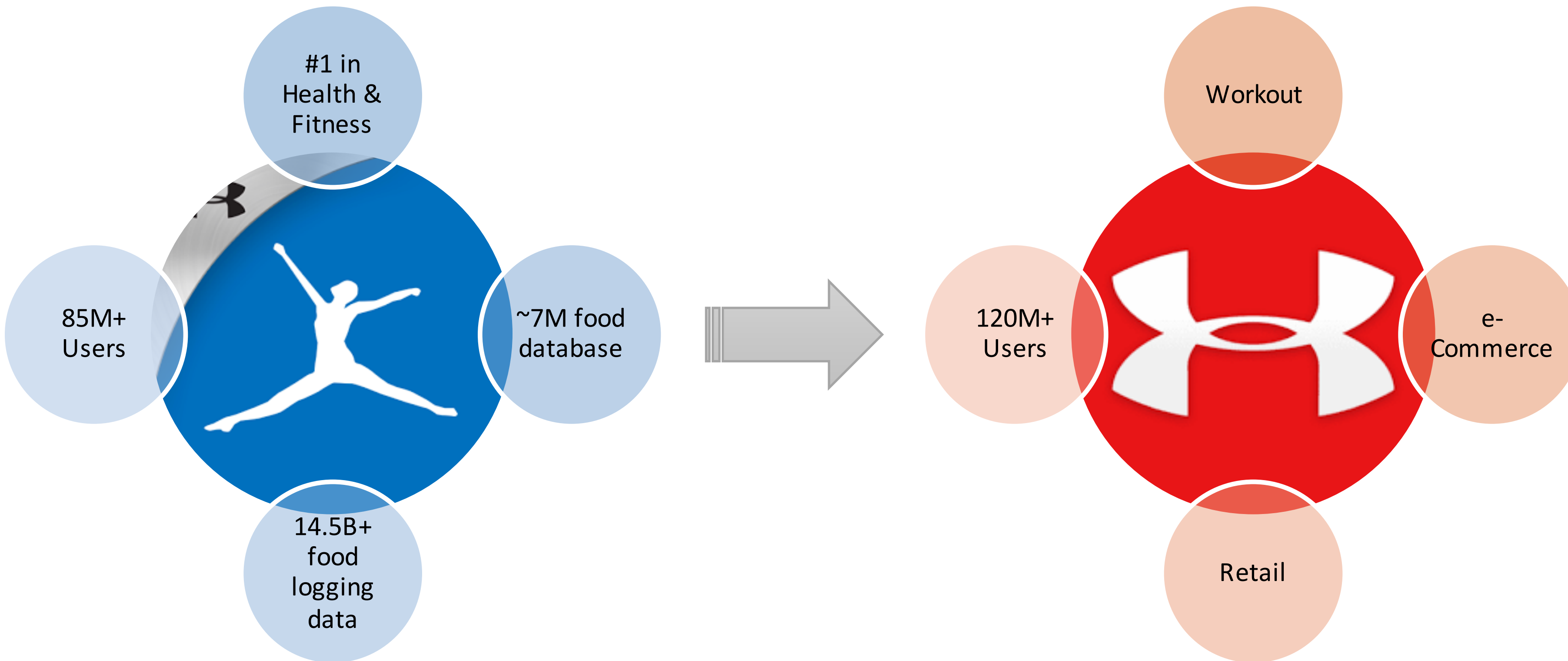
iRIS: A Large-Scale Food and Recipe Recommendation System Using Spark

Joohyun Kim

Sr. Data Scientist

MyFitnessPal – Under Armour Connected Fitness

Who are we?


#1 in Health & Fitness
85M+ Users
~7M food database
14.5B+ food logging data
Workout
120M+ Users
e-Commerce
Retail

Under Armour = Apparel Company?

