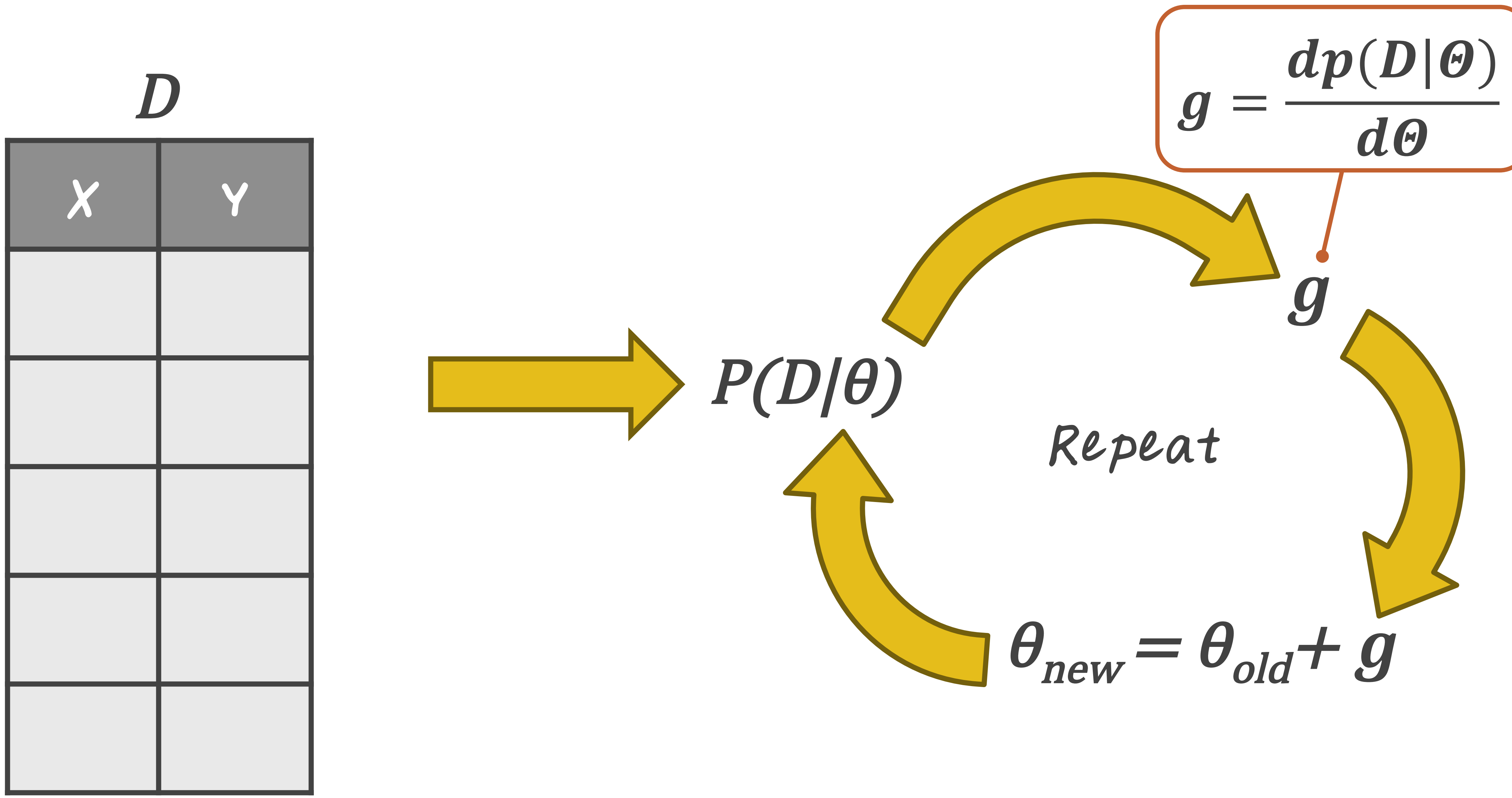
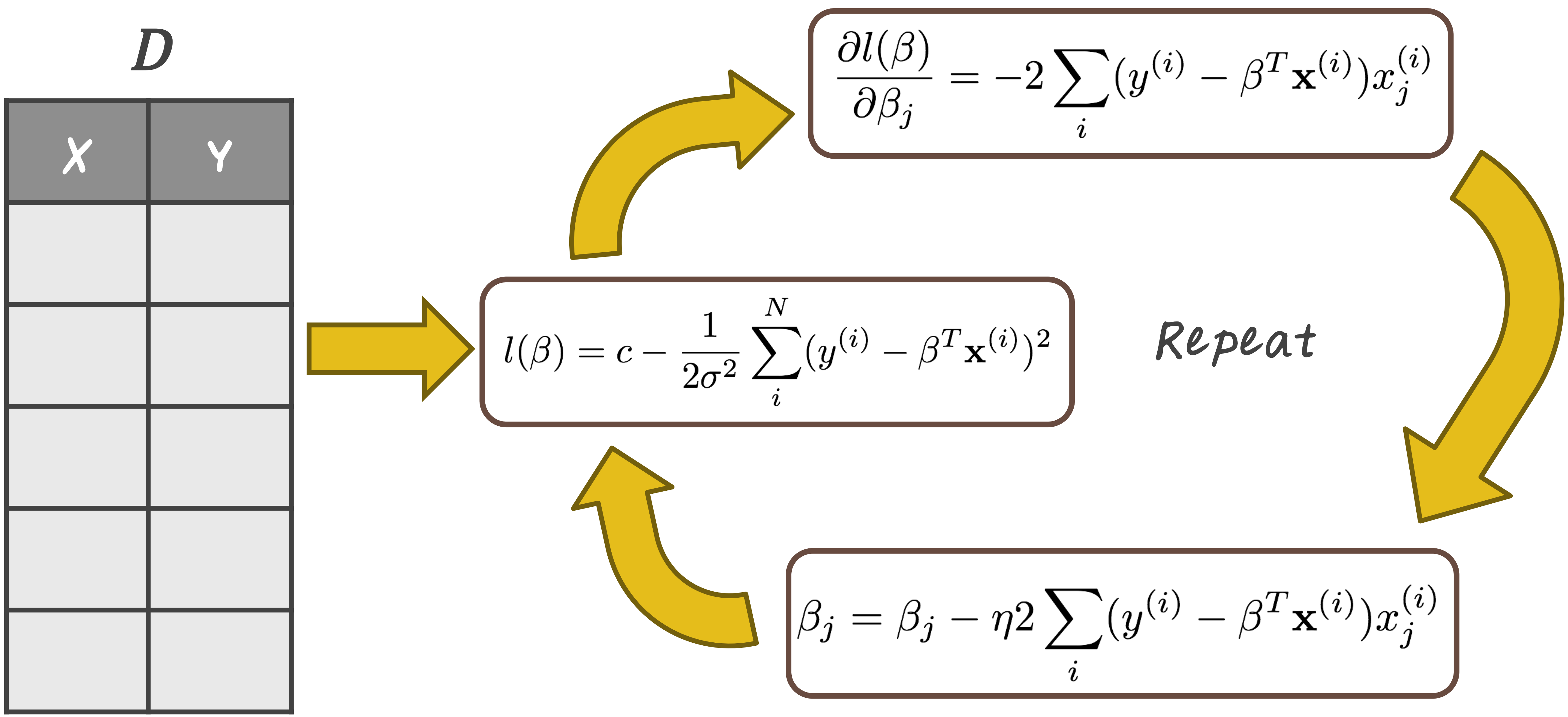
Classification methods

