Parameter Estimation in Probabilistic Models, Linear Regression and Logistic Regression

Piyush Rai

CS5350/6350: Machine Learning

September 20, 2011
Assume data generated via a probabilistic model

\[ \mathbf{d} \sim P(\mathbf{d} \mid \theta) \]

- \( P(\mathbf{d} \mid \theta) \): Probability distribution underlying the data
  - \( \theta \): fixed but unknown distribution parameter

- **Given**: \( N \) independent and identically distributed (i.i.d.) samples of the data
  \[ \mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_N\} \]

- Independent and Identically Distributed:
  - Given \( \theta \), each sample \( \mathbf{d}_n \) is independent of all other samples
  - All samples \( \mathbf{d}_n \) drawn from the same distribution

- **Goal**: Estimate parameter \( \theta \) that best models/describes the data

- Several ways to define the “best”
Maximum Likelihood Estimation (MLE)

- **Maximum Likelihood Estimation (MLE):** Choose the parameter $\theta$ that maximizes the probability of the data, *given* that parameter.

- Probability of the data, given the parameters is called the **Likelihood,** a function of $\theta$ and defined as:

  $$
  \mathcal{L}(\theta) = P(\mathcal{D} \mid \theta) = P(d_1, \ldots, d_N \mid \theta) = \prod_{n=1}^{N} P(d_n \mid \theta)
  $$

- MLE typically maximizes the **Log-likelihood** instead of the likelihood.

- Log-likelihood:

  $$
  \log \mathcal{L}(\theta) = \log P(\mathcal{D} \mid \theta) = \log \prod_{n=1}^{N} P(d_n \mid \theta) = \sum_{n=1}^{N} \log P(d_n \mid \theta)
  $$

- Maximum Likelihood parameter estimation

  $$
  \hat{\theta}_{MLE} = \arg \max_{\theta} \log \mathcal{L}(\theta) = \arg \max_{\theta} \sum_{n=1}^{N} \log P(d_n \mid \theta)
  $$
Maximum-a-Posteriori Estimation (MAP): Choose $\theta$ that maximizes the posterior probability of $\theta$ (i.e., probability in the light of the observed data).

Posterior probability of $\theta$ is given by the Bayes Rule:

$$P(\theta \mid D) = \frac{P(\theta)P(D \mid \theta)}{P(D)}$$

- $P(\theta)$: Prior probability of $\theta$ (without having seen any data)
- $P(D \mid \theta)$: Likelihood
- $P(D)$: Probability of the data (independent of $\theta$)

$$P(D) = \int P(\theta)P(D \mid \theta)d\theta \quad \text{(sum over all $\theta$’s)}$$

The Bayes Rule lets us update our belief about $\theta$ in the light of observed data.

While doing MAP, we usually maximize the log of the posterior probability.
Maximum-a-Posteriori Estimation (MAP)

- Maximum-a-Posteriori parameter estimation

\[ \hat{\theta}_{MAP} = \arg \max_{\theta} P(\theta \mid D) = \arg \max_{\theta} \frac{P(\theta)P(D \mid \theta)}{P(D)} = \arg \max_{\theta} P(\theta)P(D \mid \theta) = \arg \max_{\theta} \log P(\theta)P(D \mid \theta) = \arg \max_{\theta} \{\log P(\theta) + \log P(D \mid \theta)\} \]

\[ \hat{\theta}_{MAP} = \arg \max_{\theta} \{\log P(\theta) + \sum_{n=1}^{N} \log P(d_n \mid \theta)\} \]

- Same as MLE except the extra log-prior-distribution term!

- MAP allows incorporating our prior knowledge about \( \theta \) in its estimation
Linear Regression: The Probabilistic Formulation

- Each response generated by a linear model plus some Gaussian noise

\[ y = \mathbf{w}^\top \mathbf{x} + \epsilon \]

- Noise \( \epsilon \) is drawn from a Gaussian distribution:

\[ \epsilon \sim \mathcal{N}(0, \sigma^2) \]

- Each response \( y \) then becomes a draw from the following Gaussian:

\[ y \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2) \]

- Probability of each response variable

\[
P(y \mid \mathbf{x}, \mathbf{w}) = \mathcal{N}(y \mid \mathbf{w}^\top \mathbf{x}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y - \mathbf{w}^\top \mathbf{x})^2}{2\sigma^2} \right]
\]

- Given data \( \mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\} \), we want to estimate the weight vector \( \mathbf{w} \)
Linear Regression: Maximum Likelihood Solution

- **Log-likelihood:**
  \[
  \log L(w) = \log P(D \mid w) = \log P(Y \mid X, w) = \log \prod_{n=1}^{N} P(y_n \mid x_n, w)
  \]
  \[
  = \sum_{n=1}^{N} \log P(y_n \mid x_n, w)
  \]
  \[
  = \sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y_n - w^\top x_n)^2}{2\sigma^2} \right]
  \]
  \[
  = \sum_{n=1}^{N} \left\{ -\frac{1}{2} \log(2\pi\sigma^2) - \frac{(y_n - w^\top x_n)^2}{2\sigma^2} \right\}
  \]