- http://www.fool.com/investing/general/2015/06/07/how-under-armour-is-becoming-a-tech-company.aspx

How Under Armour Is Becoming a Tech Company

By Bradley Seth McNew | More Articles
June 7, 2015 | Comments (0)

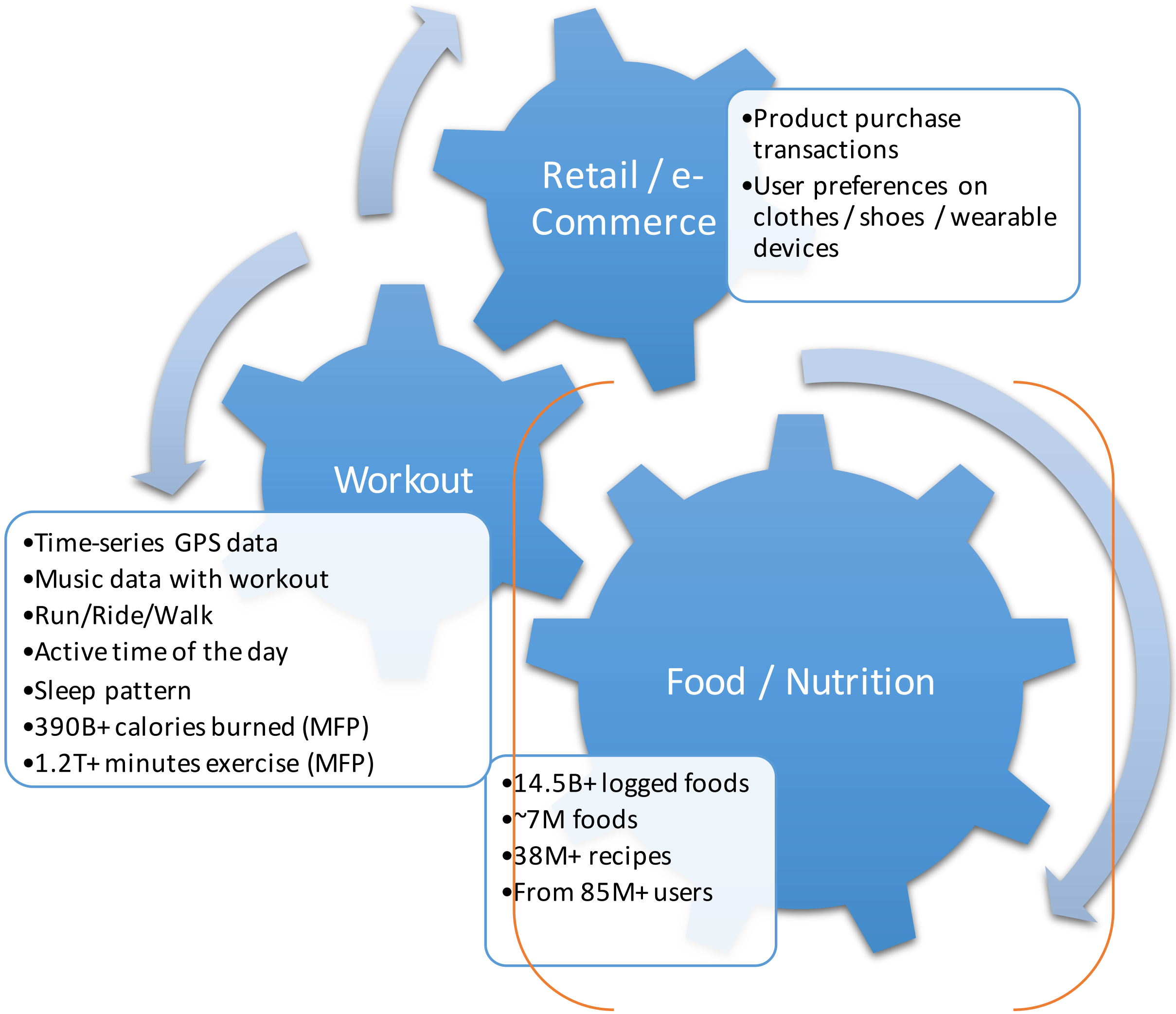
Under Armour (NYSE: UA) has grown its market share by leaps and bounds in recent years to become the second-largest athletic apparel seller in the U.S. as of 2014, behind only **Nike**.

