Naive Bayes and Perceptron

S Pradeep Kumar

2018-04-12 Thu

Outline

Bayes Theorem

Prior Probability

Likelihood of the evidence 'E' if the Hypothesis 'H' is true

$$P(H|E) = \frac{P(H) * P(E|H)}{P(E)}$$

Posterior Probability of 'H' given the evidence

Priori probability that the evidence itself is true

We know how to produce the evidence given the hypothesis.
Now, we want to produce the hypothesis given the evidence.

Simplification: Independent instances

Assuming that the points were independently sampled makes our lives easier.

$$
\begin{aligned}
L(\theta|D) &= L(\theta|x(1),...,x(n)) \\
&= p(x(1),...,x(n)|\theta) \\
&= \prod_{i=1}^{n} p(x(i)|\theta)
\end{aligned}
$$

If instances are independent, likelihood is product of probs

If each point has k features, then we need to learn at least 2^k parameters. However, that is still too much to learn from a limited training set.

The Naive Bayes assumption: Conditional Independence

$P(X_1 \mid Y, X_2) = P(X_1 \mid Y)$
where X_1 and X_2 are features of the data and Y is the class label.
Then, the number of parameters we have to learn becomes $2k$, not 2^k.

The Naive Bayes assumption: Conditional Independence

$P(X_1 \mid Y, X_2) = P(X_1 \mid Y)$
where X_1 and X_2 are features of the data and Y is the class label.
Then, the number of parameters we have to learn becomes 2k, not 2^k.

$$L(\theta|D) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i;\theta) \qquad \textbf{General likelihood}$$

$$\propto \prod_{i=1}^{n} p(\mathbf{x}_i|y_i;\theta)p(y_i|\theta) \qquad \textbf{Bayes rule}$$

$$\propto \prod_{i=1}^{n}\prod_{j=1}^{p} p(x_{ij}|y_i;\theta)p(y_i|\theta) \qquad \textbf{Naive assumption}$$

Parameters and Prediction

Parameters:

- $P(X_i \mid Y = 0)$ and $P(X_i \mid Y = 1)$ for each feature X_i
- $P(Y = 0)$

Prediction:
Given the parameters, how would we classify a new instance?
Pick the value of Y for which $P(Y) \times \Pi_i P(X_i \mid Y)$ is maximum.

Algorithm: Maximum Likelihood Estimation

Basically, $P(X_i = a \mid Y = b) = N(X_i = a, Y = b)/N(Y = b)$
$P(Y = a) = N(Y = a) / N$

Laplace Smoothing

What about zero frequencies?

Laplace Smoothing

What about zero frequencies?

$P(X = a \mid Y = b) = (N(X = a, Y = b) + 1)/(N(Y = b) + k)$

- Numerator: add 1
- Denominator: add k, where k = number of possible values of X

Worked-out example: Spam!

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

Worked-out example: Spam!

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

$P(\text{spam}) = 2/5$

Worked-out example: Spam!

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

P(spam) = 2/5
P(CS373 | spam) = (0 + 1) / (2 + 2) = 1/4
P(CS373 | not-spam) = (1 + 1) / (3 + 2) = 2/5

Worked-out example: Spam!

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

P(spam) = 2/5
P(CS373 | spam) = (0 + 1) / (2 + 2) = 1/4
P(CS373 | not-spam) = (1 + 1) / (3 + 2) = 2/5
P(F = high | spam) = (1 + 1) / (2 + 3) = 2/5
P(F = medium | spam) = (0 + 1) / (2 + 3) = 1/5
P(F = low | spam) = 1 - 2/5 - 1/5 = 2/5

Worked-out example: Your turn

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

$P(F = \text{high} \mid \text{not-spam}) = ?$
$P(F = \text{medium} \mid \text{not-spam}) = ?$
$P(F = \text{low} \mid \text{not-spam}) = ?$
$P(\text{investment} \mid \text{spam}) = ?$
$P(\text{investment} \mid \text{not-spam}) = ?$

Worked-out example: continued

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

$P(F = \text{high} \mid \text{not-spam}) = 2/6$
$P(F = \text{medium} \mid \text{not-spam}) = 2/6$
$P(F = \text{low} \mid \text{not-spam}) = 2/6$
$P(\text{investment} \mid \text{spam}) = 3/4$
$P(\text{investment} \mid \text{not-spam}) = 2/5$

Prediction

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

New point: "CS373" = 1, "investment" = 1, and Familiarity level = high. Is it spam or not-spam?

