Interpretable Machine Learning Exercises

Motivation

1. [Past Experience] The need for interpretable machine learning arises across many areas of study. This question asks you to share your experience with some of these areas.

   - Introduce yourself to your neighbors. What is your name and degree program? What are your areas of interest? How might interpretability or explainability be helpful in the work that you do?
   - Have you encountered a situation in the past where you think interpretability or explainability would be useful? Describe the situation and your overall goals. How did you address them and are there any ways the approach could have been improved?

2. [Jargon-free summary] Imagine you are working as a statistical consultant and have chosen to include a model explanation technique from this week. Your clients are familiar with the basics of machine learning (e.g., supervised vs. unsupervised learning) but not the interpretability literature. Give a jargon-free but precise description of {filled in during class}. What does it output and what is the correct interpretation?

3. [Example Critique] Summarize example {given in class} from {this week's reading}. What kinds of real-world situations is that example supposed to reflect? Do you think the proposed method would actually work in those situations, and what kinds of difficulties can you imagine arising in practice?

4. [Clear/Confusing] For this week's material, describe,

   - A concept that you learned for the first time.
   - A check you can apply to ensure your interpretations are reliable.
   - A point that are not confident you could apply in practice.

   Be as specific as possible.

Intrinsically Interpretable Models

1. [Linear Model Interpretability] A researcher uses a sparse linear regression to predict how a patient will respond to a new drug. The model uses the expression levels of 5K genes as potential features. After training, the model selects 40 genes with non-zero coefficients.

   a. The model reduced the features from 5K to 40. According to the definitions given in Murdoch et al. (2019) or Lipton (2018), is this model simulatable?

   b. In these data, many genes belong to the same molecular pathways. They are therefore highly correlated. How does this affect the modularity of the resulting model?

2. [Interpreting Lasso Regularization Paths] We will investigate how regularization strength affects model selection and prediction error in a concrete example. The code below predicts log ozone concentration from weather features using the lasso objective:

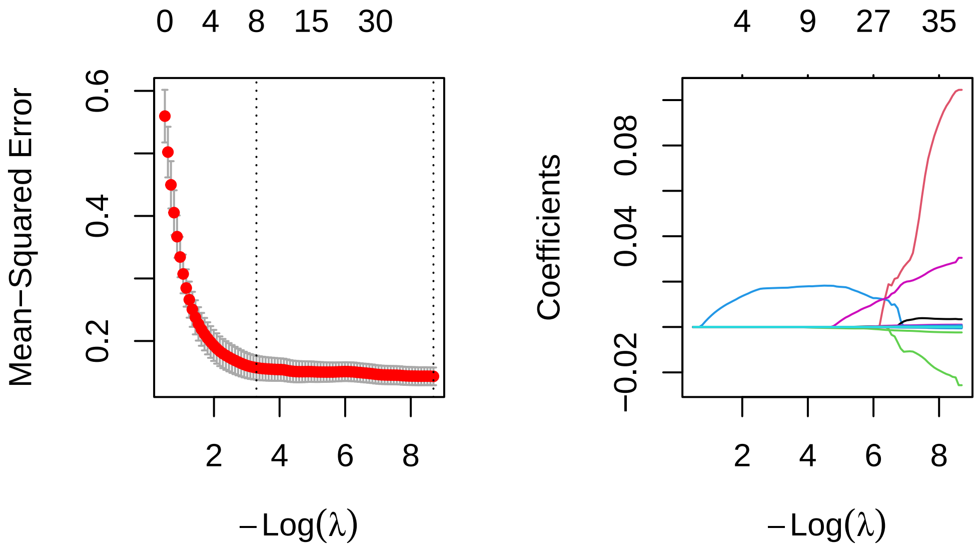
$$\min_{\beta_0,\beta\in\mathbf{R}^{45}}\left\{\frac{1}{2N}\sum_{n=1}^{N}(y_n-\beta_0-x_n^T\beta)^2+\lambda\|\beta\|_1\right\}$$

   where $N = 330$ samples, $y_n = \log(\text{ozone})_n$, and $x_n \in \mathbb{R}^{45}$ contains weather features and all pairwise interactions. Figure **??** displays cross-validation error (left) and coefficient paths (right) as functions of λ.

```r
library(faraway)
library(glmnet)
data(ozone)

# create features using first-order interactions
x <- model.matrix(O3 ~ .^2, data = ozone)[, -1]
cv_fit <- cv.glmnet(x, log(ozone$O3), alpha = 1)

# plot the results
par(mfrow = c(1, 2))
plot(cv_fit)
abline(v = log(cv_fit$lambda.1se), col = "blue", lty = 2)
plot(cv_fit$glmnet.fit, xvar = "lambda", label = TRUE)
```



```r
par(mfrow = c(1, 1))
```

a. Identify the two vertical dashed lines in the left panel. Which corresponds to λ_{min} and which to λ_{1se}? When should you prefer λ_{1se} over λ_{min}?

b. Explain the bias-variance tradeoff visible in these plots. As $-\log\lambda$ increases (i.e., λ decreases), how do model bias and variance change? Point to specific features in both panels that support your answer.

c. Explain the non-monotonic coefficient path.

