Parameter Estimation

UW ECE 457B/657A - Core Topic

Mark Crowley

Outline

Descriptive versus Inferential Analysis

- We have data (samples). This data is a sample of a population (more than just the measured or observed sample).
- **Descriptive Analysis** is the type of analysis and measures that seek to *describe and summarize* the data, the available samples.
- We can not in general use it for interpretation of unobserved data.
- **Inferential Analysis (predictive)** is the type of analysis that can describe measures over the population of data whether they are observed or unobserved.
- So it goes beyond the data we can see, make inferences about the larger population.

Example: the mean of a sample

- Take for example calculating the mean of a sample.
- The mean is correct for just the sample as it is descriptive of the sample.
- For inferential analysis it can be only make an estimate of the mean of the population as a whole.

Outline

Point Estimation (Parameter Estimation)

- Given a set of **independent and identically distributed (i.i.d.)** data points $\{x_1, x_2, \ldots, x_n\}$ about a **random variable** X.
- We can define a **point estimator** or **statistic** as a function of the data.

$$\theta = g(x_1, x_2, \cdots, x_n)$$

- We don't know θ ("theta") so we estimate it and call the estimate $\hat{\theta}$ ("theta hat").
- This is usually done by calculating the parameter value for the *population sample*.
- For example, if we assume a Gaussian distribution for the we can estimate the mean and variance of a population, then $\theta = \{\mu, \sigma\}$

Unbiased Estimator Meaning

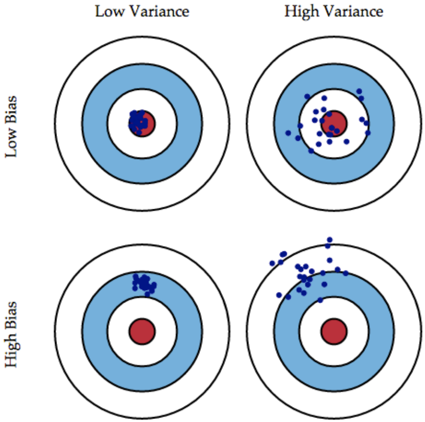
- For a estimator which is based on repeated samples of the population,
  - if the **expected value** of the estimate
    - *equals* the **actual value** of the parameter
  - then the estimator is called **unbiased**
  - otherwise the difference is called the **bias**.
  - **IOW:** We want an estimator that is *correct on average*. This is true of an unbiased estimator.
    - *(Of course, this isn't the only important measure of an good estimator: Variance, MSE, ...)*

Bias of estimator

$$Bias(\hat{\theta}) = B(\hat{\theta}) = E(\hat{\theta}) - \theta$$

- where the expectation $E(\hat{\theta})$ is taken over all the data,
- and θ is the true value of the parameter used to generate the data we have.
- An *unbiased estimator* has a bias of 0.

Bias vs. Variance Tradeoff



[From http://scott.fortmann-roe.com/docs/BiasVariance.html]

Review: Expectation Operator

- **Expectation:** For a discrete random variable, with k possible outcomes x_i, each with probability p_i of occurring, the **expected value** is:

$$E[x] = \sum_{i=1}^{k} x_i p_i(x_i)$$

- For equal probabilities, then **expectation=mean**

Bernoulli Distribution

- Suppose a coin (Heads, Tails) is tossed in the air 5 times resulting in $\mathbf{X}_{1:5}$ =*(H,H,H,H,T)*=(1,1,1,1,0).
- We can model this to follow the **Bernoulli distribution** if we know the probability p_H of the coin coming up Heads. *For a fair coin,* $p_H = p_T = .5$

Then for each coin toss we have a probability:

$$P(X_i = 1|p_H) = p_H^{x_i}(1 - p_H)^{1-x_i}$$

$$p_H : \text{Probability of H}$$

$$X_i : \text{Observation (coin toss number } i\text{)}$$

Is this estimator unbiased?

Task: Estimate the parameter $\theta = p_H = p$, the probability $P(X_i = H)$ for the Bernoulli Distribution from data.

