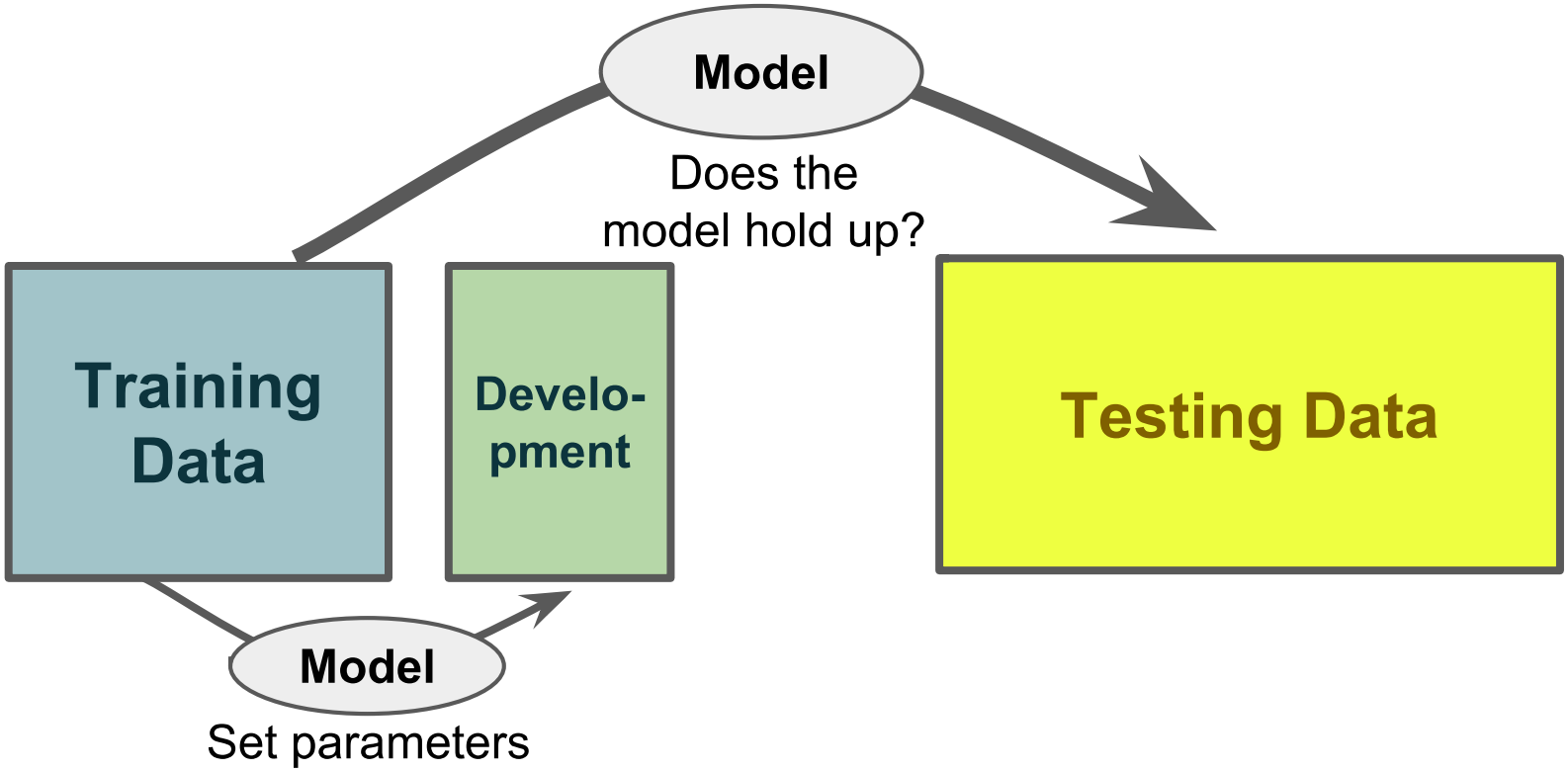
Regularization Comparison

Review, 3/31 - 4/5

- Confidence intervals
- Bootstrap

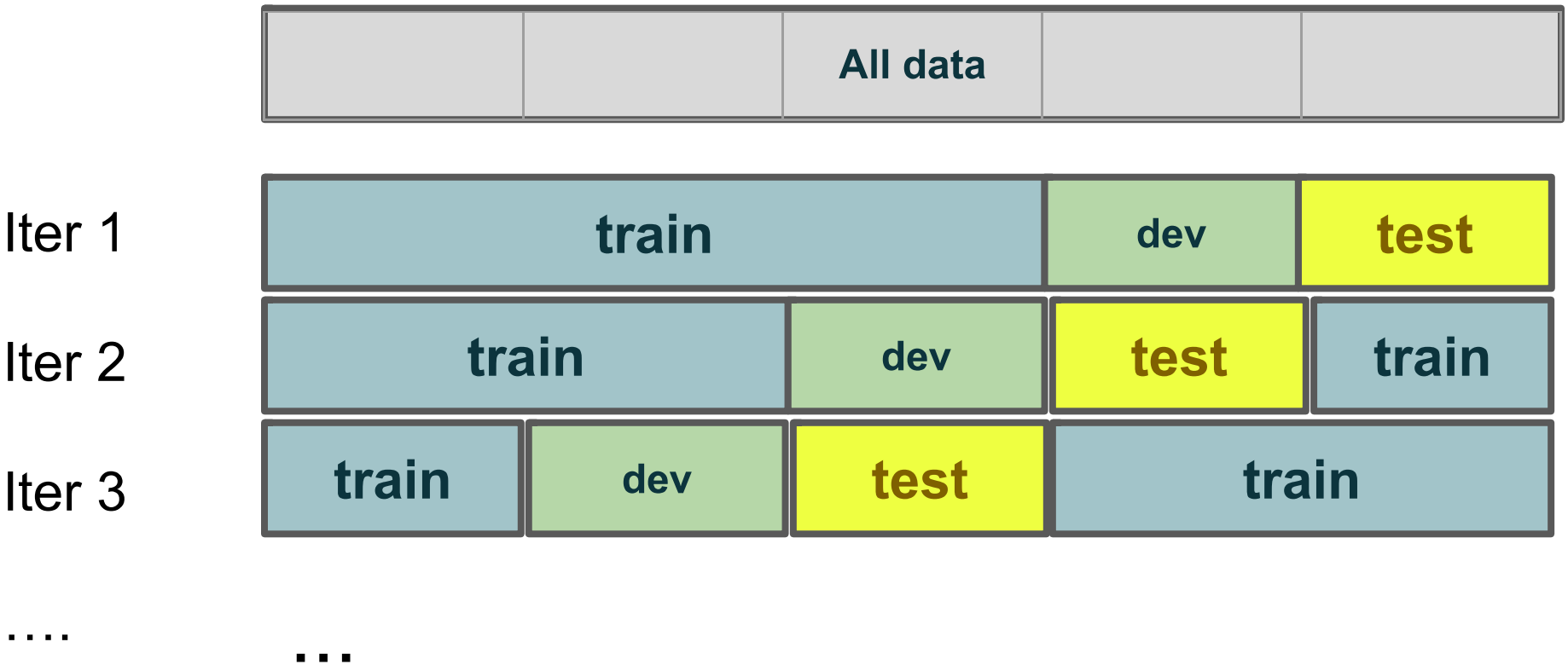
- Prediction Framework: Train, Development, Test
- Overfitting: Bias versus Variance
- Feature Selection: Forward Stepwise Regression
- Ridge Regression (L2 regularization)
- Lasso Regression (L1 regulatization)

Common Goal: Generalize to new data



N-Fold Cross-Validation

Goal: Decent estimate of model accuracy



Supervised vs. Unsupervised

Supervised

- Predicting an outcome $E(y|X)$
- Loss function used to characterize quality of prediction

$$L(y, \hat{y}) = (y - \hat{y})^2$$

Supervised vs. Unsupervised

Supervised

- Predicting an outcome $E(y|X)$
- Loss function used to characterize quality of prediction

$$L(y, \hat{y}) = (y - \hat{y})^2$$

Unsupervised

- No outcome to predict
- Goal: Infer properties of $P(X)$ without a supervised loss function.
- Often larger data.
- Don’t need to worry about conditioning on another variable.

