

Model Selection:

Regularization, Cross-Validation, Information Criteria

Fantastic models and ~~where~~ how to find them (Part Deux)

From Lecture 6

Fantastic models

- *Often, we are interested in creating high-quality but also “**generalizable**” models.*
- *We want to the right balance of complexity given the data available*
- *Train-test splits allow us to assess the quality of a model in a less biased manner*
- *Cross-validation allows us to identify model hyperparameters or report model quality in a less biased way*

Evaluating Model Complexity

In general, the model complexity should be justified by the data

Lex Parsimoniae

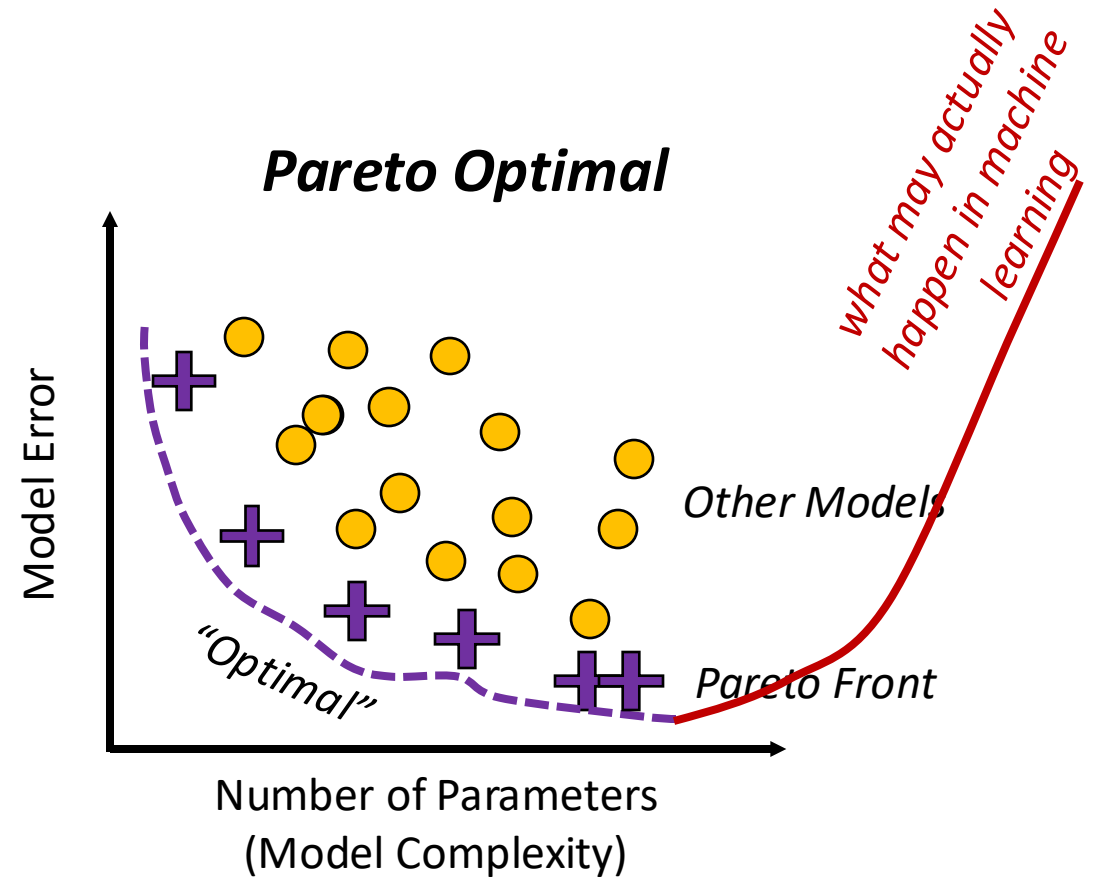
"Law of Parsimony"

Occam's Razor

"Given two machine learning models of (approximately) equal accuracy, it's better to choose the simpler one."

features can be
treated as a
hyperparameter

*Too many parameters for too little data can cause problems;
one way to reduce the number of parameters is to reduce the number of features*



