ML for System

Jinming Hu

2021.6.5

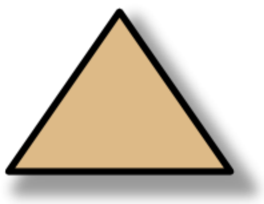
Many slides from Tim Kraska

Brief intro to ML

- Supervised learning: inputs $\boldsymbol{x}$ to outputs $\boldsymbol{y}$
- Classification: y is a categorical variable
- Regression: y is real-valued
  - This is more often used in learned index

Fundamental Building Blocks

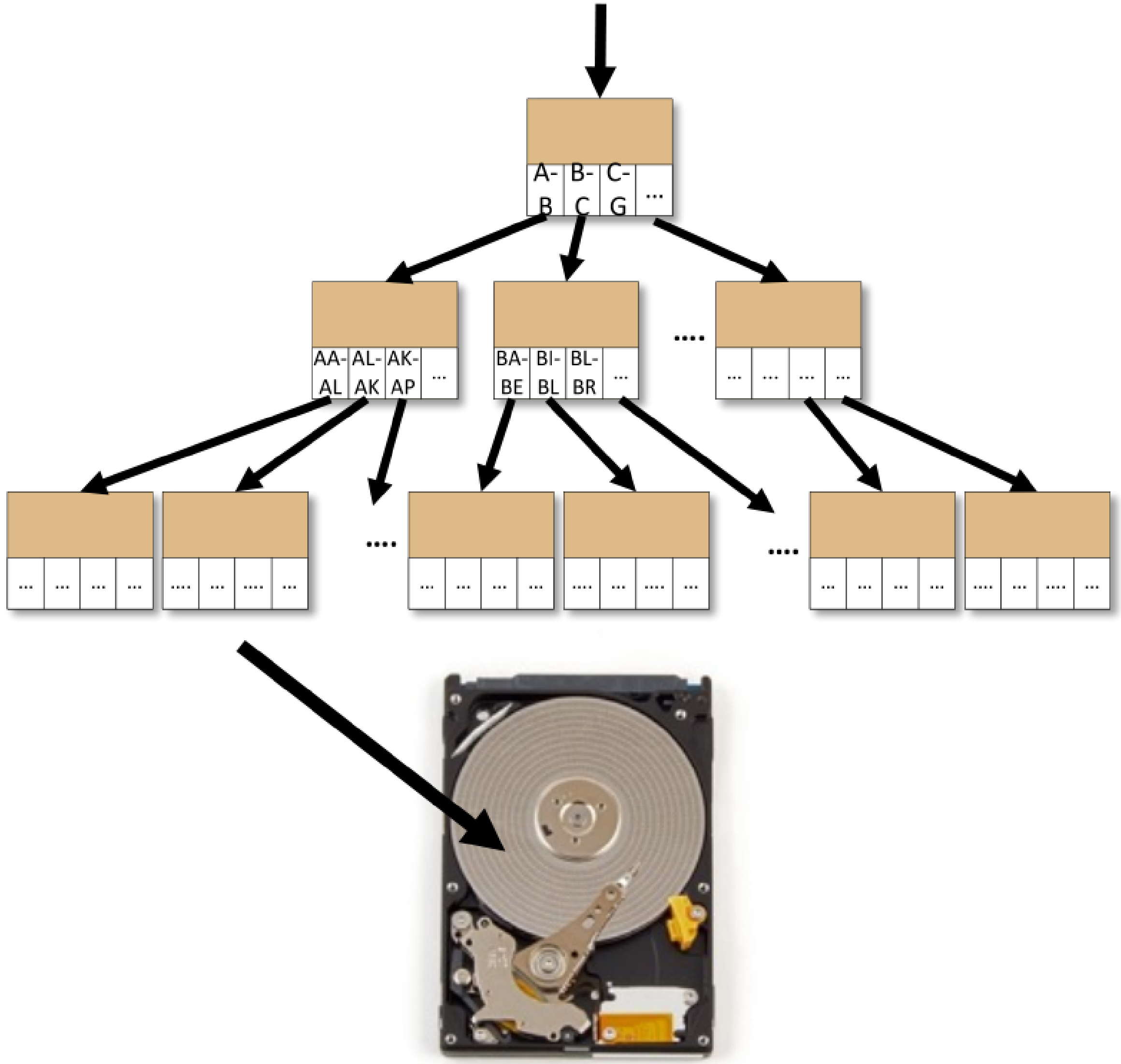
Fundamental Algorithms & Data Structures



Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, Neoklis Polyzotis:
The Case for Learned Index Structures. SIGMOD Conference 2018: 489-504

Key

(e.g., spoon #1)



Key

(e.g., spoon #1)

Model

How To Build Models To Predict the Location

id	date	first_name	last_name	email	address	zip	state	credit_card_nb	amount
1000	2017-01-01	Hobart	Spracklin	hspracklin0@dailymotion.com	20565 High Crossing Plaza	56372	Minnesota	4405-6975-7285-5160	$ 611.00
1001	2017-01-02	Billye	Binnion	bbinnion1@123-reg.co.uk	3698 Upham Point	20260	District of Columbia	3533-7150-7728-9850	$ 244.00
1002	2017-01-02	Johann	Brockley	jbrockley2@bizjournals.com	23844 Artisan Place	98516	Washington	67597-1193-7985-5100	$ 233.00
1003	2017-01-03	Artie	MacMenami	amacmenamin3@hao123.com	6276 Toban Trail	78759	Texas	3537-4829-6134-5000	$ 210.00
1004	2017-01-03	Delilah	O'Currigan	docurrigan4@chron.com	86016 New Castle Avenue	72199	Arkansas	3555-2017-2226-5780	$ 286.00
1005	2017-01-04	Gretta	Will	gwill5@yelp.com	0 Dottie Circle	68524	Nebraska	503844-1984-2085-5000	$ 870.00
1006	2017-01-04	Gordon	Kirsopp	gkirsopp6@utexas.edu	64060 Scott Park	20370	District of Columbia	633332-1895-2414-5000	$ 687.00
1007	2017-01-05	Bendick	Fagg	bfagg7@army.mil	94 Florence Hill	45440	Ohio	3528-9673-1815-8420	$ 733.00
1008	2017-01-05	Dimitry	Boyet	dboyet8@sakura.ne.jp	35886 Golf Plaza	30066	Georgia	3576-6991-4041-3170	$ 382.00
1009	2017-01-06	Ailsun	Beinke	abeinke9@si.edu	1 Badeau Place	46295	Indiana	56022-2011-8072-1400	$ 854.00
1010	2017-01-07	Lou	Hallows	lhallowsa@theguardian.com	1 Twin Pines Junction	91125	California	5602-2364-4079-0250	$ 150.00
1011	2017-01-09	Tiffani	Mathew	tmathewb@seattletimes.com	0456 Meadow Vale Lane	75260	Texas	6387-6943-8910-4580	$ 313.00
1012	2017-01-09	Perl	Bridie	pbridiec@hubpages.com	07 Bluestem Junction	33124	Florida	3539-8662-2397-5880	$ 558.00
1013	2017-01-09	Rosabelle	Blasik	rblasikd@delicious.com	7 Fairfield Pass	79699	Texas	5602-2297-6599-8560	$ 941.00
1014	2017-01-10	Meggi	Belamy	mbelamye@ask.com	0995 Manufacturers Street	10170	New York	3557-5094-7405-8340	$ 875.00
1015	2017-01-10	Tadio	Balderston	tbalderstonf@apache.org	80 Novick Road	75260	Texas	60485-3728-7119-9300	$ 954.00
1016	2017-01-11	Gianina	Oxteby	goxtebyg@google.pl	72674 Fuller Avenue	89505	Nevada	4-0415-9268-2397	$ 239.00
1017	2017-01-12	Brendan	Doody	bdoodyh@craigslist.org	87414 Golden Leaf Street	11480	New York	201-6348-4121-1314	$ 308.00
1018	2017-01-13	Conway	Coombs	ccoombsi@blogger.com	2810 Oakridge Park	32859	Florida	3529-1514-0357-9120	$ 60.00
1019	2017-01-14	Germaine	Bere	gberej@bravesites.com	82802 Oakridge Park	20041	District of Columbia	670961-0240-4054-9000	$ 95.00
1020	2017-01-15	Davide	Tolcharde	dtolchardek@redcross.org	89 Continental Avenue	79165	Texas	5018-7748-4325-9510	$ 137.00
1021	2017-01-16	Nigel	Artharg	nartharg l@gizmodo.com	31 Mcbride Point	22301	Virginia	560225-6965-2870-0000	$ 496.00
1022	2017-01-17	Rickard	Trenholm	rtrenholmm@cbslocal.com	93 Hoepker Parkway	70593	Louisiana	3541-5241-5383-9970	$ 760.00
1023	2017-01-18	Juditha	Dwane	jdwanen@vk.com	7914 Eliot Lane	14276	New York	5456-4410-0914-3180	$ 474.00
1024	2017-01-19	Susan	Ilden	sildeno@aol.com	25204 Huxley Road	21684	Maryland	3574-8586-6367-9920	$ 83.00
1025	2017-01-20	Abbey	Triggle	atrigglep@google.com.au	47 Debra Pass	74184	Oklahoma	3538-6047-6315-7710	$ 513.00
1026	2017-01-21	Zsazsa	Dunster	zdunsterq@nature.com	7 Gerald Alley	40576	Kentucky	3562-0325-7709-3490	$ 952.00
1027	2017-01-22	Grantham	Friatt	gfriattr@seattletimes.com	774 Prairieview Circle	29225	South Carolina	3571-1171-9476-8780	$ 942.00
1028	2017-01-22	Ross	Gaudin	rgaudins@samsung.com	3102 Loeprich Trail	68197	Nebraska	5108-7578-4665-2710	$ 572.00
1029	2017-01-22	Aluino	Drover	adrovert@dagondesign.com	2717 Northridge Avenue	72199	Arkansas	670999-3171-8848-0000	$ 318.00
1030	2017-01-23	Shurlock	Braker	sbrakeru@huffingtonpost.com	30783 Jenna Alley	80945	Colorado	6331106-1894-9878-0000	$ 166.00
1031	2017-01-24	Glenda	Goodbody	ggoodbodyv@economist.com	720 Pierstorff Way	7522	New Jersey	36-0593-2719-1684	$ 412.00
1032	2017-01-24	Rollin	Reddie	rreddiew@tinypic.com	09 Gina Park	65810	Missouri	4665-9188-1324-1040	$ 383.00
1033	2017-01-26	Dorry	Jenks	djenksx@virginia.edu	1 Butterfield Road	85210	Arizona	3578-9195-0297-7730	$ 636.00
1034	2017-01-26	Patti	Emby	pembyy@weather.com	26 Hoard Drive	91210	California	3585-8243-7506-2470	$ 957.00

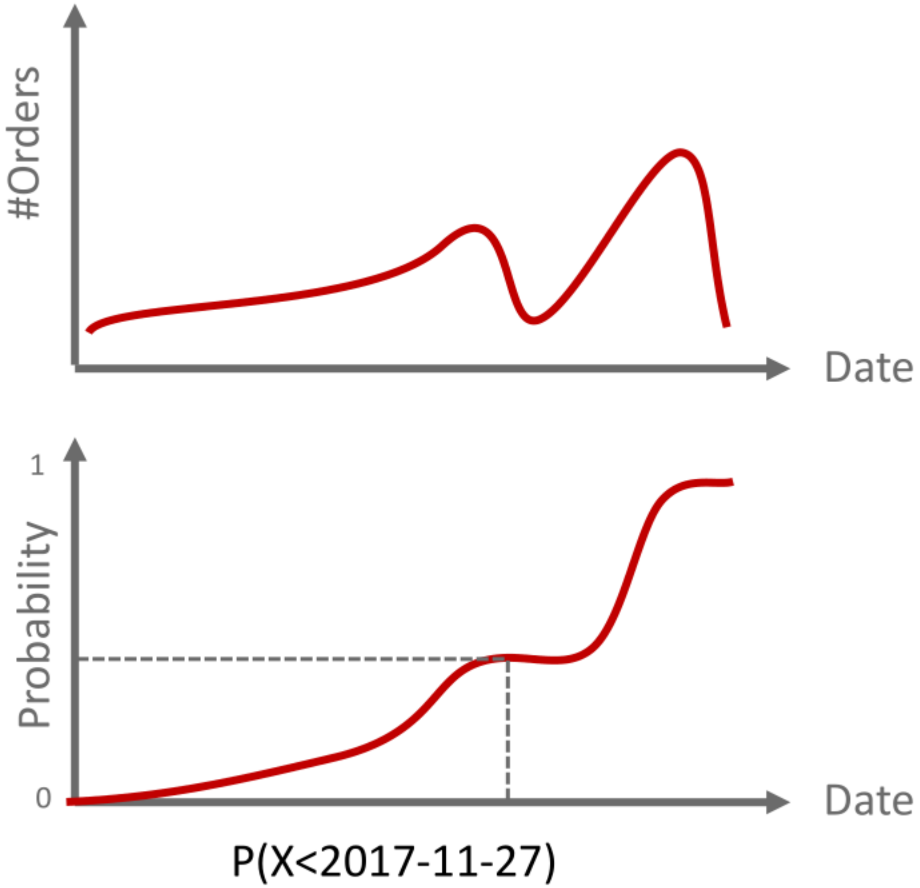
How To Build Models To Predict the Location

date
2017-01-01
2017-01-02
2017-01-02
2017-01-03
2017-01-03
2017-01-04
2017-01-04
2017-01-05
2017-01-05
2017-01-06
2017-01-07
2017-01-09
2017-01-09
2017-01-09
2017-01-10
2017-01-10
2017-01-11
2017-01-12
2017-01-13
2017-01-14
2017-01-15
2017-01-16
2017-01-17
2017-01-18
2017-01-19
2017-01-20
2017-01-21
2017-01-22
2017-01-22
2017-01-22
2017-01-23
2017-01-24
2017-01-24
2017-01-26
2017-01-26

How To Build Models To Predict the Location

Frequency Distribution

Cumulative Distribution Function (CDF)

