product-99, product-301, …
searched terms:	term-10, term--21, term-49, …

What do we recommend...

ActionML

CORRELATED CROSS-OCCURRENCE: K-NEAREST NEIGHBORS

$$r = [A^tA]h_a + [A^tB]h_b + [A^tC]h_c + \ldots$$

1. The dot product of two normalized (length = 1) vectors = the cosine of the angle between
2. The cosine of the angle between two vectors is the Machine Learning heavy lifter for similarity and therefore used by just about all search engines: https://en.wikipedia.org/wiki/Cosine_similarity and https://lucene.apache.org/core/3_0_3/api/core/org/apache/lucene/search/Similarity.html
3. $[A^tA]h_a$ and $[A^tB]h_b$ is the dot product of every row in the model with h_a and h_b
4. Take the sum of dot products for each item and sort them for ranking recommendations
5. Step #4 is exactly what Lucene does!
    - it is fast! using sparsity, sharding, and parallel execution of queries to accelerate
    - It is scalable and HA with Elasticsearch and Solr

ActionML

CORRELATED CROSS-OCCURRENCE: Find the most similar product to the user history

Lucene Indexes multi-field documents, one doc per product, one field per indicator:

product-j:

bought field:	product-1, product-5, …
viewed field:	product-1, product-3, product-5, …
category-preference field:	category-9, category-21, category-38, …
shared field:	product-50, product-99, product-301, …
searched field:	term-10, term--21, term-49, …

User history query

user-i history of all behavior:

bought products → bought fields:	product-1, product-5, …
viewed products → viewed field:	product-1, product-3, product-5, …
categories-prefered → category-preference field:	category-9, category-21, category-38, …
shared products → shared fields:	product-50, product-99, product-301, …
searched terms → searched field:	term-10, term--21, term-49, …

Search results:

product-j, product-k, …

ActionML

CORRELATED CROSS-OCCURRENCE: Find the most similar product to the user history

Lucene Indexes multi-field documents, one doc per product, one field per indicator:

product-j:

bought field:	product-1, product-5, …
viewed field:	product-1, product-3, product-5, …
category-preference field:	category-9, category-21, category-38, …
shared field:	product-50, product-99, product-301, …
searched field:	term-10, term--21, term-49, …

User history query

user-i history of all behavior:

bought products → bought fields:	product-1, product-5, …
viewed products → viewed field:	product-1, product-3, product-5, …
categories-prefered → category-preference field:	category-9, category-21, category-38, …
shared products → shared fields:	product-50, product-99, product-301, …
searched terms → searched field:	term-10, term--21, term-49, …

Search results:

product-j, product-k, …

Search ranks all products most similar to the user's multi-modal history.

ActionML

Uses:

- Better E-Commerce Recommender
  - sure, you saw that coming
- Search index augmentation
  - some terms that lead to conversions are not in the content like trendy slang or jargon or common misspellings
- Behavioral augmentation of search indexes
  - search terms + user history = results that might lead to a purchase
- Business Rules, it's only a query on documents
- Blend Collaborative Filtering and Content-based Recs
- With enough data?

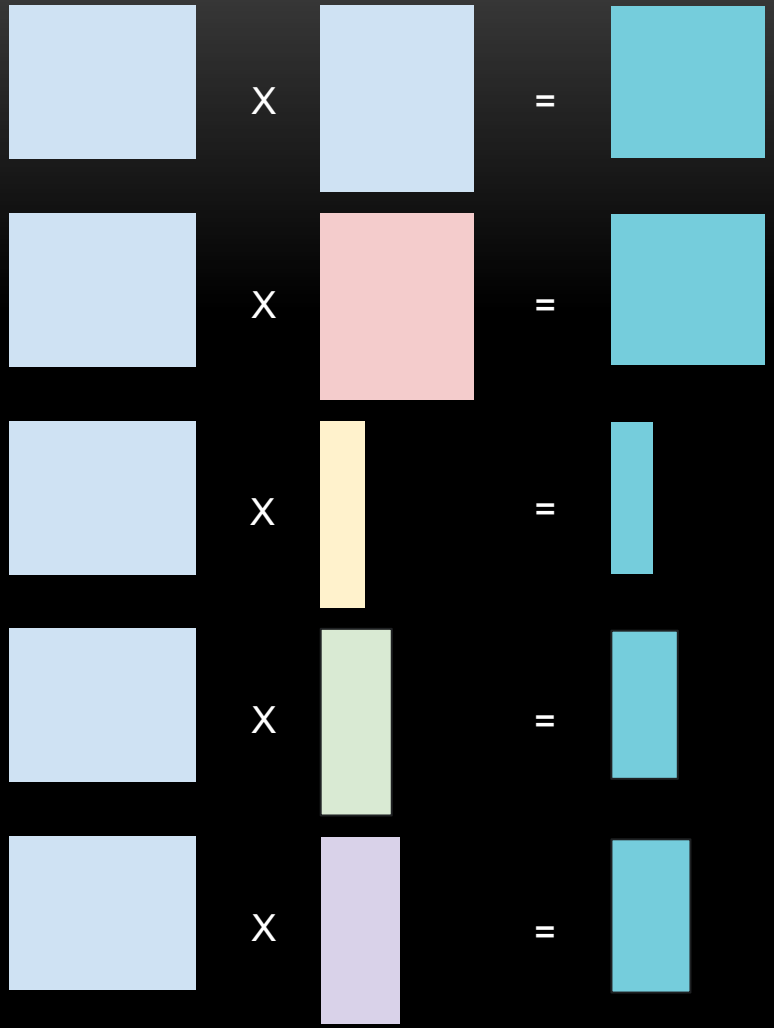
ActionML

Uses:

- Better E-Commerce Recommender
  - sure, you saw that coming
- Search index augmentation
  - some terms that lead to conversions are not in the content like trendy slang or jargon or common misspellings
- Behavioral augmentation of search indexes
  - search terms + user history = results that might lead to a purchase
- Business Rules, it's only a query on documents
- Blend Collaborative Filtering and Content-based Recs
- With enough data? Mind reading?

ActionML

Why GPUs



each matrix may be 1,000,000 x 1,000,000

calculation time is too expensive!

‘nuff said?

ActionML

Questions?

Speaker Change

Andy--give-em GPUs?

ActionML