Machine Learning (CSE 446): Probabilistic Machine Learning

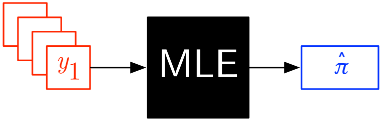
Noah Smith

© 2017

University of Washington
nasmith@cs.washington.edu

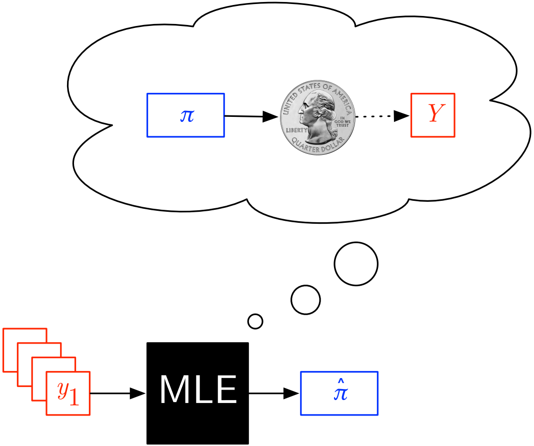
November 1, 2017

Understanding MLE

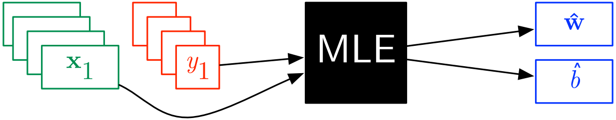


You can think of MLE as a “black box” for choosing parameter values.