Jimeng Sun

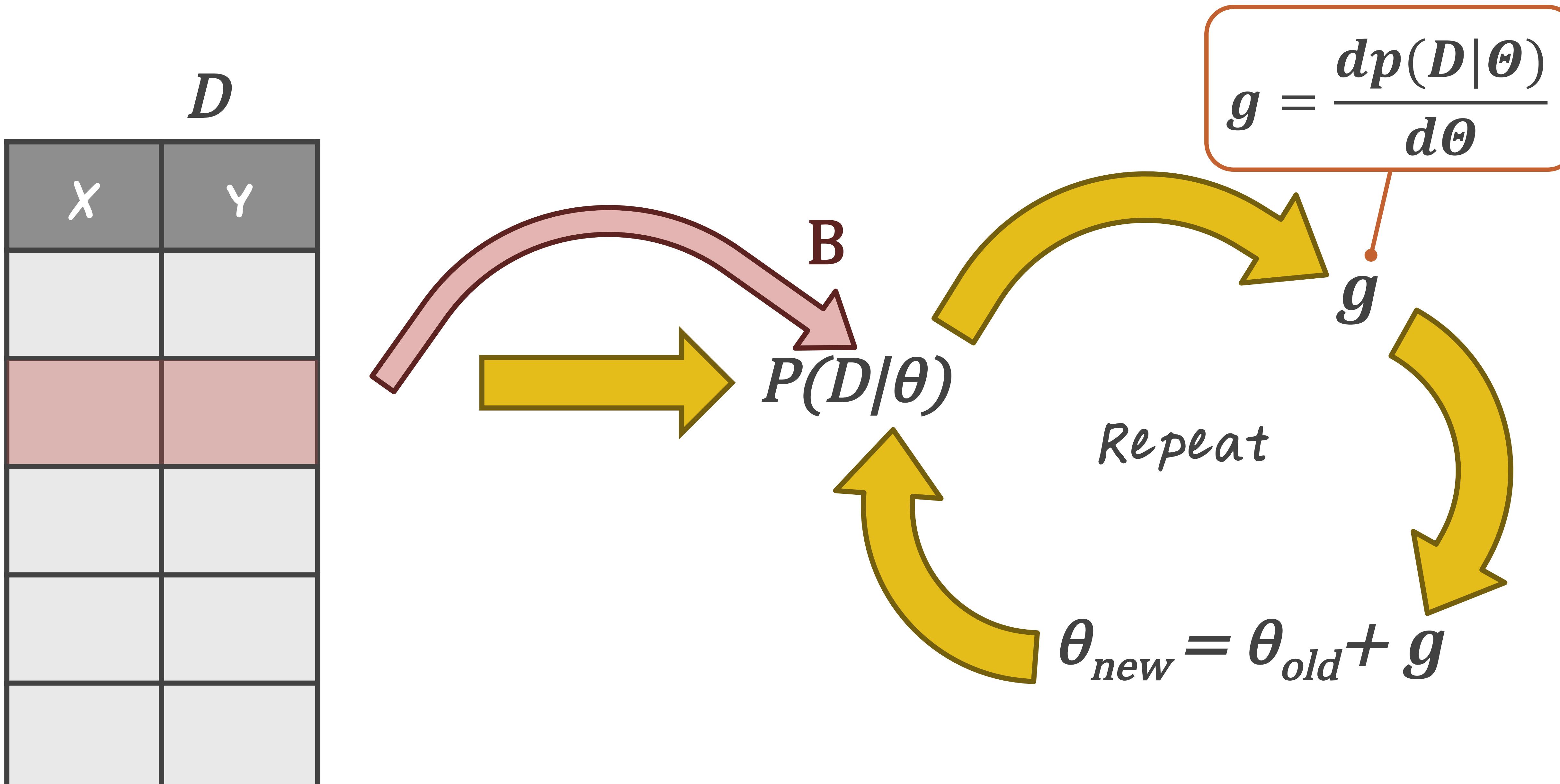
GRADIENT DESCENT METHOD



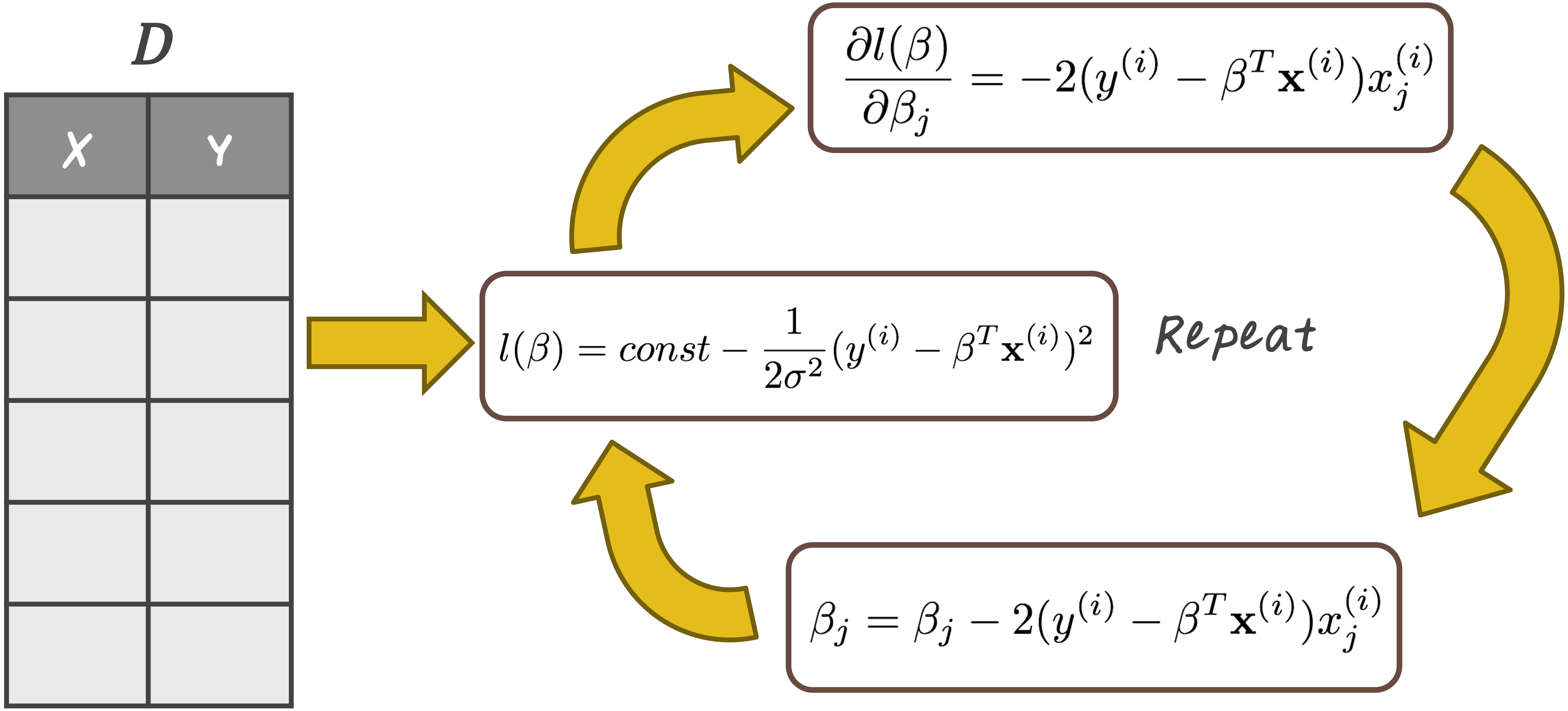
GDM FOR LINEAR REGRESSION



STOCHASTIC GRADIENT DESCENT (SGD) METHOD



SGD FOR LINEAR REGRESSION



Learn Linear regression model

- Likelihood is the joint probability of D as a function of parameters

$$L(\beta) = \prod_{i=1}^{m} p(y^{(i)}|\mathbf{x}^{(i)};\beta)$$

$$= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)}-\beta^T\mathbf{x}^{(i)})^2}{2\sigma^2}}$$

- Log-likelihood

$$l(\beta) = \log L(\beta)$$

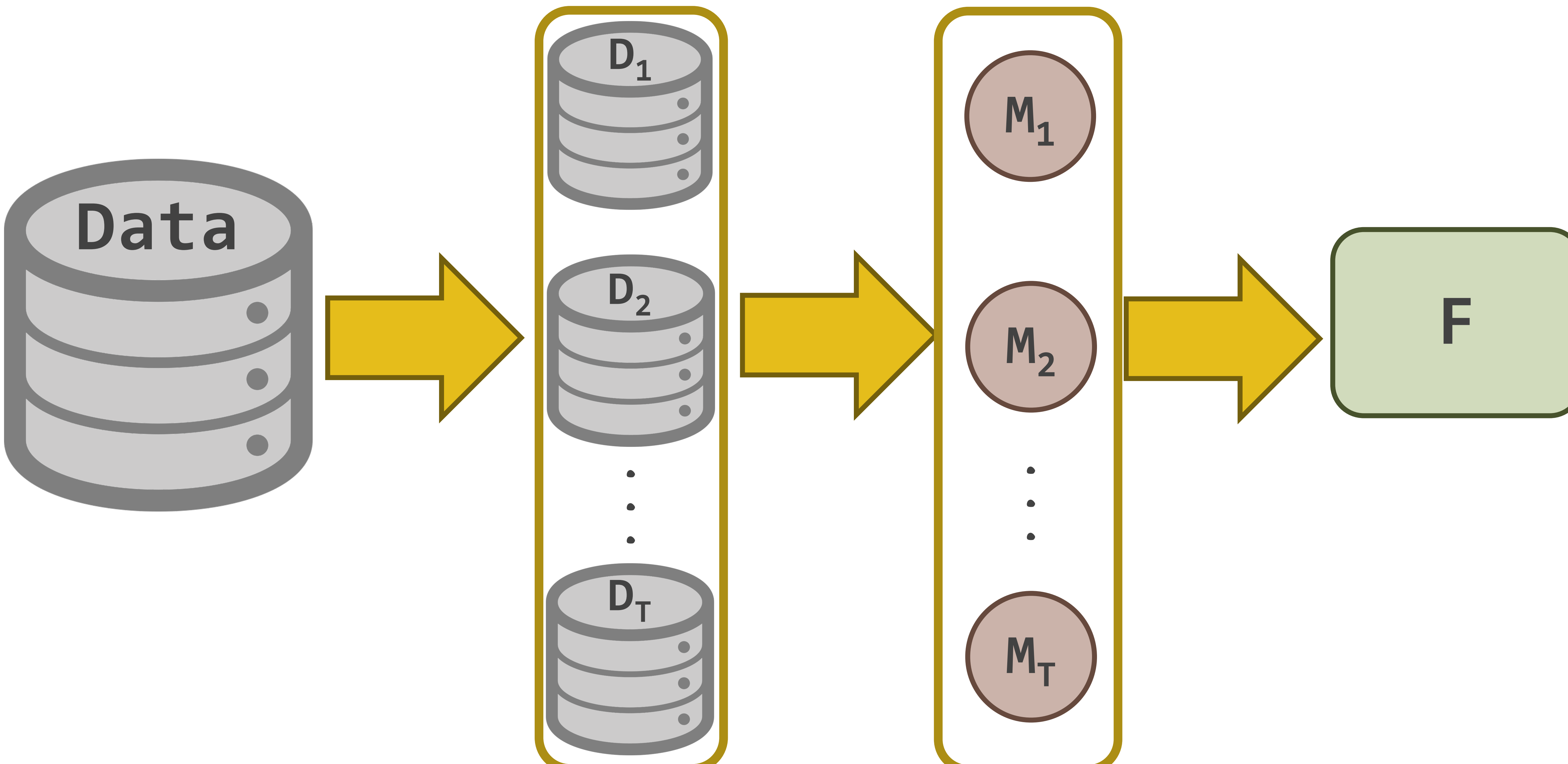
$$= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)}-\beta^T\mathbf{x}^{(i)})^2}{2\sigma^2}}$$