- The blue trajectory in the right panel initially increases in magnitude, but near $-\log\lambda \approx 5$ it *decreases*, even as the penalty weakens.
- Why might this occur? What does it suggest about relationships between this predictor and others?

d. Identify the limiting behavior. As $-\log\lambda \to \infty$ (equivalently $\lambda \to 0$), the coefficient paths converge. What estimator do they approach?

3. [Cross-Validation for Lasso] Consider the lasso estimator

$$\min_{\beta_0,\beta}\left(\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda\|\beta\|_1\right)$$

where $\lambda \geq 0$ is the sparsity penalty. In this problem, you will implement cross-validation to guide selection of λ.

a. You have asked your colleagues to help prepare pseudocode for your cross-validation study. Are their designs correct? If not, how would it affect your later data analysis. Explain your reasoning.

Implementation 1:

```
Input: data (X, y), folds {F1, ..., FK}, lambda_grid

for each lambda in lambda_grid:
    fit lasso on (X, y) using lambda
    store coefficients beta_hat[lambda]

for k = 1 to K:
    X_val, y_val = data in fold Fk
    for each lambda in lambda_grid:
        y_pred = predict(X_val, beta_hat[lambda])
        error[k, lambda] = mean_squared_error(y_val, y_pred)

cv_error[lambda] = average_k error[k, lambda]
lambda_star = argmin_lambda cv_error[lambda]
```

Implementation 2:

```
Input: data (X, y), folds {F1, ..., FK}, lambda_grid

for k = 1 to K:
    X_train, y_train = data not in fold Fk
    X_val, y_val = data in fold Fk

    for each lambda in lambda_grid:
        fit lasso on (X_train, y_train) using lambda
        y_pred = predict(X_val)
        error[k, lambda] = mean_squared_error(y_val, y_pred)

cv_error[lambda] = average_k error[k, lambda]
lambda_star = argmin_lambda cv_error[lambda]
```

b. Implement K-fold cross-validation for the lasso from scratch. You may use existing packages to fit the lasso at a single choice of λ. Your code should take X, y, K, and a grid of λ as input. It should return the mean cross-validated error at each λ, as well as an optimal λ_{min}.

c. How would you modify your implementation if you wanted to identify λ_{1se}? A new code implementation is not necessary, but explain your reasoning. How does using λ_{1se} relate to the interpretability criteria of sparsity and simulatability?

4. PCS - feature engineering and model comparison

5. [Prediction stability plots] In this exercise, you will investigate prediction stability for lasso regression under data perturbations. You will consider the `prostate` dataset, which reports prostate specific antigen levels along with other lab measurements among men about to undergo prostatectomy. You will compare predictive accuracy and stability across different modeling choices.

   a. Sketch (by hand) the prediction stability plot associated with a model with low variance but high bias in its predcitions. Repeat the exercise for a model with high variance but low bias.

   b. Is it possible for a model to have stable predictions but unstable coefficients? Explain why or why not.

   c. The code below fits a lasso model with to the prostate dataset.

```r
library(glmnet)

x <- model.matrix(lpsa ~ ., data = prostate)[, -1]
y <- prostate$lpsa

cv_fit <- cv.glmnet(x, y, alpha = 1)
fit <- cv_fit$glmnet.model
```

Select 20 samples as a held-out test set. Repeat the following steps $B = 100$ times:

- Apply bootstrap resampling to the training data
- Refit the lasso on each bootstrap, using CV to tune λ each time
- Make predictions on the 20 test points.

Create the associated prediction stability plot.

   d. Repeat the steps above with ridge regression, which can be run using the following code:

```r
cv_fit <- cv.glmnet(x, y, alpha = 0)
fit <- cv_fit$glmnet.model
```

Compare the models with respect to prediction accuracy, prediction stability, and the sparsity interpretability criterion.

6. [Prior and Posterior Predictive Distributions] Study how observing data transforms the distribution over functions in a GP model. Consider a hierarchical GP prior where kernel hyperparameters are themselves random:

$$A \sim \mathrm{LogNormal}(-1, 1)$$
$$\ell \sim \mathrm{LogNormal}(0, 1)$$
$$f \mid A, \ell \sim \mathrm{GP}\left(0, ; k_{\mathrm{RBF}}(x, x')\right), \quad k_{\mathrm{RBF}}(x, x') = A^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right).$$

a. Interpret the hyperparameter priors.