K-Means Clustering

Clustering: Group similar observations, often over unlabeled data.

K-means: A “prototype” method
(i.e. not based on an algebraic model).

Euclidean Distance: $$d(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{m}(x_{ij} - x_{i'j})^2} = ||x_i - x_{i'}||$$

```
centers = a random selection of k cluster centers
until centers converge:
    1. For all x_i, find the closest center (according to d)
    2. Recalculate centers based on mean of euclidean distance
```

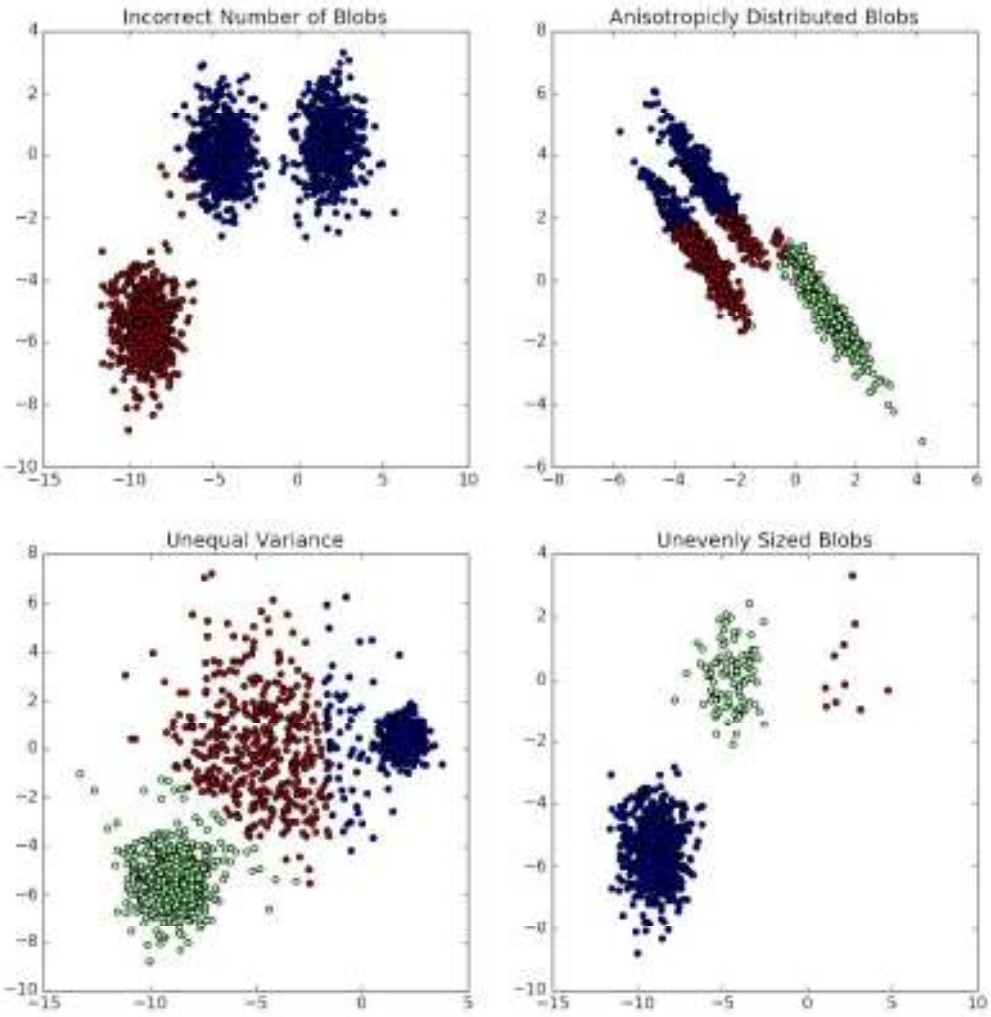
Review 4-7

- Cross-validation
- Supervised Learning
- Euclidean distance in m-dimensional space
- K-Means clustering

K-Means Clustering

Understanding K-Means

(source: Scikit-Learn)



Dimensionality Reduction - Concept

Dimensionality Reduction - PCA

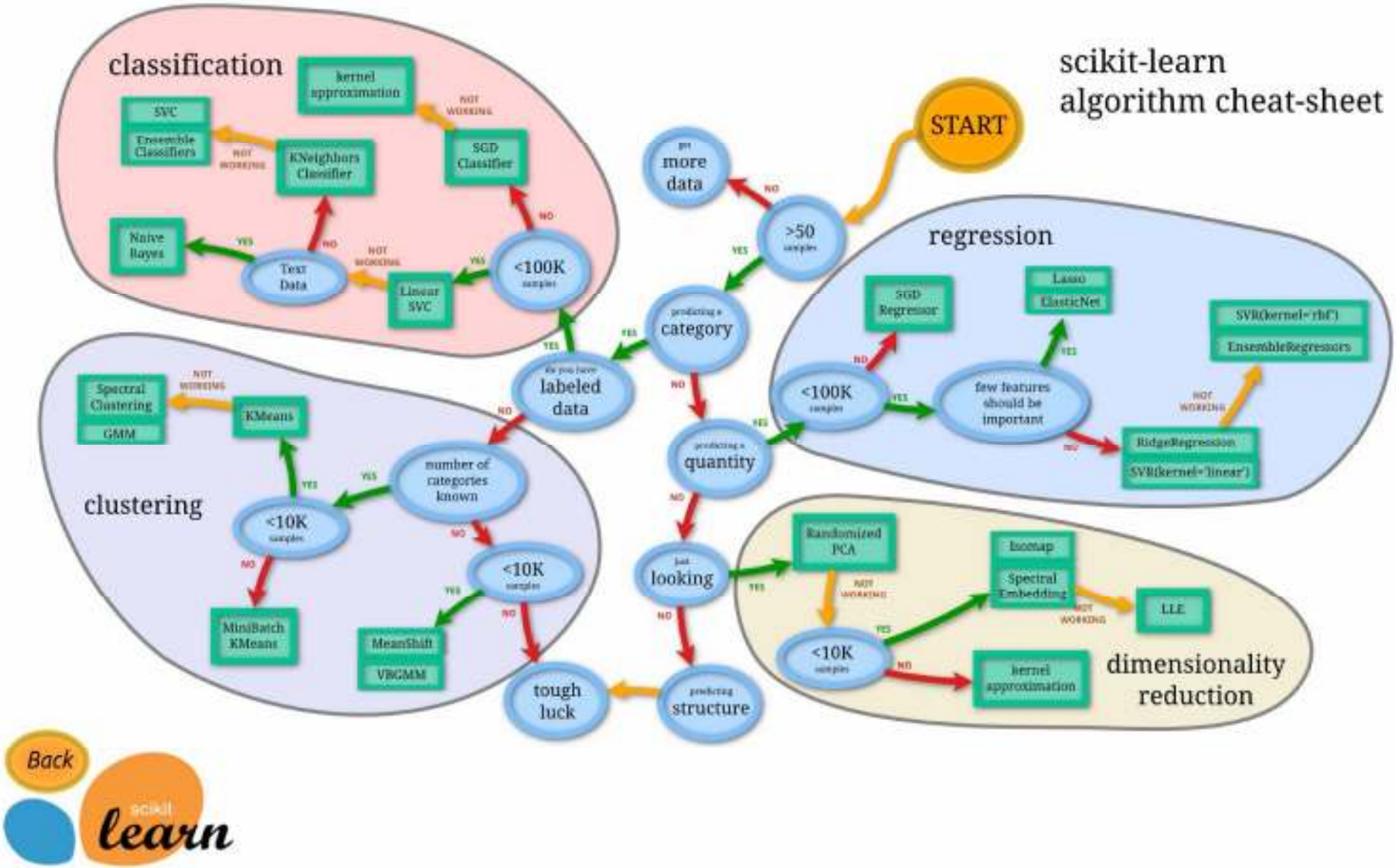
Linear approximates of data in q dimensions.

