Machine Learning Basics II

Benjamin Roth

CIS LMU München

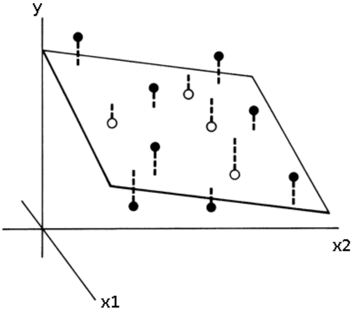
Outline

1. Regression
   - Linear Regression
   - Optimizing Mean Squared Error
   - Maximum Likelihood Estimation
   - Linear Regression as Maximum Likelihood (optional)

Outline

1. Regression
   - Linear Regression
   - Optimizing Mean Squared Error
   - Maximum Likelihood Estimation
   - Linear Regression as Maximum Likelihood (optional)

Linear Regression: Recap



- Linear function:

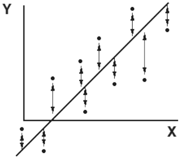
$$\hat{y} = \mathbf{w}^T \mathbf{x} = \sum_{j=1}^{n} w_j x_j$$

- Parameter vector $\mathbf{w} \in \mathbb{R}^n$
  Weight w_j decides if value of feature x_j increases or decreases prediction $\hat{y}$.

Linear Regression: Mean Squared Error

- Mean squared error of training (or test) data set is the sum of squared differences between the predictions and labels of all m instances.

$$MSE := \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$$



- In matrix notation:

$$MSE := \frac{1}{m}||\hat{\mathbf{y}} - \mathbf{y}||_2^2$$

$$= \frac{1}{m}||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$$

Outline

Learning: Improving on MSE

- Gradient: Vector whose components are the n partial derivatives of f.

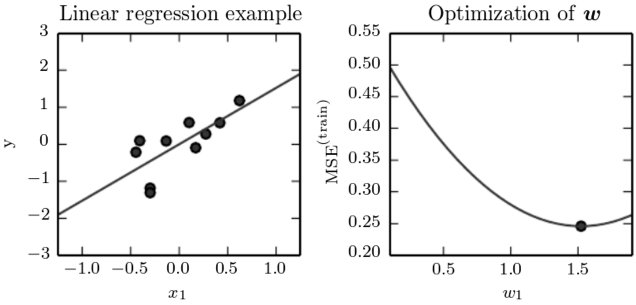
$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \begin{bmatrix} \frac{\partial f(\mathbf{w})}{\partial w_1} \\ \frac{\partial f(\mathbf{w})}{\partial w_2} \\ \vdots \\ \frac{\partial f(\mathbf{w})}{\partial w_n} \end{bmatrix}$$

- View *MSE* as a function of **w**
- Minimum is where gradient is **0**.

$$\nabla_{\mathbf{w}} MSE = \mathbf{0}$$

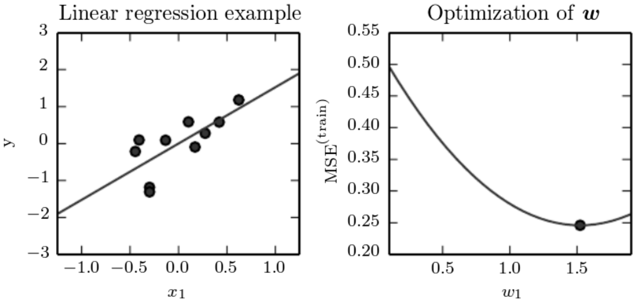
Learning: Improving on MSE

- View *MSE* as a function of **w**



- Minimum is where gradient $\nabla_{\mathbf{w}} MSE = \mathbf{0}$.
- Why minimum and and not maximum or saddle point?
  - Because it is a quadratic function...
  - Check convexity for 1 dimensional function: Second derivative > 0.
  - Check for vector valued function: Hessian is positive-semidefinite.

Second Derivative Test



Second derivative of Mean Squared Error for Linear model with only one feature:

$$\frac{d^2}{dw^2}\sum_{i=1}^{m}(x^{(i)}w - y^{(i)})^2 = \frac{d^2}{dw^2}\sum_{i=1}^{m}(x^{(i)2}w^2 - 2x^{(i)}w + y^{(i)2}) = 2\sum_{i=1}^{m}x^{(i)2} > 0$$

Solving for **w**

- We now know that minimum is where gradient is **0**.

$$\nabla_{\mathbf{w}} MSE = \mathbf{0}$$

$$\Rightarrow \nabla_{\mathbf{w}} \frac{1}{m} ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 = \mathbf{0}$$

- Solve for **w**:

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

(Normal Equation)

Deriving the Normal Equation

- Function to minimize:

$$\begin{aligned} & ||\mathbf{Xw} - \mathbf{y}||_2^2 \\ &= (\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y}) \\ &= \mathbf{w}^T\mathbf{X}^T\mathbf{Xw} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{Xw} + \mathbf{y}^T\mathbf{y} \\ &= \mathbf{w}^T\mathbf{X}^T\mathbf{Xw} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y} \end{aligned}$$

- Take the gradient[1] w.r.t. $\mathbf{w}$ and set equal to $\mathbf{0}$:

$$\begin{aligned} & 2\mathbf{X}^T\mathbf{Xw} - 2\mathbf{X}^T\mathbf{y} = \mathbf{0} \\ & \Rightarrow \mathbf{X}^T\mathbf{Xw} = \mathbf{X}^T\mathbf{y} \\ & \Rightarrow (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Xw} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \end{aligned}$$

[1] [Matrix Cookbook. Petersen and Pedersen, 2012]:

$\nabla_{\mathbf{w}}\mathbf{w}^T\mathbf{a} = \mathbf{a}$

$\nabla_{\mathbf{w}}\mathbf{w}^T\mathbf{Bw} = 2\mathbf{Bw}$ for symmetric $\mathbf{B}$

Linear Regression: Summary

- Model simple linear relationships between $\mathbf{X}$ and $\mathbf{y}$
- Mean squared error is a quadratic function of the parameter vector $\mathbf{w}$, and has a unique minimum.
- Normal equations: Find that minimum by setting the gradient to zero and solving for $\mathbf{w}$.
- Linear algebra packages have special routines for solving least squares linear regression.

Outline

Maximum Likelihood Estimation

- Machine learning models are often more interpretable if they are stated in a probabilistic way.
- Performance measure: What is the probability of the training data given the model parameters?
- Likelihood: Probability of data as a function of model parameters
- $\Rightarrow$ Maximum Likelihood Estimation
- Many models can be formulated in a probabilistic way!

Probability of Data Set

- Data:
  - Set of m examples $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots \mathbf{x}^{(m)}\}$
  - Sometimes written as design matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)T} \\ \vdots \\ \mathbf{x}^{(m)T} \end{bmatrix}$$

- Probability of dataset $\mathbf{X}$, parametrized by $\boldsymbol{\theta}$:

$$p_{model}(\mathbf{X}; \boldsymbol{\theta})$$

Probability of Data Set

- Probability of dataset $\mathbf{X}$, parametrized by theta:

$$p_{model}(\mathbf{X}; \boldsymbol{\theta})$$

- Data points are independent and identically distributed random variables (i.i.d.)
  - Assumption made by many ML models.
  - Identically distributed: Examples come from same distribution.
  - Independent: Value of one example doesn't influence other example.
  - $\Rightarrow$ Probability of data set is product of example probabilities.

$$p_{model}(\mathbf{X}; \boldsymbol{\theta}) = \prod_{i=1}^{m} p_{model}(\mathbf{x}^{(i)}; \boldsymbol{\theta})$$

Maximum Likelihood Estimation

- Likelihood: Probability of data viewed as function of parameters $\boldsymbol{\theta}$
- (Negative) Log-Likelihood (NLL):
  - Logarithm is monotonically increasing
    - Maximum of function stays the same
    - Easier to do arithmetic with (sums vs. products)
  - Optimization is often formulated as minimization $\Rightarrow$ take negative of function.
- Maximum likelihood estimator for $\boldsymbol{\theta}$:

$$\begin{aligned}\boldsymbol{\theta}_{ML} &= \operatorname{argmax}_{\theta} p_{model}(\mathbf{X}; \boldsymbol{\theta}) \\ &= \operatorname{argmax}_{\theta} \prod_{i=1}^{m} p_{model}(\mathbf{x}^{(i)}; \boldsymbol{\theta}) \\ &= \operatorname{argmax}_{\theta} \sum_{i=1}^{m} \log p_{model}(\mathbf{x}^{(i)}; \boldsymbol{\theta})\end{aligned}$$

Conditional Log-Likelihood

- Log-likelihood can be stated for supervised and unsupervised tasks.
- Unsupervised learning (e.g. density estimation).
  - Task: model $p_{model}(\mathbf{X}; \boldsymbol{\theta})$ (as before)
  - $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots \mathbf{x}^{(m)}\}$
- Supervised learning (Predictive modelling):
  - Task: model $p_{model}(\mathbf{y}|\mathbf{X}; \boldsymbol{\theta})$
  - $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots \mathbf{x}^{(m)}\}, \quad \mathbf{y} = \{y^{(1)}, \ldots y^{(m)}\}$
- Maximum likelihood estimation for the supervised i.i.d. case:

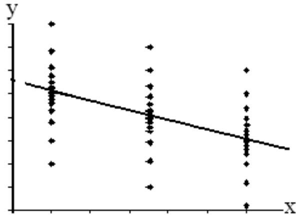
$$\boldsymbol{\theta}_{ML} = \text{argmax}_{\boldsymbol{\theta}} P(\mathbf{y}|\mathbf{X}; \boldsymbol{\theta})$$

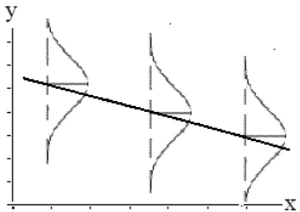
$$= \text{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log P(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta})$$

Outline

Linear Regression as Maximum Likelihood

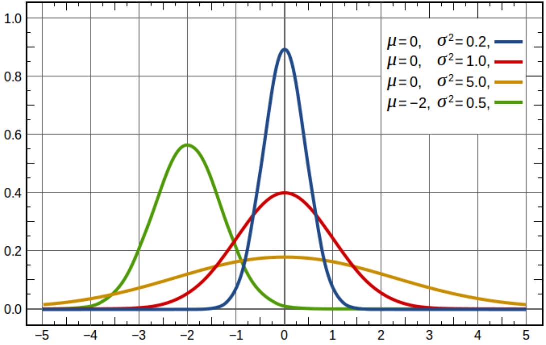
- Instead of predicting one value $\hat{y}$ for an input $\mathbf{x}$, model probability distribution $p(y|\mathbf{x})$.
- For the same value of $\mathbf{x}$, different values of y may occur (with different probability).





Gaussian Distribution

- Gaussian distribution: $N(y|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right]$
  - Quadratic function as negative exponent, scaled by variance
  - Normalization factor $\frac{1}{\sigma\sqrt{2\pi}}$



Linear Regression as Maximum Likelihood

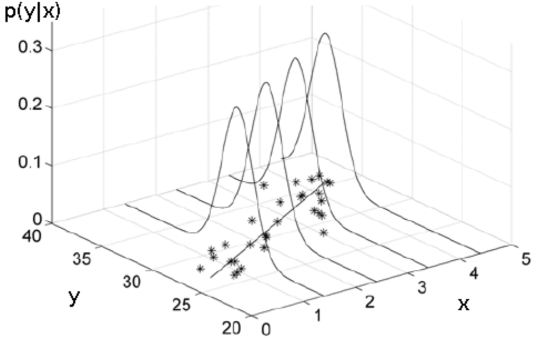
- Assume label y is distributed by a Gaussian, depending on features $\mathbf{x}$

$$p(y|\mathbf{x}) = N(y|\mu, \sigma^2)$$

where the mean is determined by the linear transformation

$$\mu = \boldsymbol{\theta}^T \mathbf{x}$$

and σ is a constant.



Linear Regression as Maximum Likelihood

- Gaussian distribution: $N(y|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right]$
  - Taking the log makes it a quadratic function!
- Conditional log-likelihood:

$$
\begin{aligned}
& -\log P(\mathbf{y}|\mathbf{X};\boldsymbol{\theta}) \\
&= -\sum_{i=1}^{m} \log p(y^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta}) \\
&= m \log\sigma + \frac{m}{2}\log(2\pi) + \sum_{i=1}^{m} \frac{(y^{(i)} - \boldsymbol{\theta}^T\mathbf{x}^{(i)})^2}{2\sigma^2} \\
&= \text{const} + \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y^{(i)} - \boldsymbol{\theta}^T\mathbf{x}^{(i)})^2
\end{aligned}
$$

- What is optimal $\boldsymbol{\theta}$?

Linear Regression as Maximum Likelihood

- Conditional negative log-likelihood:

$$NLL(\boldsymbol{\theta}) = \text{const} + \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)})^2$$

- Compare to previous result:

$$MSE(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{n} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2$$

- Minimizing NLL under these assumptions is equivalent to minimizing MSE!

Maximum Likelihood: Summary

- Many machine learning problems can be stated in a probabilistic way.
- Mean squared error linear regression can be stated as a probabilistic model that allows for Gaussian random noise around the predicted value $\hat{y}$.
- A straightforward optimization is to maximize the likelihood of the training data.
- Maximum likelihood is not Bayesian, and may give undesirable results (e.g. if there is only little training data).
- In practice, MLE and point estimates are often used to solve machine learning problems.