The **greta.gp** syntax below implements this model. For each of amplitude, lengthscale, and sigma: what is the effect of increasing the first argument (the log-scale mean) in the lognormal call?

```r
set.seed(479)
amplitude <- lognormal(-1, 1)
lengthscale <- lognormal(0, 1)
sigma <- lognormal(0, 1)
k <- rbf(lengthscale, amplitude)
f <- gp(x_obs, k)
```

b. Sample from the prior predictive distribution. Specifically, draw 50 function realizations $f^{(i)} \sim p(f)$ by marginalizing over hyperparameters: sample $(A^{(i)}, \ell^{(i)})$, then sample $f^{(i)}$ from the corresponding GP. Then plot these samples over a grid covering the domain of interest.

Hint: The code sketch below gives a prior draw.

```r
f <- gp(x_obs, k) # GP at observed locations
f_grid <- project(f, x_grid) # GP at plotting grid
prior_samples <- calculate(f_grid)
```

c. Condition on the observed data. First download the dataset $\mathcal{D} = (x_n, y_n)_{n=1}^{100}$ from this link. The data are plotted in @observed-data. Note the heterogeneous sampling density. Then draw 50 samples from the posterior predictive distribution $p(f \mid \mathcal{D})$ using the same hierarchical prior.

```r
library(tidyverse)
theme_set(theme_classic())

x_obs <- c(
  runif(50, 1, 3), # dense
  runif(25, -1, 1), # medium
  runif(25, 7, 12) # sparse
)
```

```
data_obs <- tibble(x = x_obs) |>
  mutate(y = sin(x) + 0.8 * cos(0.5 * x + pi / 2) + rnorm(n(), 0, 0.2))
ggplot(data_obs, aes(x, y)) +
 geom_point()
```

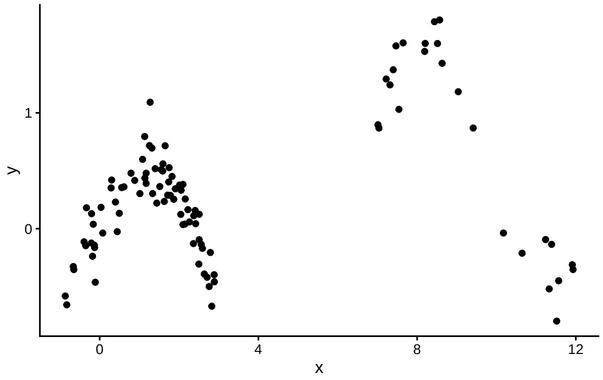


Figure 1

d. Overlay the samples from (b) and (c) in separate panels (compare with Figure **??**). Describe how the posterior differs from the prior: where does uncertainty concentrate? How does sampling density in $\mathcal{D}$ affect posterior behavior?

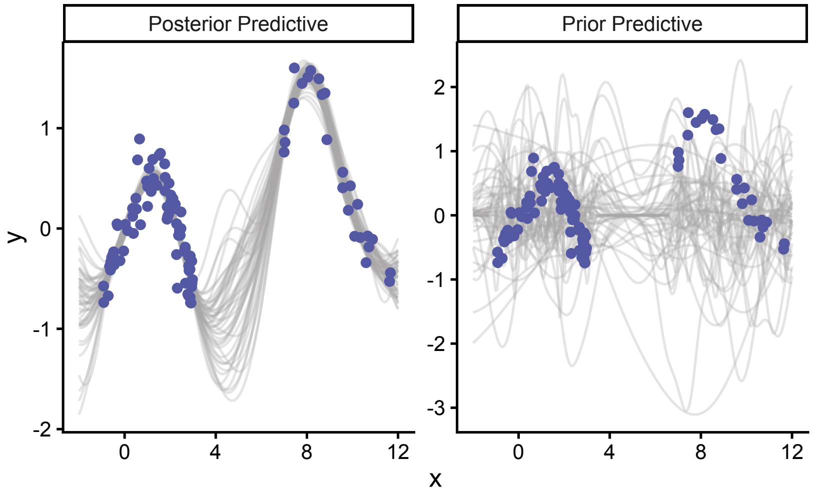


Figure 2

7. [GPs Marginal Likelihood]. The marginal likelihood surface of a GP can exhibit multiple local maxima corresponding to qualitatively different fits.

   a. The code below computes the log marginal likelihood for given hyperparameters.

Write the formula for K_{ij} and identify the kernel.

```r
compute_ll <- function(length_scale, sigma_y, sigma_f=1) {
    K <- sigma_f^2 * exp(-0.5 * outer(x, x, "-")^2 / length_scale^2) +
        sigma_y^2 * diag(length(x))

    L <- chol(K)
    alpha <- backsolve(L, forwardsolve(t(L), y))
    -0.5 * sum(y * alpha) - sum(log(diag(L))) - 0.5 * length(x) * log(2 * pi)
}
```

b. Figure Figure 3(a) shows the log marginal likelihood over (l, σ_y). Panels in Figure 3(b) show posteriors at local maxima A and B. Which panel corresponds to A, and which to B?

c. Using the data and hyperparameters below, compute both GP posteriors and replicate Figure 3(b).

```r
x <- c(-1.3089, 6.7612, 1.0553, -1.1734, -2.9339, 7.2530, -6.5843)
y <- c(1.6218, 1.8558, 0.4102, 1.2526, -0.0133, 1.6380, 0.2189)
x_test <- seq(-7.5, 7.5, length.out = 201)
df <- data.frame(x = x, y = y)

paramsA <- list(length_scale = 12.4, sigma_y = 0.62)
paramsB <- list(length_scale = 1.2, sigma_y = 0.23)
```

d. In practice, how would you choose between option (A) and (B)?

