1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation

5. More...
Different coin-based models

- Beta-Binomial model (last lecture):
  - one coin parameter per class $y$
  - multiple flips of the same coin parameter to generate $x$

- Logistic regression model (this lecture):
  - one coin parameter for every $x$
  - flip coin once to generate $y$
Bernoulli decision function

- Recap: Bernoulli Model $P(Y = 1 \mid \theta) = \theta$ and $P(Y = 0 \mid \theta) = 1 - \theta$

  $Y \sim \text{Ber}(\theta), \theta \in [0, 1]$

- **Given $\theta$, what are the odds of drawing a $Y = 1$?**

  $\frac{P(Y = 1 \mid \theta)}{P(Y = 0 \mid \theta)}$

- In the case of Bernoulli

  $\frac{\theta}{1 - \theta}$

- Perform in log space

  $\ln(\theta) - \ln(1 - \theta)$
Bernoulli odds

Bernoulli odds decision function

\[ \frac{\theta}{1 - \theta} > 1 \]
**Bernoulli log odds**

In log space: Bernoulli **log odds**

\[
\ln \frac{\theta}{1 - \theta} > \ln 1 \\
\ln(\theta) - \ln(1 - \theta) > 0
\]

(1)  

(2)
Inverse to Bernoulli log odds?

- Bernoulli log odds $\mu = f(\theta) = \ln(\theta) - \ln(1 - \theta)$
- Inverse: $\theta = f^{-1}(\mu)$?
- $\theta = \frac{\exp(\mu)}{1 + \exp(\mu)} = \frac{1}{\exp(-\mu) + 1} = f^{-1}(\mu)$ is the **logistic function**.
- Remember: The logistic function is the inverse to the Bernoulli log odds function!
Outline

1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation

5. More...
Logistic regression

- supervised model
- classification model: given $x_i$, predict $y_i$
- discriminative model
- probabilistic predictions
- “log odds = linear function of inputs”
Logistic regression model

- Given an input \( x_i \), a \( D \)-dimensional feature vector \( \{x_{ij}\}_{j=1}^{D} \)
- Predict an output label \( y_i \in \{0, 1\} \)
- Model parameters are weights \( w_1, w_2, \ldots, w_D \) and intercept \( w_0 \) (all to be determined)
- Model “log odds = linear function of \( x \)”

\[
P(Y_i = 1 \mid x_i) = \frac{1}{1 + \exp(w_0 + \sum_{j=i}^{D} w_j x_{ij})} \quad (3)
\]

\[
P(Y_i = 0 \mid x_i) = \frac{\exp(w_0 + \sum_{j=i}^{D} w_j x_{ij})}{1 + \exp(w_0 + \sum_{j=i}^{D} w_j x_{ij})} \quad (4)
\]

\[
= 1 - P(Y_i = 1 \mid x_i) \quad (5)
\]

- \( Y_i = 1 \) iff

\[
1 < \frac{P(Y_i = 1 \mid x_i)}{P(Y_i = 0 \mid x_i)} \iff 1 > \exp(w_0 + \sum_{j=i}^{D} w_j x_{ij})
\]
LogReg: Alternative notation

\[
\exp(w_0 \underbrace{1}_{1} + \sum w_j x_{ij})
\]

- Expand feature vector \(x_i\) to start with \(x_{i0} = 1\).
- Use Dirac’s inner product notation \(\langle \cdot, \cdot \rangle\)

\[
\exp(\langle \mathbf{w}, x_i \rangle) \text{ with } \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{pmatrix} \quad x_i = \begin{pmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{iD} \end{pmatrix}
\]

\[
P(Y_i = 1 \mid x_i) = \frac{1}{1 + \exp(\langle \mathbf{w}, x_i \rangle)}
\]
Intuition of the LogReg model

\[ P(Y_i = 1 \mid x_i) = \frac{1}{1 + \exp(\langle w, x_i \rangle)} \]

Three parts of the model (inner first):
- **Linear predictor**: \( \mu = \langle w, x_i \rangle \)
- **Link function** \( f \): \( \theta = f^{-1}(\mu) = \frac{1}{1 + \exp(-\mu)} \).
  