Even more exciting is the company's continued evolution. With recent smartphone app purchases and upgrades, a push into smart apparel, and now a new digital headquarters in Austin, Texas, Under Armour is quickly becoming a technology company.

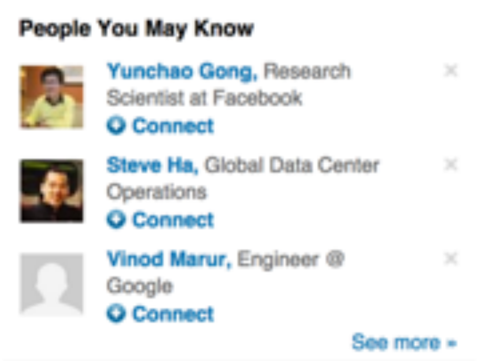
Why Under Armour spent $710 million for apps

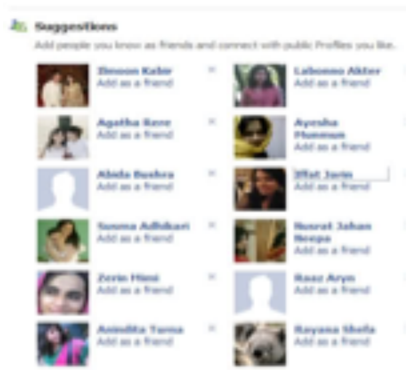
Under Armour has made 2015 the year of connected fitness apps and devices. In Q1, the company purchased two fitness-tracking apps, MyFitnessPal and Endomondo. The company's MapMyFitness, bought in 2013, also was upgraded this year with a new premium service for serious fitness tracking. In total, Under Armour spent $710 million on these apps. The company also released its own fitness app, called Record, in January.

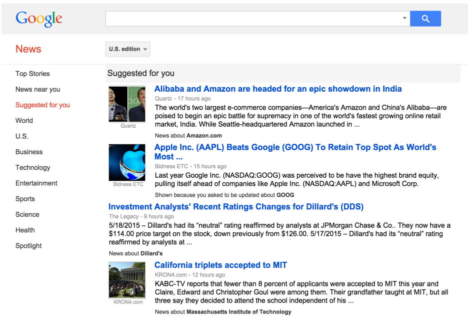
Under Armour = Data Company!