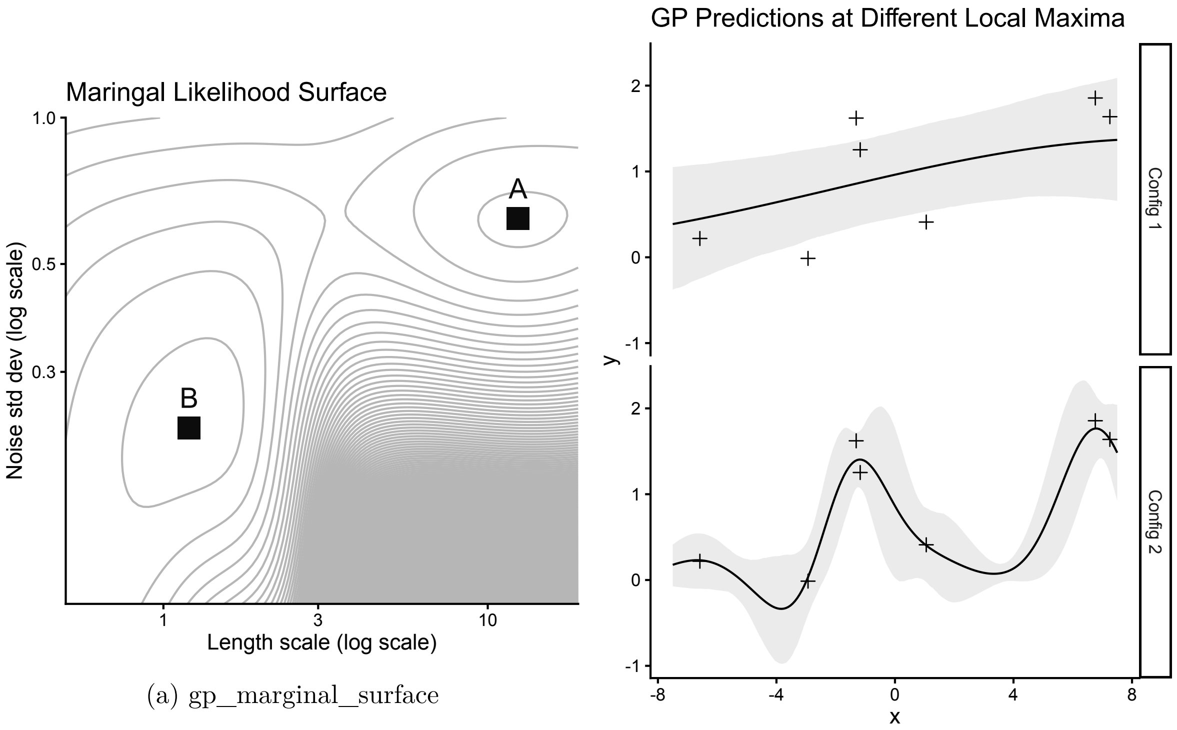
8. [Partition $\iff$ Tree] Any decision tree corresponds to a partition of the input space. This is especially easy to visualize in two dimensions.

   a. For the partition in Figure 4(a), sketch the corresponding decision tree.

   b. For the tree in Figure 4(b), sketch the corresponding partition.

9. [Kernels and GP Samples] A Gaussian Process on a set X is determined by its covariance kernel $k : X \times X \to \mathbb{R}$. The kernel encodes structural assumptions: smoothness, periodicity, polynomial growth. Since sums and products of positive definite kernels remain positive definite, one builds complex priors compositionally.

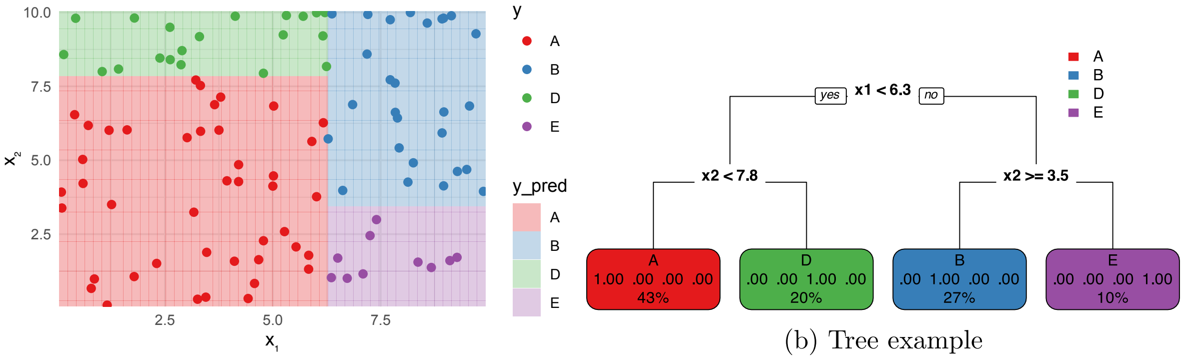
   Investigate this compositional structure through explicit computation and sampling. See Figure 5 and Figure 6 for reference—your results will differ due to hyperparameter choices.