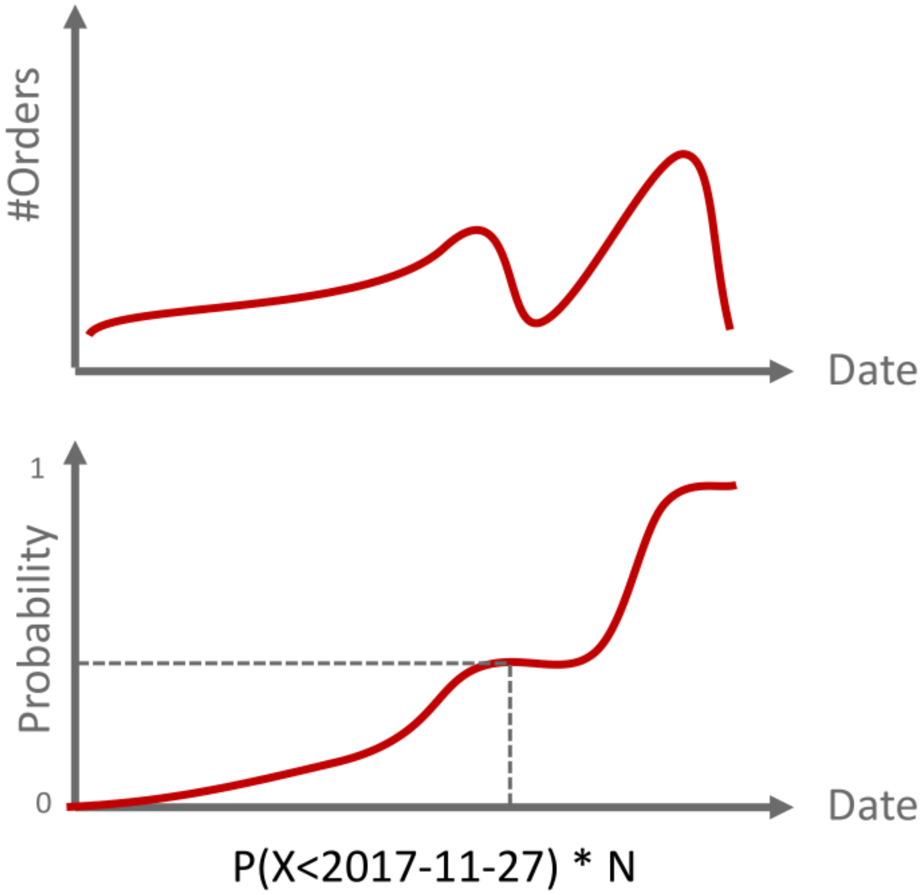


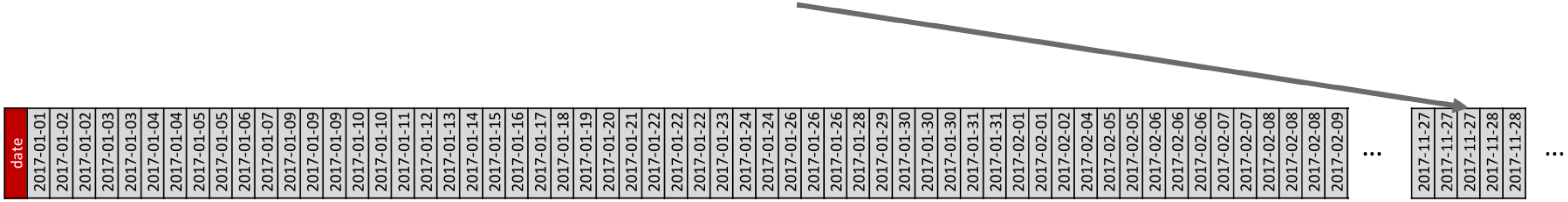
date
2017-01-01
2017-01-02
2017-01-02
2017-01-03
2017-01-03
2017-01-04
2017-01-04
2017-01-05
2017-01-05
2017-01-06
2017-01-07
2017-01-09
2017-01-09
2017-01-09
2017-01-10
2017-01-10
2017-01-11
2017-01-12
2017-01-13
2017-01-14
2017-01-15
2017-01-16
2017-01-17
2017-01-18
2017-01-19
2017-01-20
2017-01-21
2017-01-22
2017-01-22
2017-01-22
2017-01-23
2017-01-24
2017-01-24
2017-01-26
2017-01-26
2017-01-26
2017-01-28
2017-01-29
2017-01-30
2017-01-30
2017-01-30
2017-01-31
2017-01-31
2017-02-01
2017-02-01
2017-02-02
2017-02-04
2017-02-05
2017-02-05
2017-02-06
2017-02-06
2017-02-06
2017-02-07
2017-02-07
2017-02-08
2017-02-08
2017-02-08
2017-02-09
2017-02-10
2017-02-10
2017-02-11
2017-02-12
2017-02-13
2017-02-13
2017-02-14
2017-02-14
2017-02-15

How To Build Models To Predict the Location

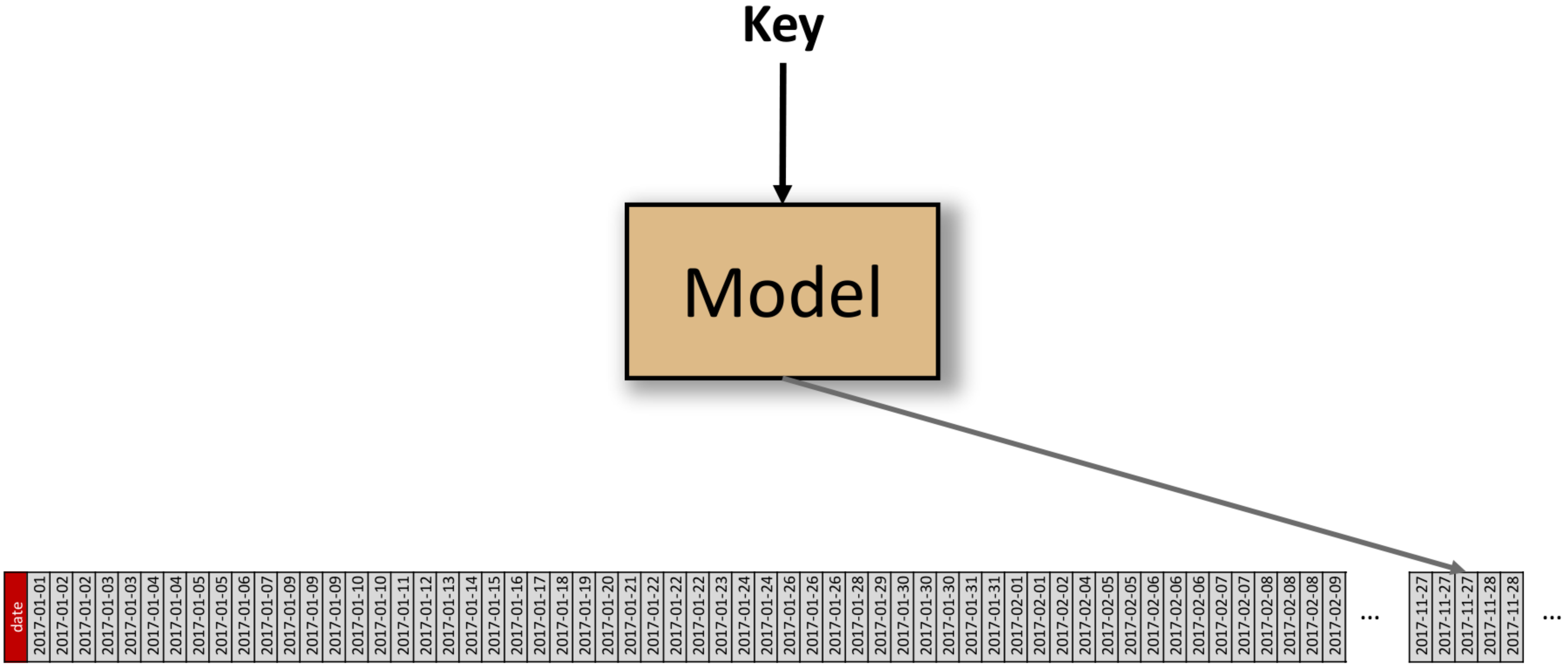
Frequency Distribution

Cumulative Distribution Function (CDF)

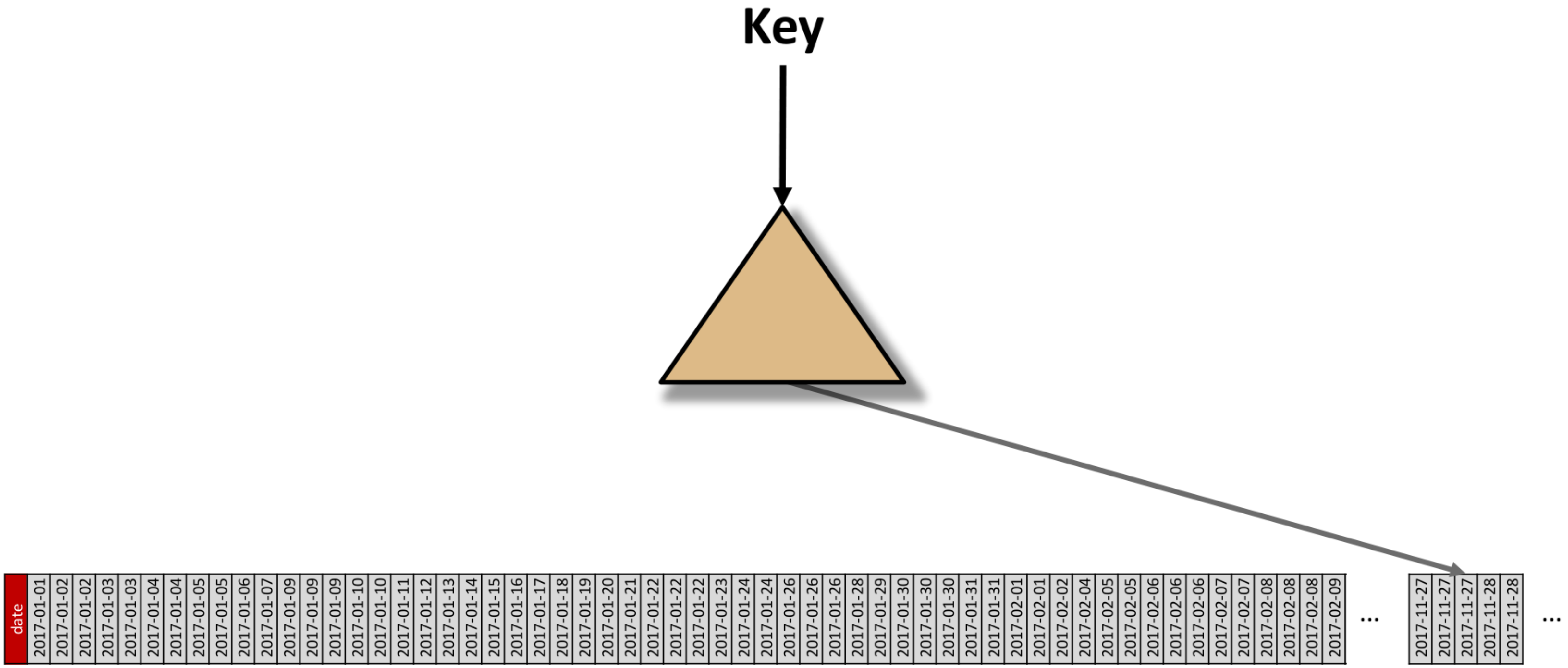




CDF Model

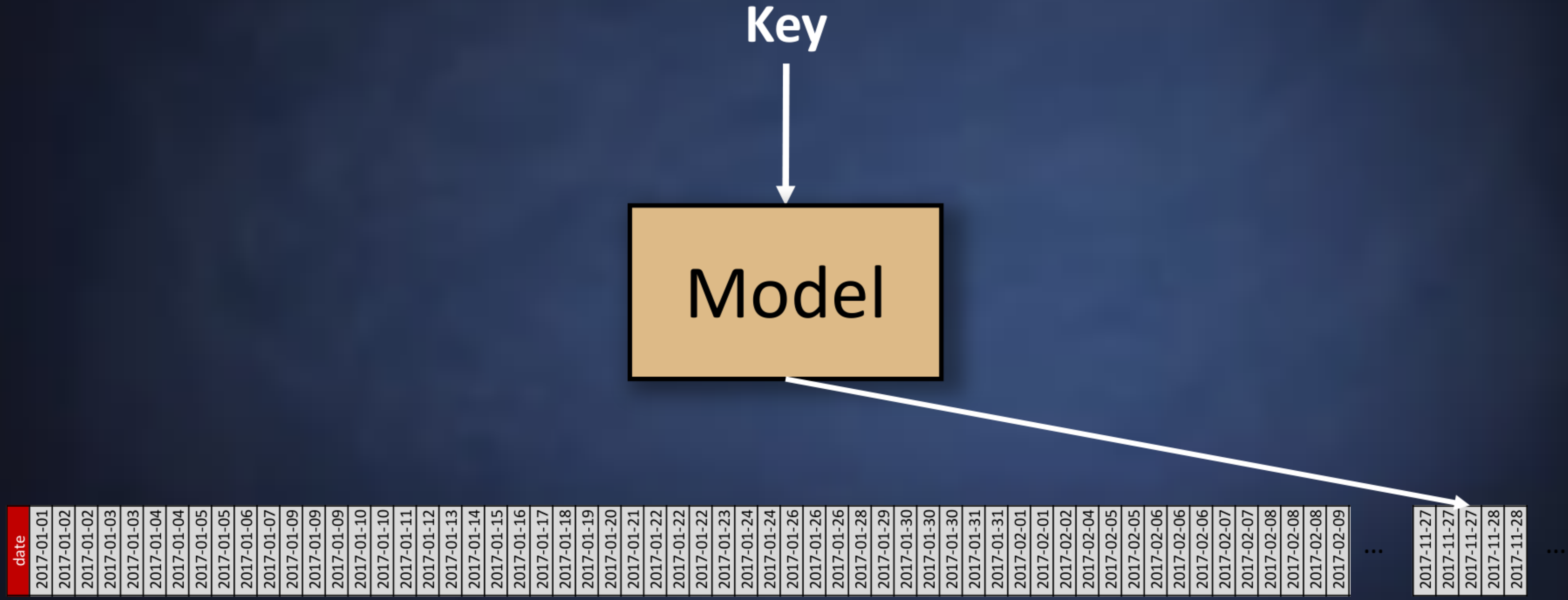

Key
Model
date
2017-01-01
2017-01-02
2017-01-02
2017-01-03
2017-01-03
2017-01-04
2017-01-04
2017-01-05
2017-01-05
2017-01-06
2017-01-07
2017-01-09
2017-01-09
2017-01-09
2017-01-10
2017-01-10
2017-01-11
2017-01-12
2017-01-13
2017-01-14
2017-01-15
2017-01-16
2017-01-17
2017-01-18
2017-01-19
2017-01-20
2017-01-21
2017-01-22
2017-01-22
2017-01-22
2017-01-23
2017-01-24
2017-01-24
2017-01-26
2017-01-26
2017-01-26
2017-01-28
2017-01-29
2017-01-30
2017-01-30
2017-01-30
2017-01-31
2017-01-31
2017-02-01
2017-02-01
2017-02-02
2017-02-04
2017-02-05
2017-02-05
2017-02-06
2017-02-06
2017-02-07
2017-02-07
2017-02-08
2017-02-08
2017-02-09
...
2017-11-27
2017-11-27
2017-11-27
2017-11-28
2017-11-28
...