Proposed Estimator - **Sample Mean**

$$\hat{\theta} = \bar{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}(X_i = H) = \frac{1}{n}\sum_{i=1}^{n} X_i$$

To check bias, substitute proposed estimator into the distribution:

$$\begin{aligned} B(\hat{\theta}) &= E[\hat{\theta}] - \theta \\ &= E[\hat{p}] - p \\ &= E\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] - p \end{aligned}$$

Is this estimator unbiased?

$$
\begin{aligned}
B(\hat{p}) &= E[\hat{p}] - p \\
&= E[\bar{\mathbf{X}}] - p \\
&= E\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] - p \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} E[X_i]\right) - p \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} p\right) - p \\
&= \left(\frac{np}{n}\right) - p \\
&= p - p = 0
\end{aligned}
$$

Aside:

$$
\begin{aligned}
&= E[X_i] \\
&= \sum_{x_i \in \{H=1, T=0\}} (x_i \text{ value})\ (x_i \text{ probability}) \\
&= \sum_{x_i \in \{H=1, T=0\}} x_i \left(p^{x_i}(1-p)^{(1-x_i)}\right) \\
&= 1p^1(1-p)^0 + 0p^0(1-p)^1 \\
&= p
\end{aligned}
$$

Is this estimator unbiased?

$$
\begin{aligned}
B(\hat{p}) &= E[\hat{p}] - p \\
&= E[\bar{\mathbf{X}}] - p \\
&= E\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] - p \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} E[X_i]\right) - p \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} p\right) - p \\
&= \left(\frac{np}{n}\right) - p \\
&= p - p = 0
\end{aligned}
$$

Aside:

$$
\begin{aligned}
&= E[X_i] \\
&= \sum_{x_i \in \{H=1, T=0\}} (x_i \text{ value})\ (x_i \text{ probability}) \\
&= \sum_{x_i \in \{H=1, T=0\}} x_i \left(p^{x_i}(1-p)^{(1-x_i)}\right) \\
&= 1p^1(1-p)^0 + 0p^0(1-p)^1 \\
&= p
\end{aligned}
$$

Therefore, the estimator is **unbiased**.

Is this estimator unbiased?

Task: Estimating the variance.

Proposed Estimator - **Sample Variance** $\hat{\sigma}^2$ ("sigma hat")

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

$$\begin{aligned}
B(\hat{\sigma}^2) &= E[\hat{\sigma}^2] - \sigma^2 \\
&= E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2\right] - \sigma^2 \\
&= \left(\frac{n-1}{n}\sigma^2\right) - \sigma^2 \\
&= \frac{-\sigma^2}{n}
\end{aligned}$$

Is this estimator unbiased?

Task: Estimating the variance.
Proposed Estimator - **Sample Variance** $\hat{\sigma}^2$ ("sigma hat")

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

$$\begin{aligned} B(\hat{\sigma}^2) &= E[\hat{\sigma}^2] - \sigma^2 \\ &= E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2\right] - \sigma^2 \\ &= \left(\frac{n-1}{n}\sigma^2\right) - \sigma^2 \\ &= \frac{-\sigma^2}{n} \end{aligned}$$

The estimator is **biased**!

Is this estimator unbiased?

Estimating the variance.
Proposed Estimator - $\tilde{\sigma}^2$ ("sigma tilde")

$$\tilde{\sigma} = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \hat{\mu})^2$$

Try it yourself.

Mean Squared Error

- The simplest way to assess quality of an estimate is **variance**:

$$V(\hat{\theta}) = E(\hat{\theta} - E(\hat{\theta}))^2$$

- A better measure for the effectiveness of an estimator is a combination of bias and variance, the **Mean Squared Error (MSE)**

$$MSE(\hat{\theta}) = E(\hat{\theta} - \theta)^2 = B^2(\hat{\theta}) + V(\hat{\theta})$$

- Alternative: **Root Mean Squared Error (RMSE)** $= \sqrt{MSE}$
  - gives a better sense of the magnitude of the error
- MSE and Bias: If the estimator θ is unbiased then

$$MSE = Variance$$

Outline

Bayesian Learning/Conjugate Methods

- **MAP:** tries to find the best seting given the parameters
- **MLE:** tries to maximize the parameters to best explain the data
- **Probabilistic Inference:** computes this posterior given that we know the parameters