(a) gp_marginal_surface

(b) gp_marginal_fits

Figure 3: Two local maxima in the marginal likelihood for problem [GP Marginal Likelihood].



(b) Tree example

(a) Partition example

Figure 4: Tree and partition for the problem [Partition $\iff$ Tree].

a. Select two kernels from [Linear, RBF, Periodic]. State the formula for each kernel $k(x, x')$ and interpret each hyperparameter geometrically or analytically. Fix hyperparameters and plot the function $x \mapsto k(x, 1)$ for each kernel. Explain: what does this cross-section $k(\cdot, x_0)$ represent?

b. Construct sum and product kernels from your choices in (a):

$$k_{\text{sum}}(x, x') = k_1(x, x') + k_2(x, x'), \quad k_{\text{prod}}(x, x') = k_1(x, x') \cdot k_2(x, x')$$

Plot $x \mapsto k_{\text{sum}}(x, 1)$ and $x \mapsto k_{\text{prod}}(x, 1)$.

c. Sample function realizations from GPs with the four kernels (the two base kernels and their sum and product). For each:

- Draw 3–5 sample paths.
- Identify which properties (smoothness, oscillation, trend) are induced by the kernel structure.
- Explain how composition (sum vs. product) combines these properties.

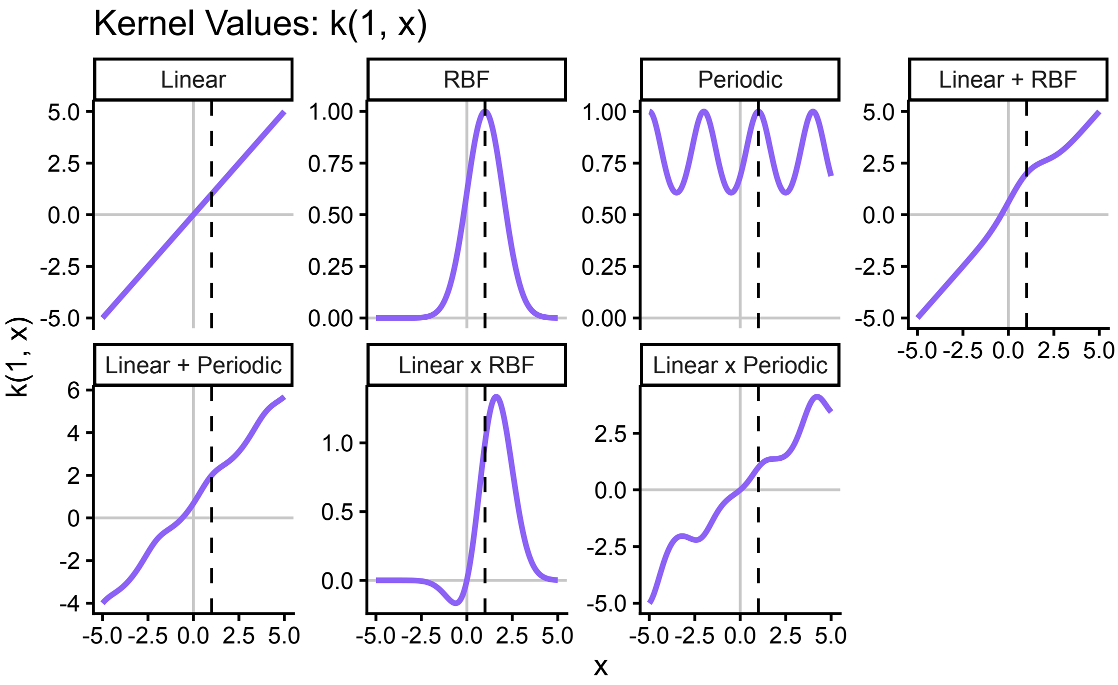


Figure 5

Hint: You may find the code sketch below helpful.

```
x_grid <- seq(-5, 5, length.out = 200)
k_lin <- linear(#...
```