B-Trees are also models (regression tree)



What Does This Mean

Why Is This A Big Deal?

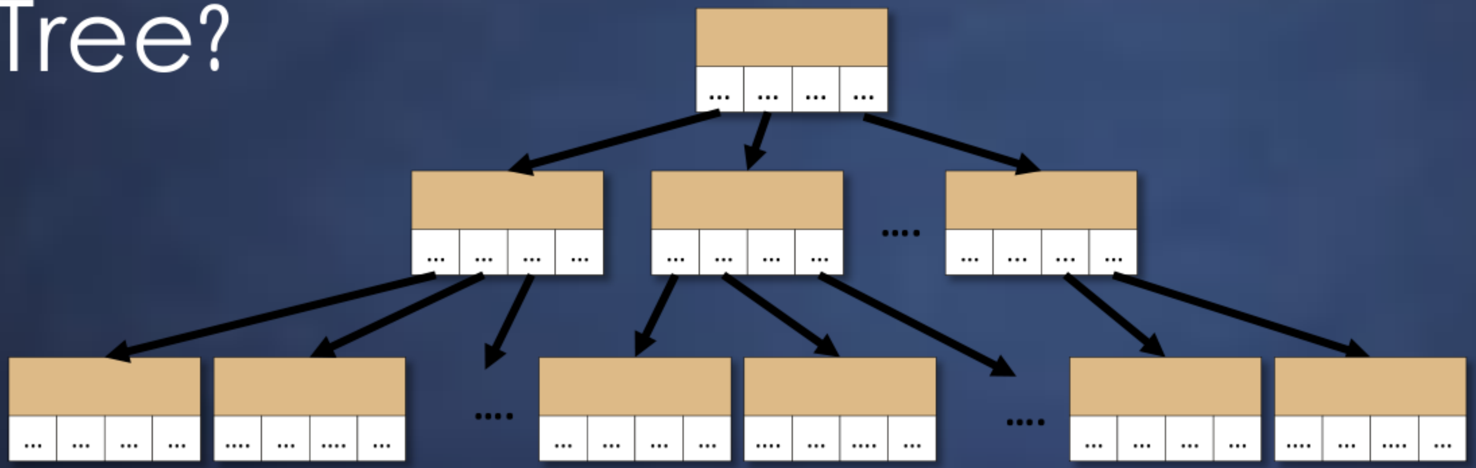


Adaptation To Application Data

id	date	first_name	last_name	email	address	zip	state	credit_card_nb	amount
1000	2017-01-01	Hobart	Spracklin	hspracklin0@dailymotion.com	20565 High Crossing Plaza	56372	Minnesota	4405-6975-7285-5160	$ 611.00
1001	2017-01-02	Billye	Binnion	bbinnion1@123-reg.co.uk	3698 Upham Point	20260	District of Columbia	3533-7150-7728-9850	$ 244.00
1002	2017-01-02	Johann	Brockley	jbrockley2@bizjournals.com	23844 Artisan Place	98516	Washington	67597-1193-7985-5100	$ 233.00
1003	2017-01-03	Artie	MacMenami	amacmenamin3@hao123.com	6276 Toban Trail	78759	Texas	3537-4829-6134-5000	$ 210.00
1004	2017-01-03	Delilah	O'Currigan	docurrigan4@chron.com	86016 New Castle Avenue	72199	Arkansas	3555-2017-2226-5780	$ 286.00
1005	2017-01-04	Gretta	Will	gwill5@yelp.com	0 Dottie Circle	68524	Nebraska	503844-1984-2085-5000	$ 870.00
1006	2017-01-04	Gordon	Kirsopp	gkirsopp6@utexas.edu	64060 Scott Park	20370	District of Columbia	633332-1895-2414-5000	$ 687.00
1007	2017-01-05	Bendick	Fagg	bfagg7@army.mil	94 Florence Hill	45440	Ohio	3528-9673-1815-8420	$ 733.00
1008	2017-01-05	Dimitry	Boyet	dboyet8@sakura.ne.jp	35886 Golf Plaza	30066	Georgia	3576-6991-4041-3170	$ 382.00
1009	2017-01-06	Ailsun	Beinke	abeinke9@si.edu	1 Badeau Place	46295	Indiana	56022-2011-8072-1400	$ 854.00
1010	2017-01-07	Lou	Hallows	lhallowsa@theguardian.com	1 Twin Pines Junction	91125	California	5602-2364-4079-0250	$ 150.00
1011	2017-01-09	Tiffani	Mathew	tmathewb@seattletimes.com	0456 Meadow Vale Lane	75260	Texas	6387-6943-8910-4580	$ 313.00
1012	2017-01-09	Perl	Bridie	pbridiec@hubpages.com	07 Bluestem Junction	33124	Florida	3539-8662-2397-5880	$ 558.00
1013	2017-01-09	Rosabelle	Blasik	rblasikd@delicious.com	7 Fairfield Pass	79699	Texas	5602-2297-6599-8560	$ 941.00
1014	2017-01-10	Meggi	Belamy	mbelamye@ask.com	0995 Manufacturers Street	10170	New York	3557-5094-7405-8340	$ 875.00
1015	2017-01-10	Tadio	Balderston	tbalderstonf@apache.org	80 Novick Road	75260	Texas	60485-3728-7119-9300	$ 954.00
1016	2017-01-11	Gianina	Oxteby	goxtebyg@google.pl	72674 Fuller Avenue	89505	Nevada	4-0415-9268-2397	$ 239.00
1017	2017-01-12	Brendan	Doody	bdoodyh@craigslist.org	87414 Golden Leaf Street	11480	New York	201-6348-4121-1314	$ 308.00
1018	2017-01-13	Conway	Coombs	ccoombsi@blogger.com	2810 Oakridge Park	32859	Florida	3529-1514-0357-9120	$ 60.00
1019	2017-01-14	Germaine	Bere	gberej@bravesites.com	82802 Oakridge Park	20041	District of Columbia	670961-0240-4054-9000	$ 95.00
1020	2017-01-15	Davide	Tolcharde	dtolchardek@redcross.org	89 Continental Avenue	79165	Texas	5018-7748-4325-9510	$ 137.00
1021	2017-01-16	Nigel	Artharg	nartharg l@gizmodo.com	31 Mcbride Point	22301	Virginia	560225-6965-2870-0000	$ 496.00
1022	2017-01-17	Rickard	Trenholm	rtrenholmm@cbslocal.com	93 Hoepker Parkway	70593	Louisiana	3541-5241-5383-9970	$ 760.00
1023	2017-01-18	Juditha	Dwane	jdwanen@vk.com	7914 Eliot Lane	14276	New York	5456-4410-0914-3180	$ 474.00
1024	2017-01-19	Susan	Ilden	sildeno@aol.com	25204 Huxley Road	21684	Maryland	3574-8586-6367-9920	$ 83.00
1025	2017-01-20	Abbey	Triggle	atrigglep@google.com.au	47 Debra Pass	74184	Oklahoma	3538-6047-6315-7710	$ 513.00
1026	2017-01-21	Zsazsa	Dunster	zdunsterq@nature.com	7 Gerald Alley	40576	Kentucky	3562-0325-7709-3490	$ 952.00

Adaptation To Application Data

B–Tree?



Adaptation To Application Data

id	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066

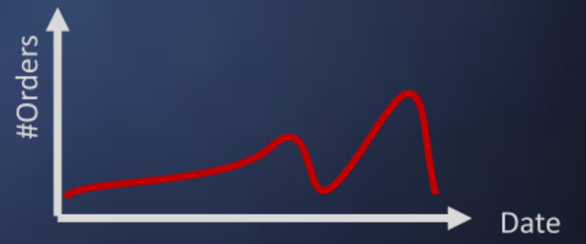
…

```
data_array[id - 1000]
```

date	2017-01-01	2017-01-02	2017-01-02	2017-01-03	2017-01-03	2017-01-04	2017-01-04	2017-01-05	2017-01-05	2017-01-06	2017-01-07	2017-01-09	2017-01-09	2017-01-09	2017-01-10	2017-01-10	2017-01-11	2017-01-12	2017-01-13	2017-01-14	2017-01-15	2017-01-16	2017-01-17	2017-01-18	2017-01-19	2017-01-20	2017-01-21	2017-01-22	2017-01-22	2017-01-22	2017-01-23	2017-01-24	2017-01-24	2017-01-26	2017-01-26	2017-01-26	2017-01-28	2017-01-29	2017-01-30	2017-01-30	2017-01-30	2017-01-31	2017-01-31	2017-02-01	2017-02-01	2017-02-02	2017-02-04	2017-02-05	2017-02-05	2017-02-06	2017-02-06	2017-02-06	2017-02-07	2017-02-07	2017-02-08	2017-02-08	2017-02-08	2017-02-09	2017-02-10	2017-02-10	2017-02-11	2017-02-12	2017-02-13	2017-02-13	2017-02-14	2017-02-14	2017-02-15

…

```
data_array[model(date)]
```



Does It Work? A First Attempt

State-Of-The-Art
B-Tree

260ns

TensorFlow

???

Does It Work? A First Attempt

State-Of-The-Art
B-Tree

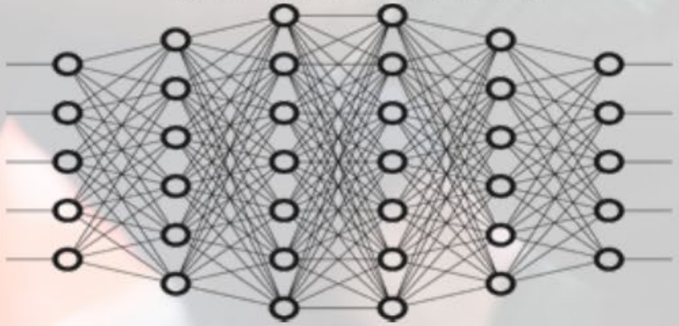
260ns

TensorFlow

>80,000ns

Challenges

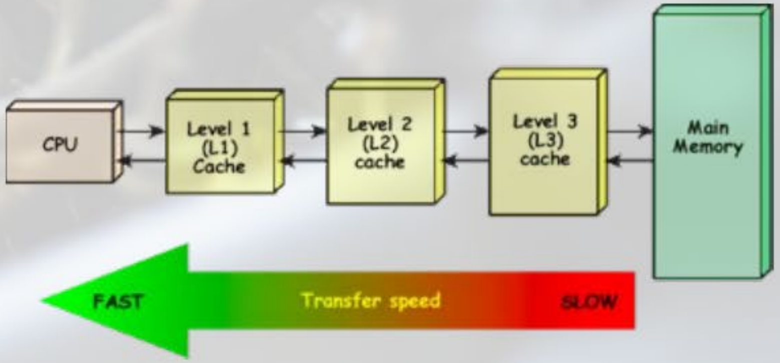
Traditional model architectures do not work



Frameworks are not designed for nano-second execution

Overfitting can be good

ML+System Co-Design



What model type should I use?

