Applied Artificial Intelligence

Session 9: Homework Discussion
Linear Algebra for AI and Machine Learning II

Fall 2018

NC State University

Lecturer: Dr. Behnam Kia

Course Website: https://appliedai.wordpress.ncsu.edu/

Sep 25, 2018

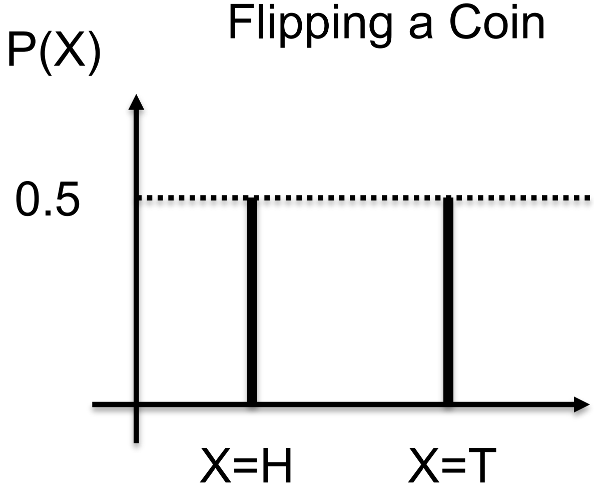
Homework 3:
Document Classification with Naïve Bayes Classifier

Discussion

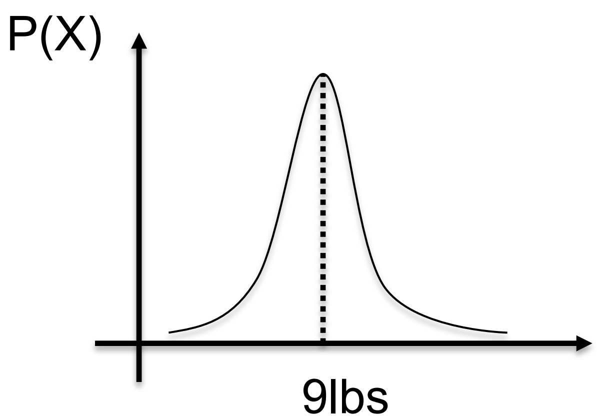
Probability Distribution Function

- Probability distribution function is a description of how likely a random variable or a set of variables is to take on each of its possible states.