Regularization

Another way to guide model selection in over/underdetermined problems is to employ **regularization**, which adds penalty terms to the loss function based on parameter values



pubs.acs.org/jcim

Article

Topology Automated Force-Field Interactions (TAFFI): A Framework for Developing Transferable Force Fields

Bumjoon Seo, Zih-Yu Lin, Qiyuan Zhao, Michael A. Webb, and Brett M. Savoie*

Lennard-Jones parameters are fit to minimize the following objective function:

$$\chi_{\text{LJ}}^2 = \omega_{\text{IE}} N_{\text{IE}}^{-1} \sum_i^{N_{\text{IE}}} (\text{IE}_{\text{QC},i} - \text{IE}_{\text{FF},i})^2 + \omega_{\epsilon} N_{\epsilon}^{-1} \sum_i^{N_{\epsilon}} (\epsilon_{\text{UFF},i} - \epsilon_{\text{FF},i})^2 + \omega_{\sigma} N_{\sigma}^{-1} \sum_i^{N_{\sigma}} (\sigma_{\text{UFF},i} - \sigma_{\text{FF},i})^2 \quad (6)$$

where the first summation corresponds to squared deviations of the force-field interaction energy (IE_{FF}) from the counterpoise corrected interaction energy (IE_{QC}) over all N_{IE} pairwise samples. The second summation corresponds to the L2 regularization of the Lennard-Jones energy parameters ($\epsilon_{\text{FF},i}$) with respect to the UFF reference values ($\epsilon_{\text{UFF},i}$), and the third summation corresponds to the L2 regularization of the Lennard-Jones atomic radii ($\sigma_{\text{FF},i}$) with respect to the UFF reference values ($\sigma_{\text{UFF},i}$). The latter terms in the objective

The Fourier coefficients are fit to minimize the residual between the quantum chemistry and force-field potentials for the constrained dihedral rotation according to the following objective function:

$$\chi_{\text{Fourier}}^2 = \sum_i \left(E_{\text{QC},i} - \sum_{\nu_j \notin \text{fit}} E_{\text{FF},i}(\nu_j) - \sum_{\nu_j \in \text{fit}} \sum_{k=1}^4 \frac{1}{2} V_{j,k} (1 + (-1)^{k+1} \cos(k\phi_{i,j})) \right)^2 + \omega_{\text{L2}} N_{\text{fit}}^{-1} \sum_{i,j \in \text{fit}} V_{i,j}^2 \quad (4)$$

where the index i runs over all scan configurations. $E_{\text{QC},i}$ is the single-point energy of the configuration. The second summation runs over all force-field terms that are not being fit (i.e., bonds, angles, unscanned dihedrals, electrostatics, and Lennard-Jones terms). The third summation runs over all dihedrals that share the scanned bond (i.e., $\nu_j \in \text{fit}$). $V_{j,k}$ are the dihedral-specific force constants, and $\phi_{i,j}$ is the angle of dihedral j in configuration i . The last summation is an L2 regularization of the average magnitude of the dihedral fit coefficients that reduces overfitting to noisy data. ω_{L2} is set to 0.1% of the range of the fit values (i.e., the difference between

usually regularization will penalize parameters from taking on large values or can be used to penalize deviations from reference values

Regularization

Another way to guide model selection in over/underdetermined problems is to employ **regularization**, which adds penalty terms to the loss function based on parameter values

Over-Determined System

$$\begin{array}{|c|} \hline X \\ \hline \end{array} \begin{array}{|c|} \hline \theta \\ \hline \end{array} = \begin{array}{|c|} \hline y \\ \hline \end{array}$$

e.g.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|\hat{y}(\theta) - y\|_2$$