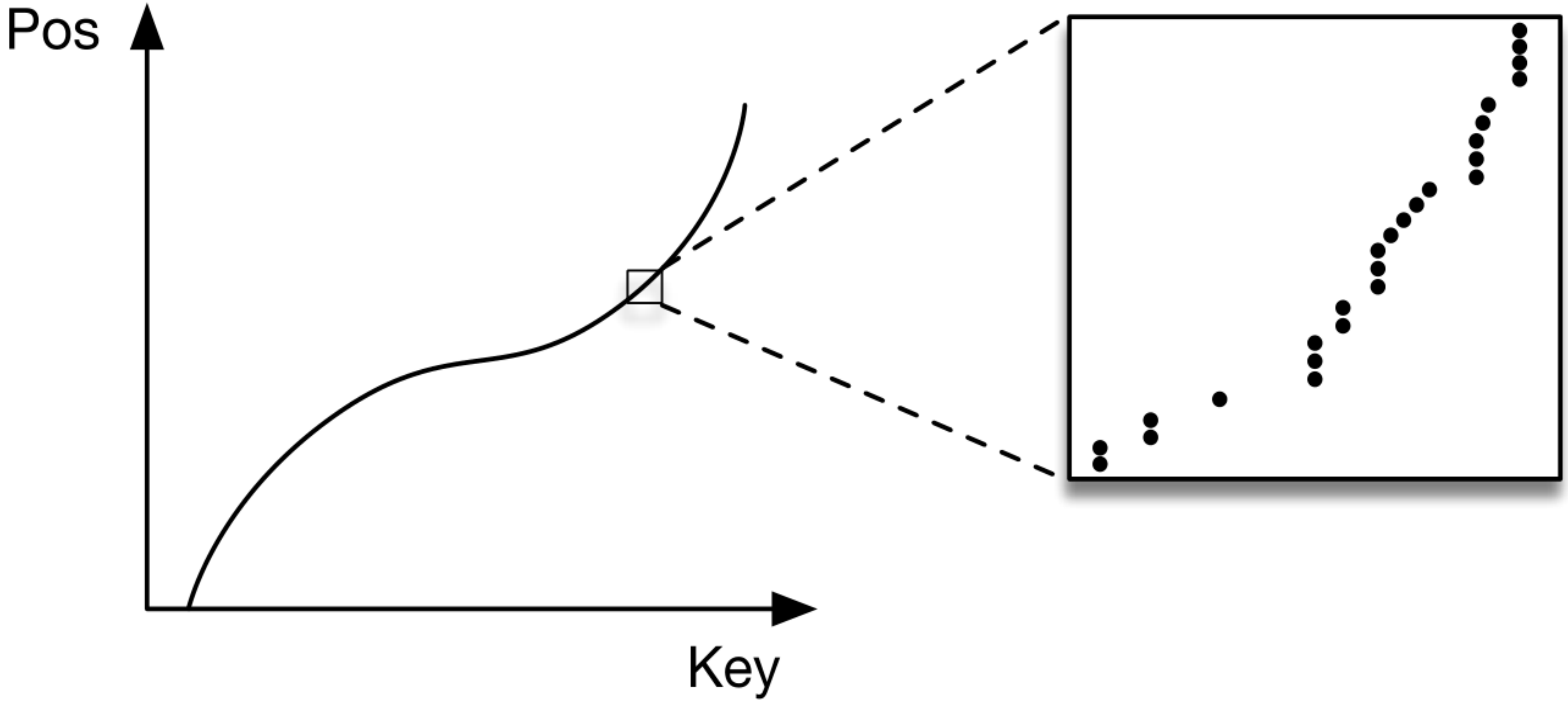
What model type should I use?

Whatever works!

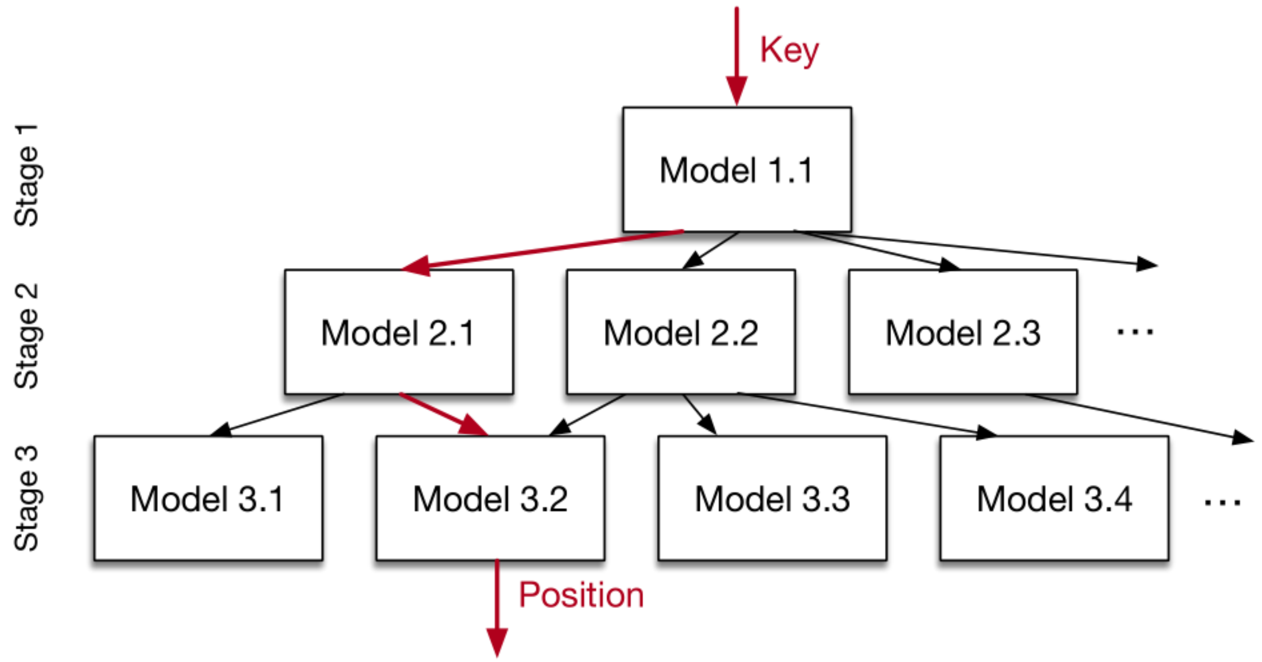
- Often continuous functions
- Possible model types include neural nets, regression models, piece-wise linear functions, among other things
- Model-type can be auto-tuned
- Opens up a complete new toolbox of building data structures and algorithms

Overfitting is a Good Thing

The Last Mile Problem



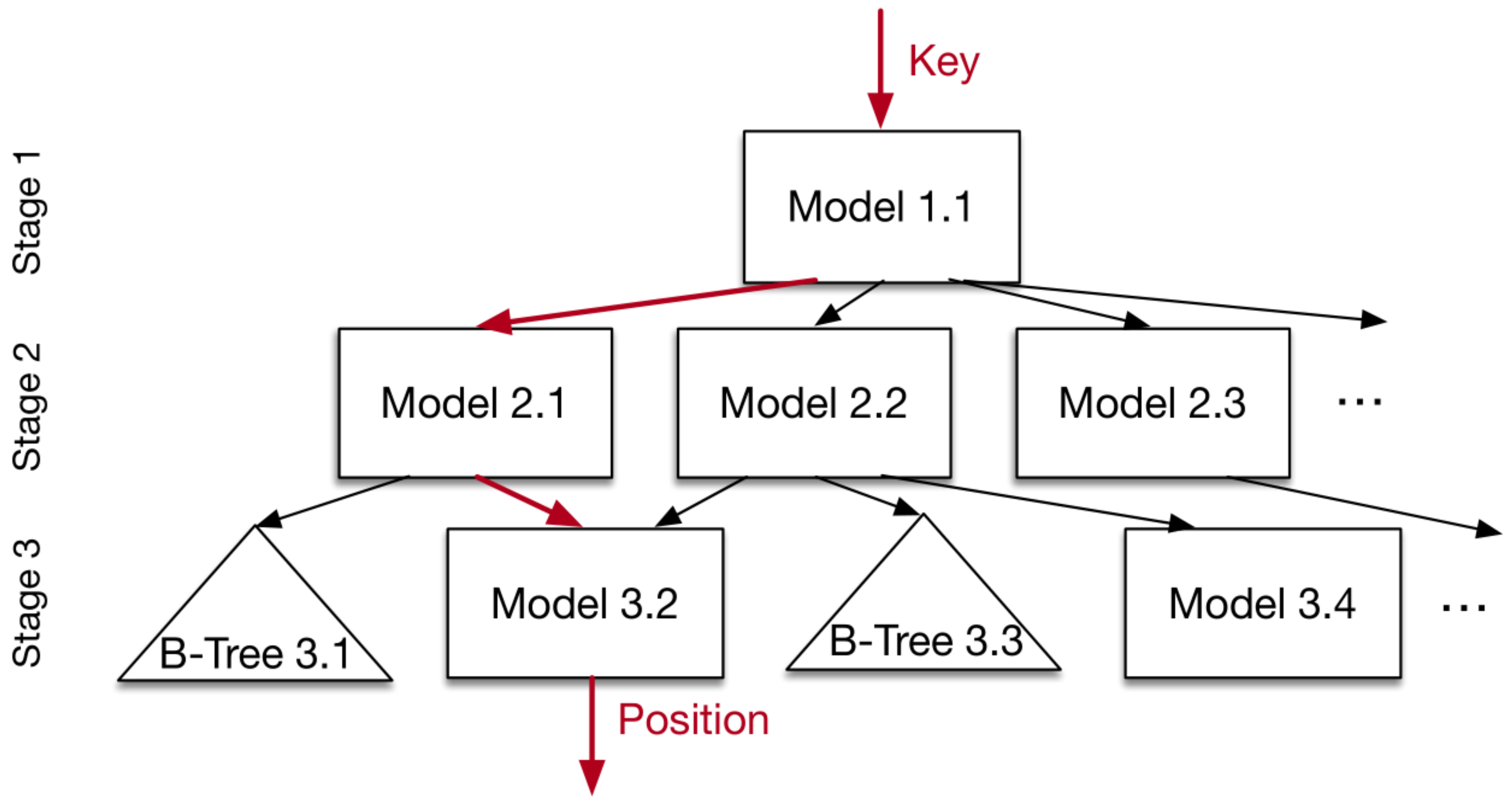
Recursive-Model Index (RMI)



2-Stage RMI with Linear Model

```
pos_0 = a_0 + b_0 * key
pos_1 = m_1[pos_0].a + m_1[pos_0].b * key
record = local-search(key, pos_1)
```

Hybrid RMI



Worst-Case Performance is the one of a B-Tree

Initial Results

TensorFlow

>80,000ns

State-Of-The-Art
B-Tree

265ns

13MB

Learned Index

85ns

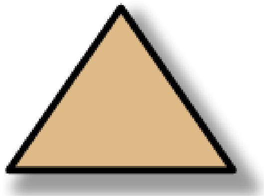
0.7MB

Learned Index

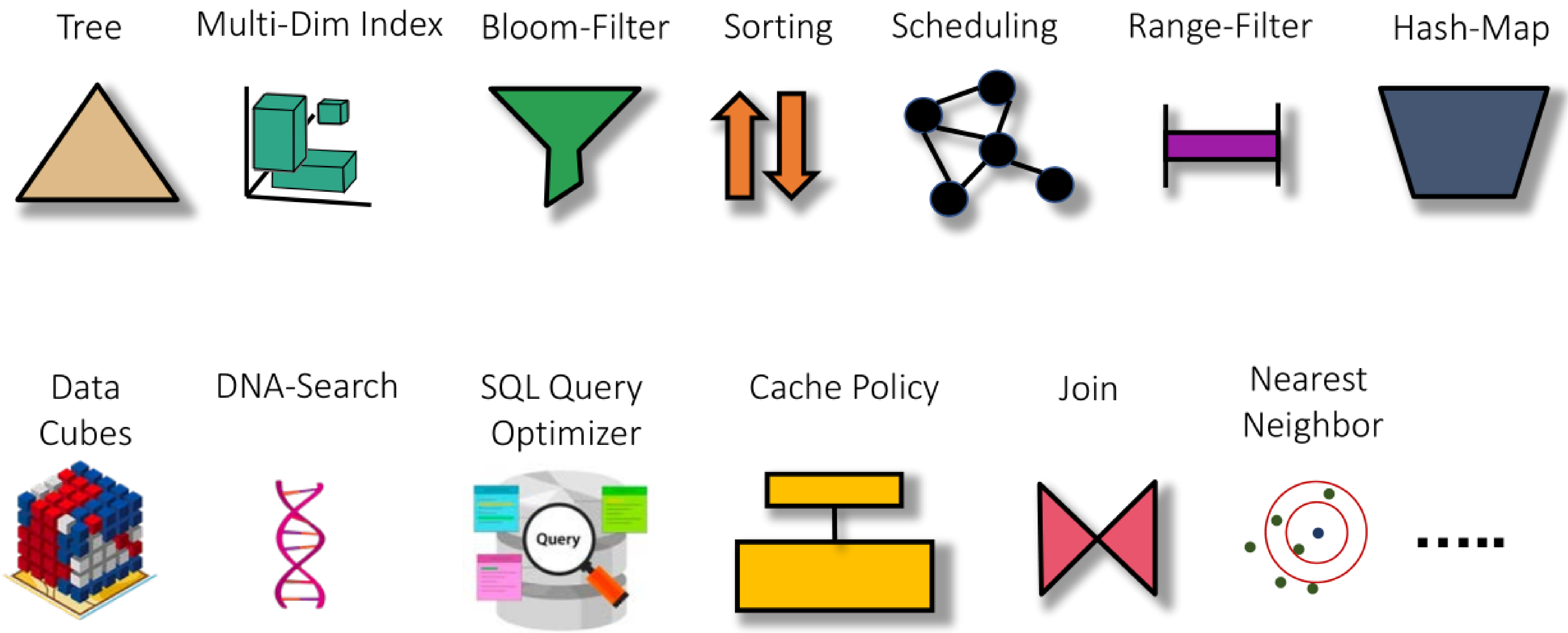
- The biggest advantage is memory size
  - These memory can be used to do other things!
- Also ML models can utilize more with the parallelizing computation, while B-Tree is essentially a lot of if-else