Found via *Singular Value Decomposition:*

$$X = UDV^T$$

Review 4-11

- K-Means Issues
- Dimensionality Reduction
- PCA
  - What is V (the components)?
  - Percentage variance explained


scikit-learn
algorithm cheat-sheet
START
more data
NO
>50
YES
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
predicting a category
YES
NO
<100K
NO
YES
few features should be important
YES
NOT WORKING
RidgeRegression
SVR(kernel='linear')
labeled data
YES
NO
Spectral Clustering
GMM
NOT WORKING
KMeans
clustering
number of categories known
YES
YES
predicting a quantity
YES
<10K
NO
NO
<10K
YES
NO
MiniBatch KMeans
MeanShift
VBGMM
just looking
NO
Randomized PCA
YES
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
<10K
YES
NO
kernel approximation
dimensionality reduction
tough luck
predicting structure
Back
scikit learn

Classification: Regularized Logistic Regression

$\lambda||\beta||_2^2$

$\lambda||\beta||_1$

Classification: Naive Bayes

Bayes classifier: choose the class most likely according to P(y|X).
(y is a class label)

Classification: Naive Bayes

Bayes classifier: choose the class most likely according to P(y|X).
(y is a class label)

Naive Bayes classifier: Assumes all predictors are independent given y.

$$P(Y = y|A = a, B = b, C = c) = p(y|a)p(y|b)p(y|c)$$

$$\mathrm{P}(y|X) = \prod_{i=1}^{m} \mathrm{P}(y|X_i)$$

Classification: Naive Bayes

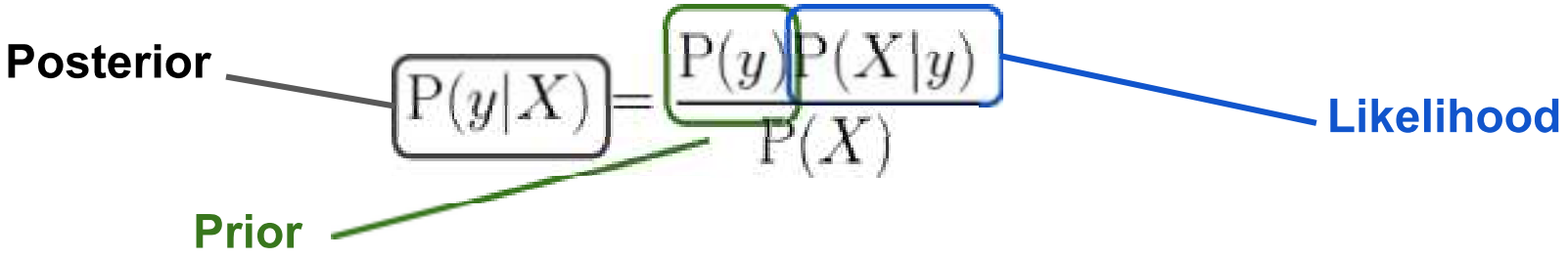
$$\mathrm{P}(y|X) = \frac{\mathrm{P}(y)\mathrm{P}(X|y)}{\mathrm{P}(X)}$$