  This \( f^{-1} \) is the logistic function, one of the sigmoid functions.
- **Probability distribution**: \( Y_i \sim \text{Ber}(\theta) = P(Y_i = 1 \mid \theta) \) Bernoulli
Example in Jupyter notebook

- Refer to the Jupyter Notebook "03-HTML-log-reg-example.ipynb" (or *py or *html)
Pros and Cons of LogReg

- good: very simple and useful classification model.
- good: any feature function can be used. E.g. bag-of-words.
- bad: if feature vectors are linearly dependent then weights are underdetermined.
- bad: all parameters depend on each other, therefore training can not be trivially parallelized.
- bad: when examples are linearly separable, $w$ may go towards infinity. Approach: add regularization of parameter.
- good: under mild conditions, intercept $w_0$ may correct for imbalanced class datasets.

More discussions in Murphy’s book in Section 8.6. and link below.
Outline

1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation

5. More...
Generalized Linear Models (GLMs)

Variations on the three parts of the model (inputs: $x_i$, outputs: $y_i$) where the log-odds of $y$ is a linear model in the inputs $x$

- **Linear predictor:**
  a linear model of $D$ independent variables $x_{*j}$ yielding outcome $\mu = \langle w, x_i \rangle$

- **Link (Mean) function:**
  a link function $f$ connecting parameter $\theta$ with the outcome $\mu$ of the linear model

  **Link function:** $\mu = f(\theta)$

  **Mean function:** $\theta = f^{-1}(\mu)$

- **Probability distribution:**
  Dependent variable $Y_i = y_i$ drawn from a distribution with parameter $\theta$

  - The link function is chosen to transform the linear predictor outcome into the parameter space of the probability distribution
## Canonical link functions (change of variable names)

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Support of distribution</th>
<th>Typical uses</th>
<th>Link name</th>
<th>Link function</th>
<th>Mean function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>real: $(-\infty, +\infty)$</td>
<td>Linear-response data</td>
<td>Identity</td>
<td>$X\beta = \mu$</td>
<td>$\mu = X\beta$</td>
</tr>
<tr>
<td>Exponential</td>
<td>real: $(0, +\infty)$</td>
<td>Exponential-response data, scale parameters</td>
<td>Inverse</td>
<td>$X\beta = -\mu^{-1}$</td>
<td>$\mu = -(X\beta)^{-1}$</td>
</tr>
<tr>
<td>Gamma</td>
<td>real: $(0, +\infty)$</td>
<td>Inverse squared</td>
<td>$X\beta = -\mu^{-2}$</td>
<td>$\mu = (-X\beta)^{-1/2}$</td>
<td></td>
</tr>
<tr>
<td>Poisson</td>
<td>integer: $0, 1, 2, \ldots$</td>
<td>count of occurrences in fixed amount of time/space</td>
<td>Log</td>
<td>$X\beta = \ln(\mu)$</td>
<td>$\mu = \exp(X\beta)$</td>
</tr>
<tr>
<td>Bernoulli</td>
<td>integer: ${0, 1}$</td>
<td>outcome of single yes/no occurrence</td>
<td>$X\beta = \ln\left(\frac{\mu}{1-\mu}\right)$</td>
<td>$\mu = \frac{\exp(X\beta)}{1 + \exp(X\beta)}$</td>
<td></td>
</tr>
<tr>
<td>Binomial</td>
<td>integer: $0, 1, \ldots, N$</td>
<td>count of &quot;yes&quot; occurrences out of N yes/no occurrences</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation
   4.1 (Steepest) Full Gradient Descent
   4.2 Gradient descent for logistic regression ML estimation
   4.3 Coordinate-wise Gradient Descent
   4.4 Stochastic Gradient Descent

5. More...
Parameter estimation

- For a given parametric model, e.g., logistic regression
- Given training data $X, y$
- Estimate weights $\hat{w}$ (includes intercept $w_0$!) that
  - Fit the training data well and
  - Generalize well to (yet unseen) test data

- This problem is called **parameter estimation**
Parameter estimation in our setting

- “Fit the training data well”
- For fixed training examples $X, y$,
- Goal: find setting $\hat{w}$ for weight parameter $w$ that maximizes the likelihood function $L$.
- Select $\hat{w}$ so that

$$\hat{w} = \arg \max_{w \in \mathbb{R}^{D+1}} L(y|X,w)$$

where $L(y|X,w)$ is the likelihood function of the model for given training examples $X, y$ and parameter settings $w$.

$$L(y|X,w) = \prod_{i=1}^{n} P(Y_i = y_i|x_i,w)$$
Many approaches to parameter estimation

- Many different approaches in ML literature focus on parameter estimation
  - in general
  - for different models
  - different data assumptions

- Close relationship to numerical optimization theory, where given an optimization criterion, optimal values are to be identified.