Prediction

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

New point: "CS373" = 1, "investment" = 1, and Familiarity level = high.
Is it spam or not-spam?
Find the value of S that maximizes
$x = P(CS373 = 1 \mid S) \times P(\text{investment} = 1 \mid S) \times P(F = \text{high} \mid S) \times P(S)$
For S = not-spam, we get x = 2/5 x 2/5 x 2/6 x 3/5 = 24/750

Prediction: Your turn

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

New point: "CS373" = 1, "investment" = 1, and Familiarity level = high.
Is it spam or not-spam?
Find the value of S that maximizes
$x = P(\text{CS373} = 1 \mid S) \times P(\text{investment} = 1 \mid S) \times P(F = \text{high} \mid S) \times P(S)$
For S = not-spam, we get x = 2/5 x 2/5 x 2/6 x 3/5 = 24/750
For S = spam, we get ?

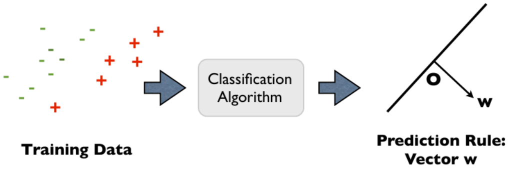
Prediction: Your turn

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

New point: "CS373" = 1, "investment" = 1, and Familiarity level = high.
Is it spam or not-spam?
Find the value of S that maximizes
$x = P(CS373{=}1|S) \times P(investment{=}1|S) \times P(F{=}high|S) \times P(S)$
For S = not-spam, we get x = 2/5 x 2/5 x 2/6 x 3/5 = 24/750
For S = spam, we get $x = 1/4 \times 3/4 \times 2/5 \times 2/5 = 12/400$
The value is higher for not-spam, so we predict that the mail is not spam.

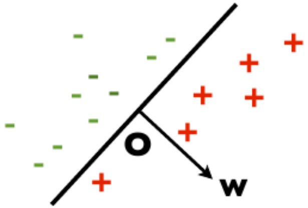
Perceptron: Problem Statement

Want to separate labeled points in D-dimensional space.



Perceptron: The Model

- Linear classifier



Vanilla Algorithm

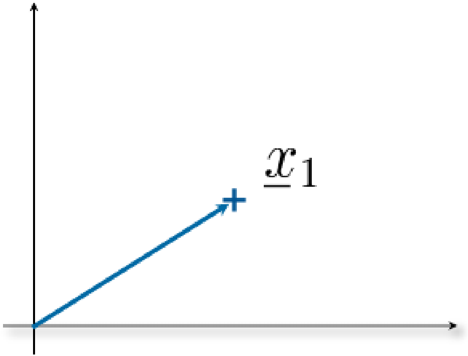
How do we learn a perceptron?

Algorithm 1 Vanilla Perceptron

function TRAIN($\mathbf{D}$, $MaxIter$)
    $w_i \leftarrow 0$, for all $i = 1, \ldots, n$ ▷ Initialize Weights
    $b \leftarrow 0$ ▷ Initialize Bias
    for $iter = 1, \ldots, MaxIter$ **do**
        for all $(x, y) \in \mathbf{D}$ **do**
            $error \leftarrow y - f(x)$
            if $error$ **then**
                $b \leftarrow b + error$ ▷ Update Bias
                $w_i \leftarrow w_i + (error \times x_i)$, for all $i = 1, \ldots, n$ ▷ Update Weights
    return $b, w_0, \ldots, w_n$
function PREDICT($b, w_0, \ldots, w_n, \hat{x}$)
    return f(x)

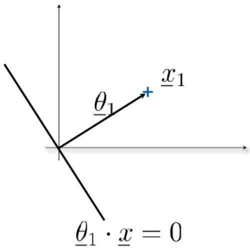
$$f(x) = \begin{cases} 1, & \sum w_j x_j \geq 0 \\ 0, & \sum w_j x_j < 0 \end{cases}$$