$$= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)}-\beta^T\mathbf{x}^{(i)})^2}{2\sigma^2}}$$

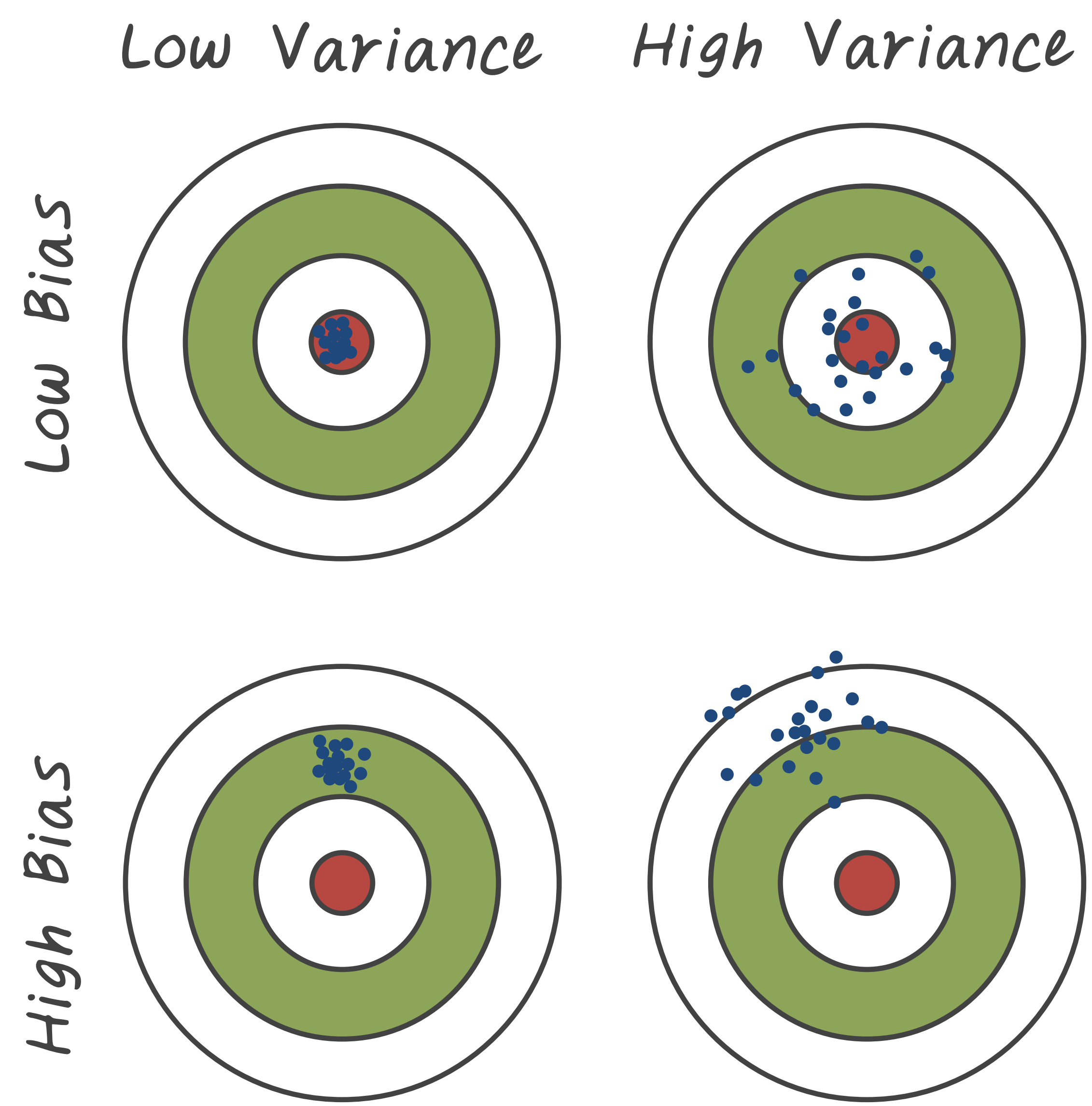
$$= \underbrace{\cancel{m \log \frac{1}{\sqrt{2\pi}\sigma}}}_{\text{constant}} - \frac{1}{2\sigma^2} \underbrace{\sum_{i=1}^{m} (y^{(i)} - \beta^T\mathbf{x}^{(i)})^2}_{\text{minimize}}$$