Maximum Likelihood Estimate (MLE)

- Assumes that the samples are from a specific distribution with some unknown parameter(s) $\mathbf{X} \sim P(X_i|\theta)$
- **Likelihood** is the probability that the samples observed come from the given distribution. $\mathcal{L}(\theta) = p(X|\theta)$
  - IOW: How *likely* is would the data *given* the proposed distribution?
- We can estimate the parameter θ by maximizing the likelihood function.

Maximum Likelihood Estimate (MLE)

Given a known distribution form $P(X_i|\theta)$
Given the sample $\mathbf{X} = \{x_1, \cdots, x_n\}$
The likelihood can be formulated

$$\mathcal{L}(\theta) = P(X_1, \ldots, X_n|\theta) = \prod_{i=1}^{n} P(X_i|\theta)$$

Maximum Likelihood Estimate (MLE)

- The MLE is the value of parameter θ that **maximizes** the likelihood
  - IOW: the parameter that explains the data generating process the best, given the evidence.
- To find the MLE analytically:
  - find the derivative of the log-likelihood with respect to the parameter θ
  - equate the resulting formula to zero
  - solve for θ
- *Note:* if there are multiple parameters then you can maximize each one seperately or attempt to find a mulidimensional maximum.

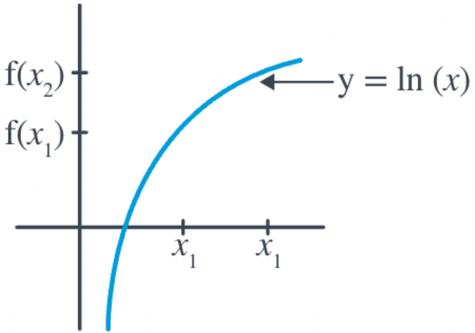
Logification

Since max $\mathcal{L}$ is the same as max log $\mathcal{L}$ we can use

$$\mathcal{L}(\theta) = \log \mathcal{L}(\theta) = \log \prod_{i=1}^{n} P(x_i|\theta)$$

$$= \sum_{i=1}^{n} \log P(x_i|\theta)$$

...it also happens to make many types of derivations easier.



MLE on the Bernoulli Distribution

Back to the Bernoulli with a "different" proposed estimator, MLE:

- Suppose a coin is tossed in the air 5 times (H,H,H,H,T).
- Assuming a fair coin, the values follow the **Bernoulli distribution**

$$
\begin{aligned}
P(x_i|p) &= p^{x_i}(1-p)^{1-x_i} \\
p &- \text{probability of H or T} \\
x_i &- \text{Observation (sample)}
\end{aligned}
$$

Example for MLE

Using the Bernoulli Disitrbution:

$$\begin{aligned}\mathcal{L}(p|x_1, \cdots, x_5) &= \prod_{i=1}^{5} p^{x_i}(1-p)^{1-x_i} \\ &= p^{\sum_{i=1}^{5} x_i}(1-p)^{5-\sum_{i=1}^{5} x_i}\end{aligned}$$

Taking the log *(actually* $\ln(x)$*...)* of both sides we get

$$\mathcal{L}(p) = \log \mathcal{L}(p) = \sum_{i=1}^{5} x_i \log(p) + (5 - \sum_{i=1}^{5} x_i) \log(1-p)$$

Example for MLE

To Maximize the likelihood, take the derivative and equate to zero

$$\frac{\partial \mathcal{L}(p)}{\partial p} = \frac{\sum_{i=}^{5} x_i}{p} - \frac{5 - \sum_{i=1}^{5} x_i}{1-p} = 0$$

$$\frac{\sum_{i=1}^{5} x_i}{p} = \frac{5 - \sum_{i=1}^{5} x_i}{1-p}$$

$$p = \frac{\sum_{i=1}^{5} x_i}{5}$$

If we define heads $= 1$, tails $= 0$ and we see 4 heads then $\hat{p} = \frac{4}{5} = 0.8$
This $\hat{p}$ is the estimate for p that maximizes the likelihood that the given sequence will be observed.