Flipping a Coin

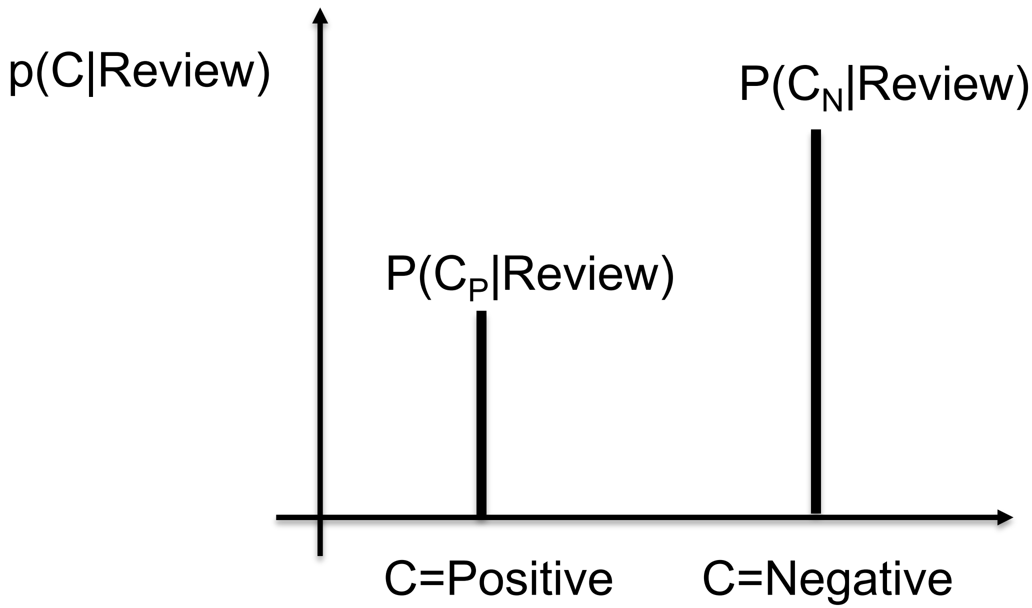


Cat's weight



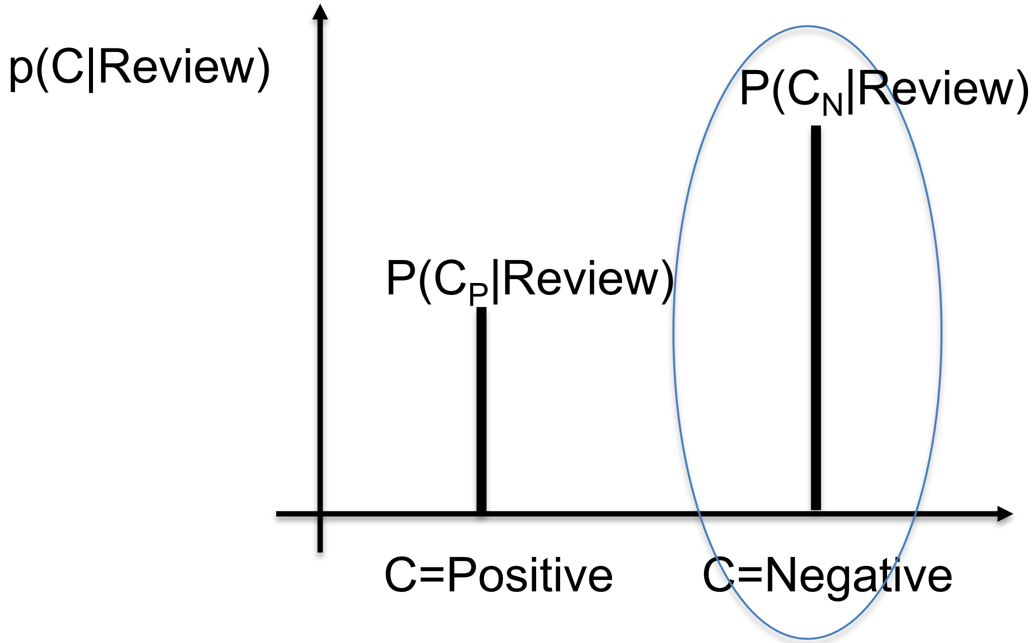
Homework 3

- You calculated conditional probability distribution function for each class:



Homework 3

- You calculated conditional probability distribution function for each class:
- And picked the class with the highest conditional probability.