ENSEMBLE METHOD

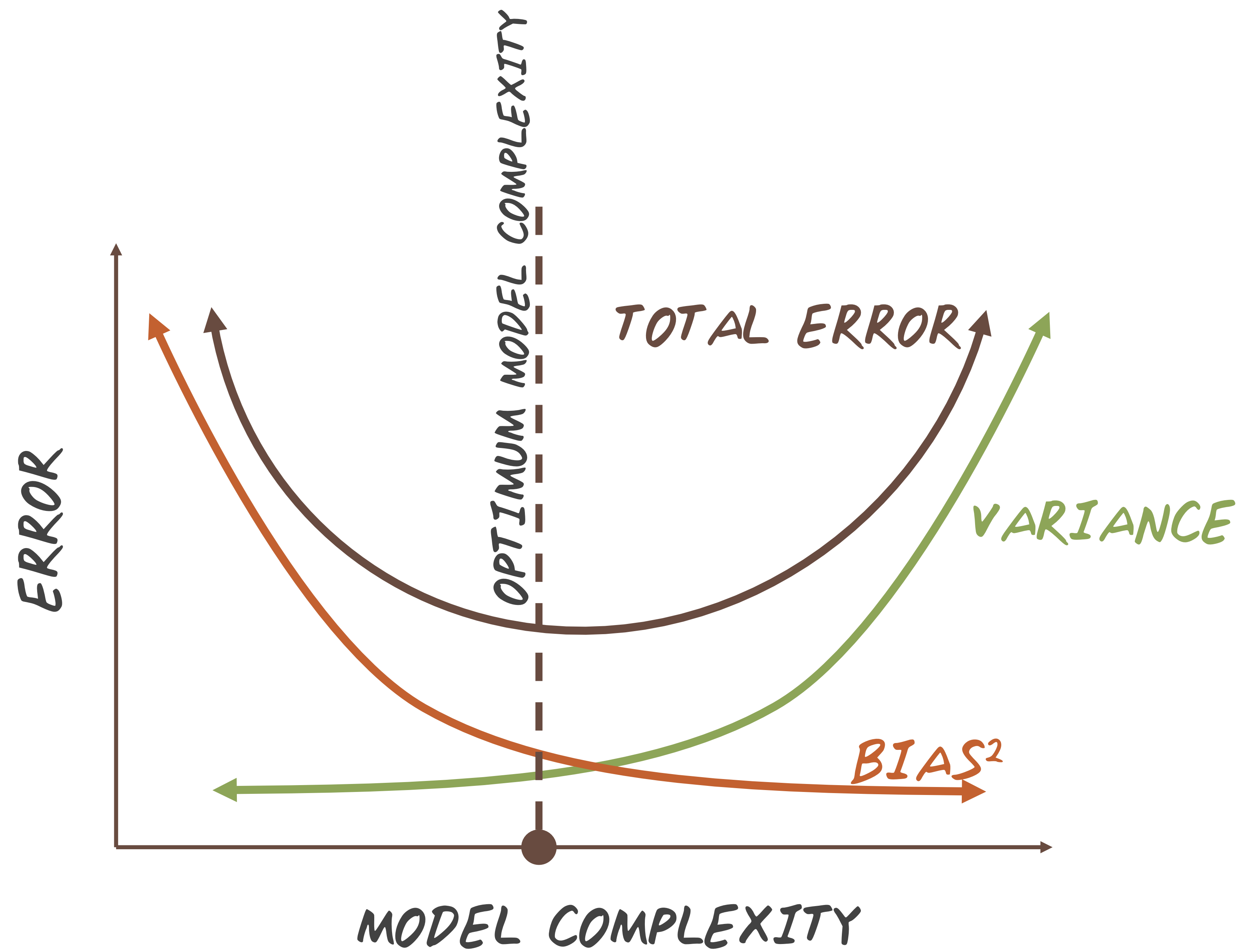
ENSEMBLE METHOD



BIAS VARIANCE TRADEOFF

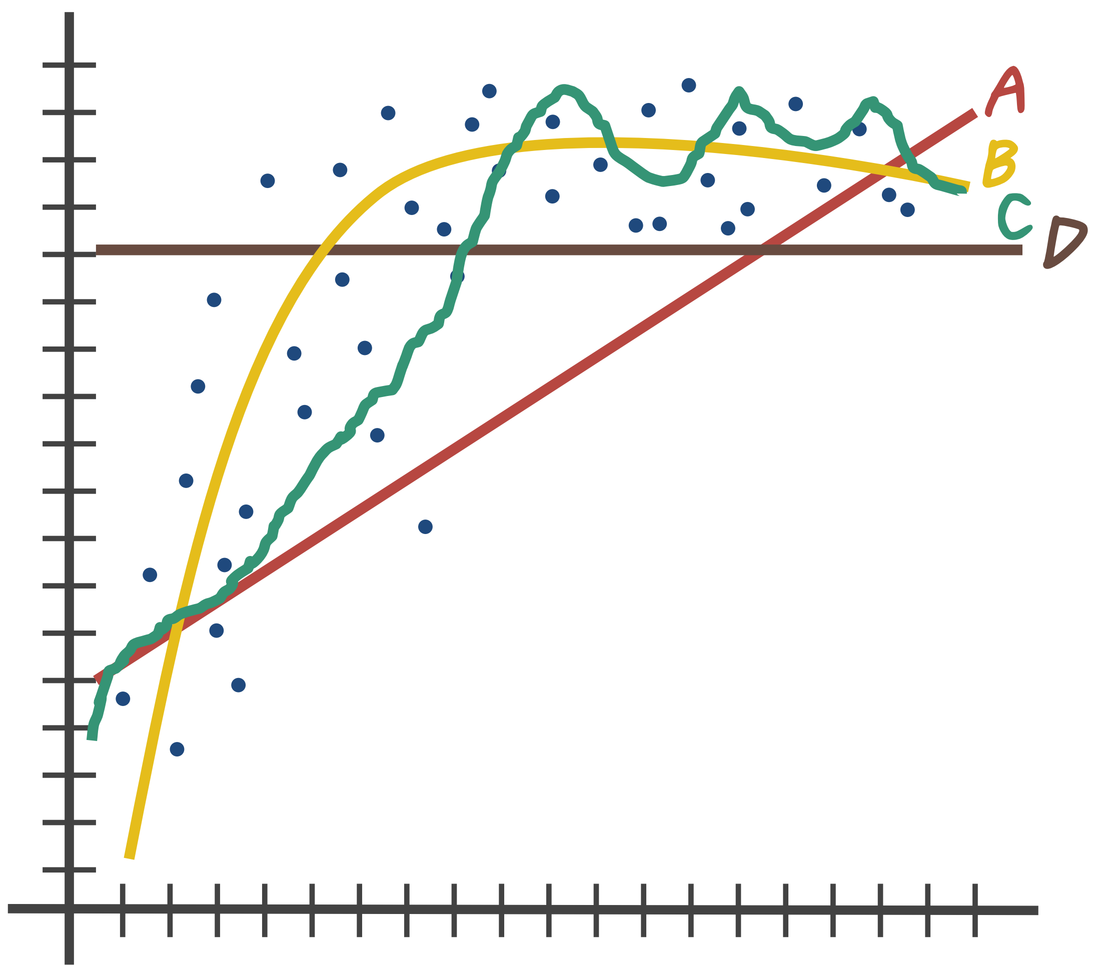


BIAS VARIANCE TRADEOFF



BIAS VARIANCE TRADEOFF QUIZ

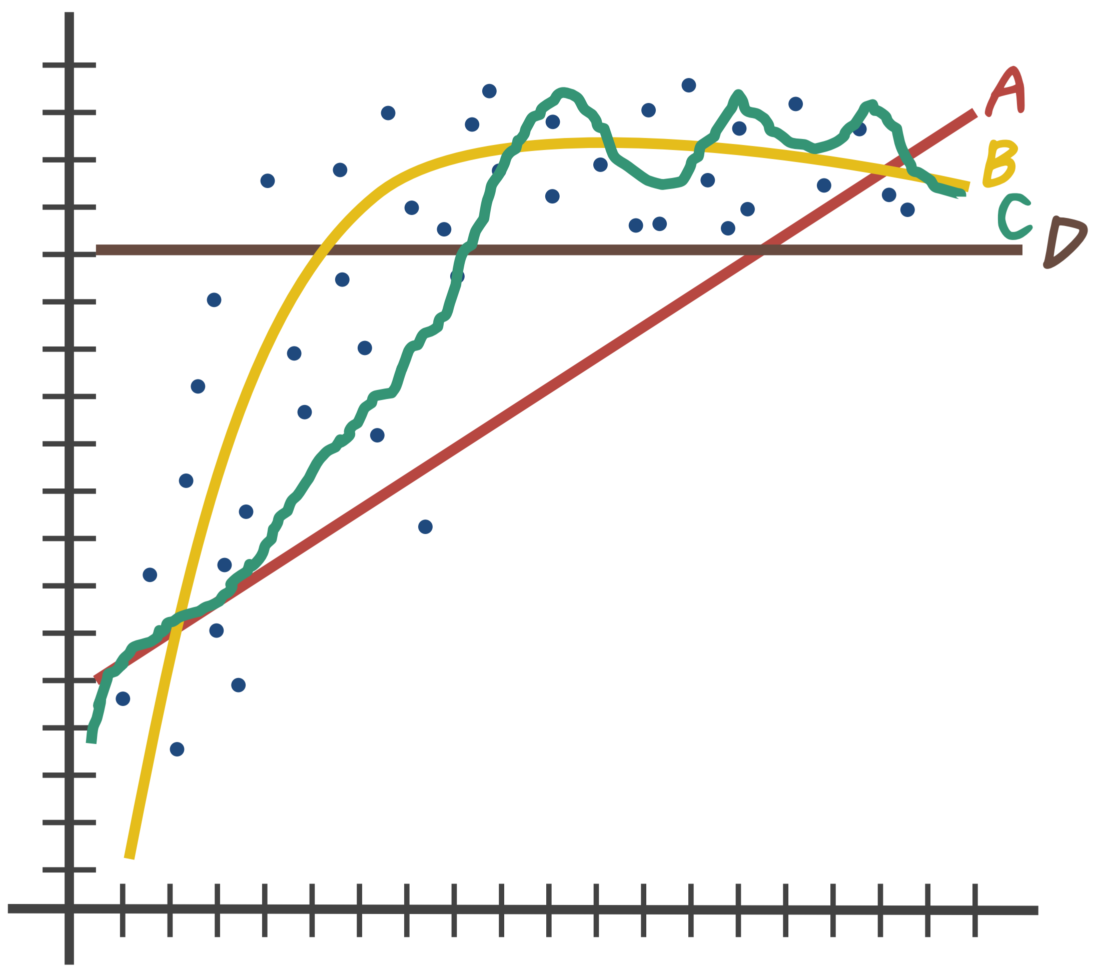
Rank from lowest to highest model complexity.



1. D
2. A
3. B
4. C

BIAS VARIANCE TRADEOFF QUIZ 2

Which is the best model?