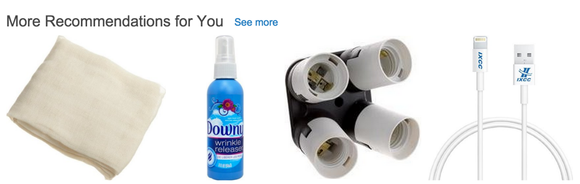


Importance of Data

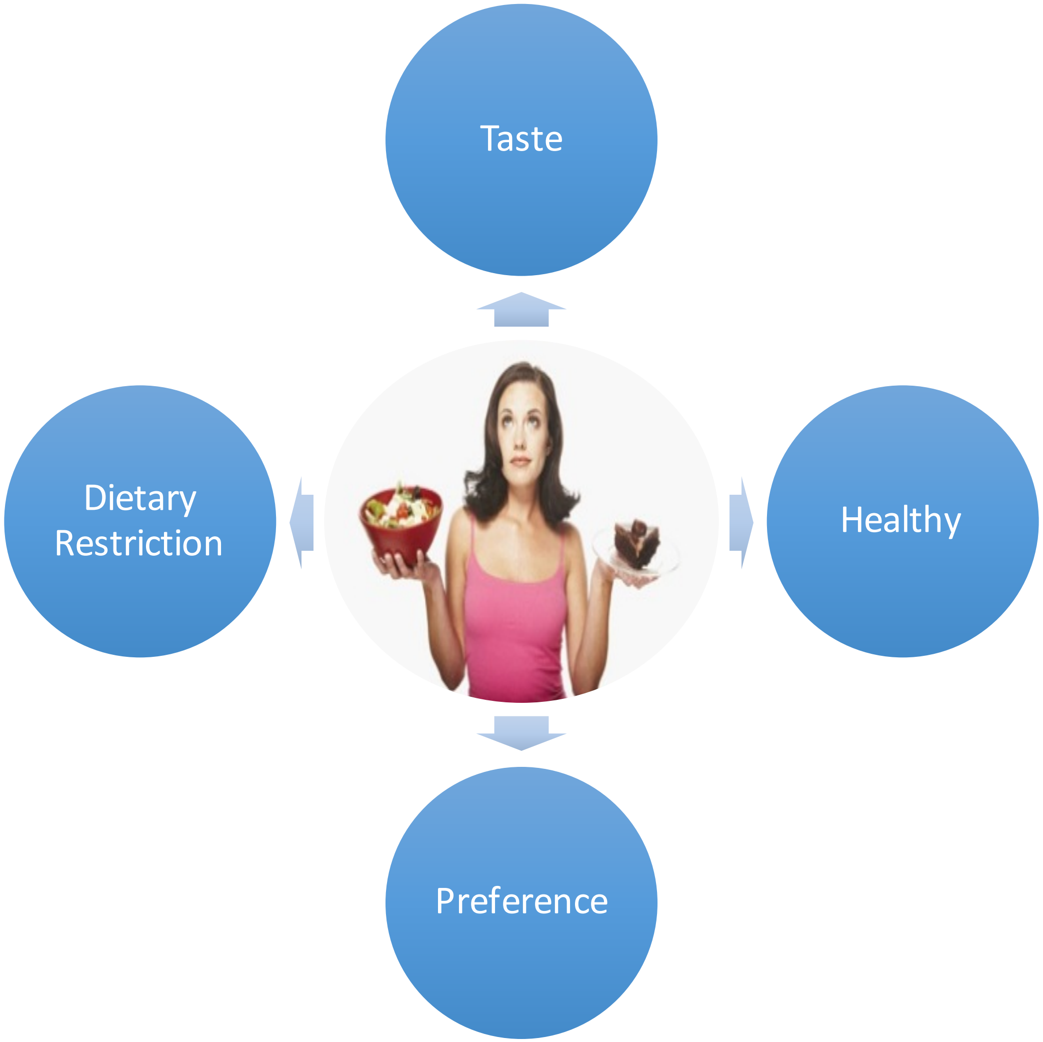








Biggest Concern of Life: What to Eat?



Recommender System?