Another MLE Example : Poisson Distribution

Poisson Distribution : the probability of obtaining exactly n successes in N trials of a **Poisson Process**.

$$P(x_i, \ldots, x_n|\lambda) = \frac{e^{-n\lambda}\lambda^{\sum x_i}}{\prod x_i!}$$

Take log: $$\ln f = -n\lambda + (\ln\lambda)\sum x_i - \ln\left(\prod x_i!\right)$$

Derivative, set to zero: $$\frac{d(\ln f)}{\lambda} = -n + \frac{\sum x_i}{\lambda} = 0$$

Then solve for λ.

MLE Formulation

$$\hat{\lambda} = \frac{\sum x_i}{n}$$

A MAP to the treasure...

MAP vs. MLE

There is a another way to estimate a particular parameter value *if you have prior knowledge about the distribution*.

Maximum-a-Posteriori Estimation (MAP)

- **Maximum-a-Posteriori Estimation (MAP):** Choose θ that maximizes the posterior probability of θ (i.e., probability in the light of the observed data)
- Posterior probability of θ is given by the Bayes Rule

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

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

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

- The Bayes Rule lets us update our belief about θ in the light of observed data
- While doing MAP, we usually maximize the log of the posterior probability

Maximum A Posteriori

Estimated parameter

Log prior

$$\theta_{\text{MAP}} = \underset{\theta}{\operatorname{argmax}} \left(\log(g(\theta)) + \sum_{i=1}^{n} \log(f(X_i|\theta)) \right)$$

Chose the value of theta that maximizes:

Sum of log likelihood

Maximum-a-Posteriori Estimation (MAP)

- Maximum-a-Posteriori parameter estimation

$$\begin{aligned}
\hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta \mid \mathcal{D}) &= \arg\max_{\theta} \frac{P(\theta)P(\mathcal{D} \mid \theta)}{P(\mathcal{D})} \\
&= \arg\max_{\theta} P(\theta)P(\mathcal{D} \mid \theta) \\
&= \arg\max_{\theta} \log P(\theta)P(\mathcal{D} \mid \theta) \\
&= \arg\max_{\theta} \{\log P(\theta) + \log P(\mathcal{D} \mid \theta)\}
\end{aligned}$$

$$\boxed{\hat{\theta}_{MAP} = \arg\max_{\theta} \{\log P(\theta) + \sum_{n=1}^{N} \log P(\mathbf{d}_n \mid \theta)\}}$$

- Same as MLE except the extra log-prior-distribution term!
- MAP allows incorporating our prior knowledge about θ 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 ϵ is drawn from a Gaussian distribution:

$$\epsilon \sim \mathcal{N}or(0, \sigma^2)$$

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

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

- Probability of each response variable

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

- 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: MLE vs MAP (summary)

- MLE solution:

$$\boxed{\hat{\mathbf{w}}_{MLE} = \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbf{w}^\top \mathbf{x}_n)^2}$$

- MAP solution:

$$\boxed{\hat{\mathbf{w}}_{MAP} = \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbf{w}^\top \mathbf{x}_n)^2 + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w}}$$

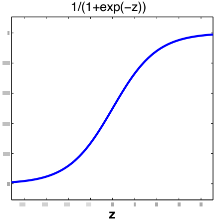
- **Take-home messages:**
  - MLE estimation of a parameter leads to unregularized solutions
  - MAP estimation of a parameter leads to regularized solutions
  - The prior distribution acts as a regularizer in MAP estimation
- Note: For MAP, different prior distributions lead to different regularizers
  - Gaussian prior on $\mathbf{w}$ regularizes the ℓ_2 norm of $\mathbf{w}$
  - Laplace prior $\exp(-C||\mathbf{w}||_1)$ on $\mathbf{w}$ regularizes the ℓ_1 norm of $\mathbf{w}$