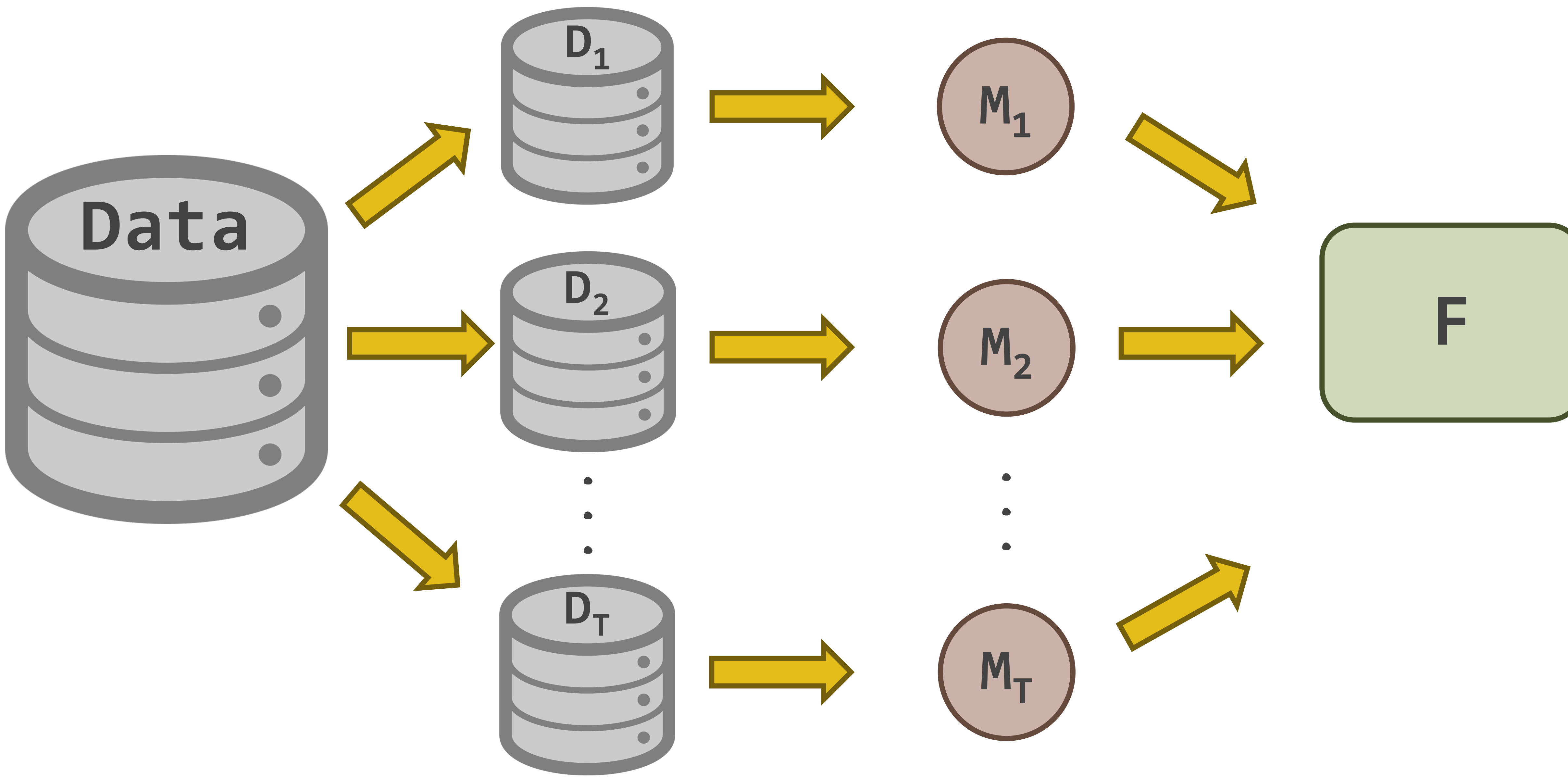
A. ☐

B. ☑

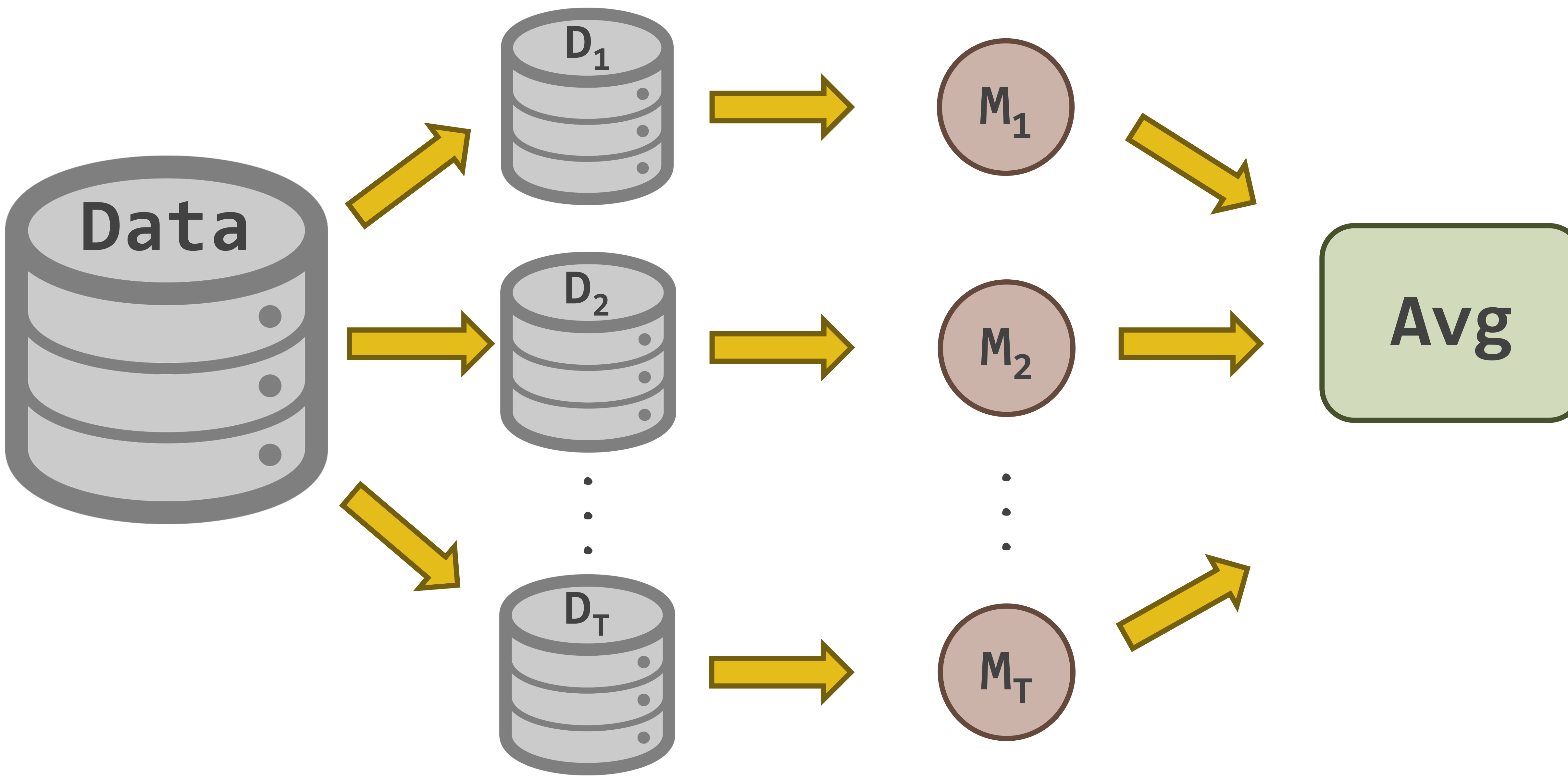
C. ☐

D. ☐

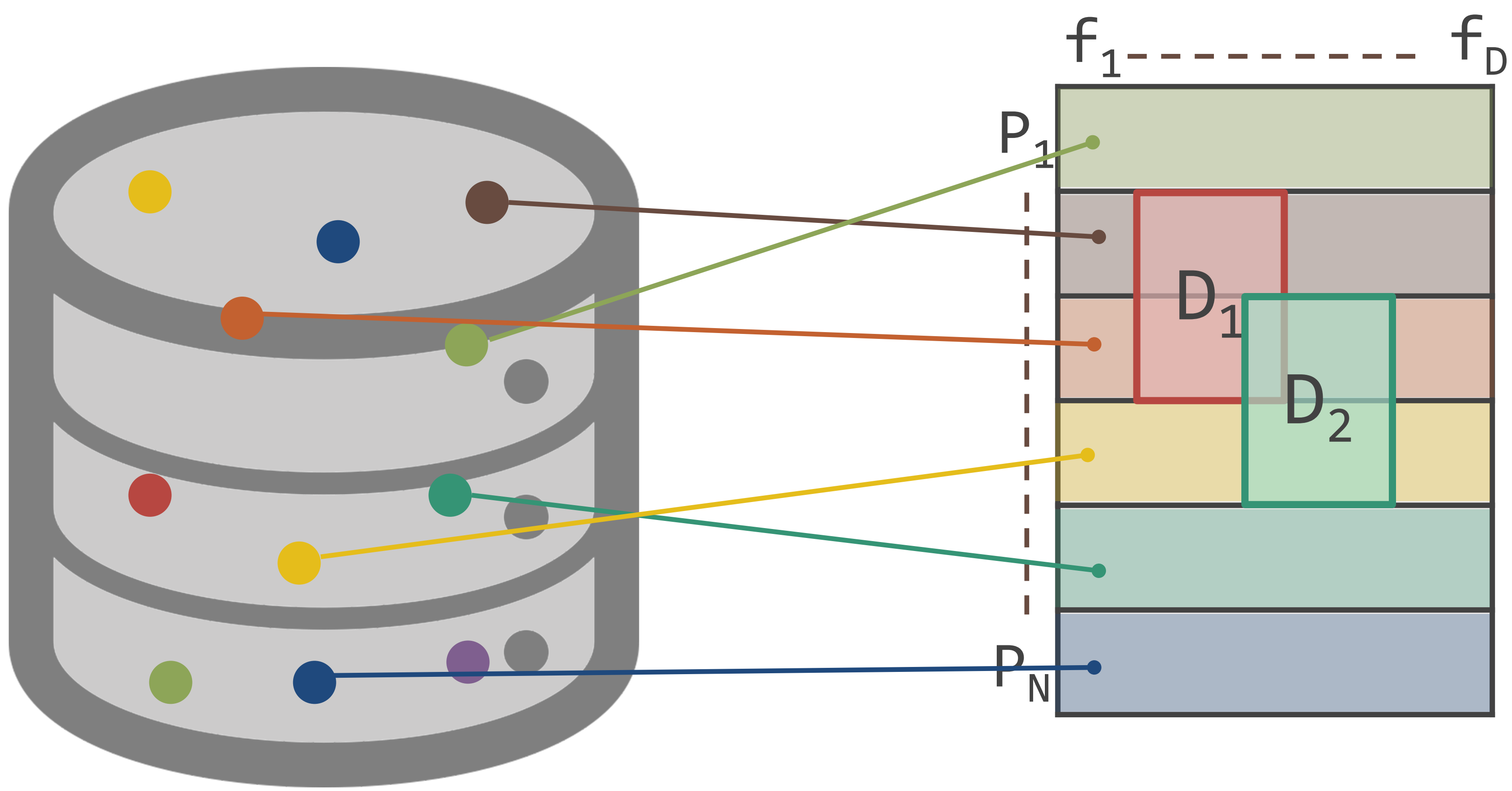
ENSEMBLE METHOD



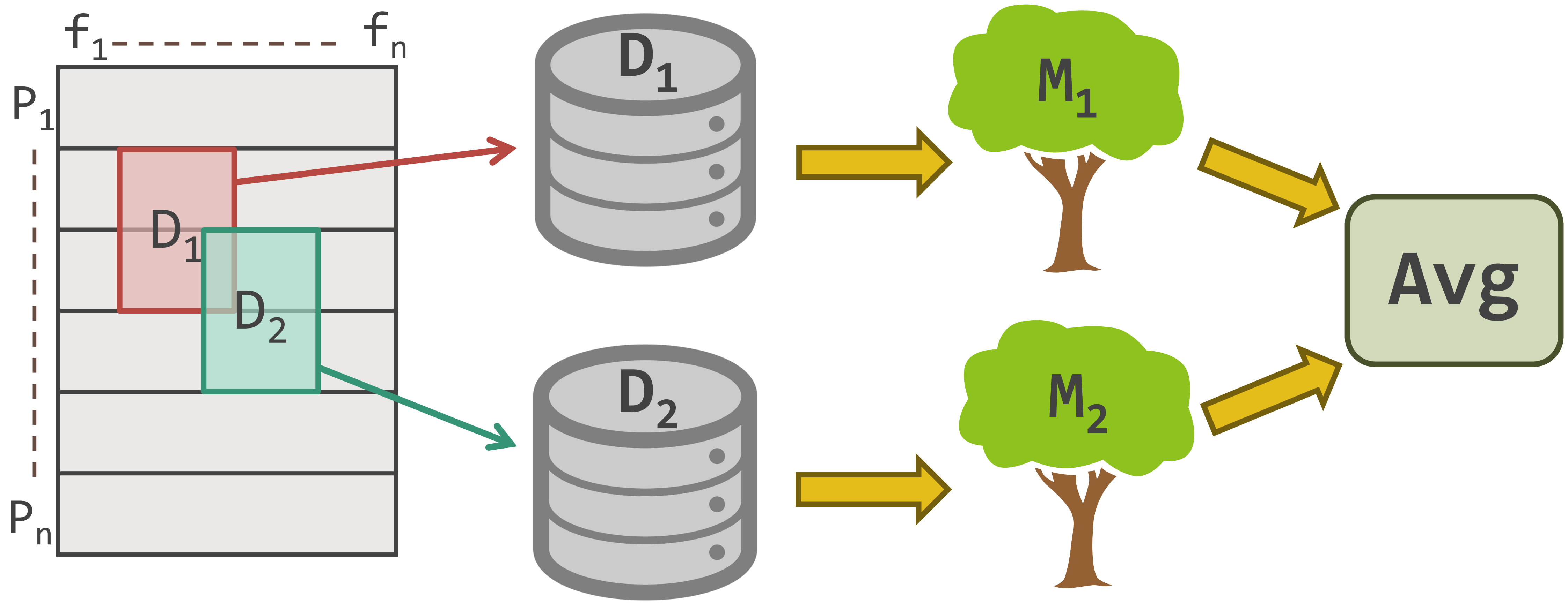
BAGGING



RANDOM FOREST



RANDOM FOREST



Random Forrest Algorithm

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.

   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

