Machine learning problem formulation

- The machine learning approach expresses NLP as an optimization problem:

$$\hat{\mathbf{Y}} = \underset{\mathbf{y} \in f(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})}{\operatorname{argmax}} \; \Psi(\mathbf{Y}, \mathbf{X}; \boldsymbol{\theta})$$

where $\mathbf{x} \in X$ is the input

$\mathbf{y} \in Y$ is the output

$\Psi(\cdot) \rightarrow \mathbb{R}$ is a function expressing the learning objective

$f(\cdot)$ is the function, or model, that maps $\mathbf{x}$ to $\mathbf{y}$

$\boldsymbol{\theta}$ parameterizes $f(\cdot)$

Maximum likelihood estimation

- Estimates the parameters, $\boldsymbol{\theta}$, of a distribution using a likelihood function, $\mathscr{L}(\mathbf{D};\boldsymbol{\theta})$, given some data, ***D.***

- We maximize the likelihood function by minimizing its -logarithm:

$$\begin{aligned}\mathscr{L}(\boldsymbol{\theta}\,|\,\mathbf{D}) &= P(\mathbf{D};\boldsymbol{\theta}) \\ &= \left(\prod_{i=1}^{M} P(\mathbf{y}_i\,|\,\mathbf{x}_i;\boldsymbol{\theta})\right)^{\frac{1}{M}} \\ &= \frac{1}{M}\sum_{i=1}^{M} \log P(\mathbf{y}_i\,|\,\mathbf{x}_i;\boldsymbol{\theta})\end{aligned}$$

Note: technically natural log, but true to within a constant

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^{M} -\log P(\mathbf{y}_i\,|\,\mathbf{x}_i;\boldsymbol{\theta})$$

- This expresses an optimization problem; the form of $p(\mathbf{D};\boldsymbol{\theta})$ dictates how we solve it
  - In deep learning, this function is a neural network; we compute its gradient w.r.t. $\boldsymbol{\theta}$, and then estimate $\hat{\boldsymbol{\theta}}$ using stochastic gradient descent (SGD).