Outline

Probabilistic Classification: Logistic Regression

- Often we don't just care about predicting the label y for an example
- Rather, we want to predict the label probabilities $P(y \mid \mathbf{x}, \mathbf{w})$
  - E.g., $P(y = +1 \mid \mathbf{x}, \mathbf{w})$: 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 ($y = -1/+1$):

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



- σ 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 :-)

Linear Regression vs. Logistic Regression

- A simple type of **Generalized Linear Model**
- Linear regression learns a function to predict a continuous variable output of continuous or discrete input variables

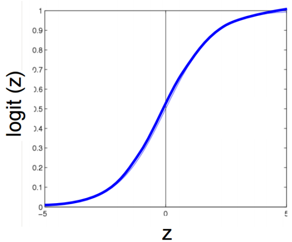
$$= b_0 + \sum(b_i X_i) + \epsilon$$

- Logistic regression predicts the probability of an outcome, the appropriate class for an input vector or the **odds** of one outcome being more likely than another.

The Logistic (Sigmoid) Function

Define probability of label being 1 by fitting a linear weight vector to the logistic (a.k.a sigmoid) function.

$$P(Y=1|X) = \frac{1}{1+e^{-(w_0+\sum_i w_i x_i)}}$$
$$= \frac{1}{1+\exp(-(w_0+\sum_i w_i x_i))}$$

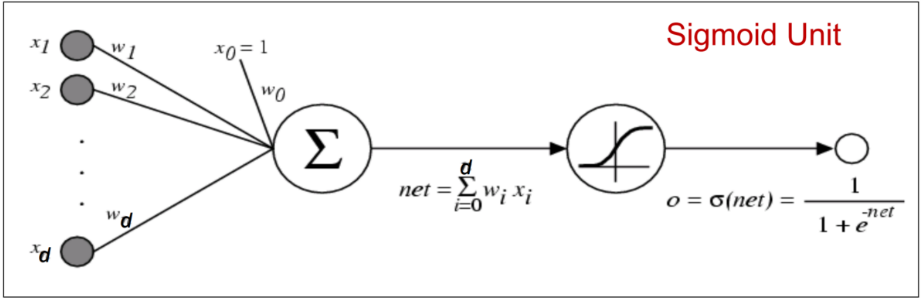


Advantage of this is you can turn the continuous $[-\infty, \infty]$ feature information into $[0, 1]$ and treat it like a probability.

Bias: w_0 is called the bias, it basically adjusts the sigmoid curve to the left or right, biasing what the expected outcome is. The other weights w_i adjust the steepness of the curve in each dimension.

Logistic Regression as a Graphical Model

$$P(\mathbf{x}) = \sigma(w^T x_i) = \sigma(w_0 + \sum_i w_i x_i) = \frac{1}{1 + \exp(-(w_0 + \sum_i w_i x_i))}$$



Properties of Logistic Regression

- Very simple yet powerful classification method
- LATER: Relatively easy to fit to data using gradient descent, many methods for doing this
- Learned parameters can be interpreted as computing the **log odds**, the logarithm of the ratio of successes to failures

$$
\begin{aligned}
LO &\sim \log \frac{p(Y=1|X)}{p(Y=0|X)} \\
&= w^T \mathbf{x}
\end{aligned}
$$

- If x_0 : number of cigarettes per day
- x_1 : minutes of excercise
- $y = 1$: getting lung cancer
- Then if parameters learned at $\hat{w} = (1.3, -1.1)$, it means each cigarette raises odds of cancer by factor of $e^{1.3}$

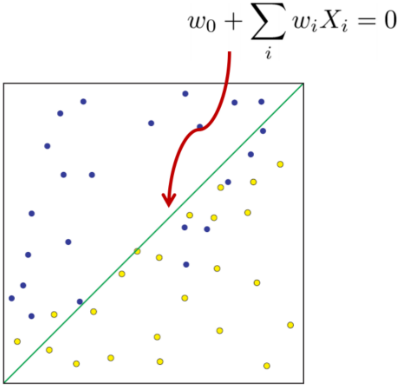
Logistic Regression Used as a Classifier

Logistic Regression can be used as a simple linear classifier.