      i. Select m variables at random from the p variables.

      ii. Pick the best variable/split-point among the m.

      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

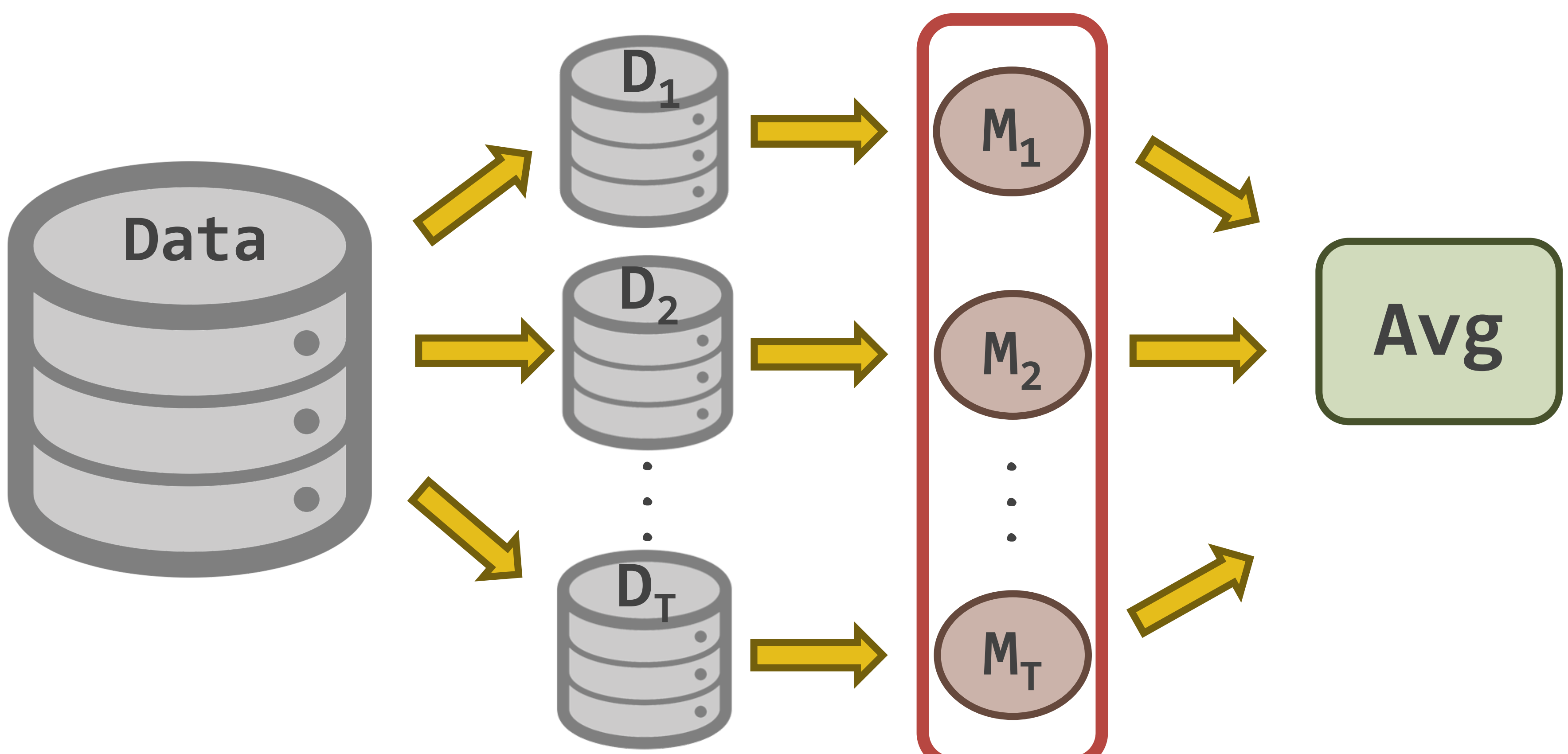
Regression: $\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\mathrm{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