- For logistic regression we focus on gradient descent methods:
  - (Steepest) Gradient descent
  - Coordinate-wise gradient descent
  - Stochastic gradient descent
Outline

1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation
   4.1 (Steepest) Full Gradient Descent
   4.2 Gradient descent for logistic regression ML estimation
   4.3 Coordinate-wise Gradient Descent
   4.4 Stochastic Gradient Descent

5. More...
Continuous Gradient descent

- Find minimum $\theta^*$ of function $f$
- Pick a starting point $\theta(t = 0)$
- Compute gradient $f'(\theta(t = 0))$
- Over time $t$,
  walk downhill in tiny steps $\alpha \to 0$
  to obtain updates $\Delta t \to 0$

$$\theta(t + \Delta t) = \theta(t) - \alpha f'(\theta(t))$$

- Under certain conditions

$$\lim_{t \to \infty} \theta(t) \to \theta^*$$
Discrete gradient descent

- Find minimum $\theta^*$ of function $f$
- Pick a starting point $\theta^{(0)}$
- Compute gradient $f'(\theta)$
  (also denoted $\nabla f(\theta)$)
- In step $k$, jump downhill with step length $\alpha_k$
  in the direction of the gradient

$$
\theta^{(k+1)} = \theta^{(k)} - \alpha^{(k)} f'(\theta^{(k)})
$$

- Under certain conditions, approximates continuous GD
- $\alpha^{(k)}$ is also called the learning rate.
How to choose the step size?

Given a direction of the gradient $d^{(k)} = f'(\theta^{(k)})$

How to set step-size $\alpha^{(k)}$ in the $k$’th step?

- a) guess a step size, keep it constant.
- b) **Bold driver heuristic**: After every epoch
  - Increase step size slightly when loss decreased (by, say, 5%)
  - Decrease step size sharply when loss increased (by, say, 50%)
- c) optimize the step size directly (called line search).

\[
\hat{\alpha}^{(k)} = \arg \min_{\alpha \in \mathbb{R}} f(\theta^{(k)} - \alpha d^{(k)})
\]

- d) and further avoid zig-zagging with **Wolfe conditions**

\[
1) \quad f(\theta^{(k)} + \alpha^{(k)} d^{(k)}) - f(\theta^{(k)}) \leq c_1 \alpha^{(k)} \langle d^{(k)}, f'(\theta^{(k)}) \rangle \tag{6}
\]

\[
2) \quad \frac{\langle d^{(k)}, f'(\theta^{(k)} + \alpha^{(k)} d^{(k)}) \rangle}{\langle d^{(k)}, f'(\theta^{(k)}) \rangle} \geq c_2 \tag{7}
\]

for fixed constants $0 < c_1 < c_2 < 1$, e.g., ($c_1 = 10^{-4}$ and $c_2 = 0.9$)
Outline

1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation
   4.1 (Steepest) Full Gradient Descent
   4.2 Gradient descent for logistic regression ML estimation
   4.3 Coordinate-wise Gradient Descent
   4.4 Stochastic Gradient Descent

5. More...
Maximum likelihood estimation for logistic regression

- \( P(Y_i = 1 \mid x_i) = \frac{1}{1 + \exp(\langle w, x_i \rangle)} = \text{logistic}(\langle w, x_i \rangle) \)
- For \( n \) examples \( X, y \) the \textbf{likelihood function} is:

\[
\mathcal{L}(y \mid X, w) = \prod_{i=1}^{n} P(Y_i = y_i \mid x_i, w)
\]

- \( \ln \mathcal{L}(y \mid X, w) = \sum_{i=1}^{n} (y_i \ln P(Y_i = 1 \mid x_i, w) + (1 - y_i) \ln P(Y_i = 0 \mid x_i, w)) \)

Examples \((x,y)\): \([(0.6,1), (0.3,1), (0.2,0), (0.1,0)]\)
How to apply GD to logistic regression?