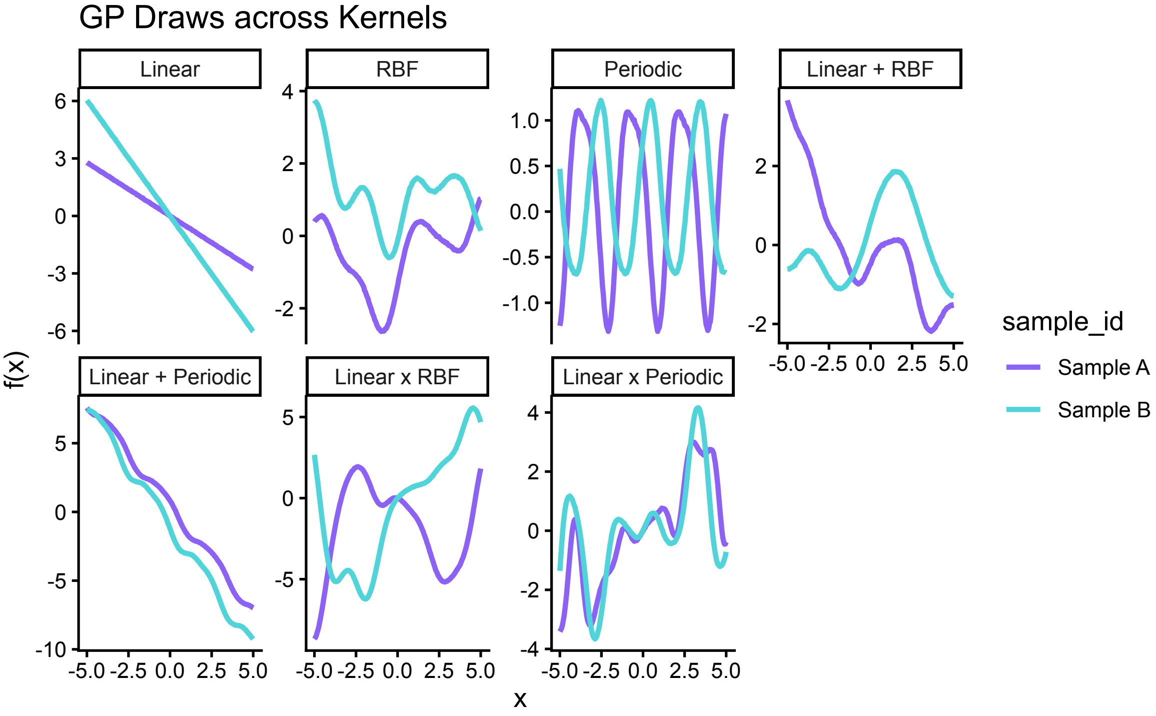

GP Draws across Kernels
Linear
RBF
Periodic
Linear + RBF
Linear + Periodic
Linear x RBF
Linear x Periodic
f(x)
x
sample_id
Sample A
Sample B
6
3
0
-3
-6
4
2
0
-2
1.0
0.5
0.0
-0.5
-1.0
2
0
-2
5
0
-5
-10
5
0
-5
4
2
0
-2
-4
-5.0 -2.5 0.0 2.5 5.0

Figure 6

```
kernel_values <- calculate(#...

f <- gp(x_grid, kernel = k_lin)
draws <- calculate(f, #...
samples_matrix <- draws$f[,,1]
```

10. [Tree Importances] In your own words, summarize how one of these techniques measures variable importance in tree models: Permutation VI, MDI, MDI+. For your chosen approach, what is one strength and one weakness?

11. [Split Choice Visual Explanation] Develop a visual explanation of the following excerpt from our reading (equations 8.2 - 8.3 in ISLR). Make sure to explain how the mathematical symbols relate to the parts of your visual explanation. Precisely describe any simplifying assumptions to make the figure clearer.

    For any j and s, we define the pair of half-planes

    $$R_1(j,s) = \left\{X \mid X_j < s\right\} \text{ and } R_2(j,s) = \left\{X \mid X_j \geq s\right\},$$

    and we seek the value of j and s that minimize the equation

    $$\sum_{i:x_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2.$$

12. [Intrinsic Interpretability Portfolio] The goal of this exercise is to give you a chance to experiment with two intrinsic interpretability techniques in a problem of your choice. The most important aspect of the exercise is that you have a chance to get hands-on experience with the topics discussed in this course, so that you can develop a more critical eye towards what does and does not work well in a topic that you care about.

    a. **Problem Formulation**. Describe the problem context, a representative dataset, and why intrinsic interpretability is important. What types of people would be the main beneficiaries of effective interpretation/explanation, and what exactly would they be able to get out of it? Search for related literature applying interpretable ML methods to this type of data and briefly summarize the main approaches you identify.

    b. **Method 1 Application**. Train an intrinsically interpretable model to the dataset you described above. Report the out-of-sample prediction accuracy and an interpretability-related output (e.g., random forest variable importance plot or GP posterior samples).

    c. **Method 2 Application**. Repeat the previous part with the a new intrinsically interpretable model.

d. **Discussion**. Compare the results from the two methods. Which conclusions are stable and which are not, and can you offer potential reasons for differences? What would you share in a report to one of the beneficiaries you identified above? Include 1 - 2 figures resulting from your analysis and summarize how to read them.

e. **Sanity Check**. Prepare a follow-up check to evaluate the reliability of one of the models. It is okay if the results raise potential problems, just make sure to discuss how they affect the takeaways. For example, you may:

- Repeat the analysis with alternative hyperparameters to see that the key conclusions are robust.
- Permute labels in a labeled dataset to serve as a no-signal control.
- Repeat the analysis on bootstrap versions of the data, to check that conclusions are stable.