Typical ways of getting recommendation		
Limited	Biased	Lack of Source

Collaborative Filtering

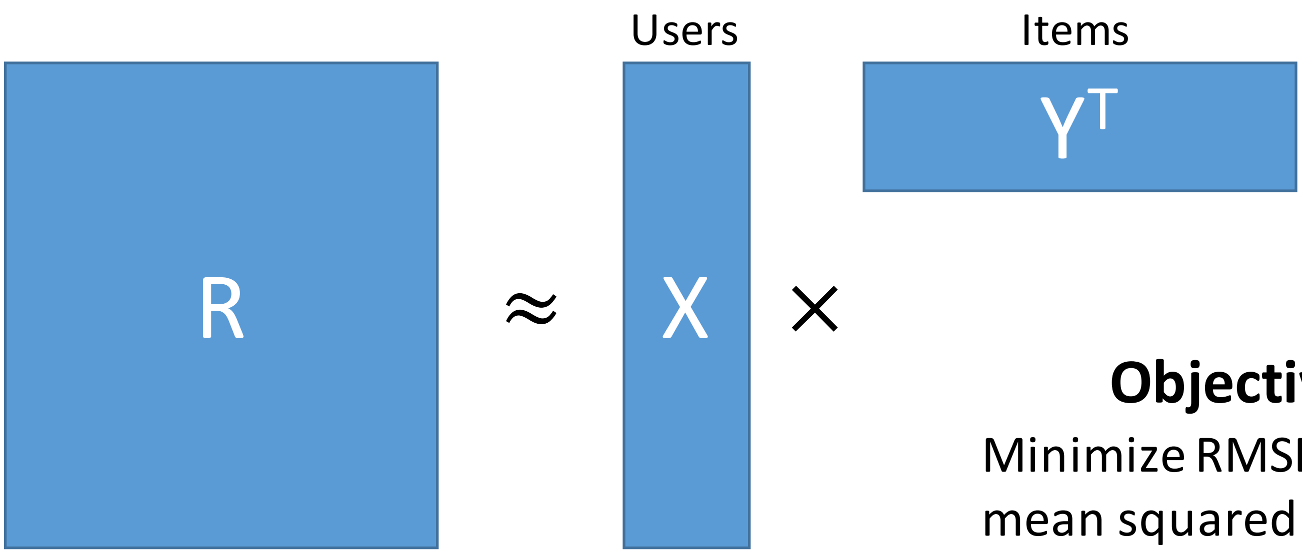
- Predict how a user may like a new item based on prior user behaviors with similar preference

User – Food Logged Counts Table

	John	Mary	Mike	Jane
Banana	5	3	1	4
Blueberry	3	-	-	2
Apple	-	-	2	- (?)
Melon	1	-	-	- (?)

Matrix Factorization

- Filling in the missing entries in ratings (user-food logged counts) matrix
  - Formulate as low-rank matrix factorization
  - Factorize user-item matrix to user-feature and feature-item matrix (# features $\ll$ # users or # items)



Objective

Minimize RMSE (Root mean squared error) between R and $X \times Y^T$

"Implicit" Matrix Factorization

- Explicit ratings are available for movies / songs
  - Typically 1~5 stars (ratings) given
- For MFP food logging events, there are only "logged" foods. No negative feedback
  - Can't assume 0 count (no entry) as negative
  - Reference: Hu, Koren, and Volinsky, **Collaborative Filtering with Implicit Feedback Dataset**, ICDM 08
- Construct "binary" ratings matrix P, and factorize P instead of R (original ratings matrix)

$$\underset{\textbf{R}}{\begin{bmatrix} 5 & 3 & ? & 3 & ? \\ ? & 2 & ? & ? & 4 \\ 1 & ? & ? & ? & ? \\ ? & ? & 1 & 2 & ? \\ ? & ? & 2 & ? & 1 \\ ? & 3 & ? & ? & 1 \end{bmatrix}} \Rightarrow \underset{\textbf{P}}{\begin{bmatrix} 1 & 1 & ? & 1 & ? \\ ? & 1 & ? & ? & 1 \\ 1 & ? & ? & ? & ? \\ ? & ? & 1 & 1 & ? \\ ? & ? & 1 & ? & 1 \\ ? & 1 & ? & ? & 1 \end{bmatrix}} \approx \underset{\textbf{X}}{\begin{bmatrix} \quad & \quad \\ & \\ & \\ & \\ & \\ & \end{bmatrix}} \times \underset{\textbf{Y}}{\begin{bmatrix} \quad & \quad & \quad & \quad & \quad \\ & & & & \end{bmatrix}}$$

Alternating Least Squares

- Optimizing X (user) and Y (item) at the same time is hard
- Fix X or Y $\Longrightarrow$ Solve for the other
  - Solve the system of linear equations
  - Take the derivative of objective function w.r.t X or Y, set 0, and solve
  - Starting with random initialization of Y
  - EM-like iterative process
- Iterate until the change is very small (or stop with fixed iteration number)

Scalability? ⇒ Parallelization!