Visualize



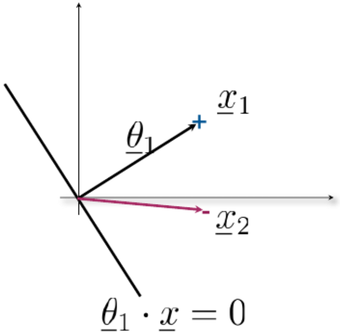
Visualize

$$\underline{\theta}_0 = 0$$
$$\underline{\theta}_1 = \underline{\theta}_0 + 1\,\underline{x}_1$$



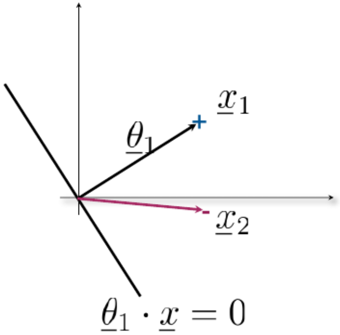
The sign indicates which side of the plane the point is on.
We want all positive instances to fall on one side and the negative instances to fall on the other side.

Visualize



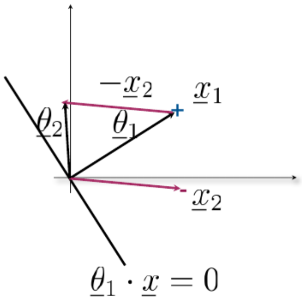
What is the predicted label for x_2?

Visualize



What is the predicted label for x_2?
It's predicted as $+$, but it is actually -. So, our model made an error.

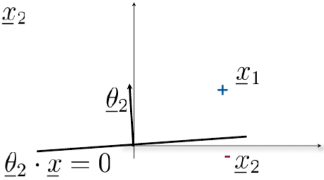
Visualize



Every time it makes an error, it will update as per the algorithm.
This will rotate the plane so that the misclassified point is more likely to fall on the right side.
If there is a bias term, then it will also try to shift the plane.

Visualize

$$\underline{\theta}_0 = 0$$
$$\underline{\theta}_1 = \underline{\theta}_0 + 1\,\underline{x}_1$$
$$\underline{\theta}_2 = \underline{\theta}_1 + (-1)\,\underline{x}_2$$



Worked-out example

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

Worked-out example

"CS373"	"investment"	Familiarity level	Spam
0	1	low	spam
0	0	high	not-spam
0	1	high	spam
0	1	medium	not-spam
1	0	low	not-spam

Binarize the features.
(Note: "CS373" and "investment" should be binarized, but weren't.)

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Vanilla Perceptron Algorithm: Refresher

How do we learn a perceptron?

Algorithm 1 Vanilla Perceptron

function TRAIN($\mathbf{D}$, $MaxIter$)
    $w_i \leftarrow 0$, for all $i = 1, \ldots, n$ ▷ Initialize Weights
    $b \leftarrow 0$ ▷ Initialize Bias
    for $iter = 1, \ldots, MaxIter$ **do**
        for all $(x, y) \in \mathbf{D}$ **do**
            $error \leftarrow y - f(x)$
            if $error$ **then**
                $b \leftarrow b + error$ ▷ Update Bias
                $w_i \leftarrow w_i + (error \times x_i)$, for all $i = 1, \ldots, n$ ▷ Update Weights
    return $b, w_0, \ldots, w_n$
function PREDICT($b, w_0, \ldots, w_n, \hat{x}$)
    return f(x)

$$f(x) = \begin{cases} 1, & \sum w_j x_j \geq 0 \\ 0, & \sum w_j x_j < 0 \end{cases}$$

Worked-out example: Iteration 1

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Initialize: $w = (0, 0, 0, 0, 0)$ and $b = 0$
Iteration 1:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
0	1	yes	0	(0,0,0,0,0)	(0,0,0,0,0)	0

Worked-out example: Iteration 1

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Initialize: $w = (0, 0, 0, 0, 0)$ and $b = 0$
Iteration 1:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
0	1	yes	0	(0,0,0,0,0)	(0,0,0,0,0)	0
0	1	no	-1	(0,0,-1,0,0)	(0,0,-1,0,0)	-1