13. [Efron's Experiment] This exercise explores the phenomenon where, in a random forest model, a variable may be important without necessarily being indispensible. It is based on an experiment from the paper, "Prediction, Estimation, and Attribution" (Efron, ?), which used a real medical dataset. We will use a small simulation instead, but will recover the same essential finding.

a. Describe the permutation variable importance method used when interpreting random forest models.

b. Consider the data at this link. Use K-fold cross-validation to select the `mtry` hyperparameter. Report the RMSE this full model.

c. Identify the top 10 most important variables. Remove these 10 variables from the dataset. Re-run the tuning process from part b on the remaining 90 variables. Report the RMSE of this trimmed model.

d. Report the relative change in performance between the models in (b) and (c), $\frac{RMSE_{\text{trim}} - RMSE_{\text{full}}}{RMSE_{\text{full}}}$. You should see a drop of less than 5%. Investigate the original data to explain how the model could still perform well despite losing its "best" variables. How does this influence how you might explain variable importance outputs to your non-statisticial colleagues in future projects?

14. [Importance in Arcene]. Here you will study how feature correlation affects variable importance estimates in lasso regression and CART. The Arcene dataset (NeurIPS 2003 Feature Selection Challenge) contains mass spectrometry data: $n = 100$ samples, $10K$ spectral features (we consider a subset $p = 200$), binary outcome (cancer/healthy). The high-dimensional regime ($p > n$) makes variable selection nontrivial. Download: [MS features], [cancer status].

a. Fit lasso and CART to predict cancer status. Select hyperparameters λ (lasso) and cp (CART) via cross-validation. Report holdout accuracy.

b. Find:

  - Feature $j_L^* = \arg\max_j |\widehat{\beta}_j|$ (largest lasso coefficient magnitude)
  - Feature j_T^* with maximum CART variable importance (MDI)

c. Stability analysis: For $b = 1, \ldots, B = 50$:

- Resample data with replacement: (X_b, y_b)
- Refit both models using λ and cp from part (a)

Record coefficients $\widehat{\beta}^{(b)}$ and importances $I^{(b)}$. For features j_L^* and j_T^*, what fraction of bootstrap samples yielded zero coefficient or zero importance?

Hint: The pseudocode below may help.

```
FUNCTION stability(X, y, lambda_star, cp_star, B=50):

    INIT coef[B x p], imp[B x p]
    FOR b = 1 to B:
        idx = sample_with_replacement(n)
        X_b, y_b = X[idx], y[idx]

        lasso = fit_lasso(X_b, y_b, lambda_star, family = "binomial")
        coef[b] = lasso.coef

        tree = fit_tree(X_b, y_b, cp_star)
        imp[b] = tree.importance

RETURN coef, imp
```

d. Create a plot showing how coefficients and MDI scores vary across bootstrap samples for each of the original features. An example output is given in Figure 7. Discuss the implications for using the top lasso coefficients or MDI scores in these dataset.

e. Compute the feature correlation matrix $C \in \mathbf{R}^{p \times p}$ where $C_{jk} = \text{cor}(X_{\cdot j}, X_{\cdot k})$. Let $\rho_j = \text{median}_{k \neq j} |C_{jk}|$ denote the median absolute correlation between feature j and any other feature. Visualize the correlation matrix C, ordering rows if helpful. For the top 25 features by mean $\left|\widehat{\beta}_j^{(b)}\right|$ across bootstraps, report ρ_j. Similarly for the top 10 features by mean MDI. Does high ρ_j relate to importance estimates? Discuss your findings.

15. [Boosting Iterations] Consider boosting on data from $y = 3\sin(x) + \epsilon$, $\epsilon \sim \mathcal{N}(0, 0.25)$.