- Rating matrix: 85M users × 7M foods ≅ 595T entries
  - Impossible to fit in a single machine
  - Sparse representation: billions of entries
- ~~ALS can be easily parallelized with map-reduce framework~~
  - ~~Sharding users and items vectors~~
  - ~~Mapper on individual sub-matrix~~
  - Reducer on aggregation over users/items
- Spark MLLib
  - Parallelized version of ALS ready to use
  - Fast computation with DataFrame

```
val model = new ALS()
              .setRank(20)
              .setImplicitPrefs(true)
              .setAlpha(40)
              .setRegParam(0.1)
              .setMaxIter(10)
              .fit(ratings)
```

Food Recommendation Pipeline

Logged foods data (user / food)

Predict food preference by matrix factorization

Generate top K food recommendation

Generating Top K Recommendations

- We need to serve top recommended foods to users
  - With the trained factorized matrix model,
  - Predict top K foods for each user (in the order of their own preference)
- Seem trivial, but the computation is huge
  - For each user, retrieve food preference by $R = X \times Y^T$
  - Get top K per each user: min(O(Kmn), O($mn \log n$))
    - m: # of users, n: # of items
    - Same order of constructing whole ratings matrix
  - Major bottleneck of the entire pipeline
  - No easy way to get around the computation

Some Numbers

- Spark cluster
  - 72 nodes (1 master + 71 workers)
  - 2TB memory ⟸ One of the largest clusters in production
- Dataset
  - User : 85M+
  - Item (food) : ~7M
  - Rating (food log counts): 6.5B+ (aggregated per user/food)
- Time
  - ALS model training: 4 hours
  - Generating top K food recommendation for every user: 48 hours
  - More than **20x** speed improvement over Mahout in conventional Hadoop cluster

Advantages Using Spark

- Faster development cycle
  - MLLib
  - Parallelization provided via RDD with abstraction
  - Easy to construct data pipeline with DataFrame
  - Easy to load / export data in and out of S3 / Redshift
- Faster model optimization
  - In-memory, distributed computation
  - Faster model training / testing
  - Significant reduction in parameter tuning / optimization on validation dataset
- Easy scalability
  - By launching more worker instances
- Enables frequent model updates
  - Reflect user preference change more often

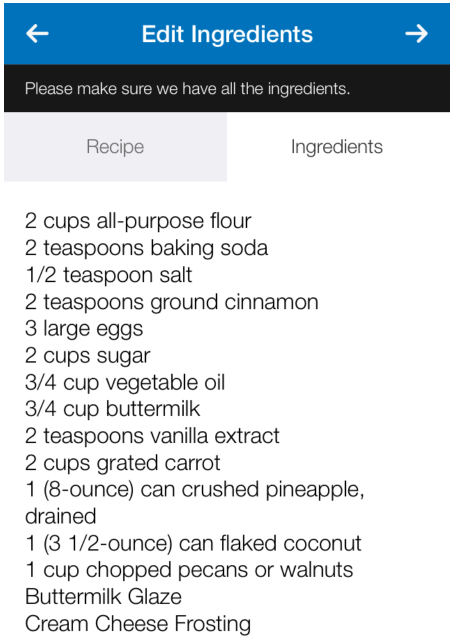
Sample Food Recommendation

Logged Foods
Korean soy milk with high calcium
Coke 12oz
Fried rice
Korean mixed grain shake
Korean Rice Cake
Sweet Soy Milk
Blackberries - Raw
Blueberries - Raw
Korean Melon (Chameh / 참외)
Japchae (Korean Stir-Fried Sweet Potato Noodles)

Recommended Foods
Cooked White Jasmine Rice
Steamed White Rice (Unenriched)
Pho
Kimchi
Tofu - Fried
Miso Soup With Seaweed and Tofu
Shrimp Dumplings
Miso Soup
Salmon Nigiri
Sunny Side Up