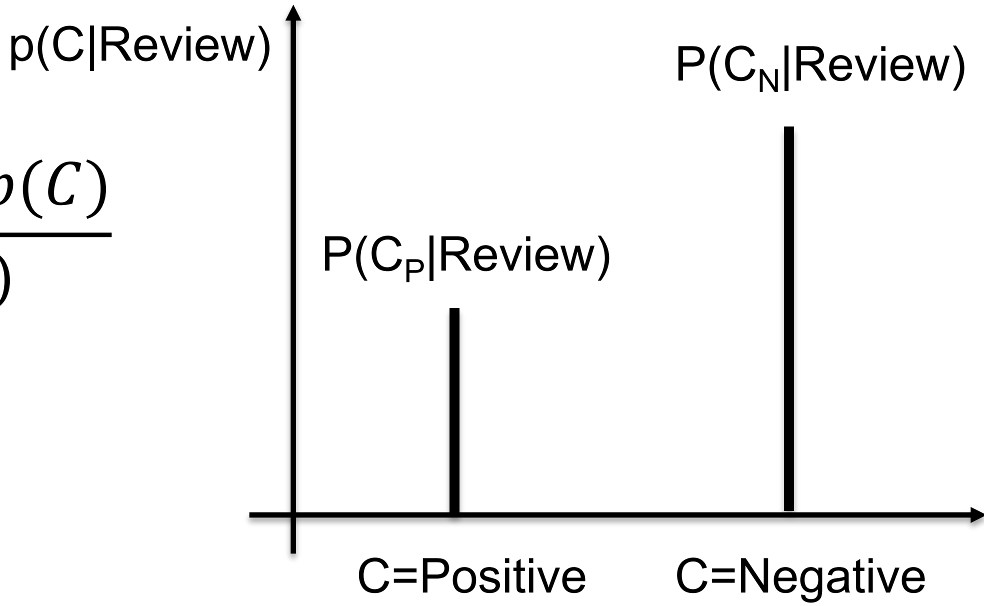


Homework 3

- You calculated conditional probability distribution function for each class:

$$p(C|Review) = \frac{p(Review|C)p(C)}{p(Review)}$$

$$posterior = \frac{likelihood \times prior}{evidence}$$



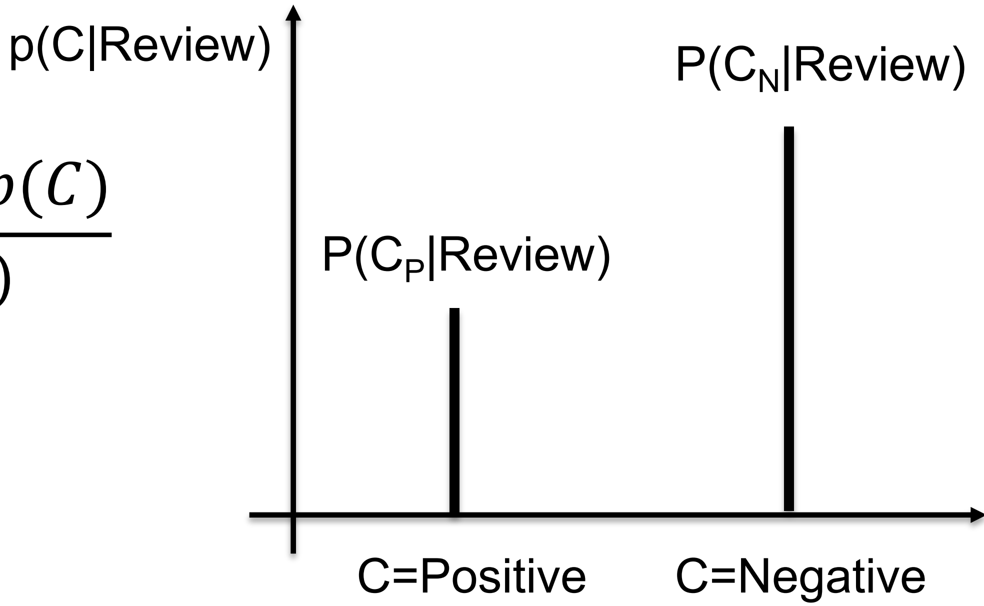
Homework 3

- You calculated conditional probability distribution function for each class:

Naïve Bayes to Approximate

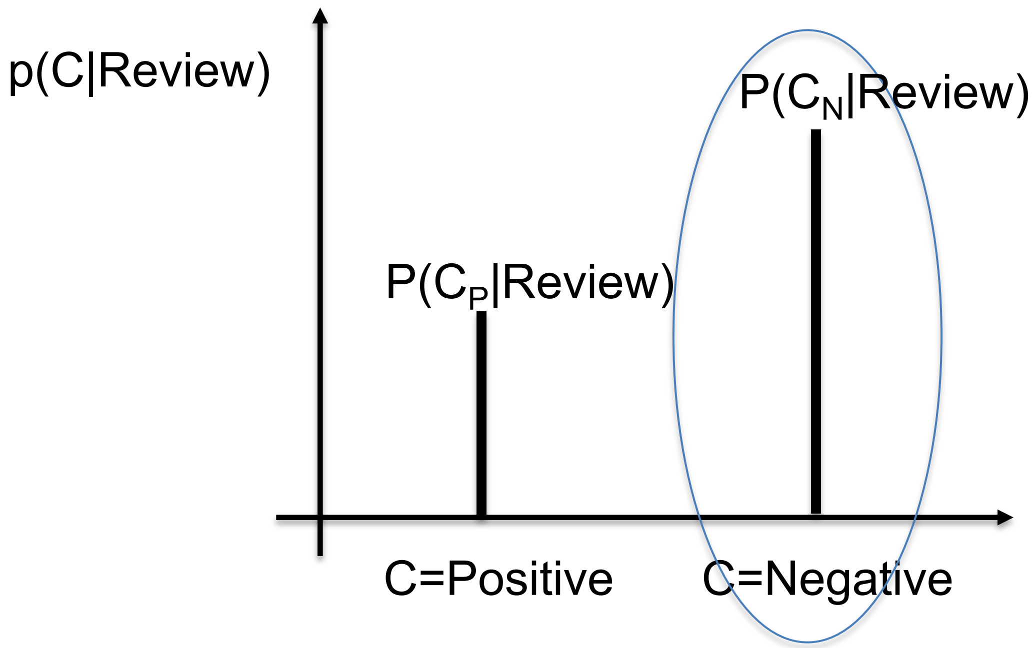
$$p(C|Review) = \frac{p(Review|C)p(C)}{p(Review)}$$

$$posterior = \frac{likelihood \times prior}{evidence}$$



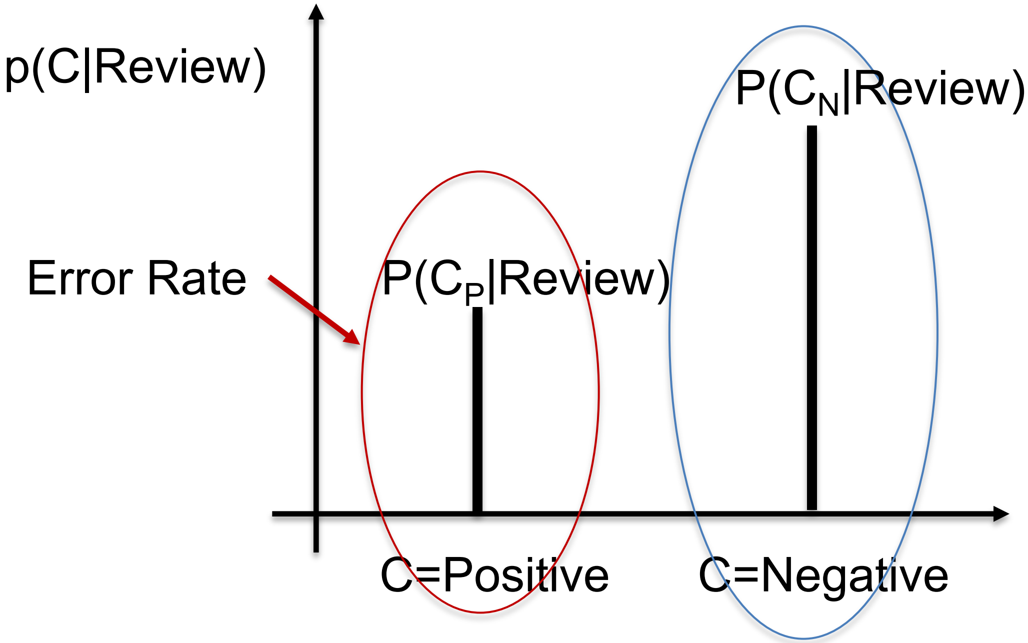
What is the Error Rate?

- You calculated conditional probability distribution function for each class:
- And picked the class with the highest conditional probability.