$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|\hat{y}(\theta) - y\|_2 + \omega_1 \|\theta\|_1 + \omega_2 \|\theta\|_2$$

constraint on error

constraints on solution vector

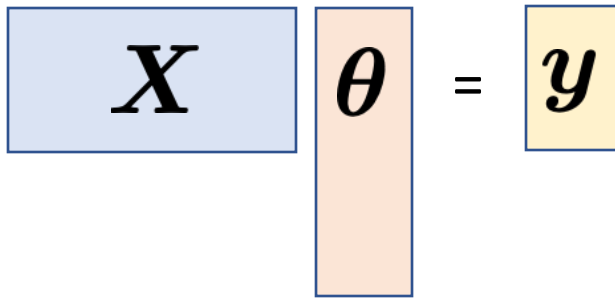
- constraints on solution vector can be *designed* (there are no rules!)
- nonetheless, penalizing magnitudes via norms is common

Regularization

Another way to guide model selection in over/underdetermined problems is to employ **regularization**, which adds penalty terms to the loss function based on parameter values

Under-Determined System

e.g.,


$$X \theta = y$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|\theta\|_p ; X\theta = y$$

- constraints on solution vector can be *designed* (there are no rules!)
- nonetheless, penalizing magnitudes via norms is common

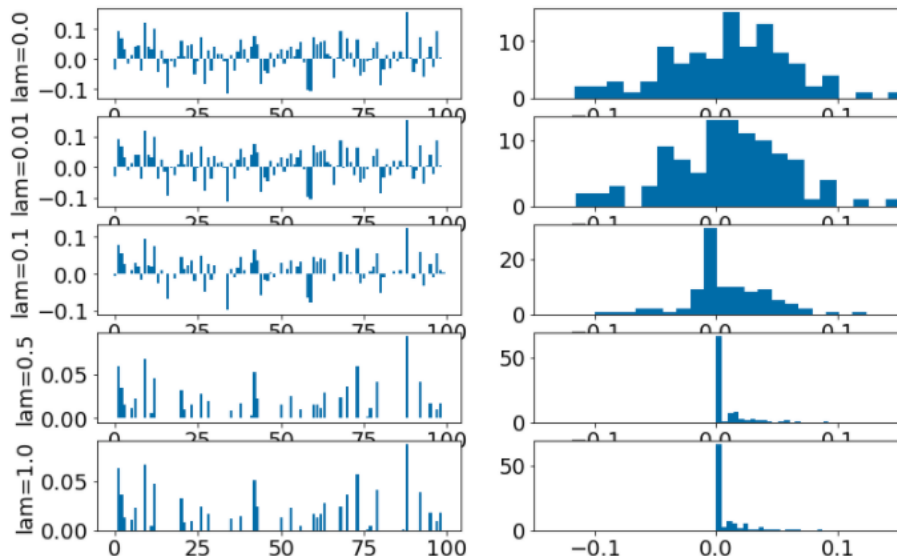
Regularization in optimization

L1 Regularization promotes solution sparsity

```
lam = np.array([0,0.01,0.1,0.5,1.])

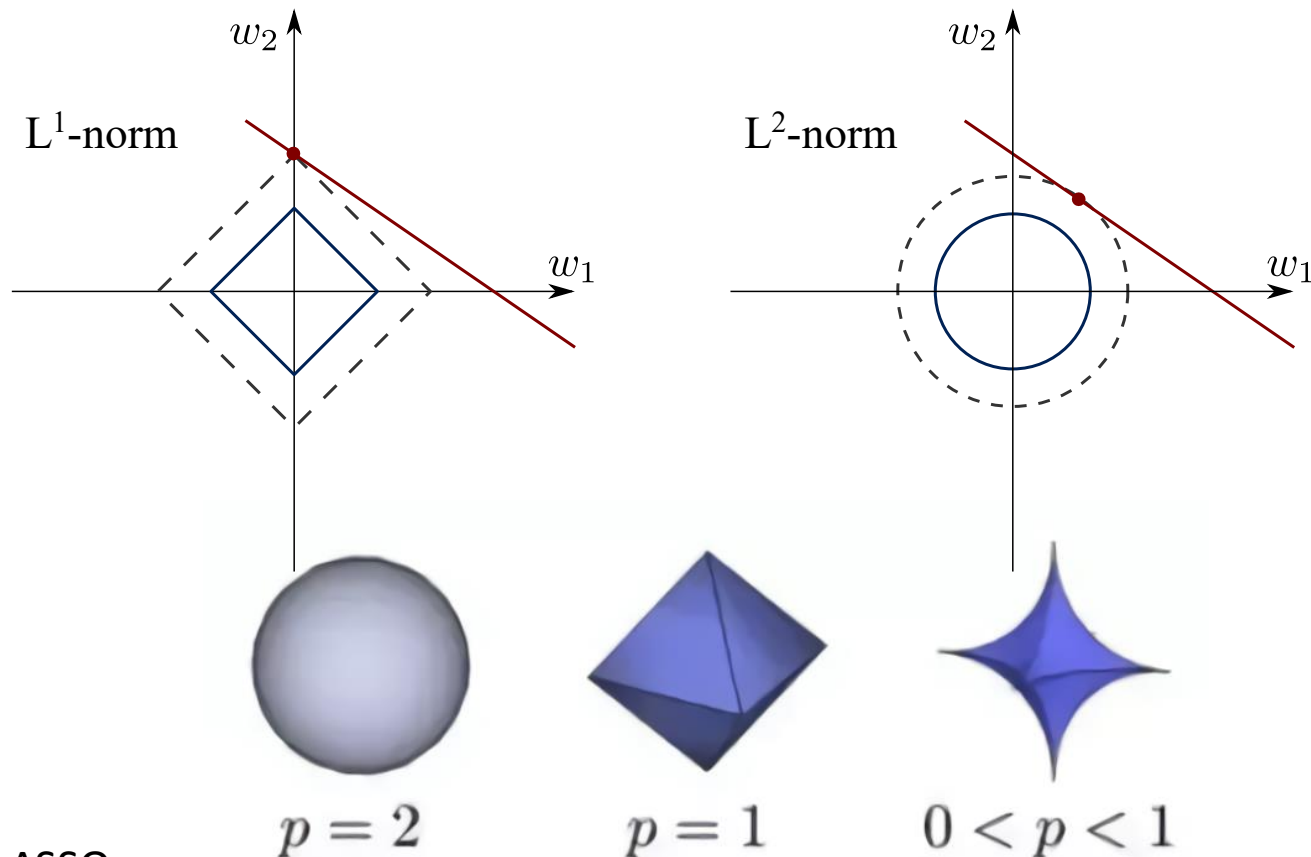
def reg_norm(x,A,b,lam):
    return np.linalg.norm(A*x-b,ord=2) + lam*np.linalg.norm(x,1)
### end solution

fig,axs = plt.subplots(len(lam),2)
for j in range(len(lam)):
    res = minimize(reg_norm,args=(A,b,lam[j]),x0=xdag)
    x = res.x
    axs[j,0].bar(range(m),x)
    axs[j,0].set_ylabel('lam='+str(lam[j]))
    axs[j,1].hist(x,20)
    axs[j,1].set_xlim(-0.15,0.15)
plt.show()
```



This type of regularization is more or less the premise of LASSO (least absolute shrinkage and selection operator) regression