Bayes Rule:

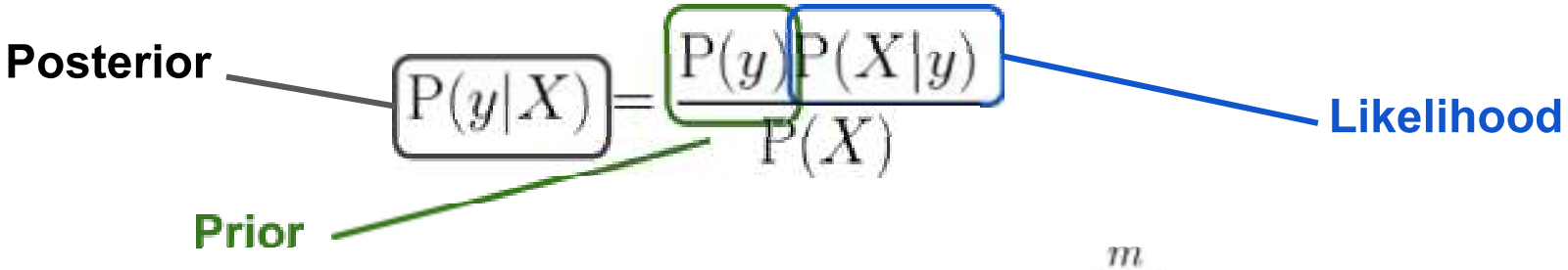
P(*A*|*B*) = P(*B*|*A*)P(*A*) / P(*B*)

$$\mathrm{P}(y|X) = \prod_{i=1}^{m} \mathrm{P}(y|X_i)$$

Classification: Naive Bayes



Classification: Naive Bayes

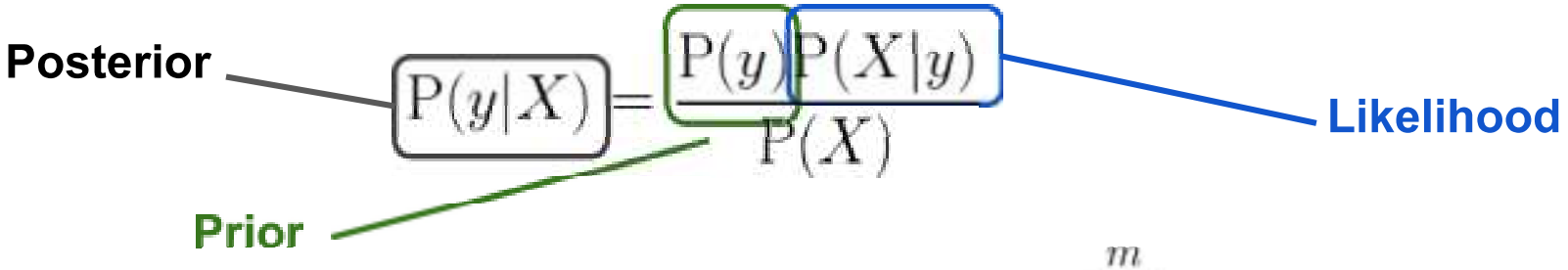


$$P(y|X) \propto P(y, X_1, ..., X_m) \propto P(y) \prod_{i=1}^{m} P(X_i|y)$$

Maximum a Posteriori (MAP): Pick the class with the maximum posterior probability.

$$\hat{y} = arg\max_{y} \; P(y) \prod_{i=1}^{m} P(X_i|y)$$

Classification: Naive Bayes



$$P(y|X) \propto P(y, X_1, ..., X_m) \propto P(y) \prod_{i=1}^{m} P(X_i|y)$$

Maximum a Posteriori (MAP): Pick the class with the maximum posterior probability.

Unnormalized Posterior

$$\hat{y} = arg\max_{y} P(y) \prod_{i=1}^{m} P(X_i|y)$$