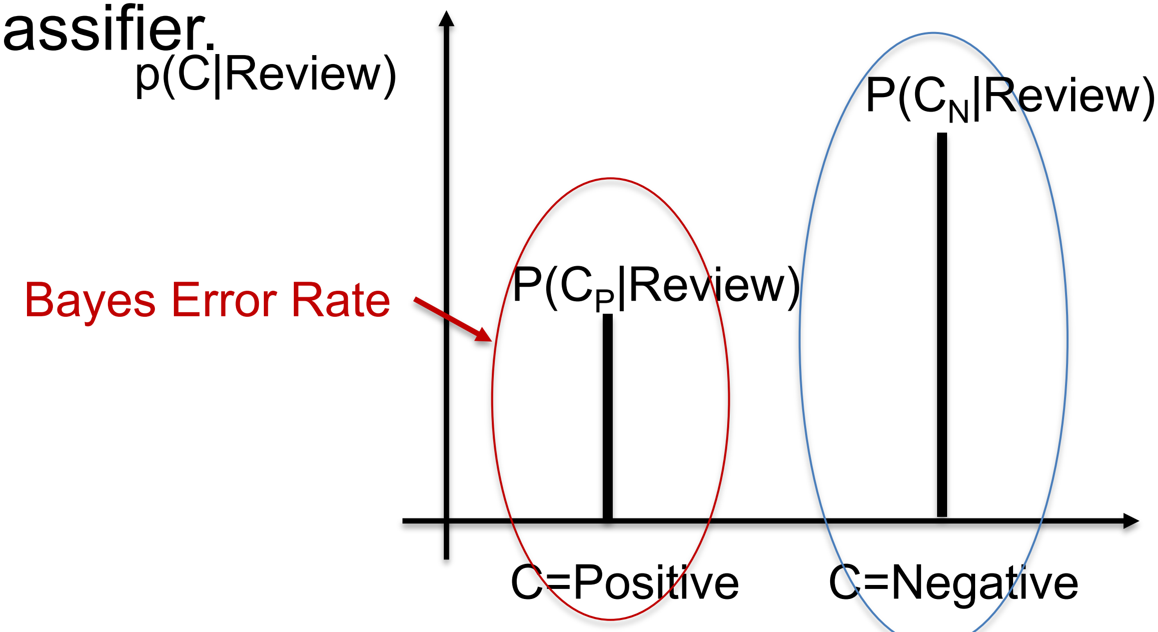
What is the Error Rate?

- You calculated conditional probability distribution function for each class:
- And picked the class with the highest conditional probability.