Worked-out example: Iteration 1

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Initialize: $w = (0, 0, 0, 0, 0)$ and $b = 0$

Iteration 1:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
0	1	yes	0	(0,0,0,0,0)	(0,0,0,0,0)	0
0	1	no	-1	(0,0,-1,0,0)	(0,0,-1,0,0)	-1
-2	0	no	1	(0,1,1,0,0)	(0,1,0,0,0)	0

Worked-out example: Iteration 1

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Initialize: $w = (0, 0, 0, 0, 0)$ and $b = 0$

Iteration 1:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
0	1	yes	0	(0,0,0,0,0)	(0,0,0,0,0)	0
0	1	no	-1	(0,0,-1,0,0)	(0,0,-1,0,0)	-1
-2	0	no	1	(0,1,1,0,0)	(0,1,0,0,0)	0
1	1	no	-1	(0,-1,0,-1,0)	(0,0,0,-1,0)	-1

Worked-out example: Iteration 1

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Initialize: $w = (0, 0, 0, 0, 0)$ and $b = 0$

Iteration 1:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
0	1	yes	0	(0,0,0,0,0)	(0,0,0,0,0)	0
0	1	no	-1	(0,0,-1,0,0)	(0,0,-1,0,0)	-1
-2	0	no	1	(0,1,1,0,0)	(0,1,0,0,0)	0
1	1	no	-1	(0,-1,0,-1,0)	(0,0,0,-1,0)	-1
-1	0	yes	0	(0,0,0,0,0)	(0,0,0,-1,0)	-1

Worked-out example: Prediction after Iteration 1

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

After iteration 1: w = (0,0,0,-1,0) and b = -1
Test case: "CS373" = 1, "investment" = 1, Familiarity level = high.
This is encoded as (1,1,1,0,0)
sign(w.x + b) = sign(0 + -1) = 0
Prediction: Not spam! (Same as NBC.)

Worked-out example: Iteration 2

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Continue from: w = (0,0,0,-1,0) and b = -1
Iteration 2:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b

Worked-out example: Iteration 2

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

Continue from: w = (0,0,0,-1,0) and b = -1

Iteration 2:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
-1	0	no	1	(0,1,0,0,1)	(0,1,0,-1,1)	0
0	1	no	-1	(0,0,-1,0,0)	(0,1,-1,-1,1)	-1
-1	0	no	1	(0,1,1,0,0)	(0,2,0,-1,1)	0
1	1	no	-1	(0,-1,0,-1,0)	(0,1,0,-2,1)	-1
0	1	no	-1	(-1,0,0,0,-1)	(-1,1,0,-2,0)	-2

Worked-out example: Prediction after Iteration 2

"CS373"	"investment"	F_{high}	F_{medium}	F_{low}	Spam
0	1	0	0	1	1
0	0	1	0	0	0
0	1	1	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0

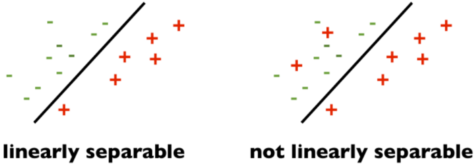
After iteration 2: w = (-1,1,0,-2,0) and b = -2
Predict on the test case: "CS373" = 1, "investment" = 1, and Familiarity level = high.
This is encoded as (1,1,1,0,0)
sign(w.x + b) = sign(0 + -2) = sign(-2) = 0
Prediction: Not spam!

Counter-example

Any guesses?

Counter-example

Any guesses?



- Also, famously, XOR

Properties of Perceptron

- online
- error-driven
- hyper-parameter: MaxIter
- susceptible to noise?
- discriminative

Averaged Perceptron: How?

- Averaged perceptron is just $(\sum_k w_k \ / \ nT)$.