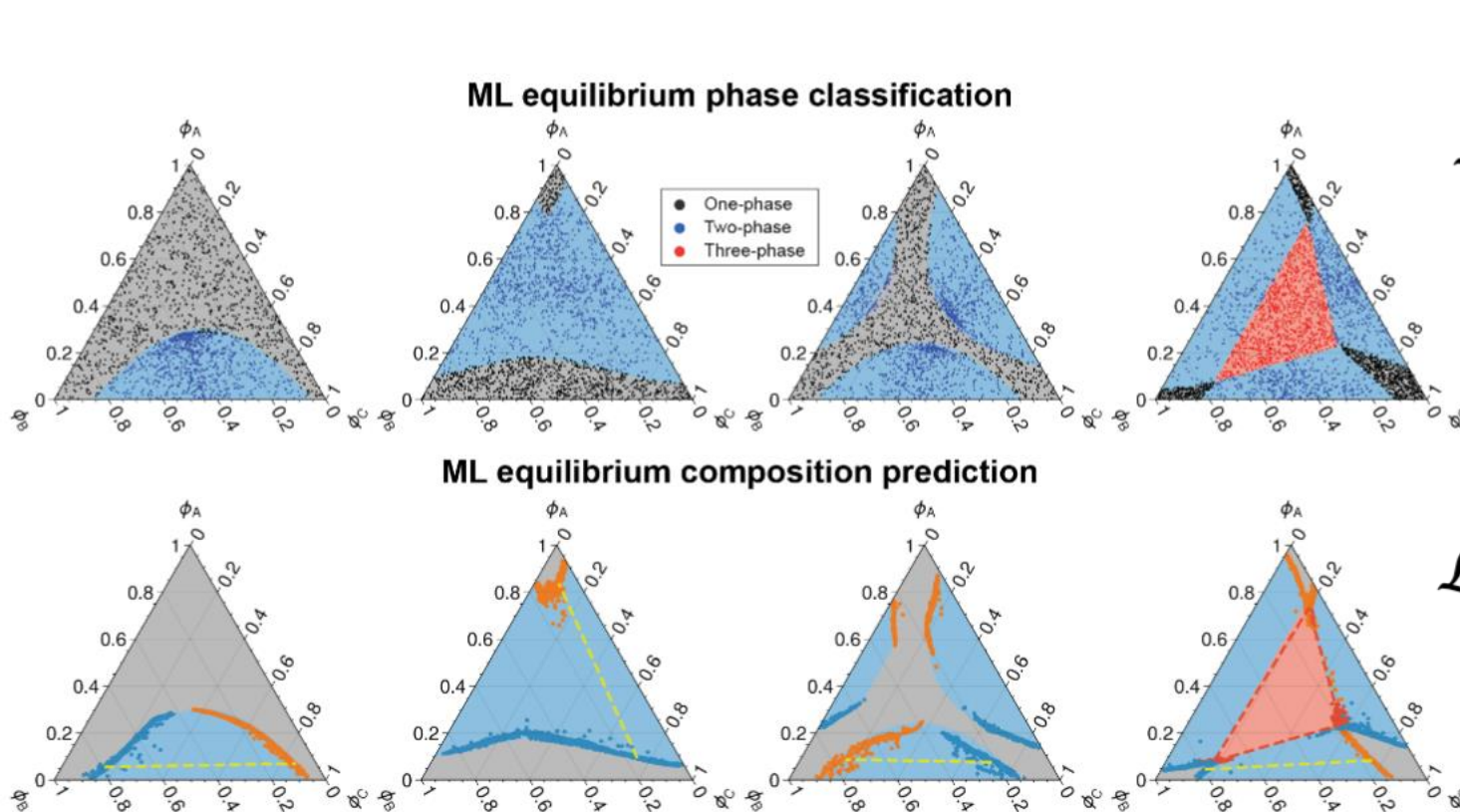
Why? We can understand based on shapes of norm surfaces...



this would be even more sparse but is not as easy to calculate

Regularization in **Physics-informed** Machine Learning

One strategy for selecting towards more *physical models* is to incorporate physical constraints/targets as *restraints in regularization*



$$\mathcal{L}_{PI} = \underbrace{\mathcal{L}_{base}}_{\text{data-based}} + \underbrace{\lambda_{split} \mathcal{L}_{split} + \lambda_{\Delta\mu} \mathcal{L}_{\Delta\mu} + \lambda_f \mathcal{L}_f}_{\text{physics-informed}}$$