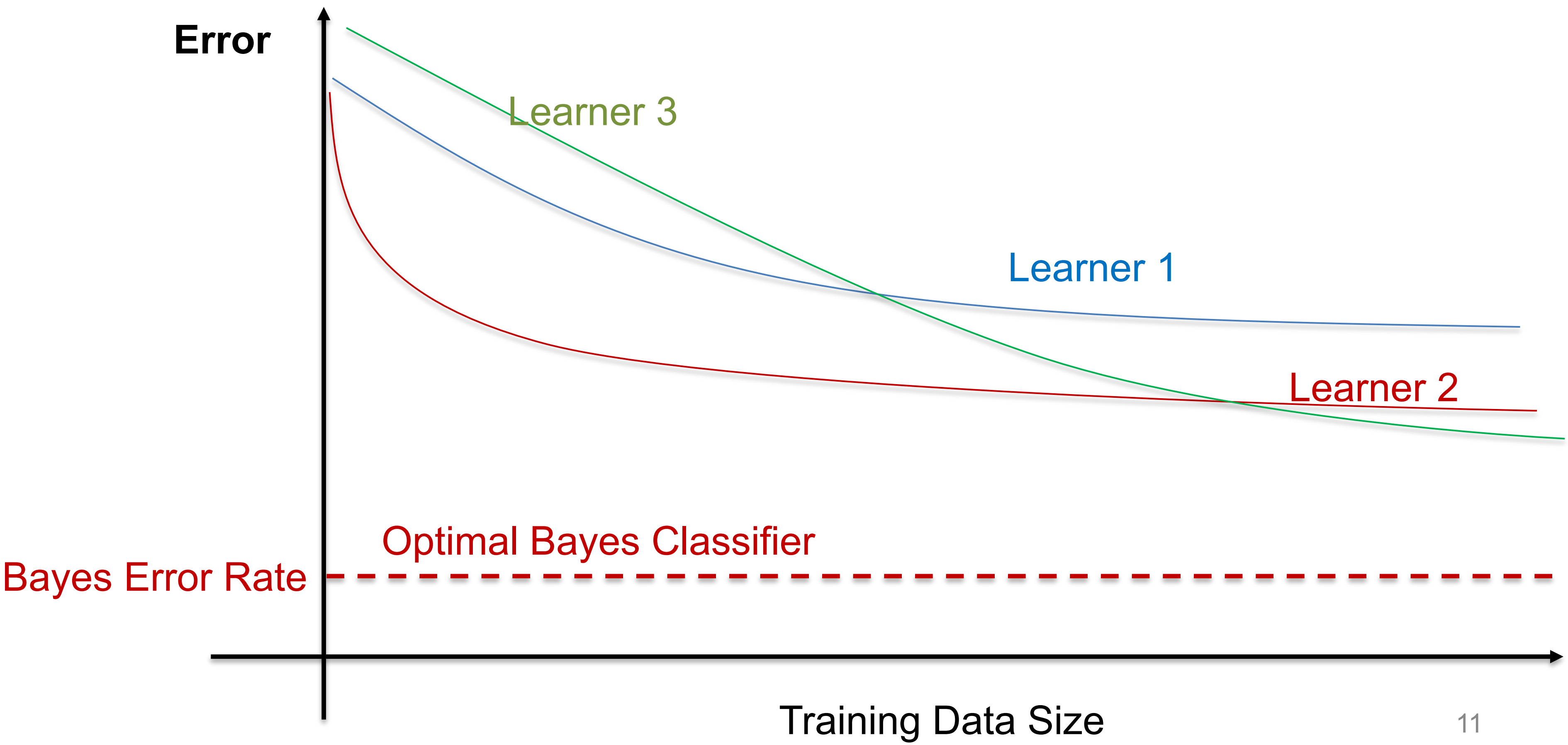


Bayes Error Rate

- You calculated conditional probability distribution function for each class:
- And picked the class with the highest conditional probability.
- **Bayes error rate** is the lowest possible error rate for ANY classifier.



Performance of Classifiers



Let’s don’t confuse luck with statistical performance!

Homework 2 vs Homework 3

How Do you Compare Homework 2 with 3?

Homework 3

$$c_{NB} = \underset{c \in \{Positive, Negative\}}{\operatorname{argmax}} p(c) \prod_i p(x_i \mid c)$$

Homework 2

How Do you Compare Homework 2 with 3?

Homework 3

$$c_{NB} = \underset{c \in \{Positive, Negative\}}{\operatorname{argmax}} \; p(c) \prod_i p(x_i \mid c)$$

Generative Model

Homework 2

Discriminative Model

Recommended Reading

Andrew Y. NG, and Michael I. Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes." Advances in neural information processing systems. 2002.

On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes

What is a Generative model, and What is a Discriminative Model.

On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes

Starting Point of the Article:

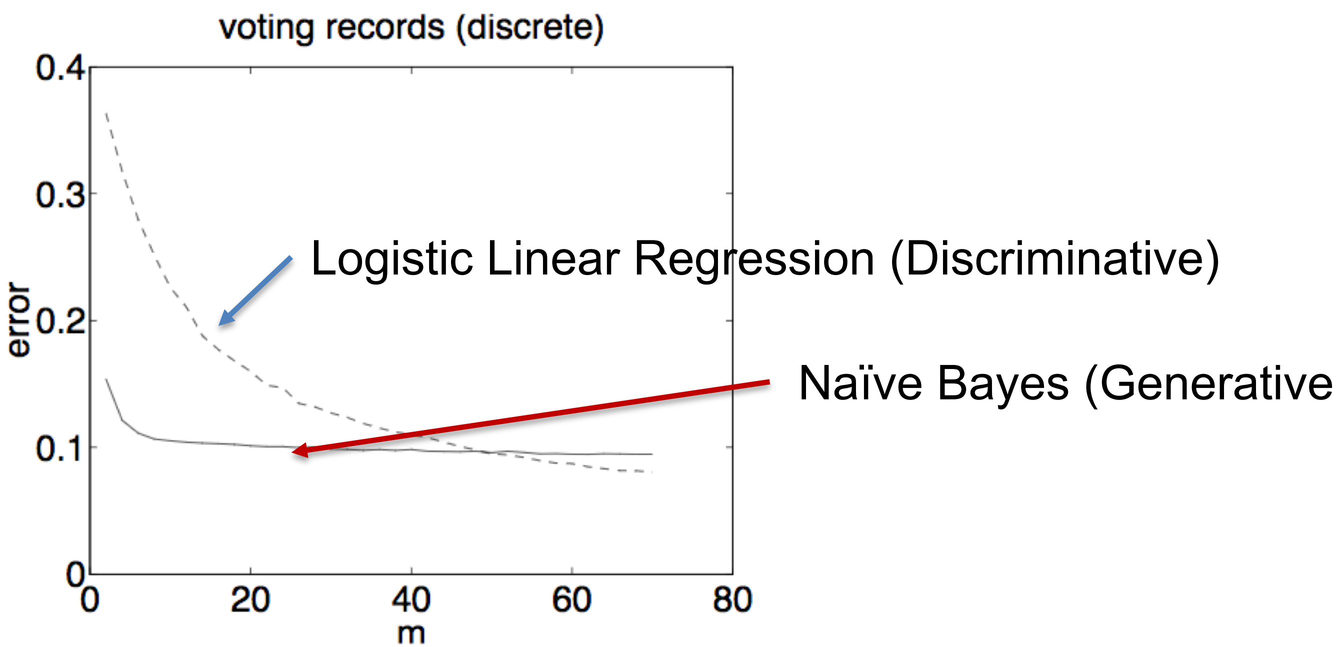
Common Wisdom: “One Should Solve the Classification Problem Directly and never [?] solve a more general problem as an intermediate step [such as modeling p(Review|C)]”

On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes

Results:

1: When m, the size of training data increases:

Asymptotic error of Discriminative Models$\leq$ Asymptotic error of Generative Models



