Introduction to Machine Learning
Summary

Philip Hartout

February 29, 2020

1 Linear Regression

Objective, approximate:

$$\begin{aligned} f(x) &= w_1 x_1 + \ldots + w_d x_d + w_0 \\ &= \sum_{i=1}^{d} w_i x_i + w_0 \\ &= \mathbf{w}^T \mathbf{x} + w_0 \end{aligned}$$

$\forall \mathbf{x}, \mathbf{w} \in \mathbb{R}^d$. This expression can be further compressed to the homogeneous representation where $\forall \tilde{\mathbf{x}}, \tilde{\mathbf{w}} \in \mathbb{R}^{d+1}$, i.e. $\tilde{x}_{d+1} = 1$. We have w.l.o.g.:

$$f(x) = \mathbf{w}^T \mathbf{x}$$

Quantify errors using residuals:

$$\begin{aligned} r_i &= y_i - f(x_i) \\ &= y_i - \mathbf{w}^T \mathbf{x_i} \end{aligned}$$

We can use squared residuals and sum over all residuals to get the cost:

$$\hat{R}(w) = \sum_{i=1}^{n} r_i^2 \tag{1}$$

$$= \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x_i})^2 \tag{2}$$

Optimization objective to find optimal weight vector $\mathbf{w}$ with least squares is the following:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \left(y_i - w^T x_i \right)^2$$

1.1 Closed form solution

This can be solved in closed form:

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

where:

$$X = \begin{bmatrix} X_{1,1} & \ldots & X_{1,d} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \ldots & X_{n,d} \end{bmatrix} \text{ and } y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

1.2 Optimization

1.2.1 Requirements

Requires a convex objective function.

Definition 1.1 (Convexity)**.** *A function is convex iff* $\forall \mathbf{x}, \mathbf{x}', \lambda \in [0,1]$ *it holds that* $f(\lambda x + (1-\lambda)x') \leq \lambda f(x) + (1-\lambda)f(x')$

Note that the least squares objective function defined in 1 is convex.

1.2.2 Gradient descent

We start with an arbitrary $w_0 \in \mathbb{R}^d$, then for $t = 0, 1, 2, \ldots$ we perform the following operation:

$$w_{t+1} = w_t - \eta_t \nabla \hat{R}(w_t)$$

where η_t is the learning rate.
Under mild assumptions, if the step size is sufficiently small, the gradient descent procedure converges to a stationary point, where the gradient is zero. For convex objectives, it therefore finds the optimal solution. In the case of the squared loss and a constant step size (e.g. 0.5), the algorithm converges at linear rate. If you look at the difference in empirical value at iteration t and compare that with the optimal value, then the gap is going to shrink at linear rate. If we look for a solution within a margin ϵ, it is found in $\mathcal{O}(\ln(\frac{1}{\epsilon}))$ iterations. The fact that the objective function congerges at linear rate can be formally described as follows:

$$\exists t_0 \forall t \geq t_0, \exists \alpha < 1 \text{ s.t. } (\hat{R}(w_{t+1}) - \hat{R}(\hat{w})) \leq \alpha(\hat{R}(w_t) - \hat{R}(\hat{w}))$$

where $\hat{w}$ is the optimal value for the hyperparameters.
For computing the gradient, we recall that:

$$\nabla \hat{R}(\hat{w}) = \begin{bmatrix} \frac{\partial}{\partial w_1}\hat{R}(w) & \ldots & \frac{\partial}{\partial w_d}\hat{R}(w) \end{bmatrix}$$

In one dimension, we have that:

$$\begin{aligned} \nabla \hat{R}(w) = \frac{d}{dw}\hat{R}(w) &= \frac{d}{dw}\sum_{i=1}^{n}(y_i - w \cdot x_i)^2 \\ &= \sum_{i=1}^{n}\frac{d}{dw}(y_i - w \cdot x_i)^2 \\ &= 2(y_i - w \cdot x_i) \cdot (-x_i) \\ &= \sum_{i=1}^{n} 2(y_i - w \cdot x_i) \cdot (-x_i) \\ &= -2\sum_{i=1}^{n} r_i x_i. \end{aligned}$$

In d-dimension, we have that:

$$\nabla \hat{R}(w) = -2\sum_{i=1}^{n} r_i x_i,$$

where $r_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^d$

1.2.3 Adaptive step size for gradient descent

The step size can be updates adaptively, via either:

1. **Line search**:
   Suppose at iteration t, we have w_t, $g_t = \nabla \hat{R}(w_t)$. We then define:

$$y_t^* = \arg \min_{y \in [0,\infty)} \hat{R}(w_t) - \eta g_t$$

2. **Bold driver heuristic**:

   - If the function decreases, increase the step size.

$$\text{If } \hat{R}(w_{t+1}) < \hat{R}(w_t) : \eta_{t+1} \leftarrow \eta_t \cdot c_{acc}$$

     where $c_{acc} > 1$

   - If the function increases, decrease the step size.

$$\text{If } \hat{R}(w_{t+1}) > \hat{R}(w_t) : \eta_{t+1} \leftarrow \eta_t \cdot c_{dec}$$

     where $c_{dec} < 1$.

1.2.4 Tradeoff between gradient descent and closed form

Several reasons:

- Computational complexity:

$$\hat{w} = (X^T X)^{-1}(X^T y)$$

  $(X^T X)$ can be computed in $\mathcal{O}(nd^2)$, $(X^T X)^{-1}$ can be computed in $\mathcal{O}(d^3)$.
  By comparison, for gradient descent calculating $\nabla \hat{R}(w) = \sum_{i=1}^{n}(y_i - w^T x_i)x_i$ can be computed in $\mathcal{O}(nd)$, where $n = \ln(\frac{1}{\epsilon})$

- the problem may not require an optimal solution.
- many problems do not admit a closed form solution.

1.3 other loss functions

Least squares is part of a general case of the following general loss function, which is convex for $p \geq 1$.

$$l_p(r) = |r|^p \tag{3}$$

Least squares is where $p = 2$.

2 Generalization and model validation

2.1 Fitting nonlinear functions via linear regression

Using nonlinear features of our data (basis functions), we can fit nonlinear functions via linear regression. Then, the model takes on the form:

$$f(\mathbf{x}) = \sum_{i=1}^{d} w_i \phi(\mathbf{x}) \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^d$, $x \mapsto \tilde{x} = \phi(\mathbf{x}) \in \mathbb{R}^d$ and $w \in \mathbb{R}^d$.

- 1 dim.: $\phi(\mathbf{x}) = \left[1, x, x^2, \ldots, x^k\right]$
- 2 dim.: $\phi(\mathbf{x}) = \left[1, x_1, x_2, x_1^2, x_2^2, \ldots, x_1^k, x_2^k\right]$
- p dim.: $\phi(\mathbf{x})$ vector of all monomials in $x_1, \ldots, x_p$ of degree up to k.

3 Probability (interlude)

3.1 Gaussians

The p.d.f. of a Gaussian distribution is given by:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x-\mu}{2\sigma^2}\right) \tag{5}$$

The p.d.f. of a multivariate Gaussian distribution is given by:

$$\frac{1}{2\pi\sqrt{|\sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\sigma^{-1}(x-\mu)\right) \tag{6}$$

where:

$$\sigma = \begin{pmatrix}\sigma_1^2, \sigma_{12} \\ \sigma_{21}, \sigma_2^2\end{pmatrix} \text{ and } \begin{pmatrix}\mu_1 \\ \mu_2\end{pmatrix} \tag{7}$$