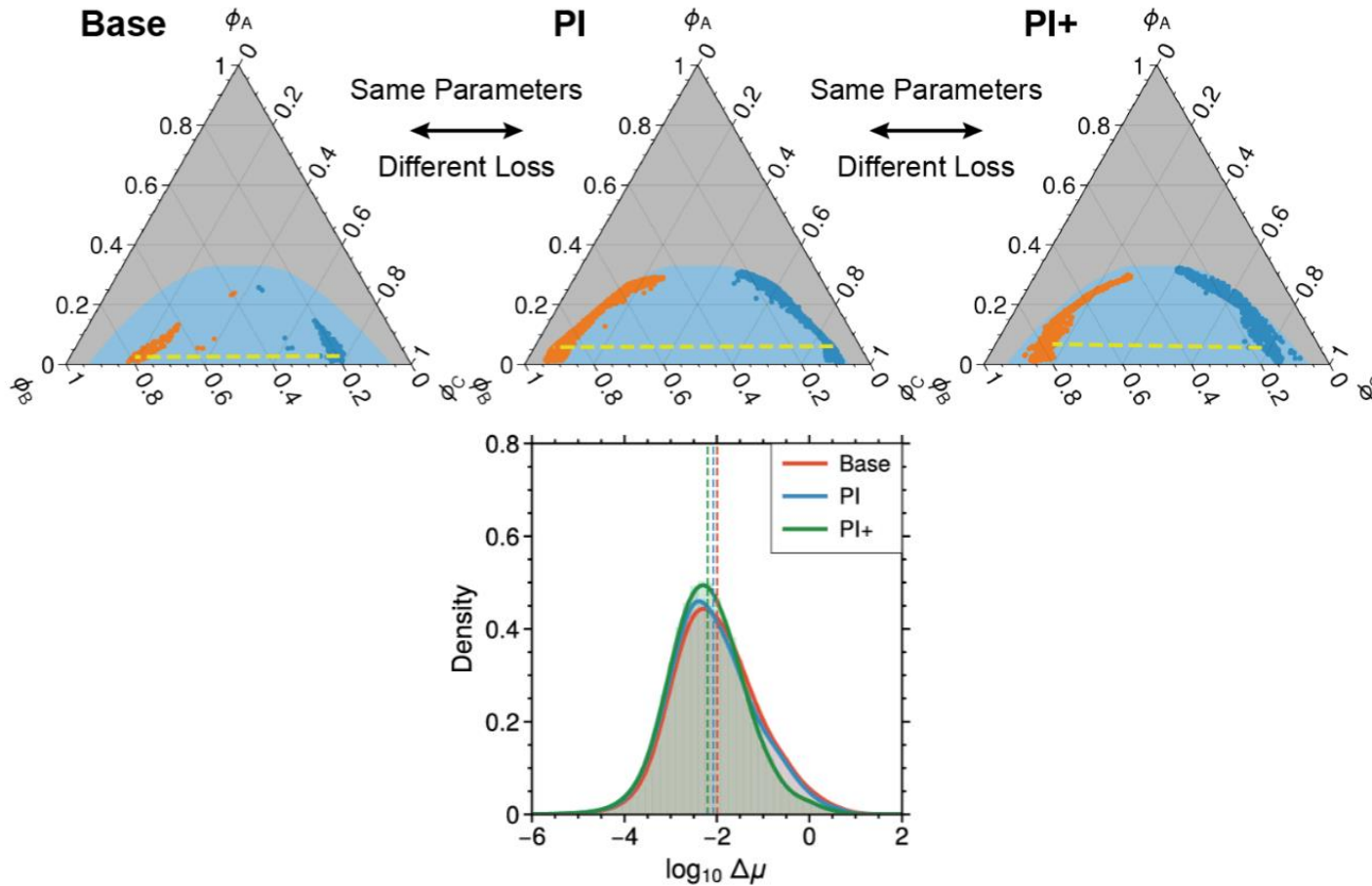
$$\mathcal{L}_{split} = \sum_{i \in \{A,B\}} \left(\phi_i - \sum_{\pi \in \{\alpha, \beta, \gamma\}} w^\pi \phi_i^\pi \right)^2$$

$$\mathcal{L}_{\Delta\mu} = \frac{1}{2} \sum_{\pi} \sum_{\pi'} \sum_{i \in \{A,B,C\}} \log \left(1 + (\Delta\mu_i^{\pi\pi'})^2 \right)$$

$$\mathcal{L}_f = \sum_{\pi} \sum_{i \in \{A,B,C\}} w^\pi \phi_i^\pi \mu_i^\pi$$

Regularization in **Physics-informed** Machine Learning

One strategy for selecting towards more *physical models* is to incorporate physical constraints/targets as *restraints in regularization*