Fundamental Algorithms & Data Structures

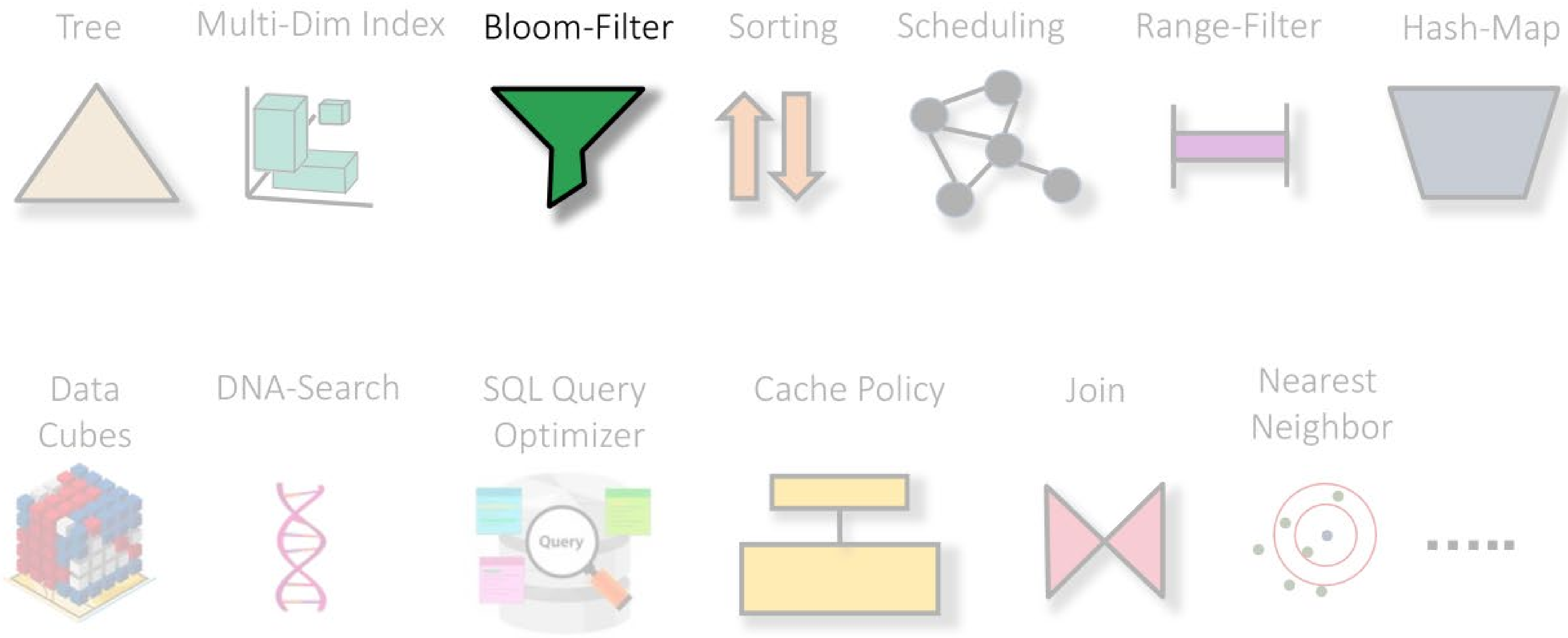
Tree



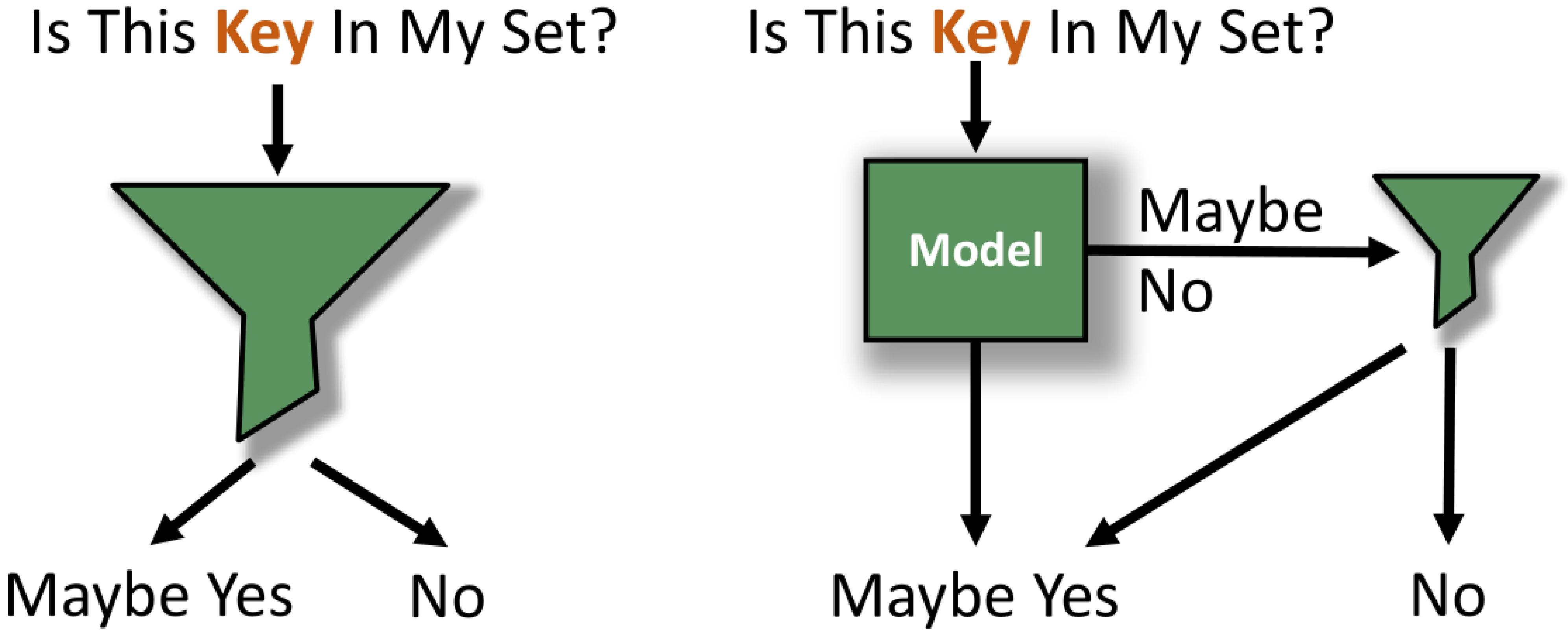
Fundamental Algorithms & Data Structures



Fundamental Algorithms & Data Structures

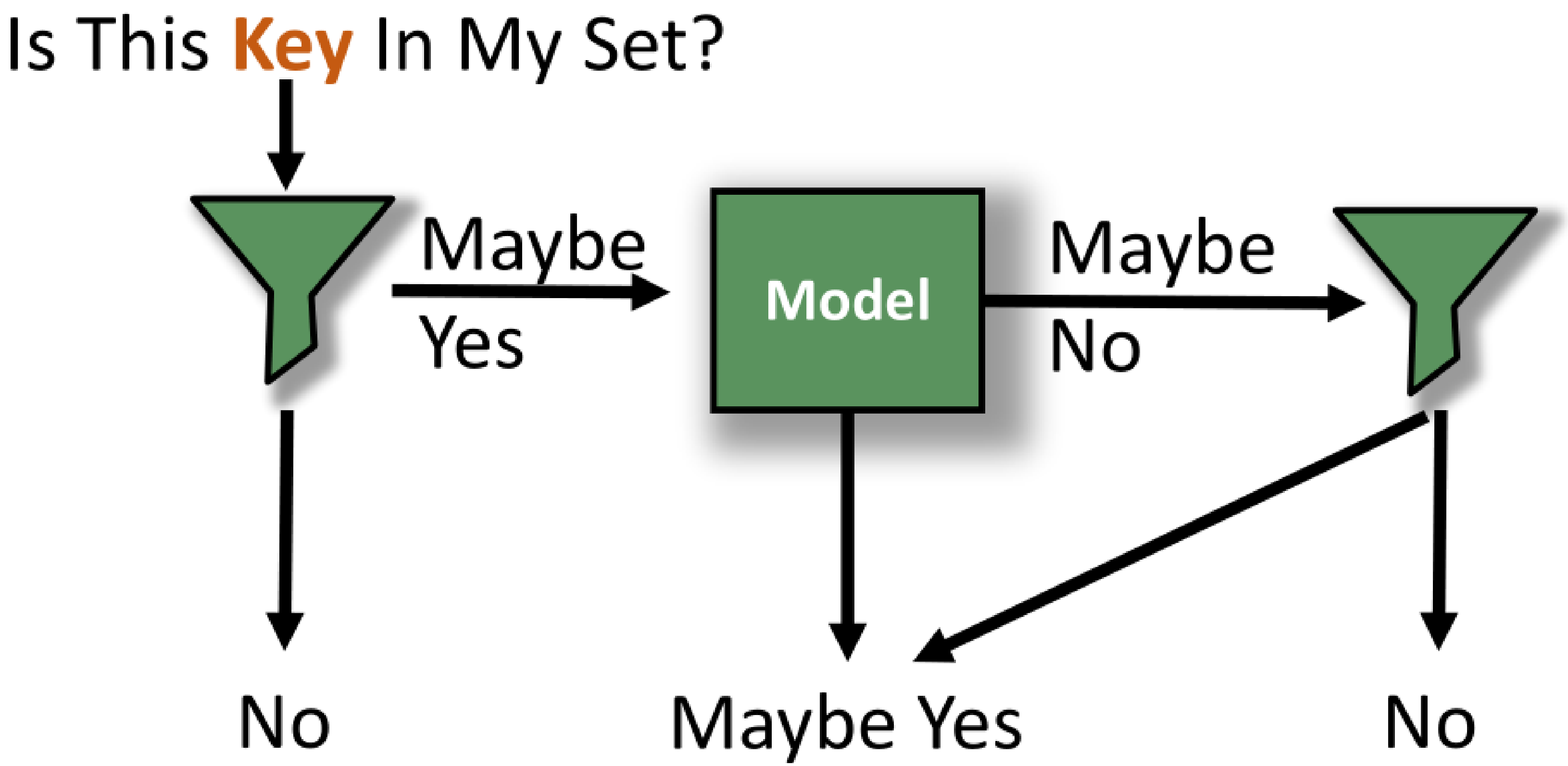


Bloom Filter- Approach 1



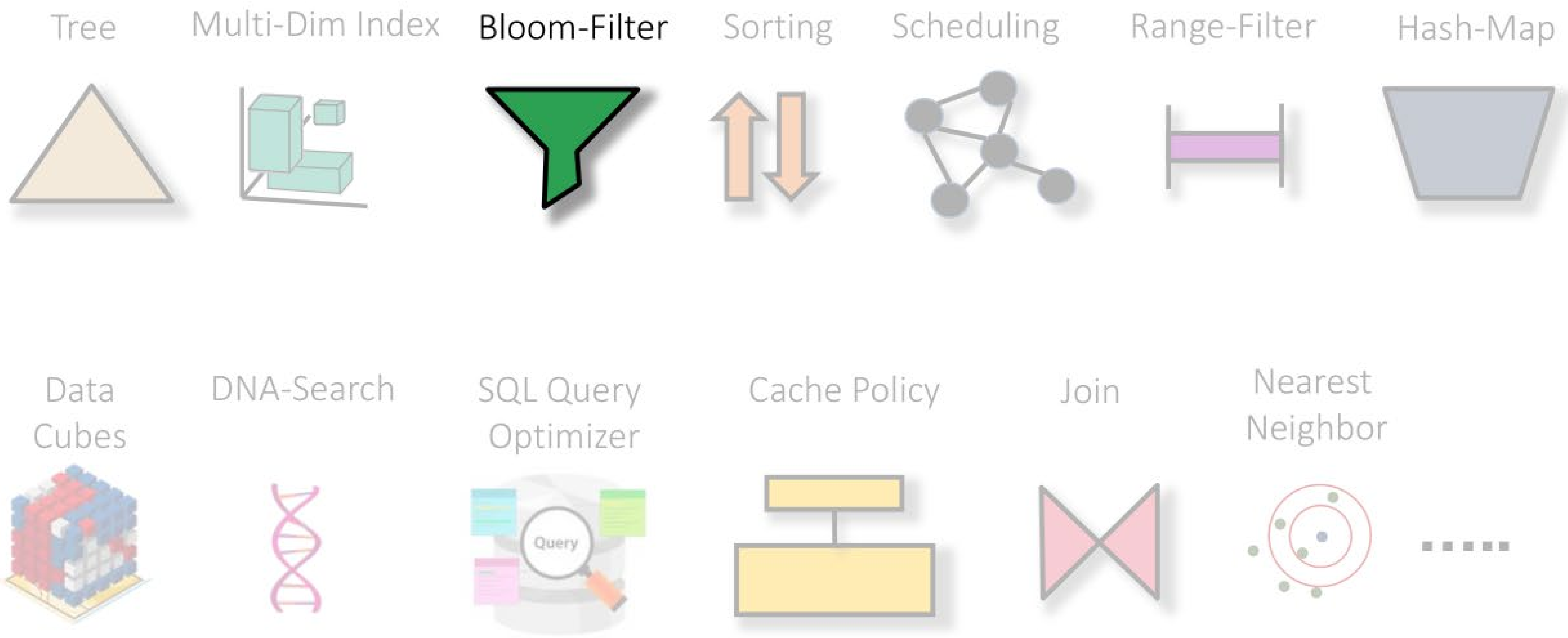
36% Space Improvement over Bloom Filter at Same False Positive Rate

Sandwiched Bloom Filter

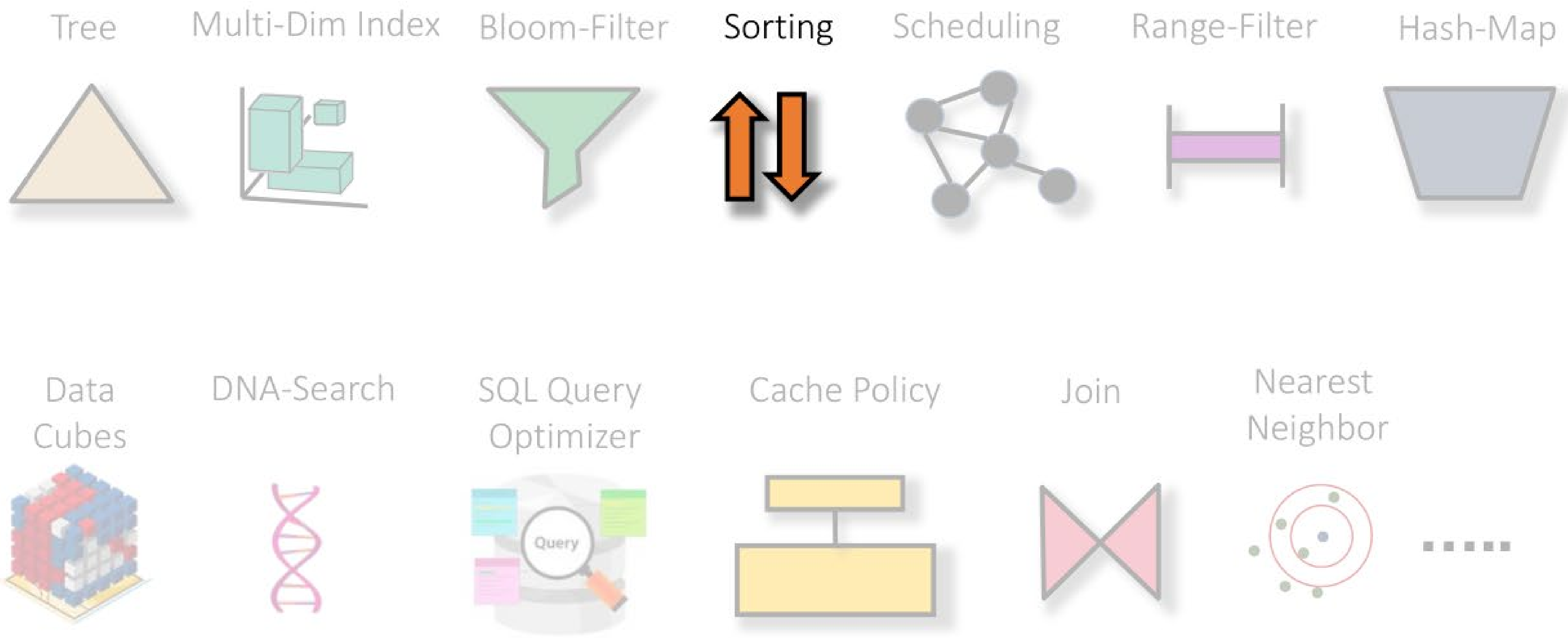


Michael Mitzenmacher: A Model for Learned Bloom Filters and Optimizing by Sandwiching. NeurIPS 2018: 462-471