Extension: Recipe Recommendation

- Advantages of recommending recipes
  - Richer metadata (instructions, ingredients, cuisines, …)
  - Complete food, home-cookable
  - Customizable with personal preference/restriction
- Recipes are not "public"
  - Currently, only foods are shared across different users
  - Recipes are "private" to individual user when created
  - Cannot construct standard user-item ratings matrix
- Solution: Recommend recipes using similarity with foods



Extension: Recipe Recommendation

Logged foods data (user / food)

Predict food preference by matrix factorization

Generate top K food recommendation

Food Recommendation

Food – Recipe similarity computation (LSH, kNN, by nutrition)

Final top K recipes from food recommendation rank and recipe similarity rank

Recipe Recommendation

Nearest Neighbor with Locality Sensitive Hashing

- Naïve nearest neighbor computation: $O(n^2)$
  - Nearly impossible with 7M foods and 38M recipes
- Locality Sensitive Hashing (LSH)
  - Hashes similar items into the same buckets with high probability
  - Similarity metric: Euclidean distance on nutrition vector
  - NN computation much faster, look up within the bucket: $O(k^2)$ (k: max size of bucket)
- Spark implementation
  - https://github.com/mrsqueeze/spark-hash

Top K Recipe Recommendation

- Order by sum of food recommendation rank and recipe similarity rank

Rank	Food
1	Banana
2	Blueberry
3	Apple
4	Pad Thai
5	…
6	…

Rank	Recipe
1	Banana Pie
2	Banana Smoothie
3	Banana Ice Cream

Rank	Recipe
1	Blueberry Yogurt
2	Blueberry Pie
3	Salad with Blueberry

…

Combined Rank	Recipe
2	Banana Pie
3	Banana Smoothie
3	Blueberry Yogurt
4	Banana Ice Cream
4	Blueberry Pie
5	…

Sample Recipe Recommendations

Recommended Foods
Cooked White Jasmine Rice
Steamed White Rice (Unenriched)
Pho
Kimchi
Tofu - Fried
Miso Soup With Seaweed and Tofu
Shrimp Dumplings
Miso Soup
Salmon Nigiri
Sunny Side Up

Recommended Recipes
Noodle sauce
Stone Ground Dijon Mustard Marinade
Citrus Dijon Miso Dressing
Shirataki Noodle Soup
dumpling sauce
Dijon Miracle Whip
Paleo Vanilla Ice-Crème
Mable's Chili Burrito
cake batter milkshake
Pita pizza

Integrating with Taste Profiles

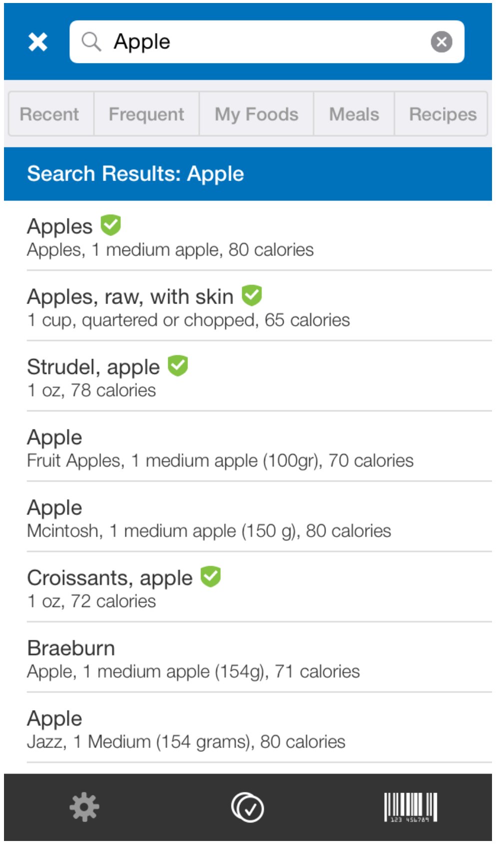
- Machine learning classifier that outputs probability distribution over 6 taste categories
  - Savory, Sweet, Sour, Spicy, Salty, Bitter
  - NN classifier performed over feature vector (semantic word vector + numeric nutritional value vector)
  - With small number (~1400) of labeled foods
  - 87% accuracy on separately labeled test set (~1200)
  - Works as additional metadata for foods
- Recommendation results can be further filtered / reordered by personal taste preferences
- With Spark, data integration with hash-join is much faster