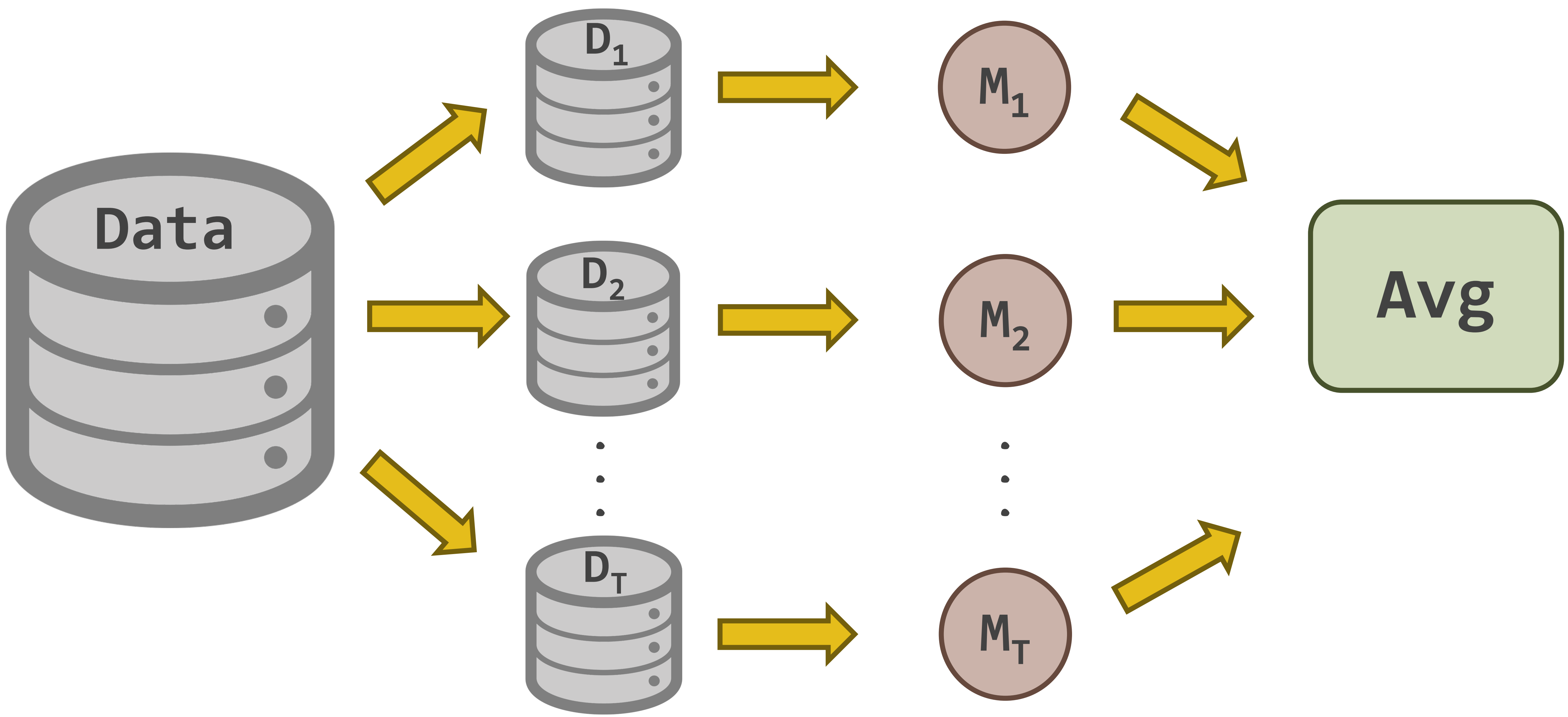
WHY BAGGING WORKS

Reduce Variance Without Increasing Bias

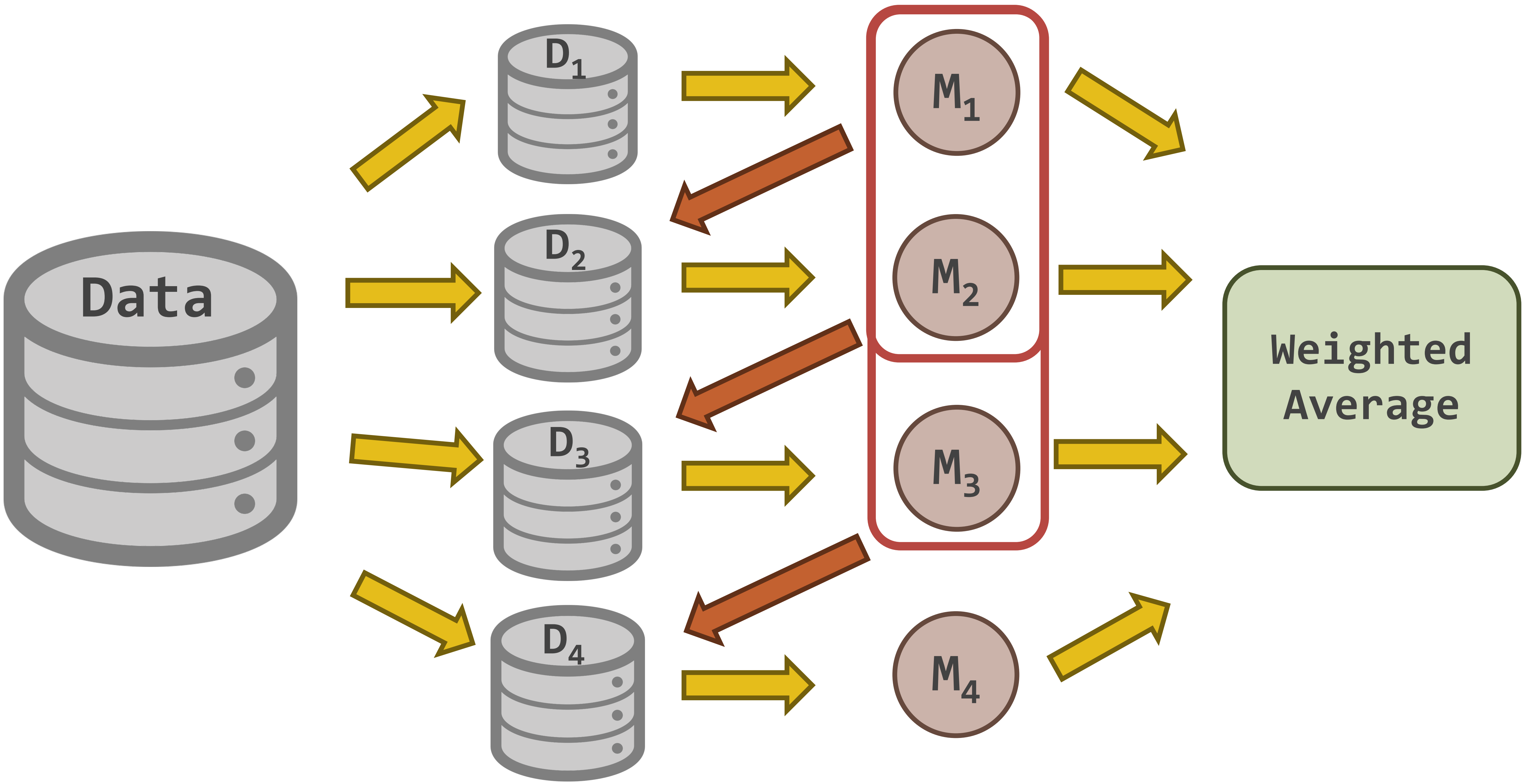
$$Var(\bar{X}) = \frac{Var(X)}{T}$$ (when X are independent)



DECREASE BIAS?