- Compare probabilities of each class $P(Y = 0|X)$ and $P(Y = 1|X)$.
- Treat the halfway point on the sigmoid as the decision boundary.

$P(Y = 1|X) > 0.5$ classify X in class 1

$$w_0 + \sum_i w_i x_i = 0$$



Outline

Logistic Regression: Maximum-a-Posteriori Solution

- Let's assume a Gaussian prior distribution over the weight vector $\mathbf{w}$

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

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

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

- 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: 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:

$$\begin{aligned}
\log \mathcal{L}(\mathbf{w}) = \log P(\mathcal{D} \mid \mathbf{w}) = \log P(\mathbf{Y} \mid \mathbf{X}, \mathbf{w}) &= \log \prod_{n=1}^{N} P(y_n \mid \mathbf{x}_n, \mathbf{w}) \\
&= \sum_{n=1}^{N} \log P(y_n \mid \mathbf{x}_n, \mathbf{w}) \\
&= \sum_{n=1}^{N} \log \frac{1}{1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)} \\
&= \sum_{n=1}^{N} - \log[1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)]
\end{aligned}$$

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

$$\begin{aligned}
\nabla_{\mathbf{w}} \log \mathcal{L}(\mathbf{w}) &= \sum_{n=1}^{N} - \frac{1}{1 + \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)} \exp(-y_n \mathbf{w}^\top \mathbf{x}_n)(-y_n \mathbf{x}_n) \\
&= \sum_{n=1}^{N} \frac{1}{1 + \exp(y_n \mathbf{w}^\top \mathbf{x}_n)} y_n \mathbf{x}_n
\end{aligned}$$

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

$$P(y = k \mid \mathbf{x}, \mathbf{w}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{x})}{\sum_k \exp(\mathbf{w}_k^\top \mathbf{x})}$$

  - We maintain a separator weight vector $\mathbf{w}_k$ for each class k
- Possible to kernelize it to learn nonlinear boundaries

Outline

General MLE Formulas

Bernoulli Distribution:

$$\text{MLE} : \hat{p} = \frac{\sum x_i}{n}$$

Normal Distribution:

$$\text{MLE} : \hat{\mu} = \frac{\sum x_i}{n}$$

$$\text{MLE} : \hat{\sigma} = \sqrt{\frac{\sum(x_i - \hat{mu})^2}{n}}$$

Notes and Limitations about MLE

- The **maximum likelihood estimator** coincides with the *most probable Bayesian estimator* given a uniform prior distribution on the parameters
- So it's correct, but why doesn't it get better with more data?
- it assumes the form of the distribution is known (gaussian, poisson, etc)
- but it also assumes that all the uncertainty is explained by that parameter you are trying to fit.
- Rather than it being is only an approximation of some other true distribution

General Properties of Bayes Classifiers

Incrementality: with each training example, the prior and the likelihood can be updated dynamically. It is flexible and robust to errors.

Uses Prior Knowledge: Combines prior knowledge and observed data.

- Prior probability of a hypothesis multiplied with probability of the hypothesis given the training data.

Probabilistic hypotheses: outputs not only a classification, but a **probability distribution** over all classes.

Meta-classification: the outputs of several classifiers can be combined

- e.g., by multiplying the probabilities that all classifiers predict for a given class.

Notes and Limitations of MAP

- Usually harder to estimate that MLE
- If we use a uniform prior distribution for θ : then MAP estimate $=$ MLE
- Given infinite data : then MAP estimate converges to MLE
- MAP is useful when you have less data, so you need additional knowledge about the domain
  - MAP estimate tends to converges to faster than MLE even with an arbitrary distribution
  - Can help prevent overfitting

Useful for model adaptation (MAP adaptation)

- Learn MLE on larger dataset, use this as your prior distribution
- Learn MAP estimate on your dataset

