Uncertainty Estimation and Sampling for Materials Data

Toby Francis

Carnegie Mellon University
College of Engineering

Follow Along

https://github.com/holmgroup/GB_Energy

What causes model uncertainty?

- Model Uncertainty: A consequence of statistics on finite datasets
  - $\gamma = f(\chi) | \chi \subseteq \mathbb{R}^n, \gamma \subseteq \mathbb{R}$
  - $\{x_i, y_i\} \in \{\mathrm{X}, \mathrm{Y}\} | \mathrm{X} \subset \chi, \mathrm{Y} \subset \gamma$
  - $h \in \mathcal{H} | h : \chi \rightarrow \gamma, h(\mathrm{X}) = h(\mathrm{Y})$
  - $|\mathrm{X}| = |\mathrm{Y}| < \infty \Rightarrow |\mathcal{H}| > 1$
  - **“All models are wrong”**
- Machine Learning: Pick the right $h \in \mathcal{H}$ given $\{\mathrm{X}, \mathrm{Y}\}$
  - Uncertainty: $|\hat{y}_n = \mathcal{H}(x_n)|$ (the number of possible outputs at x_n)

How can we use uncertainty?

- Model Criticism
  - Confidence Intervals
  - Hypothesis Testing
- Data selection
  - Active Learning

Active Learning with Uncertainty Sampling

- Central Concept:
  - Picking new points that the model is currently most uncertain about
- Practically:
  - Two cases
    - Finite pool of unlabeled points (e.g. images, possible configurations, etc.)
    - Bounded space to explore (e.g. crystallographic space)
  - Two different algorithms
    - Pools: evaluate uncertainty of all points in unlabeled pool, pick argmax
    - Bounded space: Many different algorithms
      - Draw i.i.d. samples and evaluate uncertainty
      - Monte Carlo search, Greedy optimization

How do we approximate model uncertainty?

- Backing-out Uncertainty
  - Ling et. al.
  - Committee Disagreement
  - Distance to Hyperplane (SVM)
  - Dropout Monte Carlo
  - Constant Learning-Rate SGD
- Building-in Uncertainty
  - Gaussian Processes

“Well-Calibrated Uncertainty Estimates”

- For new points:

$$\sigma_i^2(\mathbf{x}) = \mathrm{Cov}_j[n_{i,j}, t_j(\mathbf{x})]^2 + [\bar{t}_{-i}(\mathbf{x}) - \bar{t}(\mathbf{x})]^2 - \frac{ev}{T}$$
$$= \frac{ev}{T}$$

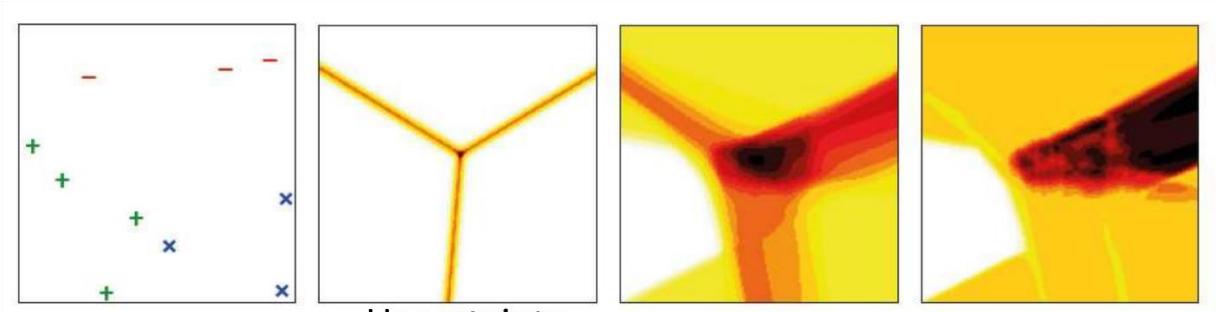
  - The “jackknife-after-bootstrap” statistical work was to provide uncertainty estimates on points that were trained on
  - Useful, but not necessary in the active setting

Committee Disagreement

- Basic idea:
  - If we have an ensemble, we can quantify the disagreement at a new point
- Mathematically (for classification)

$$D_{KL} = \frac{1}{|C|} \sum_{\theta \in C} \sum_{i} P_\theta(y_i|x) \log \frac{P_\theta(y_i|x)}{P_C(y_i|x)}$$