- **Maximum Likelihood Solution:**
  \[
  \hat{w}_{MLE} = \arg \max_w \log P(D \mid w)
  \]
  \[
  = \arg \max_w -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - w^\top x_n)^2
  \]
  \[
  = \arg \min_w \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - w^\top x_n)^2
  \]

- For \(\sigma = 1\) (or some constant) for each input, it’s equivalent to the least-squares objective for linear regression
Linear Regression: Maximum-a-Posteriori Solution

Let's assume a **Gaussian prior distribution** over the weight vector \( \mathbf{w} \)

\[
P(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid 0, \lambda^{-1} \mathbf{I}) = \frac{1}{(2\pi)^{D/2}} \exp \left( -\frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \right)
\]

**Log posterior probability:**

\[
\log P(\mathbf{w} \mid \mathcal{D}) = \log \frac{P(\mathbf{w})P(\mathcal{D} \mid \mathbf{w})}{P(\mathcal{D})} = \log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w}) - \log P(\mathcal{D})
\]

**Maximum-a-Posteriori Solution:** \( \hat{\mathbf{w}}_{MAP} = \arg \max_{\mathbf{w}} \log P(\mathbf{w} \mid \mathcal{D}) \)

\[
= \arg \max_{\mathbf{w}} \{ \log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w}) - \log P(\mathcal{D}) \}
\]

\[
= \arg \max_{\mathbf{w}} \{ \log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w}) \}
\]

\[
= \arg \max_{\mathbf{w}} \left\{ -\frac{D}{2} \log(2\pi) - \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} + \sum_{n=1}^{N} \left\{ -\frac{1}{2} \log(2\pi\sigma^2) - \frac{(y_n - \mathbf{w}^\top \mathbf{x}_n)^2}{2\sigma^2} \right\} \right\}
\]

\[
= \arg \min_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2 + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \quad \text{(ignoring constants and changing max to min)}
\]

For \( \sigma = 1 \) (or some constant) for each input, it's equivalent to the regularized least-squares objective
Linear Regression: MLE vs MAP (summary)

- **MLE solution:**
  \[
  \hat{w}_{MLE} = \arg \min_w \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - w^T x_n)^2
  \]

- **MAP solution:**
  \[
  \hat{w}_{MAP} = \arg \min_w \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - w^T x_n)^2 + \frac{\lambda}{2} w^T w
  \]

- **Take-home messages:**
  - **MLE estimation** of a parameter leads to **unregularized solutions**
  - **MAP estimation** of a parameter leads to **regularized solutions**
  - The prior distribution acts as a regularizer in MAP estimation

- **Note:** For MAP, different prior distributions lead to different regularizers
  - Gaussian prior on \( w \) regularizes the \( \ell_2 \) norm of \( w \)
  - Laplace prior \( \exp(-C||w||_1) \) on \( w \) regularizes the \( \ell_1 \) norm of \( w \)
Often we don’t just care about predicting the label $y$ for an example.

Rather, we want to predict the label probabilities $P(y \mid x, w)$

- E.g., $P(y = +1 \mid x, w)$: the probability that the label is $+1$
- In a sense, it’s our confidence in the predicted label

Probabilistic classification models allow us do that.

Consider the following function ($y = -1/1$):

$$P(y \mid x, w) = \sigma(yw^\top x) = \frac{1}{1 + \exp(-yw^\top x)}$$

$\sigma$ is the logistic function which maps all real number into (0,1)

This is the Logistic Regression model.

**Misnomer:** Logistic Regression is a classification model :-)

---

**Logistic Function:**

$$\frac{1}{1 + \exp(-z)}$$

Graph showing $\frac{1}{1 + \exp(-z)}$ for $z$ ranging from -4 to 4.
What does the decision boundary look like for Logistic Regression?

At the decision boundary labels $+1/-1$ becomes equiprobable

$$P(y = +1 \mid x, w) = P(y = -1 \mid x, w)$$

$$\frac{1}{1 + \exp(-w^\top x)} = \frac{1}{1 + \exp(w^\top x)}$$

$$\exp(-w^\top x) = \exp(w^\top x)$$

$$w^\top x = 0$$

The decision boundary is therefore linear $\Rightarrow$ Logistic Regression is a linear classifier (note: it’s possible to kernelize and make it nonlinear)
Logistic Regression: Maximum Likelihood Solution