Improving on MAP : Regularization on the loss function

- The MAP estimate:

$$\begin{aligned}\hat{\mathbf{w}}_{MAP} &= \arg\max_{\mathbf{w}} \log P(\mathbf{w} \mid \mathcal{D}) \\ &= \arg\max_{\mathbf{w}} \{\log P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w})\} \\ &= \arg\min_{\mathbf{w}} \{-\log P(\mathcal{D}|\mathbf{w}) - \log P(\mathbf{w})\}\end{aligned}$$

- Recall the regularized loss function minimization:

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

- Negative log likelihood $-\log P(\mathcal{D}|\mathbf{w})$ corresponds to the loss $L(\mathbf{Y}, \mathbf{X}, \mathbf{w})$
- Negative log prior $-\log P(\mathbf{w})$ corresponds to the regularizer $R(\mathbf{w})$

Outline

Naive Bayes Classifier

- **Naive Bayes** - probabilistic model that assumes independence of input features.
- It uses Bayes rule to update a distribution over the inputs to minimizes entropy of the outputs.
- Classification is carried out by threshold on probability.

Outline

The Problem of Missing Data

- This analytical MLE method requires that we can solve the equations exactly.
- What if we have missing data?

Expectation Maximization (EM)

- Simple and general approach for parameter estimation with *incomplete* data
- Obtains initial estimates for parameters.
- Then, iteratively use estimates for missing data and continues until convergence
- Pros: Easy to implement (other solutions are more powerful but often require gradients)
- Cons: Slow to converge, might find local maxima, sensitive to initial guess, bad in high dimensions?

The EM Algorithm

Given initial estimates and training data repeat these two steps:

1. **Estimation** - calculate a value for the missing data
2. **Maximization** - use the data and new values of the missing data to find new estimates using MLE.

Repeat steps 1 and 2 until convergence.
Convergence means the change in values is less then some small threshold ϵ.

Example for EM

- **Input** Partial data set of size $k = 4$ $X = \{1, 5, 10, 4\}$ but 2 data items missing so full data size is $n = 6$.
- **Problem:** Assume data has a normal distribution, find mean μ for data.
- For a gaussian distribution the MLE is

$$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n}$$

- Give an initial guess $\hat{\mu}_0 = 3$. Use this to fill in missing values.

Example for EM

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{k} x_i}{n} + \frac{\sum_{i=k+1}^{n} x_i}{n} = \frac{1+5+10+4}{6} + \frac{3+3}{n} = 4.33$$

Useμ_1 as new estimate

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{k} x_i}{n} + \frac{\sum_{i=k+1}^{n} x_i}{n} = \frac{1+5+10+4}{6} + \frac{4.33+4.33}{n} = 4.77$$

$$\hat{\mu}_3 = 4.92 \quad |\mu_2 - \mu_3| = .15$$

$$\hat{\mu}_4 = 4.97 \quad |\mu_3 - \mu_4| = .05$$

Let $\epsilon = .05$, then convergence occurs at $\hat{\mu}_4$.

Outline

Linear Regression: Maximum Likelihood Solution

- Log-likelihood:

$$\log \mathcal{L}(\mathbf{w}) = \log P(\mathcal{D} \mid \mathbf{w}) = \log P(\mathbf{Y} \mid \mathbf{X}, \mathbf{w}) = \log \prod_{n=1}^{N} P(y_n \mid \mathbf{x}_n, \mathbf{w})$$

$$= \sum_{n=1}^{N} \log P(y_n \mid \mathbf{x}_n, \mathbf{w})$$

$$= \sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_n - \mathbf{w}^\top \mathbf{x}_n)^2}{2\sigma^2}\right]$$

$$= \sum_{n=1}^{N} \left\{-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(y_n - \mathbf{w}^\top \mathbf{x}_n)^2}{2\sigma^2}\right\}$$

- Maximum Likelihood Solution: $\hat{\mathbf{w}}_{MLE} = \arg\max_{\mathbf{w}} \log P(\mathcal{D} \mid \mathbf{w})$

$$= \arg\max_{\mathbf{w}} -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2$$

$$= \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{w}^\top \mathbf{x}_n)^2$$

- 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 $\mathbf{w}$

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

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

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

- For $\sigma = 1$ (or some constant) for each input, it's equivalent to the regularized least-squares objective