$$\mathcal{L}_{PI} = \underbrace{\mathcal{L}_{base}}_{\text{data-based}} + \underbrace{\lambda_{split} \mathcal{L}_{split} + \lambda_{\Delta\mu} \mathcal{L}_{\Delta\mu} + \lambda_f \mathcal{L}_f}_{\text{physics-informed}}$$

$$\mathcal{L}_{split} = \sum_{i \in \{A, B\}} \left(\phi_i - \sum_{\pi \in \{\alpha, \beta, \gamma\}} w^\pi \phi_i^\pi \right)^2$$

$$\mathcal{L}_{\Delta\mu} = \frac{1}{2} \sum_{\pi} \sum_{\pi'} \sum_{i \in \{A, B, C\}} \log \left(1 + (\Delta\mu_i^{\pi\pi'})^2 \right)$$

$$\mathcal{L}_f = \sum_{\pi} \sum_{i \in \{A, B, C\}} w^\pi \phi_i^\pi \mu_i^\pi$$

Information Criteria

A primary goal of many machine learning regression problems is to achieve models that permit **generalization** (evaluation outside the domain of our dataset)

Another way to evaluate/discriminate against model complexity is to consider **Information Criteria**; typically these will penalize models that are unnecessarily complex given the evidence (data)

Some Flavors of Information Criteria

- **Kullback-Leibler divergence**

$$I(p, q) = \int p(\mathbf{x}, \boldsymbol{\theta}) \log \left[\frac{p(\mathbf{x}, \boldsymbol{\theta})}{q(\mathbf{x}, \boldsymbol{\theta})} \right] d\mathbf{x}$$

does not really address model complexity

- **Akaike Information Criterion**

$$\text{AIC} = 2m - 2 \log [\mathcal{L}(\boldsymbol{\theta}^* | \mathbf{y})]$$

*number of
parameters*

*likelihood
function*

- **Bayesian Information Criterion**

$$\text{BIC} = m \log(n) - 2 \log [\mathcal{L}(\boldsymbol{\theta}^* | \mathbf{y})]$$

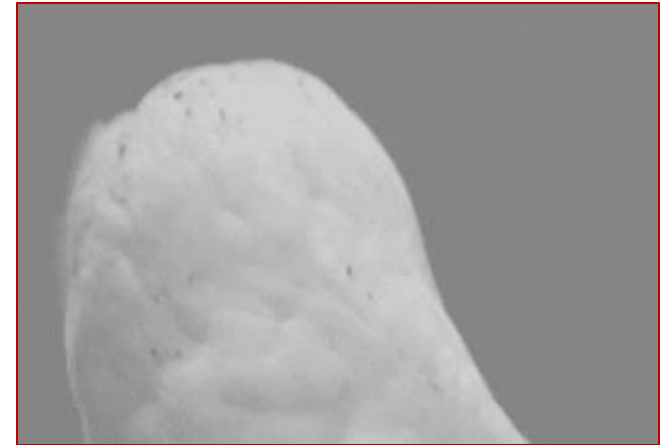
*number of
examples*

Least Squares and Maximum Likelihood Estimation

Under a particular set of assumptions... least squares estimation is equivalent to maximum likelihood estimation.

- linear model
- homoscedastic, normal, and independent errors

$$y_i | (x_{i1}, \dots, x_{ik}) \sim \mathcal{N}(\mathbf{x}^T \boldsymbol{\theta}, \sigma^2) \quad \forall \mathbf{x} \in \mathbb{R}^k$$



$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= \log(\phi(\mathbf{y}; \mathbf{X}\boldsymbol{\theta}, \sigma^2 \mathbf{I})) = \sum_{i=1}^n \log(\phi(y_i; \mathbf{x}_i^T \boldsymbol{\theta}, \sigma^2)) \\ &= -\log((2\pi)^{n/2} \sigma^n) - \frac{1}{2\sigma^2} \underbrace{(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})}_{\text{This is MSE}} \end{aligned}$$

Note: to maximize the likelihood, you would differentiate with respect to the parameters \rightarrow you don't even need to the noise/variance