Fundamental Algorithms & Data Structures

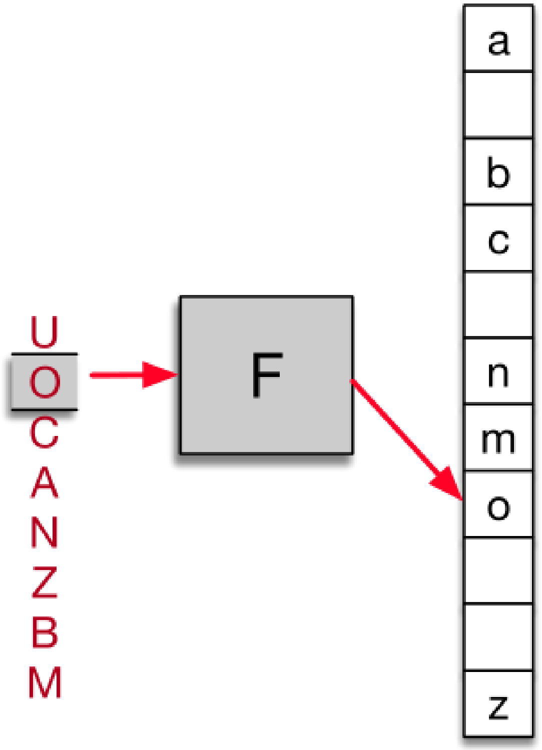


Fundamental Algorithms & Data Structures



Sorting

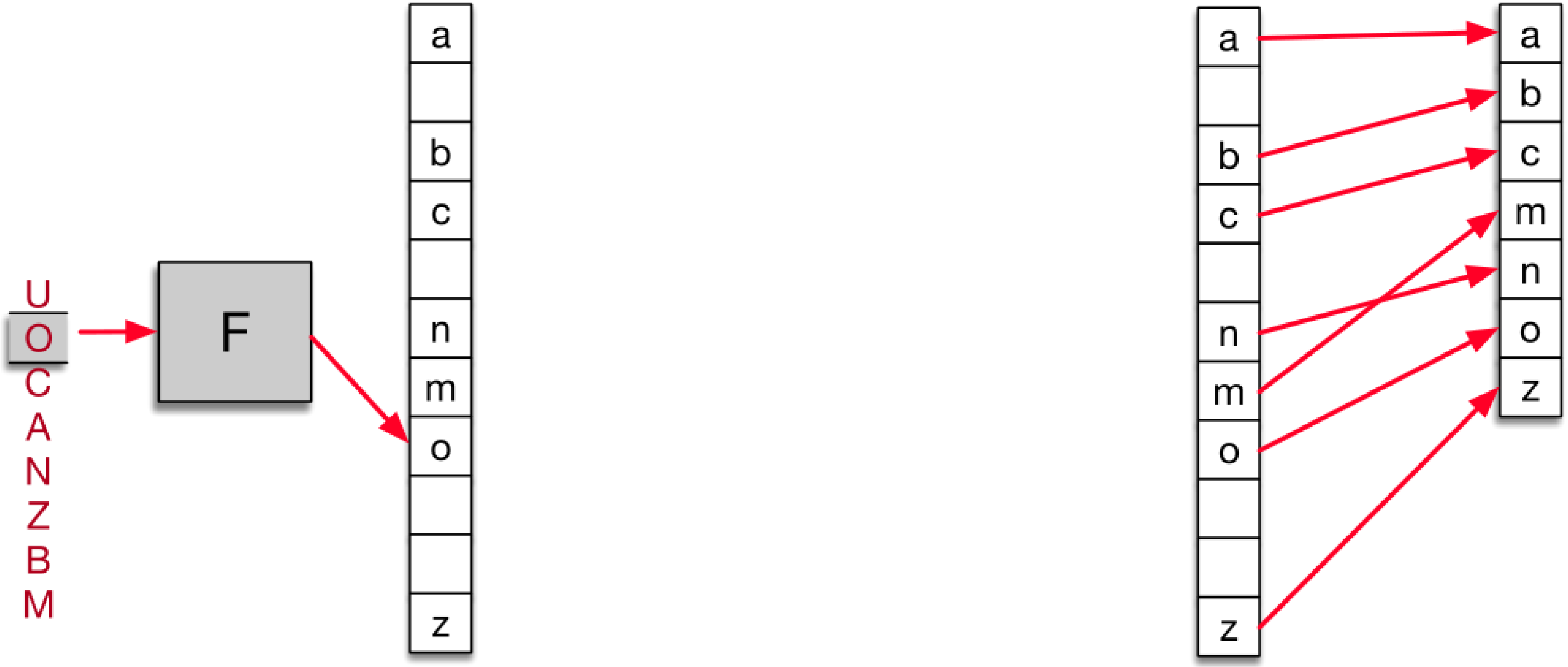
(a) CDF Model Pre-Sorts



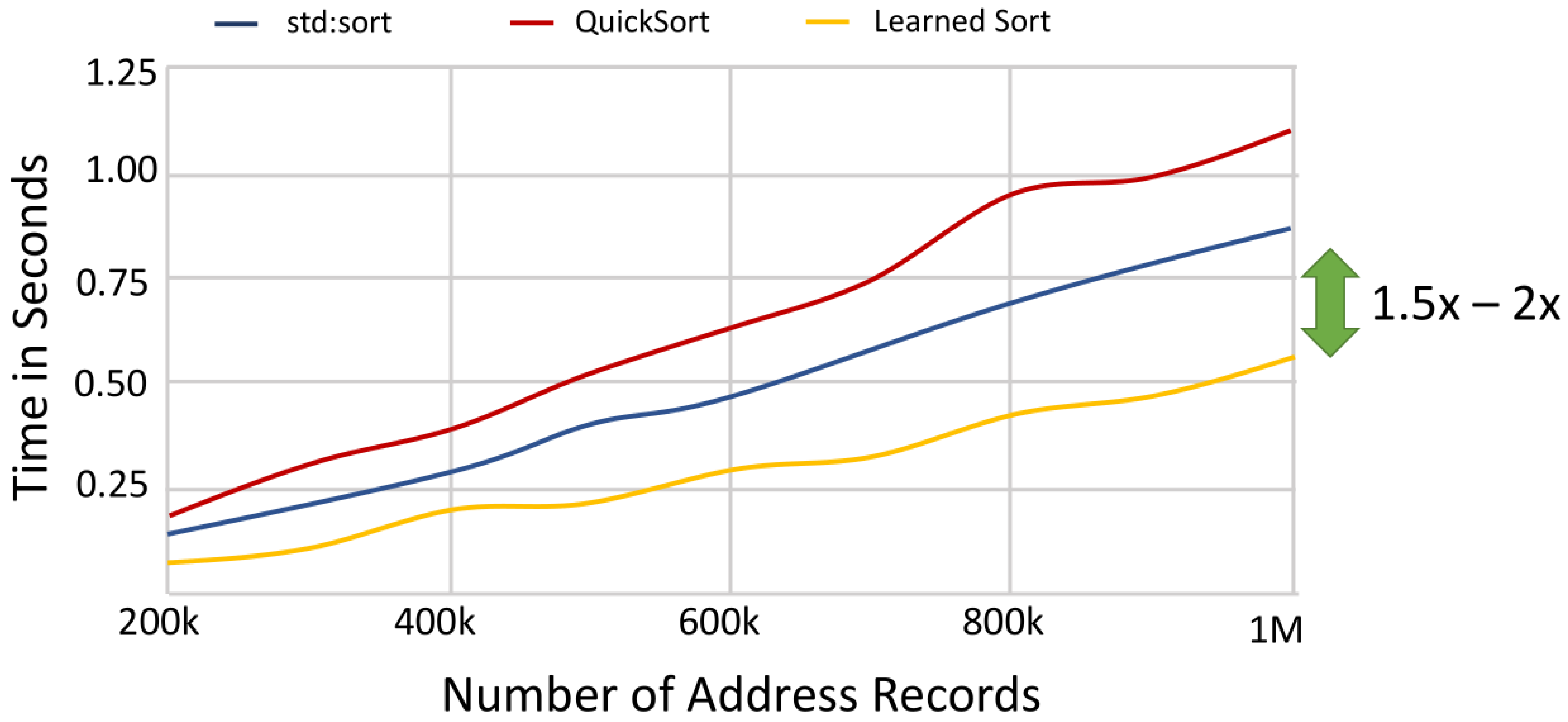
Sorting

(a) CDF Model Pre-Sorts

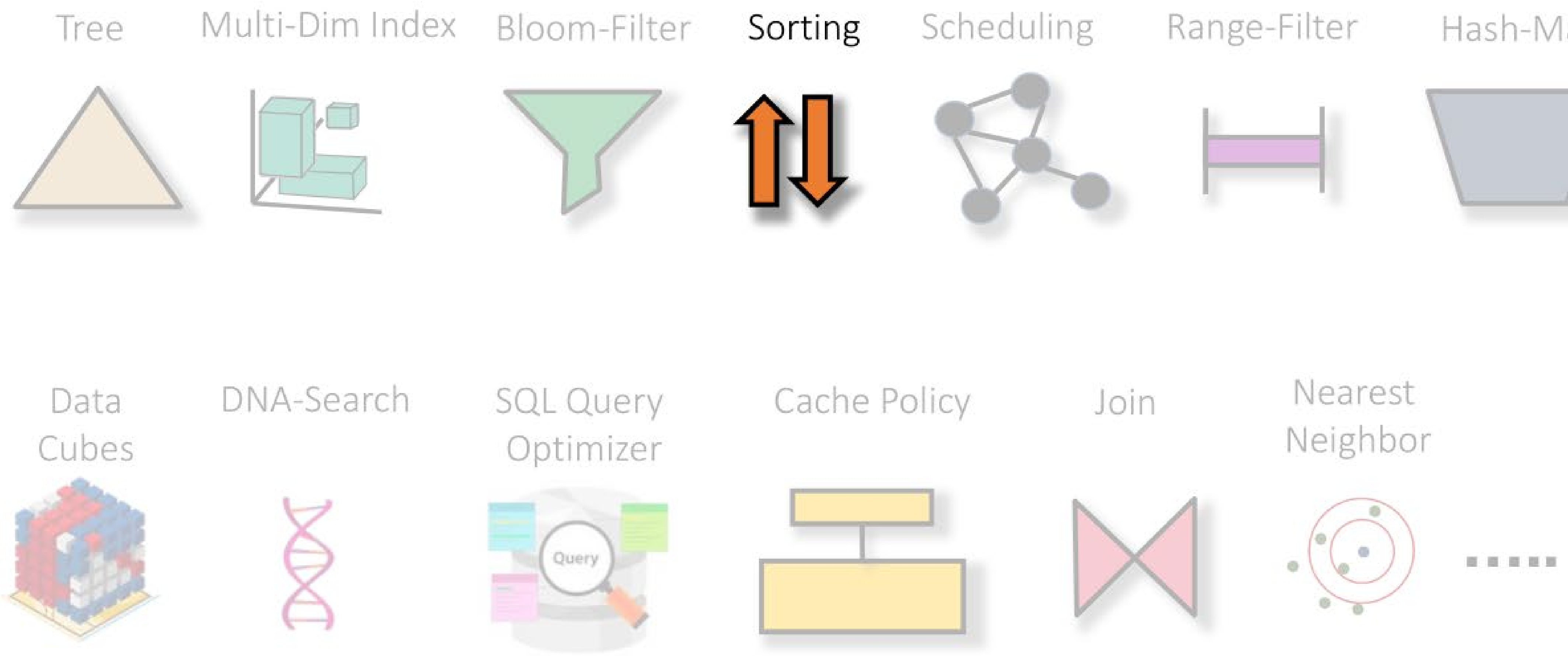
(b) Compact & local sort



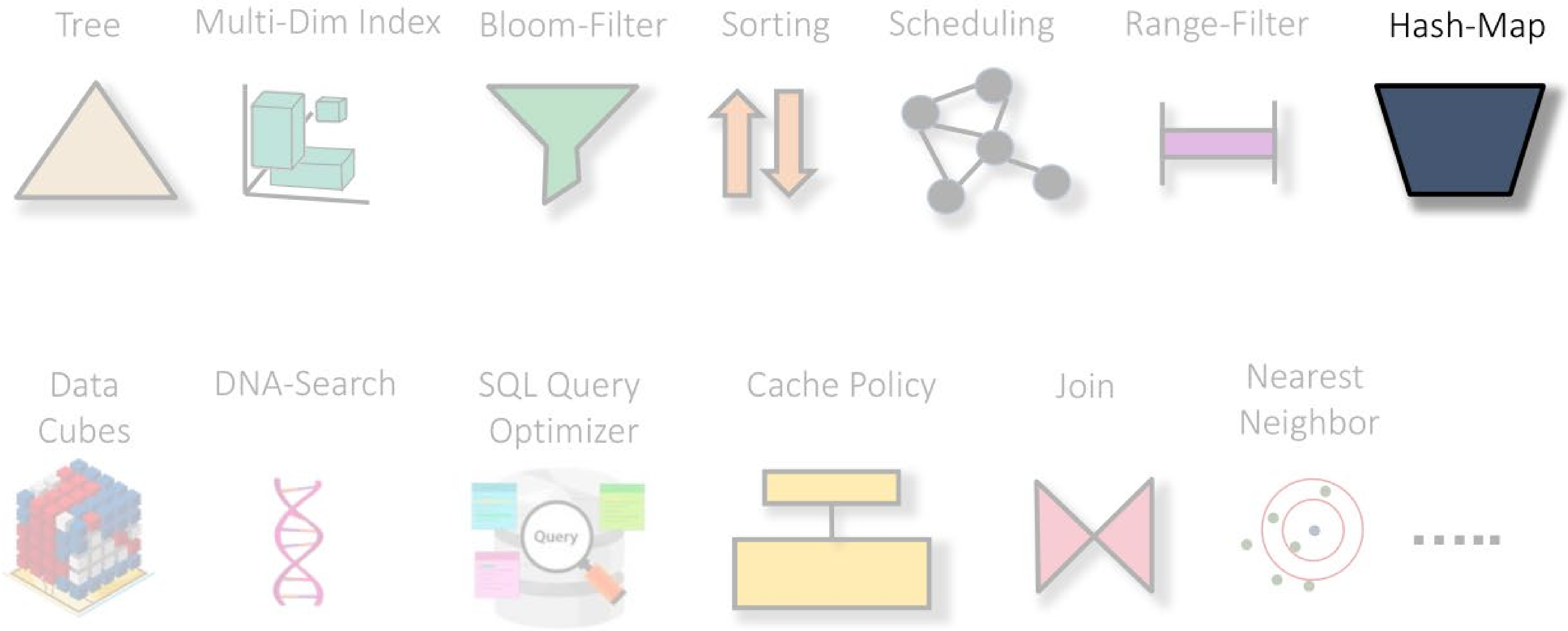
Initial Results



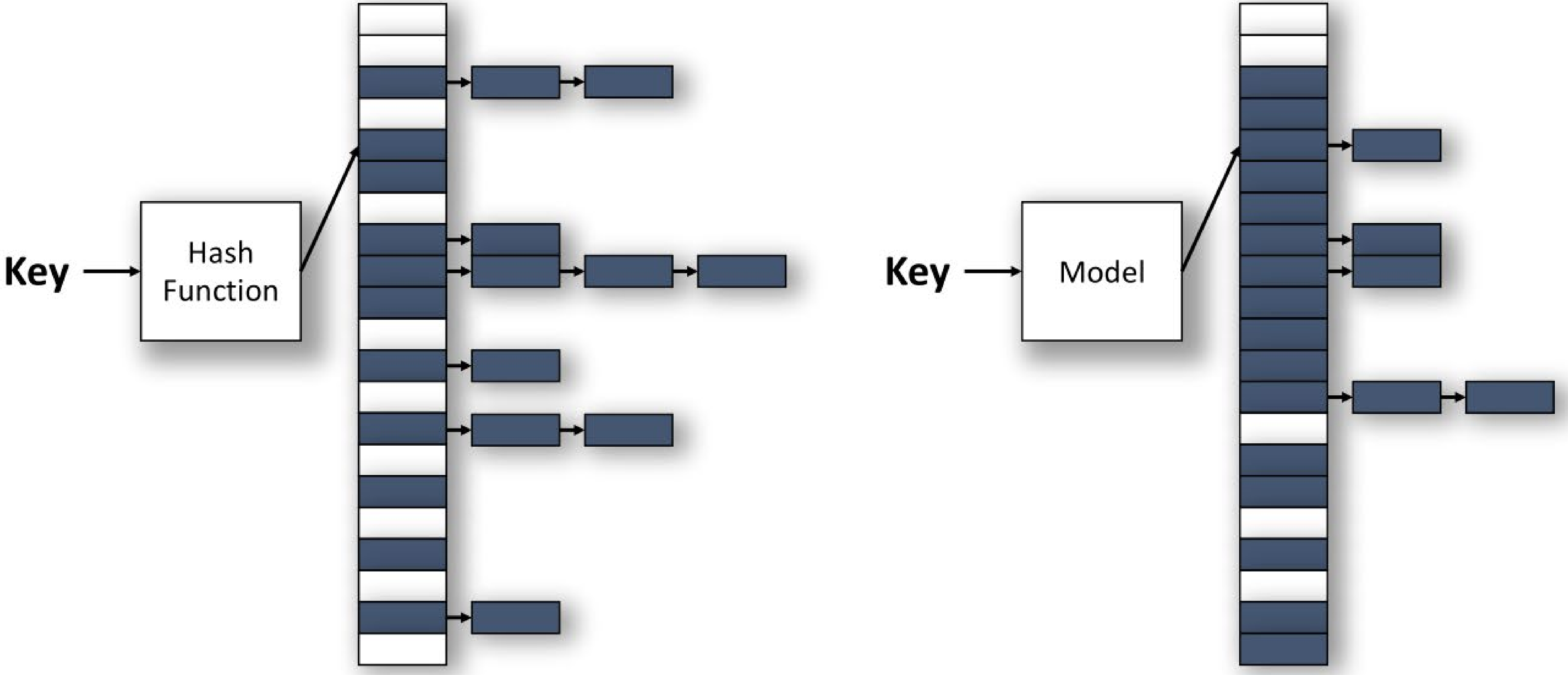
Fundamental Algorithms & Data Structures



Fundamental Algorithms & Data Structures



Hash Map



Goal: Reduce Conflicts

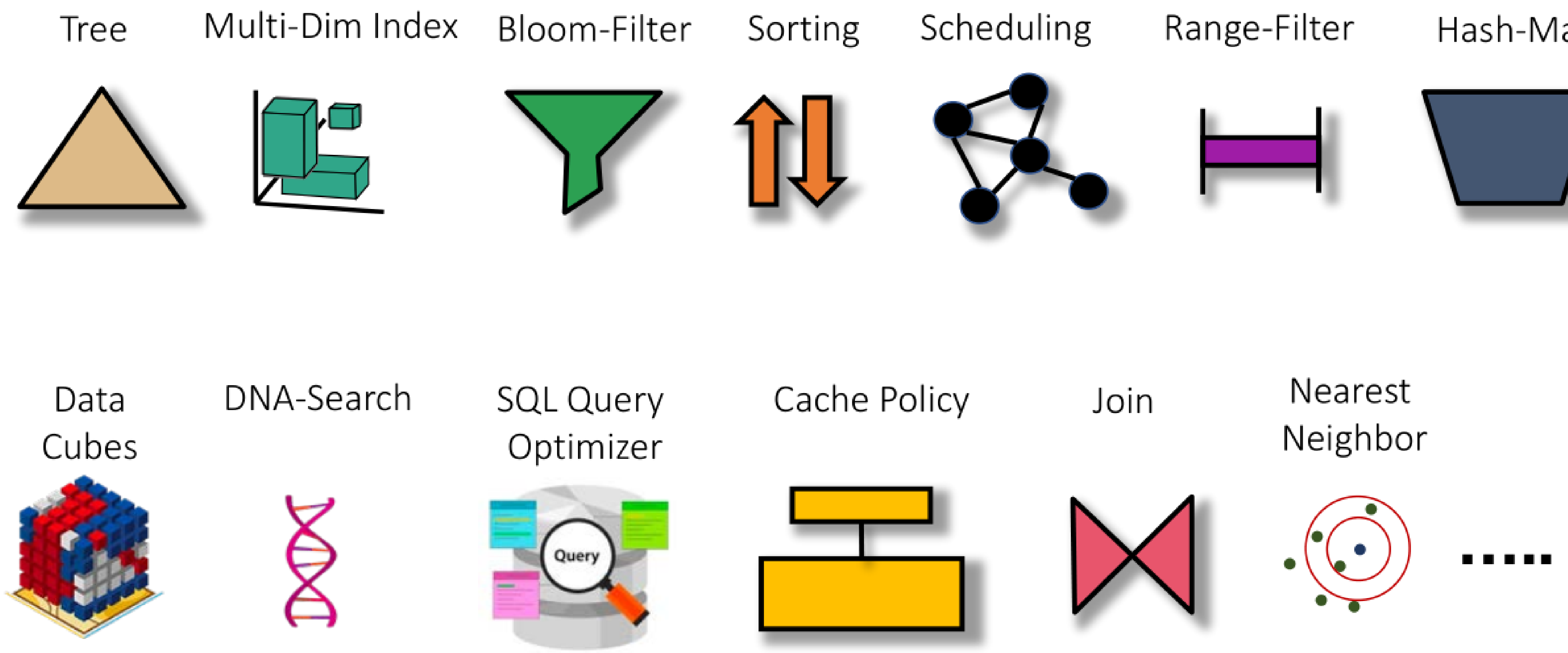
Hash Map - Results

	% Conflicts Hash Map	% Conflicts Model	Reduction
Map Data	35.3%	07.9%	77.5%
Web Data	35.3%	24.7%	30.0%
Log Normal	35.4%	25.9%	26.7%

25% - 70% Reduction in Hash-Map Conflicts