BOOSTING



Boosting algorithm (high level)

- For *t* from 1 to T
  - Learning the *t*-th weak classifier c with respect to a data distribution D
  - Assign weight to c based on c's performance
  - Adding c based on its weight to the final strong classifier C
  - Update data distribution by reweighting
- Output C

Ada-boost

- **AdaBoost**, short for "Adaptive Boosting"by Yoav Freund and Robert Schapire

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathscr{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathscr{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

https://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf

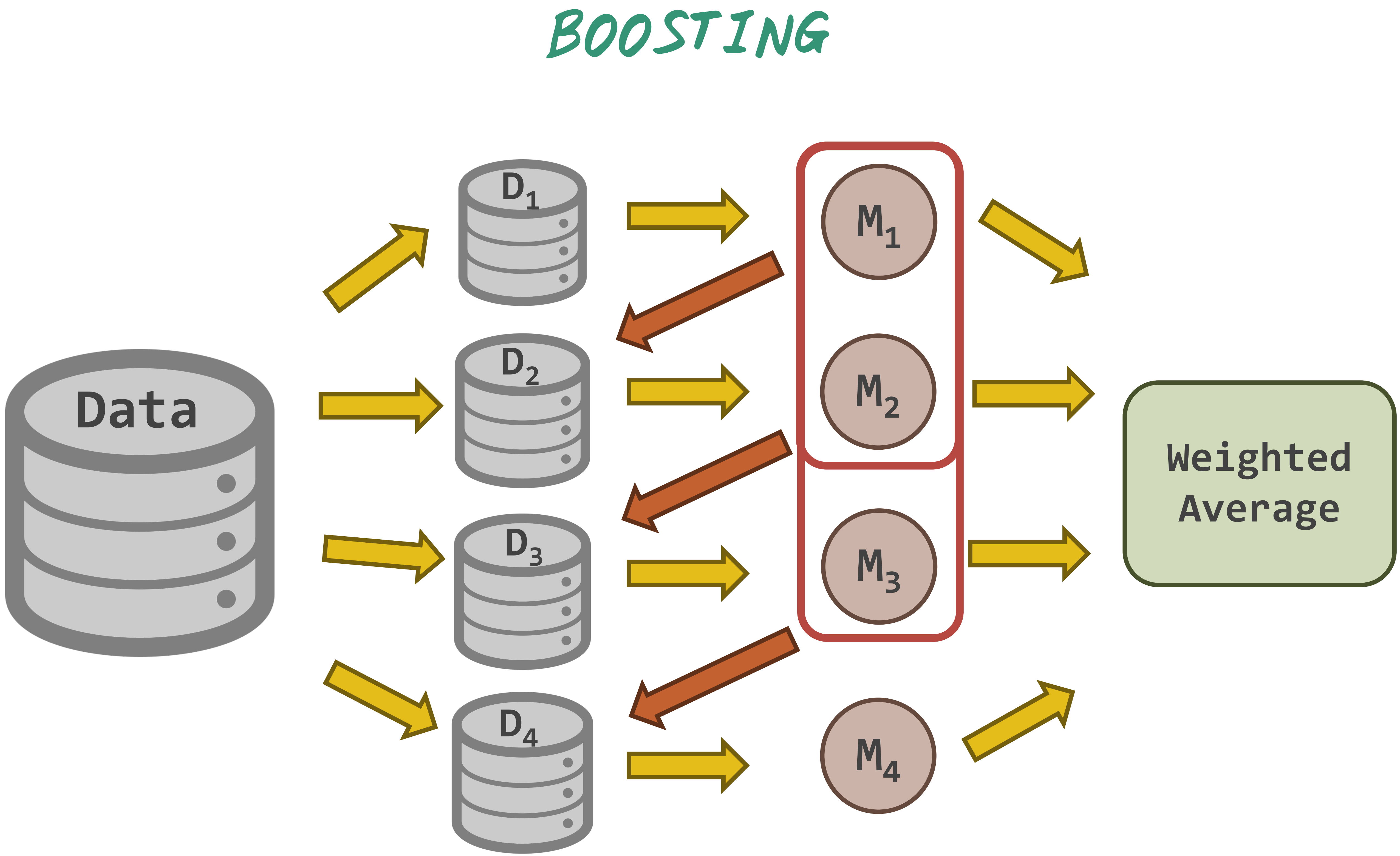
Boosting example

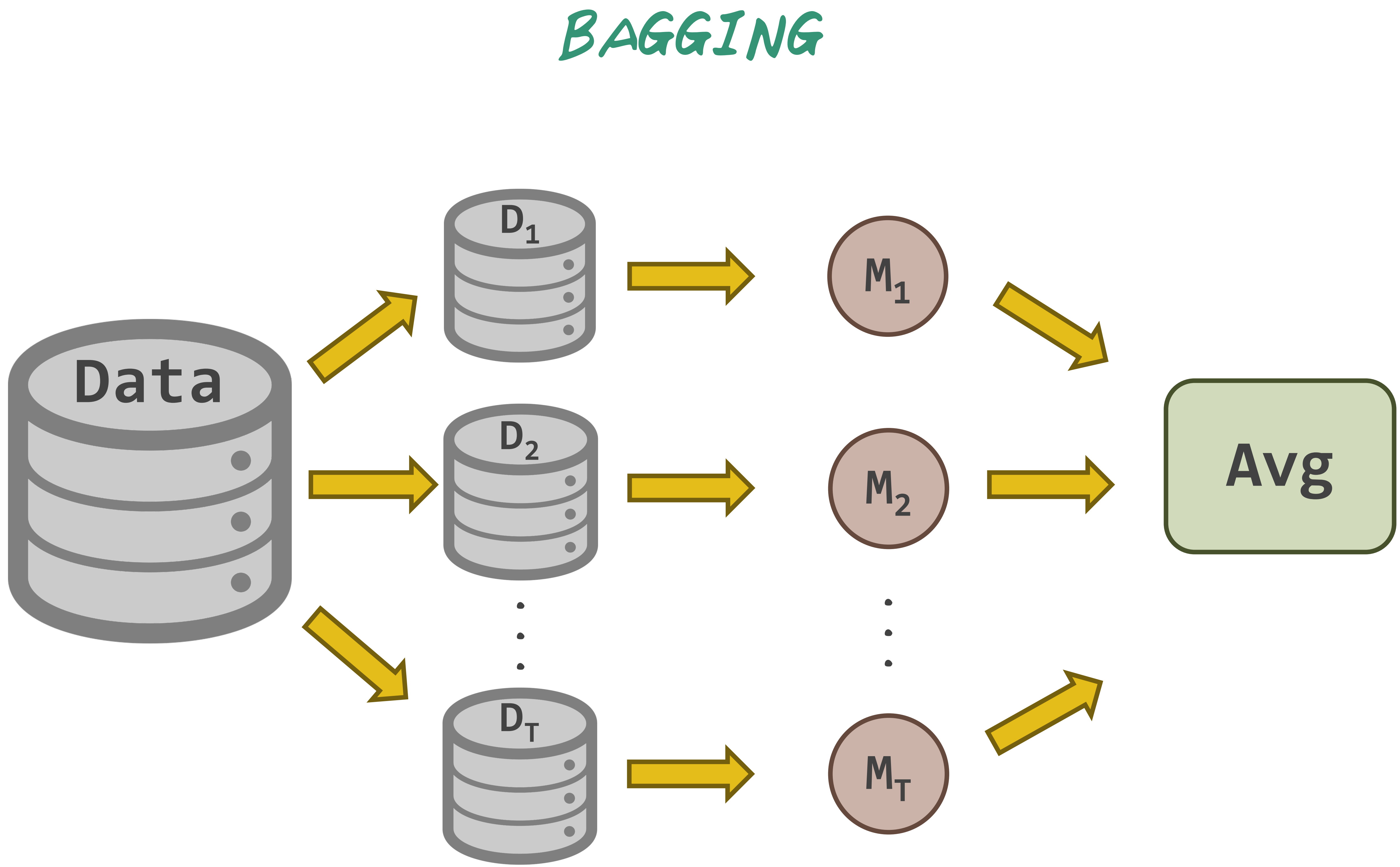
A sample of a single classifier on an imaginary set of data.	
	(Original) Training Set
Training-set-1:	1, 2, 3, 4, 5, 6, 7, 8

A sample of Boosting on the same data.	
	(Resampled) Training Set
Training-set-1:	2, 7, 8, 3, 7, 6, 3, 1
Training-set-2:	1, 4, 5, 4, 1, 5, 6, 4
Training-set-3:	7, 1, 5, 8, 1, 8, 1, 4
Training-set-4:	1, 1, 6, 1, 1, 3, 1, 5

- Sampling probability proportional to error
- Dependency between training sets
  - Errors on earlier training sets determine the sampling probability on later training sets

http://jair.org/media/614/live-614-1812-jair.pdf


BOOSTING
Data
D_1
D_2
D_3
D_4
M_1
M_2
M_3
M_4
Weighted Average



BAGGING
Data
D1
D2
DT
M1
M2
MT
Avg

BAGGING VS. BOOSTING QUIZ

	BAGGING	BOOSTING
COMBINING METHOD	✓ Simple average ○ Weighted average	○ Simple average ✓ Weighted average
PARALLEL COMPUTING	○ Hard ✓ Easy	✓ Hard ○ Easy
SENSITIVE TO NOISE	✓ Less ○ More	○ Less ✓ More
ACCURACY	✓ Good in all cases ○ Better in most cases	○ Good in all cases ✓ Better in most cases

SUMMARY FOR ENSEMBLE METHODS

PROS

- Simple
- Almost no parameter (except T)
- Flexible (combine with any algorithm)
- Theoretical guarantee

CONS

- Computational expensive due to computing multiple models
  - Both training and scoring need to deal with multiple models
- Lack of interpretation