Lecture 5b: Probabilistic Perspectives on ML Algorithms

Applied Machine Learning

Volodymyr Kuleshov
Cornell Tech

Part 1: Probabilistic Linear Regression

Previously, we derived *maximum likelihood learning* as a general way of learning machine models.

We will now seehow the algorithms we've seen so far are special cases of this principle.

Review: Probabilistic Models

A probabilistic model is a probability distribution

$$P(x, y) : \mathcal{X} \times \mathcal{Y} \to [0, 1].$$

This model can approximate the data distribution $P_{\text{data}}(x, y)$.

If we know $P(x, y)$, we can use the conditional $P(y|x)$ for prediction.

Probabilistic models may also have *parameters* $\theta \in \Theta$, which we denote as

$$P_\theta(x, y) : \mathcal{X} \times \mathcal{Y} \to [0, 1].$$

Review: Conditional Maximum Likelihood

A general approach of optimizing conditional models of the form $P_\theta(y|x)$ is by minimizing expected KL divergence with respect to the data distribution:

$$\min_\theta \mathbb{E}_{x\sim\mathbb{P}_{\text{data}}} \left[D(P_{\text{data}}(y|x) \mid\mid P_\theta(y|x))\right].$$

With a bit of math, we can show that the maximum likelihood objective becomes

$$\max_\theta \mathbb{E}_{x,y\sim\mathbb{P}_{\text{data}}} \log P_\theta(y|x).$$

This is the principle of *conditional maximum likelihood*.

Review: Least Squares

Recall that the linear regression algorithm fits a linear model of the form

$$f(x) = \sum_{j=0}^{d} \theta_j \cdot x_j = \theta^\top x.$$

It minimizes the mean squared error (MSE)

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (y^{(i)} - \theta^\top x^{(i)})^2$$

on a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\}$.

Is there a specific reason for us to be optimizing the mean squared error to fit our linear model?

The answer to this can be found by looking at the algorithm from a probabilistic perspective.

Probabilistic Least Squares

Let's derive a probabilistic algorithm by defining a class of probabilistic models and use maximum likelihood as our objective.

1. Let's choose our model family $\mathcal{M}$ to be the set of Gaussian distributions of the form
$$p(y|x;\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y - \theta^\top x)^2}{2\sigma^2} \right).$$
Each model $\mathcal{N}(y; \mu(x), \sigma)$ is a Gaussian with a standard deviation σ of one and a mean of $\mu(x) = \theta^\top x$ that is parametrized by the parameters θ.

1. We optimize the model using maximum likelihood. The log-likelihood function at a point (x, y) equals

$$\begin{aligned}\log L(\theta) = \log p(y|x;\theta) &= \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\theta^\top x)^2}{2\sigma^2}\right) \\ &= -\frac{(y-\theta^\top x)^2}{2\sigma^2} + \text{const.}\end{aligned}$$

Note how this is a mean squared error (MSE) objective!

Thus, minimizing MSE is equivalent to maximizing the log-likelihood of a Normal distribution $\mathcal{N}(y; \mu(x), \sigma)$.

Algorithm: Gaussian Ordinary Least Squares

- **Type**: Supervised learning (regression)
- **Model family**: Linear models
- **Objective function**: Mean squared error
- **Optimizer**: Normal equations
- **Probabilistic interpretation**: Conditional Gaussian fit using max-likelihood.

Extensions of Gaussian Least Squares

This is an example of how we can interpret a machine learning algorithm in a probabilistic framework.

We will see many algorithms that have these kinds of interpretations. Here are some simple extensions.

We can use a Gaussian model and also parametrize the standard deviation.

- This is called heteroscedastic regression, and allows us to obtain confidence intevals for our predictions.

We can can also parametrize other distributions, not just the Gaussian.

- Exponential or Gamma distributions for continuous variables
- Bernoulli distribution for discrete variables

This yields many new machine learning algorithms.

Part 2: Bayesian Algorithms

We can also use what we learned about Bayesian ML do interpret several algrothims that we've seen as special cases of the Bayesian framework.

Review: The Bayesian Approach

In Bayesian statistics, θ is a *random* variable whose value happens to be unknown.

We formulate two models:

- A *likelihood* model $P(x, y|\theta)$ that defines the probability of x, y for any fixed value of θ.
- A *prior* $P(\theta)$ that specifies us existing belief about the distribution of the random variable θ.

Together, these two models define the *joint* distribution

$$P(x, y, \theta) = P(x, y \mid \theta)P(\theta)$$

in which both the x, y and the parameters θ are random variables.

Review: A Posteriori Learning

Recall that in maximum a posteriori (MAP) learning, we optimize the following objective.

$$\theta_{\text{MAP}} = \arg\max_{\theta} \left(\log \prod_{i=1}^{n} P(x^{(i)}, y^{(i)} \mid \theta) + \log P(\theta) \right),$$

Note that we used the same formula as we used for maximum likelihood, except that we have added the prior term $\log P(\theta)$.

Review: Ridge Regression

Recall that the ridge regression algorithm fits a linear model

$$f(x) = \sum_{j=0}^{d} \theta_j \cdot x_j = \theta^\top x.$$

We minimize the L2-regualrized mean squared error (MSE)

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(y^{(i)} - \theta^{\top}x^{(i)})^2 + \frac{\lambda}{2}\sum_{j=1}^{d}\theta_j^2$$

on a dataset $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\}$. The term $\frac{1}{2}\sum_{j=1}^{d}\theta_j^2 = \frac{1}{2}||\theta||_2^2$ is called the regularizer.

Probabilistic Ridge Regession

We can interpet ridge regression as maximum apriori (MAP) estimation as follows.

1. First, we select our model family $\mathcal{M}$ to be the set of Gaussian distributions of the form (let's assume $x \in \mathbb{R}$ for simplicity).

$$p(y|x;\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\theta^\top x)^2}{2\sigma^2}\right).$$

1. We assume a Gaussian prior with mean zero and variance τ on the parameters θ:

$$p(\theta) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{\theta_j^2}{2\tau^2}\right).$$

1. We optimize the model using the MAP approach. The objective at a point (x, y) equals

$$\begin{aligned}\log L(\theta) &= \log p(y|x;\theta) + \log p(\theta) \\ &= \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\theta^\top x)^2}{2\sigma^2}\right) \\ &\qquad\qquad + \log \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{\theta_j^2}{2\tau^2}\right) \\ &= -\frac{(y-\theta^\top x)^2}{2\sigma^2} - \frac{1}{2\tau^2} \sum_{j=1}^{d} \theta_j^2 + \text{const.}\end{aligned}$$

Thus, we see that ridge regression actually amounts to performing MAP estimation with a Gaussian prior. The strength of the regularizer λ equals $1/\tau^2$.

Algorithm: Probabilistic Ridge Least Squares

- **Type**: Supervised learning (regression)
- **Model family**: Linear models
- **Objective function**: L2-regularized mean squared error
- **Optimizer**: Normal equations
- **Probabilistic interpretation**: Conditional Gaussian likelihood and Gaussian prior fit using MAP.

Bayesian View on ML Algorithms

Very often, we can interpret classical ML algorithms as applications of the probabilistic or Bayesian approaches (although we can derive them in other ways as well!)

- Regularization can often be seen as applying a prior on the weights.
- L1 regularization can be seen as applying a *Laplace* prior.
- Many other algorithms will have similar interpretations.