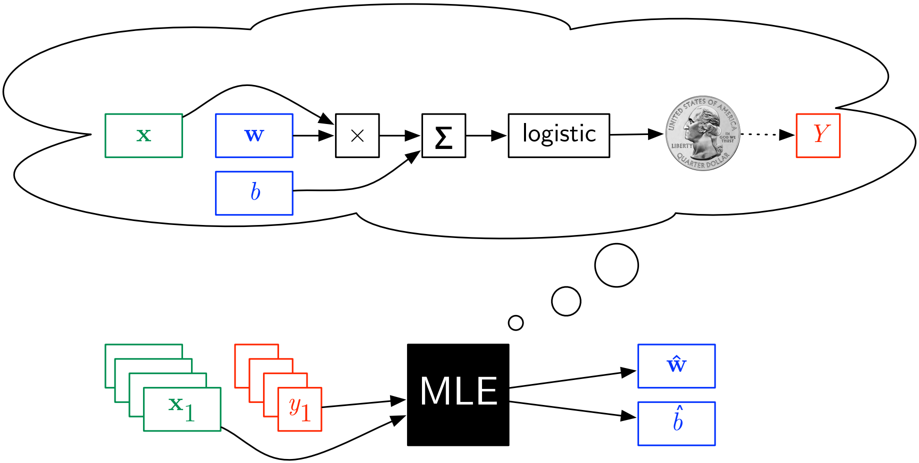
Understanding MLE



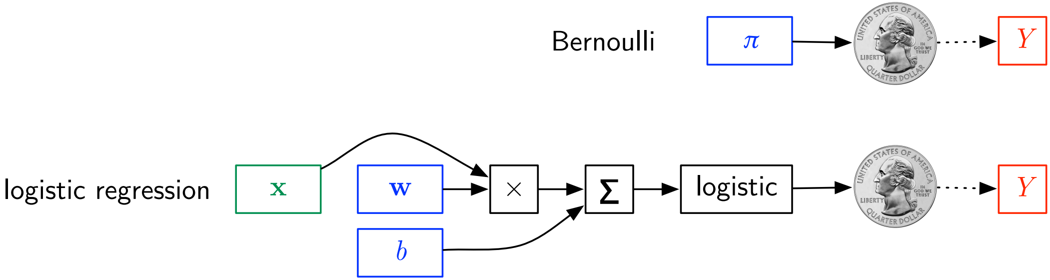
Understanding MLE



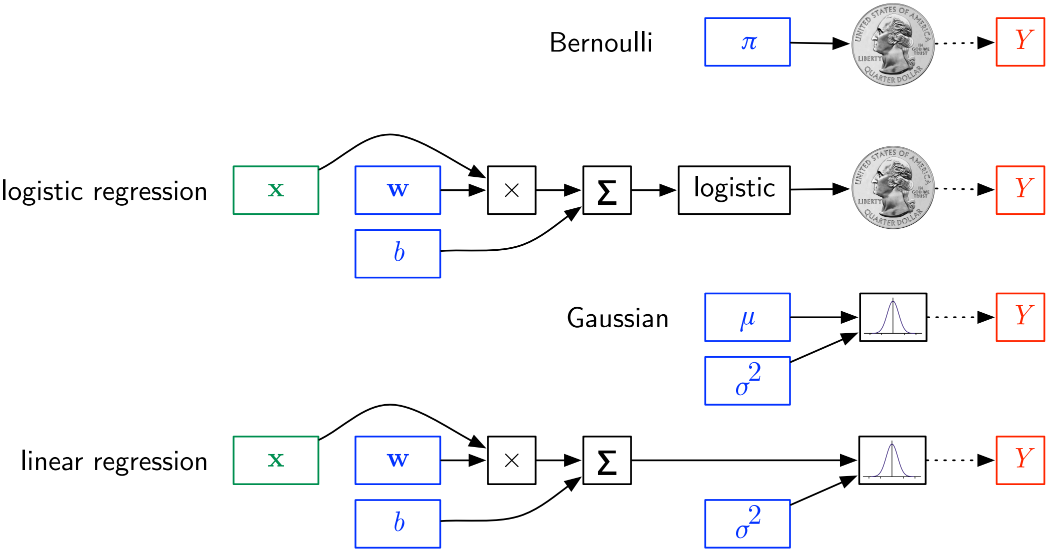
Understanding MLE



Probabilistic Stories



Probabilistic Stories



Then and Now

Before today, you knew how to do MLE:

- For a Bernoulli distribution: $\hat{\pi} = \frac{\text{count}(+1)}{\text{count}(+1)+\text{count}(-1)} = \frac{N^+}{N}$
- For a Gaussian distribution: $\hat{\mu} = \frac{\sum_{n=1}^{N} y_n}{N}$ (and similar for estimating variance, $\hat{\sigma}^2$).

Logistic regression and linear regression, respectively, generalize these so that the parameter is itself a function of $\mathbf{x}$, so that we have a **conditional model** of Y given X.

- The practical difference is that the MLE doesn't have a closed form for these models.
  (So we use SGD and friends.)

A Twist!

There *is* a closed form for the MLE of linear regression.

To keep it simple, assume $b = 0$.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the stack of training inputs and $\mathbf{y} \in \mathbb{R}^{N}$ be the stack of training outputs.

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2$$

A Twist!

There *is* a closed form for the MLE of linear regression.

To keep it simple, assume $b = 0$.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the stack of training inputs and $\mathbf{y} \in \mathbb{R}^{N}$ be the stack of training outputs.

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 \equiv \underset{\mathbf{w}}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$

A Twist!

There *is* a closed form for the MLE of linear regression.

To keep it simple, assume $b = 0$.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the stack of training inputs and $\mathbf{y} \in \mathbb{R}^{N}$ be the stack of training outputs.

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 \equiv \operatorname*{argmin}_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\overbrace{-2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w})}^{\text{gradient w.r.t. } \mathbf{w}} = \mathbf{0}$$

A Twist!

There *is* a closed form for the MLE of linear regression.

To keep it simple, assume $b = 0$.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the stack of training inputs and $\mathbf{y} \in \mathbb{R}^{N}$ be the stack of training outputs.

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 \equiv \operatorname*{argmin}_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\overbrace{-2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w})}^{\text{gradient w.r.t. } \mathbf{w}} = \mathbf{0}$$

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$$

Invertibility is fine if we have more than d linearly independent observations.

A Twist!

There *is* a closed form for the MLE of linear regression.

To keep it simple, assume $b = 0$.

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the stack of training inputs and $\mathbf{y} \in \mathbb{R}^{N}$ be the stack of training outputs.