- How to apply gradient descent to parameter estimation for logistic regression?
- Parameter to minimize: \( \theta := w \)
- Function to minimize: \( f(\theta) := -\ln \mathcal{L}(y|X, w) \) with given \( X, y \).
  (we want to maximize \( \ln \mathcal{L}(y|X, w) \))
- What is the gradient of \( \ln \mathcal{L} \) with respect to \( w \)? (add ‘-’ later)
  - Differentiate \( \ln \mathcal{L} \)
    \[
    \frac{\partial \ln \mathcal{L}(y|X, w)}{\partial w_j} \quad \forall j : 0 \leq j \leq D
    \]
  - Gradient is the vector of component-wise derivatives
    \[
    \ln \mathcal{L}'(w) = \left( \frac{\partial \ln \mathcal{L}(y|X, w)}{\partial w_0}, \ldots, \frac{\partial \ln \mathcal{L}(y|X, w)}{\partial w_j}, \ldots, \frac{\partial \ln \mathcal{L}(y|X, w)}{\partial w_D} \right)
    \]
- Exercise 05: differentiation of \( \ln \mathcal{L} \)!
Intuition for Gradient of LogReg

\[
\frac{\partial \ln \mathcal{L}(y|X, w)}{\partial w_j} = \sum_{i=1}^{n} x_{ij} \left( \begin{array}{c}
(1 - y_i) - P(Y_i = 0|x_i, w) \\
given \text{class is false} \quad predicted \text{class is false}
\end{array} \right)
\]

\[
= \sum_{i=1}^{n} x_{ij} \left( 1 - y_i - (1 - P(Y_i = 1|x_i, w)) \right)
\] (8)

\[
= - \sum_{i=1}^{n} x_{ij} \left( y_i - P(Y_i = 1|x_i, w) \right)
\] (9)
Properties of the Gradient of LogReg

\[
\frac{\partial \ln L(y|X, \mathbf{w})}{\partial w_j} = - \sum_{i=1}^{n} x_{ij} \left( y_i - P(Y_i = 1|x_i, \mathbf{w}) \right)
\]

- Zero feature values \((x_{ij} = 0)\) do not add to the gradient.
- The more perfect the predictions for \(y_i\) the less they add to the gradient.
- choices of \(\mathbf{w}\) only act through the prediction \(P(Y_i = 1|x_i, \mathbf{w})\).
- ... which is the same for all feature dimensions \(j\).
Gradient of LogReg as matrix operation

\[
\frac{\partial \ln \mathcal{L}(y|X, w)}{\partial w_j} = - \sum_{i=1}^{n} x_{ij} \left( y_i - P(Y_i = 1|x_i, w) \right)
\]

given class is true

\[
\text{predicted class is true}
\]

same for all dimensions \(j\)

\[
\text{define } g_i(w) = -(y_i - P(Y_i = 1|x_i, w))) \quad \text{error for example } i
\]

\[
g(w) = (g_1(w), \ldots, g_i(w), \ldots, g_n(w)) \quad \text{vector of errors}
\]

\[
\frac{\partial \ln \mathcal{L}(y|X, w)}{\partial w_j} = \langle X_{*j}, g(w) \rangle \quad j\text{th column of } X \text{ times errors}
\]

\[
\mathcal{L}'(w) = \left( \frac{\partial \mathcal{L}(y|X, w)}{\partial w_0}, \ldots, \frac{\partial \mathcal{L}(y|X, w)}{\partial w_j}, \ldots, \frac{\partial \mathcal{L}(y|X, w)}{\partial w_D} \right)
\]

\[
= \left( \langle X_{*0}, g(w) \rangle, \ldots, \langle X_{*j}, g(w) \rangle, \ldots, \langle X_{*D}, g(w) \rangle \right)
\]

\[
= g(w)\top X
\]
Summary: Gradient descent for Log Reg

- Goal find minimum $\mathbf{w}^*$ of function $-\ln \mathcal{L}$
- Pick a starting point $\mathbf{w}(0)$
- Iterate for step $k \in \{0,1,\ldots\}$
- In step $k$, compute direction of gradient
  \[ \mathbf{d}(k) = -\ln \mathcal{L}'(\mathbf{w}(k)) = \mathbf{g}(\mathbf{w}(k))^\top \mathbf{X} \]
- Jump downhill with step length $\alpha(k)$ in the opposite direction of the gradient
  \[ \mathbf{w}(k+1) = \mathbf{w}(k) - \alpha(k) \mathbf{d}(k) \]