- **Goal:** Want to estimate $\mathbf{w}$ from the data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_n)\}$
- **Log-likelihood:**
  \[
  \log \mathcal{L}(\mathbf{w}) = \log P(\mathcal{D} | \mathbf{w}) = \log P(\mathbf{Y} | \mathbf{X}, \mathbf{w}) = \log \prod_{n=1}^{N} P(y_n | \mathbf{x}_n, \mathbf{w})
  \]
  \[
  = \sum_{n=1}^{N} \log P(y_n | \mathbf{x}_n, \mathbf{w})
  \]
  \[
  = \sum_{n=1}^{N} \log \frac{1}{1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)}
  \]
  \[
  = \sum_{n=1}^{N} - \log[1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)]
  \]

- **Maximum Likelihood Solution:** $\hat{\mathbf{w}}_{\text{MLE}} = \arg \min_{\mathbf{w}} \log \mathcal{L}(\mathbf{w})$
- **No closed-form solution exists but we can do gradient descent on $\mathbf{w}$**

\[
\nabla_{\mathbf{w}} \log \mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N} \frac{1}{1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)} \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)(-y_n \mathbf{x}_n)
\]

\[
= \sum_{n=1}^{N} \frac{1}{1 + \exp(y_n \mathbf{w}^\top \mathbf{x}_n)} y_n \mathbf{x}_n
\]
Logistic Regression: Maximum-a-Posteriori Solution

- Let’s assume a Gaussian prior distribution over the weight vector \( \mathbf{w} \)

\[
P(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid 0, \lambda^{-1} \mathbf{I}) = \frac{1}{(2\pi)^{D/2}} \exp \left( -\frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \right)
\]

- Maximum-a-Posteriori Solution: \( \hat{\mathbf{w}}_{MAP} = \arg \max_{\mathbf{w}} \log P(\mathbf{w} \mid \mathcal{D}) \)

\[
\begin{align*}
&= \arg \max_{\mathbf{w}} \{ \log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w}) - \log P(\mathcal{D}) \} \\
&= \arg \max_{\mathbf{w}} \{ \log P(\mathbf{w}) + \log P(\mathcal{D} \mid \mathbf{w}) \} \\
&= \arg \max_{\mathbf{w}} \left\{ -\frac{D}{2} \log(2\pi) - \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} + \sum_{n=1}^{N} - \log[1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)] \right\} \\
&= \arg \min_{\mathbf{w}} \sum_{n=1}^{N} \log[1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)] + \frac{\lambda}{2} \mathbf{w}^\top \mathbf{w} \quad \text{(ignoring constants and changing max to min)}
\end{align*}
\]

- No closed-form solution exists but we can do gradient descent on \( \mathbf{w} \)

- See “A comparison of numerical optimizers for logistic regression” by Tom Minka on optimization techniques (gradient descent and others) for logistic regression (both MLE and MAP)
Logistic Regression: MLE vs MAP (summary)

- **MLE solution:**
  \[
  \hat{w}_{MLE} = \arg \min_w \sum_{n=1}^N \log[1 + \exp(-y_n w^\top x_n)]
  \]

- **MAP solution:**
  \[
  \hat{w}_{MAP} = \arg \min_w \sum_{n=1}^N \log[1 + \exp(-y_n w^\top x_n)] + \frac{\lambda}{2} w^\top w
  \]

- **Take-home messages** (we already saw these before :-) ):
  - MLE estimation of a parameter leads to unregularized solutions
  - MAP estimation of a parameter leads to regularized solutions
  - The prior distribution acts as a regularizer in MAP estimation

- Note: For MAP, different prior distributions lead to different regularizers
  - Gaussian prior on \( w \) regularizes the \( \ell_2 \) norm of \( w \)
  - Laplace prior \( \exp(-C||w||_1) \) on \( w \) regularizes the \( \ell_1 \) norm of \( w \)
Logistic Regression: some notes

- The objective function is very similar to the SVM
  - .. except for the loss function part
  - Logistic regression uses the log-loss, SVM uses the hinge-loss

- Generalization to more than 2 classes is straightforward
  - .. using the soft-max function instead of the logistic function

\[
P(y = k \mid x, w) = \frac{\exp(w_k^\top x)}{\sum_k \exp(w_k^\top x)}
\]

- We maintain a separator weight vector \( w_k \) for each class \( k \)

- Possible to kernelize it to learn nonlinear boundaries
The MAP estimate:

$$\hat{w}_{MAP} = \arg \max_w \log P(w | D)$$

$$= \arg \max_w \{ \log P(D|w) + \log P(w) \}$$

$$= \arg \min_w \{ - \log P(D|w) - \log P(w) \}$$

Recall the regularized loss function minimization:

$$\hat{w} = \arg \min_w \{ L(Y, X, w) + R(w) \}$$

Negative log likelihood $- \log P(D|w)$ corresponds to the loss $L(Y, X, w)$

Negative log prior $- \log P(w)$ corresponds to the regularizer $R(w)$