Skip

Fundamental Algorithms & Data Structures



So when can we apply ML to system?

- When you need to make a decision that different decisions may lead to significantly different running time
  - By significantly, we mean that it is far more longer than a model prediction time
  - In this case, a classification model may be your friend
  - Bloom filter is one example, we will see an example from OSDI2020 later
- When you want to $\boldsymbol{x}$ to a ordering-mattered $\boldsymbol{y}$
  - Regression model is your friend
  - Learned index, learned hashmap, learned sorting, Burbon@OSDI2020 are examples
- The penalty of misprediction is low
  - Sometimes we can combine with a conventional method to avoid misprediction

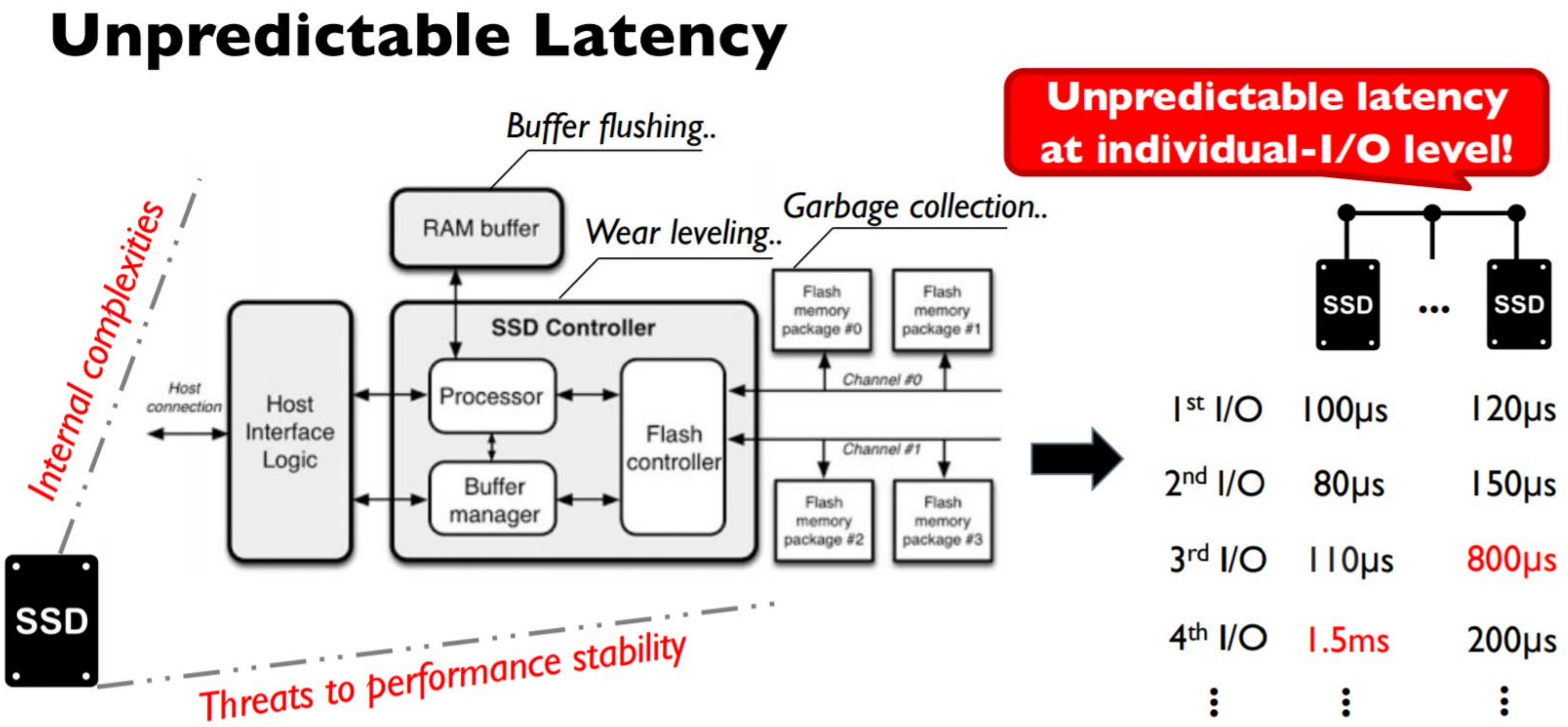
How can we apply ML?