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 \equiv \operatorname*{argmin}_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\overbrace{-2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w})}^{\text{gradient w.r.t. } \mathbf{w}} = \mathbf{0}$$

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$$

Invertibility is fine if we have more than d linearly independent observations. But it costs $O(d^3)$.

MLE is Dangerous

$$\text{Variance}(\hat{\pi}) = \frac{\pi(1-\pi)}{N} \qquad \text{(Note that } \pi \text{ is the } \textit{true} \text{ probability that } Y = 1\text{!)}$$

$$\text{Variance}(\hat{\mu}) = \frac{\sigma^2}{N} \qquad \text{(Note that } \sigma^2 \text{ is the } \textit{true} \text{ variance of the r.v.!)}$$

MLE is Dangerous

$$\text{Variance}(\hat{\pi}) = \frac{\pi(1-\pi)}{N} \qquad \text{(Note that } \pi \text{ is the } \textit{true} \text{ probability that } Y = 1\text{!)}$$

$$\text{Variance}(\hat{\mu}) = \frac{\sigma^2}{N} \qquad \text{(Note that } \sigma^2 \text{ is the } \textit{true} \text{ variance of the r.v.!)}$$

Recall the bias-variance tradeoff.

- Bias/approximation error: if your choice of features and probabilistic model align to reality, MLE is great.
- Variance/estimation error: MLE tends to overfit unless you have a lot of data.

MLE is Dangerous

$$\text{Variance}(\hat{\pi}) = \frac{\pi(1-\pi)}{N} \qquad \text{(Note that } \pi \text{ is the } \textit{true} \text{ probability that } Y = 1\text{!)}$$

$$\text{Variance}(\hat{\mu}) = \frac{\sigma^2}{N} \qquad \text{(Note that } \sigma^2 \text{ is the } \textit{true} \text{ variance of the r.v.!)}$$

Regularization reduces variance but increases bias.

Adding Regularization to the Probabilistic Story

Probabilistic story:

- For $n \in \{1, \dots, N\}$:
  - Observe $\mathbf{x}_n$.
  - Transform it using parameters $\mathbf{w}$ and b to get $p_{\mathbf{w},b}(Y \mid \mathbf{x}_n)$.
  - Sample $y_n \sim p_{\mathbf{w},b}(Y \mid \mathbf{x}_n)$.

Adding Regularization to the Probabilistic Story

Probabilistic story:

- For $n \in \{1, \ldots, N\}$:
  - Observe $\mathbf{x}_n$.
  - Transform it using parameters $\mathbf{w}$ and b to get $p_{\mathbf{w},b}(Y \mid \mathbf{x}_n)$.
  - Sample $y_n \sim p_{\mathbf{w},b}(Y \mid \mathbf{x}_n)$.

Probabilistic story with regularization:

- Use hyperparameters α to define a **prior** distribution over random variables $\boldsymbol{W}$, $p_\alpha(\boldsymbol{W})$.
- Sample $\mathbf{w} \sim p_\alpha(\boldsymbol{W})$.
- For $n \in \{1, \ldots, N\}$:
  - Observe $\mathbf{x}_n$.
  - Transform it using parameters $\mathbf{w}$ and b to get $p_{\mathbf{w},b}(Y \mid \mathbf{x}_n)$.
  - Sample $y_n \sim p_{\mathbf{w},b}(Y \mid \mathbf{x}_n)$.

Maximum a Posteriori (MAP) Estimation

$$(\hat{\mathbf{w}}, b) = \underset{\mathbf{w},b}{\operatorname{argmax}} \underbrace{\log p_\alpha(\mathbf{w})}_{\text{log prior}} + \underbrace{\sum_{n=1}^{N} \log p_{\mathbf{w},b}(y_n \mid \mathbf{x}_n)}_{\text{log likelihood}}$$

Maximum a Posteriori (MAP) Estimation

$$(\hat{\mathbf{w}}, b) = \operatorname*{argmax}_{\mathbf{w}, b} \underbrace{\log p_\alpha(\mathbf{w})}_{\text{log prior}} + \underbrace{\sum_{n=1}^{N} \log p_{\mathbf{w},b}(y_n \mid \mathbf{x}_n)}_{\text{log likelihood}}$$