Efficient Weighted Perceptron

```
D = {xi,yi} i=1,...,n
w = (0,...,0)        current function weights
a = (0,...,0)        counter of all the updates seen so far
step= nT
repeat T times
    for (xi,yi) in D
        y' = sign(wx)        prediction based
                             on current model
    if (y'!=y)
        w=w +xyr    update Rule
        a = a+(step/nT)(xyr)    update the weight
    step = step - 1             counter
return a    return the averaged result
```

Worked-out example: Averaged Perceptron

Lazy method: Just take the average of the w's that were updated after an error.

Iterations 1 and 2:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
0	1	yes	0	(0,0,0,0,0)	(0,0,0,0,0)	0
0	1	no	-1	(0,0,-1,0,0)	(0,0,-1,0,0)	-1
-2	0	no	1	(0,1,1,0,0)	(0,1,0,0,0)	0
1	1	no	-1	(0,-1,0,-1,0)	(0,0,0,-1,0)	-1
-1	0	yes	0	(0,0,0,0,0)	(0,0,0,-1,0)	-1

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
-1	0	no	1	(0,1,0,0,1)	(0,1,0,-1,1)	0
0	1	no	-1	(0,0,-1,0,0)	(0,1,-1,-1,1)	-1
-1	0	no	1	(0,1,1,0,0)	(0,2,0,-1,1)	0
1	1	no	-1	(0,-1,0,-1,0)	(0,1,0,-2,1)	-1
0	1	no	-1	(-1,0,0,0,-1)	(-1,1,0,-2,0)	-2

Worked-out example: Averaged Perceptron

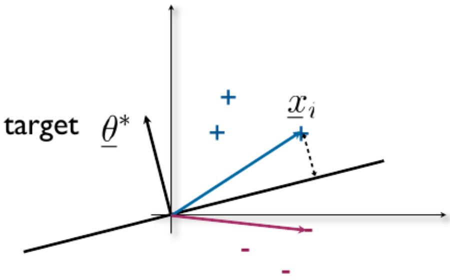
Iterations 1 and 2:

$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
0	1	yes	0	(0,0,0,0,0)	(0,0,0,0,0)	0
0	1	no	-1	(0,0,-1,0,0)	(0,0,-1,0,0)	-1
-2	0	no	1	(0,1,1,0,0)	(0,1,0,0,0)	0
1	1	no	-1	(0,-1,0,-1,0)	(0,0,0,-1,0)	-1
-1	0	yes	0	(0,0,0,0,0)	(0,0,0,-1,0)	-1

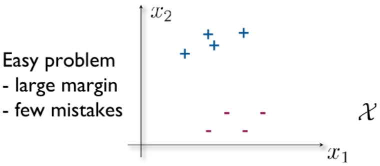
$w.x_i + b$	sign(.)	correct?	error	error $\times$ x_i	w'	b
-1	0	no	1	(0,1,0,0,1)	(0,1,0,-1,1)	0
0	1	no	-1	(0,0,-1,0,0)	(0,1,-1,-1,1)	-1
-1	0	no	1	(0,1,1,0,0)	(0,2,0,-1,1)	0
1	1	no	-1	(0,-1,0,-1,0)	(0,1,0,-2,1)	-1
0	1	no	-1	(-1,0,0,0,-1)	(-1,1,0,-2,0)	-2

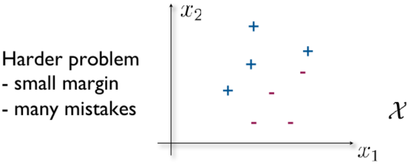
w_{avg} = (-1/8, 7/8, -2/8, -8/8, 4/8); b_{avg} = -6/8

Margin



Margin and Number of Mistakes





Zero-one Loss

$$Loss_{0/1} = \frac{1}{n} \sum_{i \in n} \left\{ \begin{array}{cc} 0 & \text{if } y(i) = \hat{y}_i \\ 1 & \text{otherwise} \end{array} \right\}$$

Squared Loss

$$Loss_sq(T) = \frac{1}{n}\sum_{i \in n}(1 - p_i)^2$$

Use logarithm

$$
\begin{aligned}
l(\theta|D) &= log\, L(\theta|D) \\
&= log \prod_{i=1}^{n} p(x(i)|\theta) \\
&= \sum_{i=1}^{n} logp(x(i)|\theta)
\end{aligned}
$$

Bias-variance tradeoff