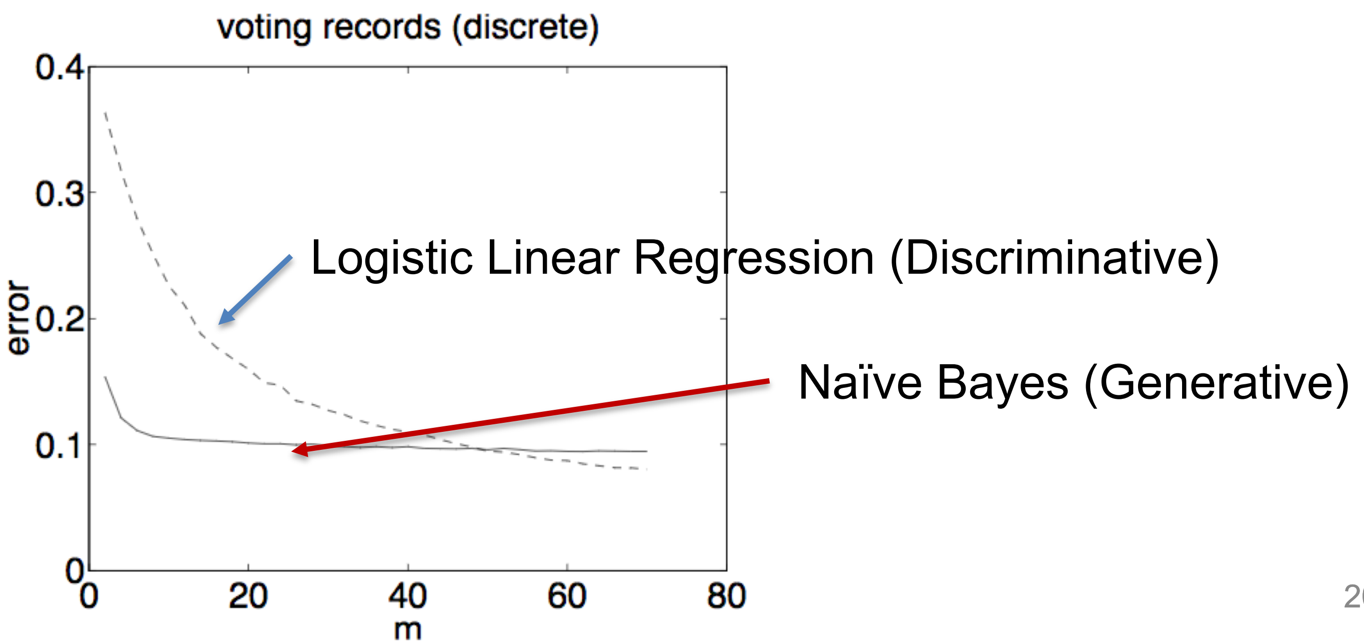
On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes

Results:

2- Generative Models converge to their asymptotic errors fasters.

For small datasets:

Error of Generative Models ≤ Error of Discriminative Models



Recommended Reading

Ng, Andrew Y., and Michael I. Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." Advances in neural information processing systems. 2002.

Linear Algebra for AI and Machine Learning II

Computational complexity for basic PCA

- Complexity of PCA: $O(n_{max}^2 n_{min})$

where $n_{max} = \max(n_{sample}, n_{features})$

$n_{min} = min(n_{sample}, n_{features})$

$$X = \underbrace{\begin{bmatrix} 0.3 & \cdots & -0.1 \\ \vdots & \ddots & \vdots \\ 0.9 & \cdots & 0.43 \end{bmatrix}}_{n_{samples}} \left.\vphantom{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}\right\} n_{features}$$

Computational complexity for basic PCA

- Computational Complexity of PCA: $O(n_{max}^2 n_{min})$

where $n_{max} = \max(n_{sample}, n_{features})$
$n_{min} = min(n_{sample}, n_{features})$

- Computational Complexity of Randomized PCA:

$O(n_{max}^2 n_{components})$

```
ipca = PCA(n_components=ncomponents,
svd_solver="randomized")
```

PCA's limitations

- It is a linear transformation

  Use Kernel PCA

Kernel functions [implicitly] transform the data to a higher dimensional feature space, where linear PCA can be applied.

```
kpca = KernelPCA(kernel="rbf",
fit_inverse_transform=True, gamma=10)
```

PCA's limitations

- It is a linear transformation

  Use Kernel PCA

Kernel functions [implicitly] transform the data to a higher dimensional feature space, where linear PCA can be applied.

```
kpca = KernelPCA(kernel="rbf",
fit_inverse_transform=True, gamma=10)
```

PCA's limitations

- It is a linear transformation.

- [In scikit-learn implementation of PCA] the entire data matrix should fit into the memory.
  - Solution: Use Incremental PCA, called I IncrementalPCA.

```
ipca = IncrementalPCA(n_components=n_components,
batch_size=10)
```

Homework

- Go to scikit-learn website, and follow any of PCA or decomposition examples

- http://scikit-learn.org/stable/auto_examples/index.html

Final Project Proposal

- Teams of three people.
- (Preferably) choose a problem from your own field to solve.
- 2-minute presentation in the class (4-5 slides).
- An extended executive summary (1-2 page word document.)