Typical assumption is that each weight is independent of the others.

$$p_\alpha(\boldsymbol{W}) = \prod_j p_\alpha(W_j)$$

Maximum a Posteriori (MAP) Estimation

$$(\hat{\mathbf{w}}, b) = \operatorname*{argmax}_{\mathbf{w},b} \underbrace{\log p_\alpha(\mathbf{w})}_{\text{log prior}} + \underbrace{\sum_{n=1}^{N} \log p_{\mathbf{w},b}(y_n \mid \mathbf{x}_n)}_{\text{log likelihood}}$$

Typical assumption is that each weight is independent of the others.

$$p_\alpha(\boldsymbol{W}) = \prod_j p_\alpha(W_j)$$

Option 1: let $p_\alpha(W_j)$ be a zero-mean Gaussian distribution with standard deviation α.

$$\log p_\alpha(\mathbf{w}) = -\frac{1}{2\alpha^2}\|\mathbf{w}\|_2^2 + \text{constant}$$

Maximum a Posteriori (MAP) Estimation

$$(\hat{\mathbf{w}}, b) = \operatorname*{argmax}_{\mathbf{w},b} \underbrace{\log p_\alpha(\mathbf{w})}_{\text{log prior}} + \underbrace{\sum_{n=1}^{N} \log p_{\mathbf{w},b}(y_n \mid \mathbf{x}_n)}_{\text{log likelihood}}$$

Typical assumption is that each weight is independent of the others.

$$p_\alpha(\boldsymbol{W}) = \prod_j p_\alpha(W_j)$$

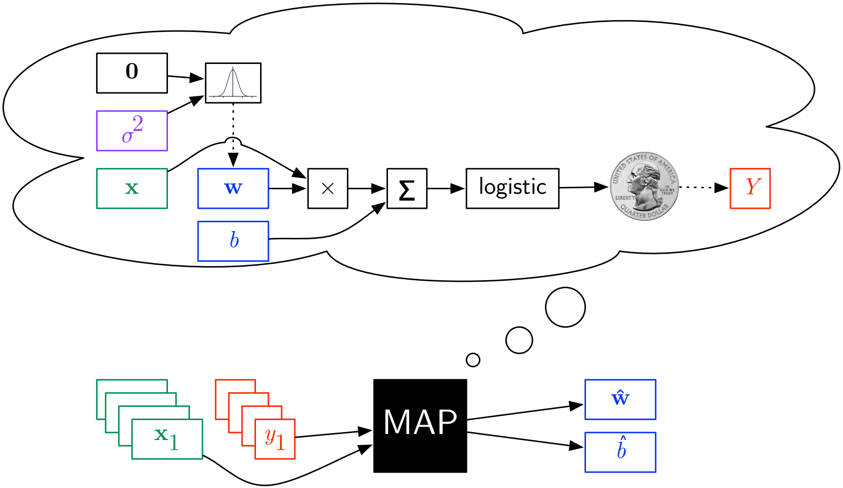
Option 1: let $p_\alpha(W_j)$ be a zero-mean Gaussian distribution with standard deviation α.

$$\log p_\alpha(\mathbf{w}) = -\frac{1}{2\alpha^2}\|\mathbf{w}\|_2^2 + \text{constant}$$

Option 2: let $p_\alpha(W_j)$ be a zero-location Laplace distribution with scale α.

$$\log p_\alpha(\mathbf{w}) = -\frac{1}{\alpha}\|\mathbf{w}\|_1 + \text{constant}$$

Probabilistic Story: L_2-Regularized Logistic Regression



Why Go Probabilistic?

- Interpret the classifier's activation function as a (log) probability (density), which encodes uncertainty.
- Interpret the regularizer as a (log) probability (density), which encodes uncertainty.
- Leverage theory from statistics to get a better understanding of the guarantees we can hope for with our learning algorithms.
- Change your assumptions, turn the optimization-crank, and get a new machine learning method.

The key to success is to tell a probabilistic story that's reasonably close to reality, including the prior(s).