Problems

- Cold-start problem
  - New user (important)
  - New food item (not too much, since they may not be popular)
- Solution
  - Hybrid with content-based recommendation
    - Construct basic profile for new users to get baseline recommendation
    - May recommend new items based on feature similarity
  - Runs the pipeline more often
    - For new users, recommend top/popular foods
    - After a while, these users will get recommendation

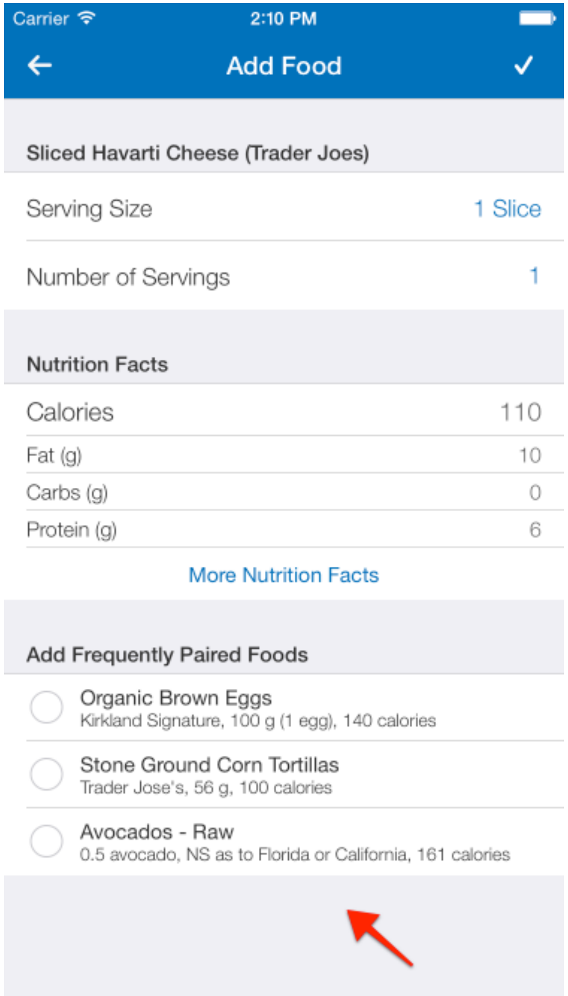
Possible Extension

- Collaborative filtering over aggregated food clusters
  - User-generated foods contain (near) duplicates
  - Combine with verified food project
    - Spark-based data deduplication / processing pipeline
    - Construct “verified” foods out of thorough clustering
  - Recommend users with high-quality food
  - Recommend representative food from each cluster, instead of individual variation
  - Reducing the computation time due to reduced dimension



Application

- Recommend frequently paired foods
- Pairing foods within a single meal depends on
  - Individual user's own preference
  - Cultural difference (region, country)
- Simple way
  - Suggest popular foods based on co-occurrence stats per individual user / overall users
- Utilize this framework to capture better personalized preference



Summary

- Spark-powered machine learning pipeline for food/recipe recommendation system
  - Faster computation help reduce the time on development cycle
  - Help data scientists focus on core problems
  - Easy extension by attaching additional data processing steps with scalability
- Only scratched a surface
  - Food / recipe recommendation
  - Extensions with other data sources
    - Workout
    - Music
    - Retail
    - e-Commerce

Questions?