- Particularly useful for read-only system
  - Levelfiles in LSMT
  - Some index for large-scale database for historical data(never change)
- Light and quick
  - We want the model be small and quick enough to plug in system
  - If there are update for learned index, then retraining is often required. So we want the training to be as fast as possible
- Formularize the problem to be easy for ML models
  - Use what(features) to predict what(target)?
  - Think about whether there are patterns in the data
  - Sometimes can combine conventional methods with ML
  - Recursive model index is an example

What models are good for system

- Classification:
  - Logistic regression
  - Neural networks(not that good)
  - …
- Regression:
  - Linear regression
  - Piece-wise linear regression
  - Polynomial fitting
  - Neural networks (not that good)
  - ……

A case study: Linnos@OSDI 2020



Hao M, Toksoz L, Li N, et al. LinnOS: Predictability on Unpredictable Flash Storage with a Light Neural Network, OSDI2020

Speculative Execution

- ❑ Black-box approaches
  1. SSD-aware filesystems and applications

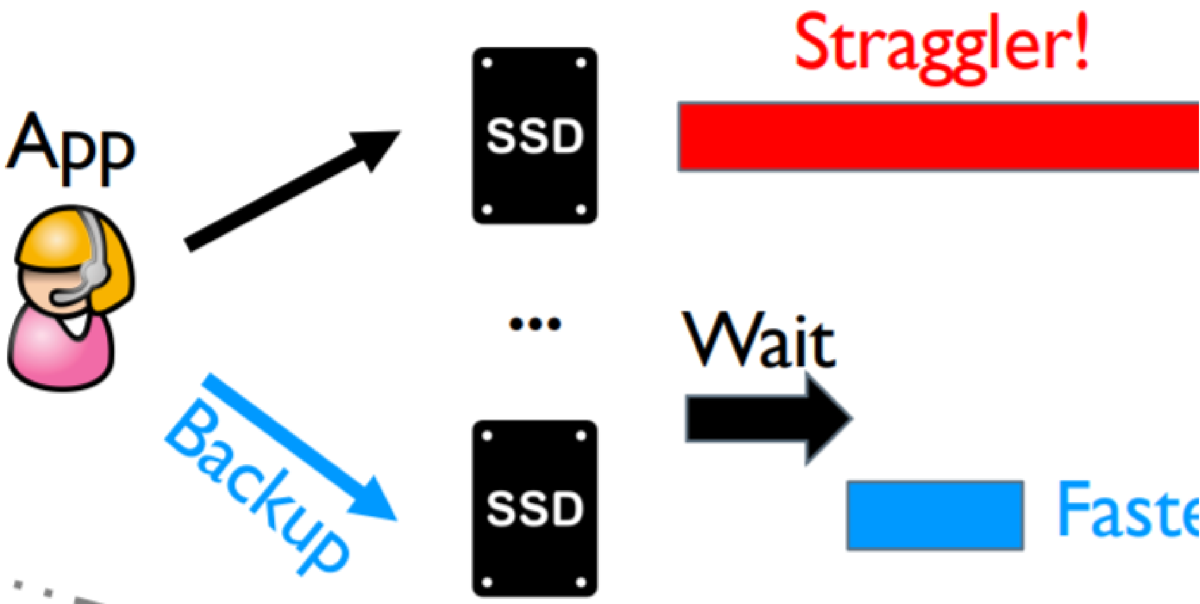
Most popular

2. **Speculative execution**

MongoDB cassandra ...

Mitigate every slow I/O in a black-box way

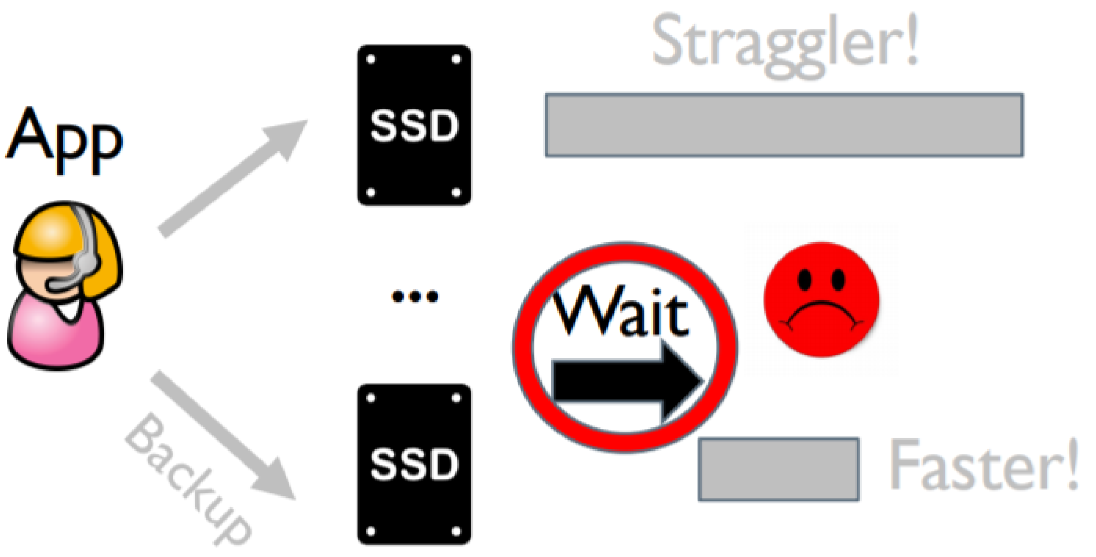
Hedged requests (hedgin



Agnostic!

Speculative execution

- Passively wait due to black-box

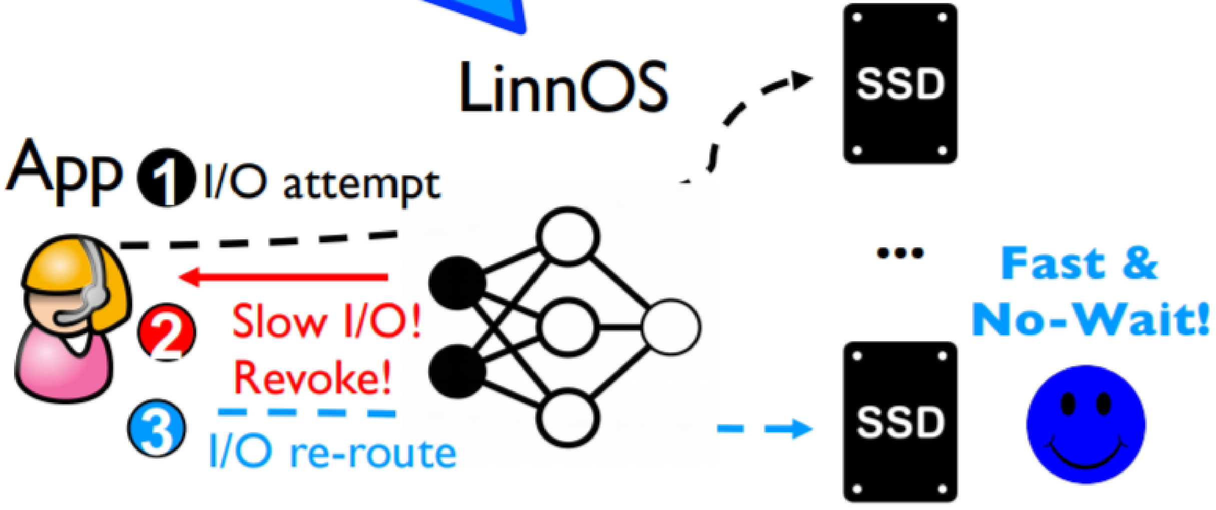


Learning!

LinnOS

- Proactively infer the black-box

Lightweight neural network for per-I/O speed inference



Output labeling

Ideal labels:

Exact latency value
(e.g. 120μs, 80μs..) + **Flexible**

− **Difficult to achieve decent accuracy**

Latency ranges
(e.g., 100-200μs,
200-400μs)

Only 60-70% accuracy

50-100μs

100-200μs ← **Mis-Prediction**

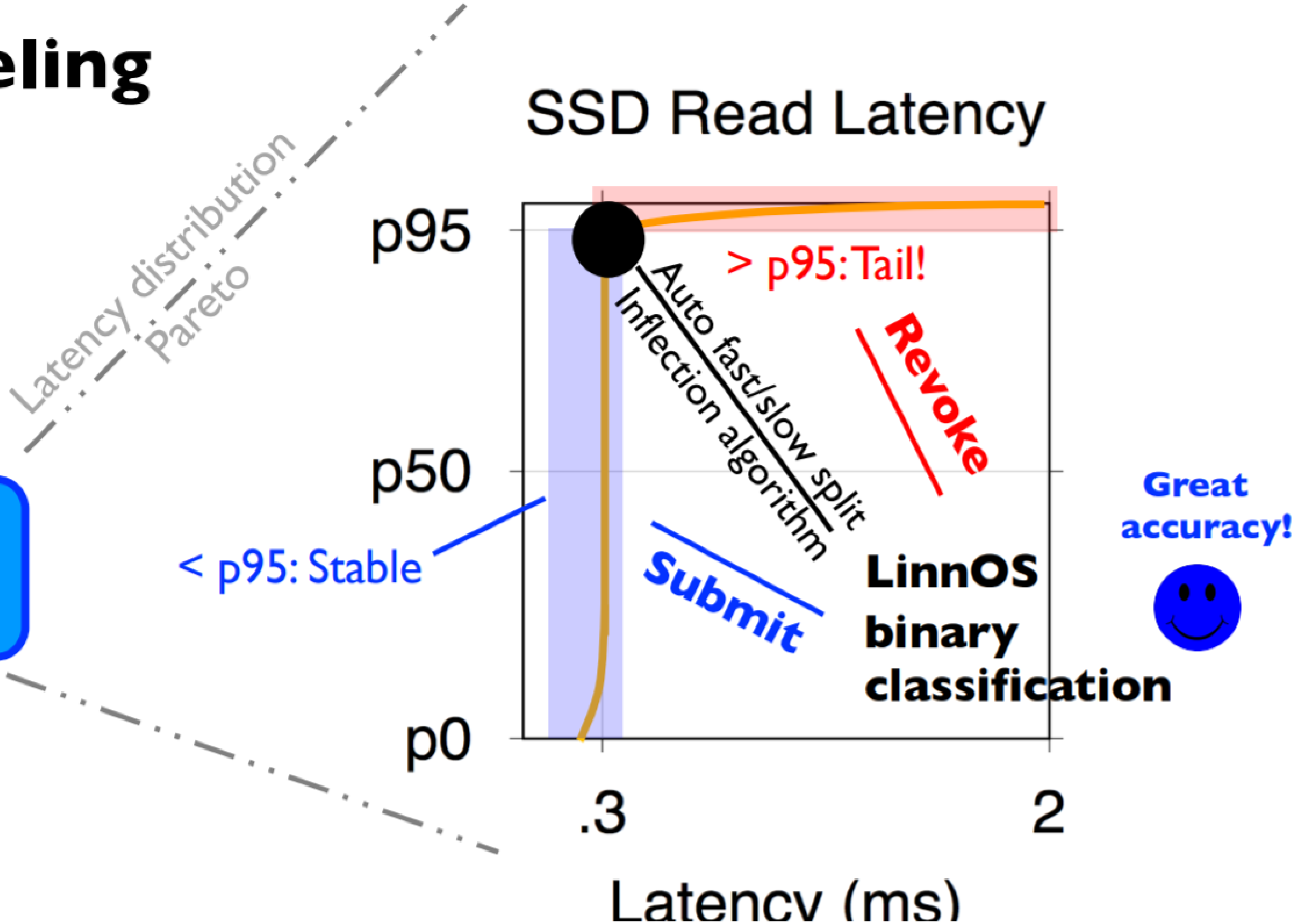
200-400μs ← **Truth**

…

Output labeling

Precise prediction **is difficult!**

Simpler and **helpful** alternatives?



Input features

Ideal features:

Addresses of related I/Os
(finest granularity)

Directly indicate the resource contention

+ High accuracy (up to 99%)

− High overhead

Thousands of features
(addresses in 32-bit binaries)

Unacceptable

Hundreds of microseconds to infer each I/O

Input features

Using **finest** features **is expensive!**

Use features that are **more aggregate**

Queue length and history I/Os

Easier to learn

For each I/O

Good latency indicator, but how about internal disruption?

Low history **queue length** + **high** history **latency** = internal **disruptions**

102	010	2184	…
Current queue length	Last I/O queue length	Last I/O latency (μs)	

Input features

Queue length and history I/Os

Last completed I/O | 2nd last | 3rd last | 4th last

102, 010, 2184, 056, 0800, 126, 1600, 368, 3920

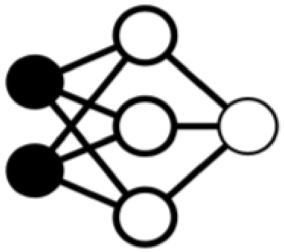
Current queue length

Queue lengths and latencies for **last four completed I/Os**

Split into individual digits

1,0,2, 0,1,0, 2,1,8,4, 0,5,6, 0,8,0,0, 1,2,6,1,6,0,0, 3,6,8,3,9,2,0

More aggregate features

31 features +  3 fully-connected layers (31-256-2) = **87-97%** accuracy **4-6μs** overhead across various SSDs/traces

Handling inaccuracy

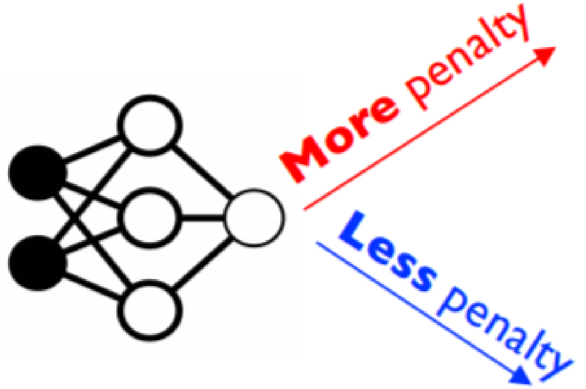
Inaccurate cases

Category		Description		Cost
False **submits**	:	Mistakenly **accept a slow I/O**	:	Up to **milliseconds!**
False **revokes**	:	Mistakenly **revoke a fast I/O**	:	**µs-level** failover

Customized loss function:

Reduce false submits by **up to 68%!**

Biased training



Conclusion

- System is always about trade-off
- ML is also often about trade-off: computation vs accuracy
- Define your problem well, find a good trade-off