- Step size (optional: Wolfe conditions)
  \[ \alpha(k) = \arg \min_{\alpha} \left( -\ln \mathcal{L}(\mathbf{y} \mid \mathbf{X}, \mathbf{w}(k) - \alpha \mathbf{d}(k)) \right) \]
  \[ = -\ln \mathcal{L}(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = -\sum_{i=1}^n ((1 - y_i) \langle \mathbf{w}, \mathbf{x}_i \rangle - \ln(1 + \exp(\langle \mathbf{w}, \mathbf{x}_i \rangle))) \]
Outline

1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation
   4.1 (Steepest) Full Gradient Descent
   4.2 Gradient descent for logistic regression ML estimation
   4.3 Coordinate-wise Gradient Descent
   4.4 Stochastic Gradient Descent

5. More...
Coordinate-wise gradient descent

- Goal find min $w^*$ of function $-\ln \mathcal{L}$
- Starting point $w^{(0)}$; iterate steps $k$
- Choose a coordinate dimension $j \in \{0, 1, \ldots, D\}$
- Compute gradient for dimension $j$
  $$-\frac{\partial}{\partial w_j} \ln \mathcal{L}(y|X, w^{(k)})$$
- Compute direction $d^{(k)}$ which is zero everywhere but
  $$d_j^{(k)} = -\frac{\partial}{\partial w_j} \ln \mathcal{L}(y|X, w^{(k)})$$
  $$= -\langle X_{*j}, g(w^{(k)}) \rangle$$
- In step $k$, jump downhill along coordinate $j$
  $$w^{(k+1)} = w^{(k)} - \alpha^{(k)} d^{(k)}$$
- Use fixed (small) step size $\alpha^{(k)}$. 

\[ \theta(t) - \theta^* \]
Coordinate-wise “line-search” descent

- Does not need gradients!
- Goal find min $\mathbf{w}^*$ of function $-\ln \mathcal{L}$
- Starting point $\mathbf{w}^{(0)}$; iterate steps $k$
- Choose a coordinate dimension $j \in \{0, 1, \ldots, D\}$
- In step $k$, choose direction $j$
  
  Choose step size by line search along coordinate $\mathbf{e}_j$
  
  $\alpha^{(k)} = \arg\min_\alpha (-\ln \mathcal{L}(\mathbf{y} \mid \mathbf{X}, \mathbf{w}^{(k)} - \alpha \mathbf{e}_j))$

- Jump along coordinate $j$ with step length $\alpha^{(k)}$
  
  $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha^{(k)} \mathbf{e}_j$

- basis $\mathbf{e}_j = \begin{pmatrix} 0, \ldots, 0, & 1, & 0, \ldots, 0 \end{pmatrix}$

\[ \begin{align*}
\theta(t) - \theta^* &= \text{stepfun}(px, py) \\
\end{align*} \]
Outline

1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation
   4.1 (Steepest) Full Gradient Descent
   4.2 Gradient descent for logistic regression ML estimation
   4.3 Coordinate-wise Gradient Descent
   4.4 Stochastic Gradient Descent

5. More...
Stochastic gradient descent

- Goal Find minimum $\mathbf{w}^*$ of function $-\ln \mathcal{L}$
- Pick a starting point $\mathbf{w}^{(0)}$
- Iterate for step $k \in \{0, 1, \ldots \}$
- Compute approx direction of gradient
  $$\hat{\mathbf{d}}^{(k)} \approx -\frac{\partial}{\partial w_j} \ln \mathcal{L}(\mathbf{y}|\mathbf{X}, \mathbf{w}^{(k)})$$
- Jump “approximately” downhill
- Stochastic update
  $$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \alpha^{(k)} \hat{\mathbf{d}}^{(k)}$$
- Under certain conditions, asymptotically approximates (continuous) gradient descent
Stochastic gradient descent for LogReg

- Approximate full gradient using the gradient of only a single example \((x_i, y_i)\).

\[
\ln \hat{L}'(y | X, w) \approx n \cdot \ln \mathcal{L}'(y_i | x_i, w)
\]

- Compute the gradient components \(\frac{\partial}{\partial w_j} \ln \mathcal{L}(y_i | x_i, w)\) (See Exercise 05!).