- Pros:
  - Committee Disagreement tends to be less "myopic" than uncertainty – models are "uncertain" about borders but "disagree" about large unlabeled regions



Training data | Uncertainty Sampling (entropy) | QBC using bagging (Soft Vote Entropy) | QBC using bagging (KL Divergence)

Gaussian Processes

- Why do we use them?
  - Provides a distribution over parameter-space, from which we can approximate the posterior distribution
- How do we build them?
  - Libraries: Edward, GPFlow, GPy
- What are the limitations?
  - Scalability

Gaussian Processes

- Given:
  - $\gamma = f(\chi) + \mathcal{N}(0, \sigma^2) | \chi \subset \mathbb{R}^n, \gamma \subset \mathbb{R}$
  - $\{x_i, y_i\} \in \{\mathrm{X}, \mathrm{Y}\} | \mathrm{X} \subset \chi, \mathrm{Y} \subset \gamma$
- Approximate: $\begin{bmatrix} Y \\ y_* \end{bmatrix} \sim \mathcal{N}(\mu, \begin{bmatrix} \Sigma, \Sigma_*^{\mathrm{T}} \\ \Sigma_*, \Sigma_{**} \end{bmatrix})$ $\Sigma = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$
  - k is our kernel
  - $k(x_1, x_2) = \sigma_f^2 \exp[\frac{-(x_1 - x_2)^2}{2l^2}] + \sigma_n^2 \delta(x_1, x_2)$
- "Formally, a Gaussian process generates data located throughout some domain such that any finite subset of the range follows a multivariate Gaussian distribution."

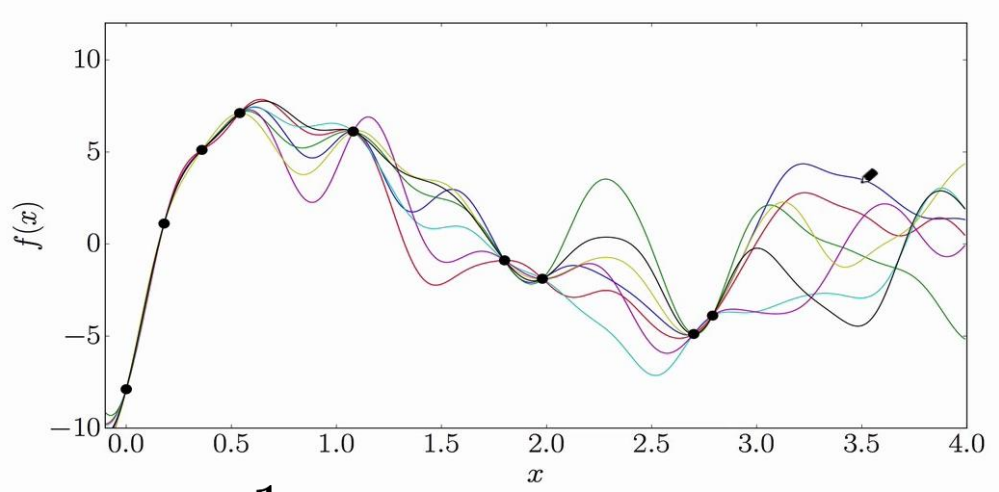
Gaussian Processes

- Infer:

$$p(y_*|X,Y,x_*) \sim \mathcal{N}(\Sigma_*\Sigma^{-1}Y, \Sigma_{**} - \Sigma_*\Sigma^{-1}\Sigma_*^{\mathrm{T}}$$

$$\Sigma_* = [k(x_*,x_1), k(x_*,x_2)...k(x_*,x_*)]$$

$$\Sigma_{**} = k(x_*,x_*)$$



$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}(y|x,\theta) = \underset{\theta}{\operatorname{argmax}}[-\frac{1}{2}y^{\mathrm{T}}K^{-1}y - \frac{1}{2}\log|K| - \frac{n}{2}log2\pi]$$

$$\theta = \{l, \sigma_f, \sigma_n\}$$

https://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf

Questions?
Concerns?
Let's set you up with GPflow.