- **SGD epoch**
  1. Pick a random example \(x_i, y_i\)
  2. Compute approximate gradient \(\ln \mathcal{L}'(y_i | x_i, w)\)
  3. Update parameters

\[
\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \alpha^{(k)} \ln \hat{L}'(y_i | x_i, \mathbf{w}^{(k)})
\]
  4. Repeat \(K\) times.
Comparison

- Per epoch, assuming $O(r)$ gradient computation per example

<table>
<thead>
<tr>
<th></th>
<th>GD</th>
<th>SGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Deterministic</td>
<td>Randomized</td>
</tr>
<tr>
<td>Gradient computations</td>
<td>1</td>
<td>$n$</td>
</tr>
<tr>
<td>Gradient types</td>
<td>Exact</td>
<td>Approximate</td>
</tr>
<tr>
<td>Parameter updates</td>
<td>1</td>
<td>$n$</td>
</tr>
<tr>
<td>Time</td>
<td>$O(rn)$</td>
<td>$O(rn)$</td>
</tr>
<tr>
<td>Space</td>
<td>$O((m+n)r)$</td>
<td>$O((m+n)r)$</td>
</tr>
</tbody>
</table>

- Why stochastic?
  - *Fast convergence* to vicinity of optimum
  - Randomization may help escape local minima
  - Exploitation of “repeated structure”
1. Bernoulli Log Odds

2. Logistic regression - Model

3. Generalization of logistic regression

4. Maximum Likelihood Parameter Estimation

5. More...
Parameter estimation – hopes and fears

• “Fit the training data well” is equivalent to selecting \( \hat{w} \) so that

\[
\hat{w} = \arg \max_{w \in \mathbb{R}^{D+1}} L(y|X, w)
\]

• Hope 1: Maximizing \( L \) will also minimize test error.
  ▶ Fear of overfitting when parameter space is high.
  ▶ Solution: Regularization (adding a penalty term) or adding a prior.

• Hope 2: Maximizing \( L \) informs about the best model.
  ▶ Fear that likelihood function does not agree with error function
  ▶ Solution: Minimize the error function instead of likelihood function. (called “risk minimization”)

• Hope 3: The model is appropriate to make predictions.
  ▶ Depends on the prediction task.
  ▶ Solution: Compare models with different model assumptions on validation data.
L2 Regularization

- A common approach to add robustness to the parameter estimation is to add **regularization** to the optimization objective.
- A wide-spread regularizer is the L2 norm of the parameter vector $\|w\|_2$, where hyperparameter $\lambda$ controls its strength.
- This changes the function to minimize to

\[- \ln \mathcal{L}(w) + \frac{\lambda}{2} \|w\|_2^2\]

- correspondingly the gradient of dimension $j$ changes to

\[- \left( \frac{\partial}{\partial w_j} \ln \mathcal{L}(y|X, w) \right) + \lambda w_j\]

- This is equivalent to using a Gaussian prior on $w$. Thus, optimization finds a MAP estimate of the parameter vector.
Lessons learned

- Logistic regression is a classification model, where it is assumed that log odds of $y$ are a linear function of the inputs $x$.

- Parameters $\mathbf{w}$ can be determined by maximizing the likelihood function $\mathcal{L}$ (or the log-likelihood $\ln \mathcal{L}$)

- Parameter estimation with gradient methods:
  - Compute the derivative $-\frac{\partial}{\partial w_j} \ln \mathcal{L}(\mathbf{y}|\mathbf{X}, \mathbf{w})$
  - Jump downhill, in opposite direction of gradient.
  - Choosing direction $\mathbf{d}^{(k)}$:
    - full gradient, coordinate-wise gradient, stochastic gradient.
  - Determine step-size $\alpha^{(k)}$:
    - constant, bold driver, line search, Wolfe conditions
  - Regularization of objective adds robustness.

- Logistic regression model is a special case of GLMs.
Suggested readings

- Koller, Section 5.4.2 “Generalized Linear Models”
- Murphy, Chapter 8 “Logistic Regression”