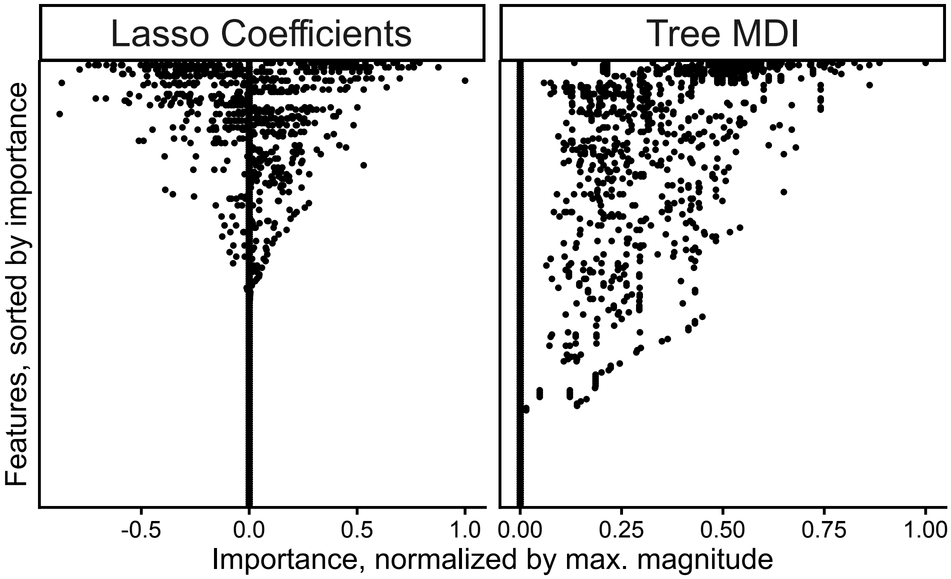


Figure 7

```r
set.seed(479)
X <- runif(100, -5, 5)
y <- 3 * sin(X) + 0.5 * rnorm(100)
df <- data.frame(X = X, y = y)
```

a. Fit a depth-2 tree to (X, y). Compute predictions $\hat{f}^1(x_i)$ and residuals $r_i^{(1)} = y_i - \hat{f}^1(x_i)$.

b. Fit a depth-2 tree to $(X, r^{(1)})$, yielding $\hat{f}^2$. Repeat on residuals $r^{(2)} = r^{(1)} - \hat{f}^2(x_i)$ to obtain $\hat{f}^3$.

c. Plot the sequence of fits $\hat{f}^b$ and ensemble predictions $\sum_{k=1}^{b} \hat{f}^k$ for $b = 1, 2, 3$ (setting $\lambda = 1$). Compare with Figure 8.

d. Comment on interpretability across: (i) a single tree, (ii) the 3-iteration ensemble from (c), and (iii) a 100-iteration ensemble.

Post-hoc Explanations

1. [SHAP Waterfall] We saw a visualization of $\varphi_x\left(f, i\right)$ for a single sample i in the census dataset. The plot below gives the analogous visualization for all N samples in that data. How do you interpret it?

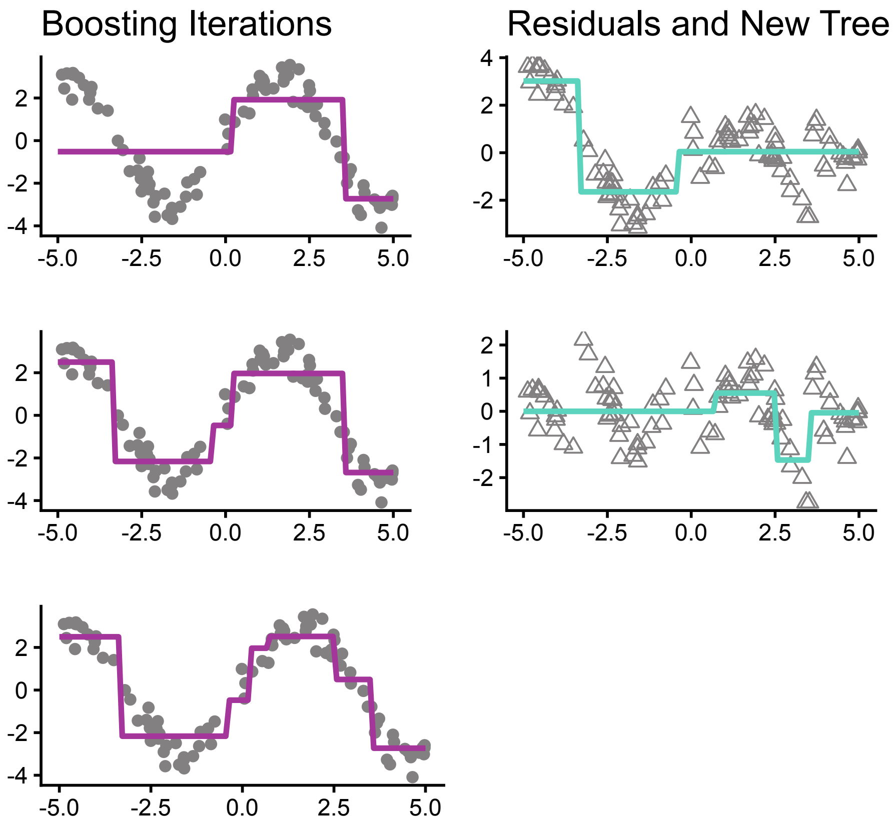
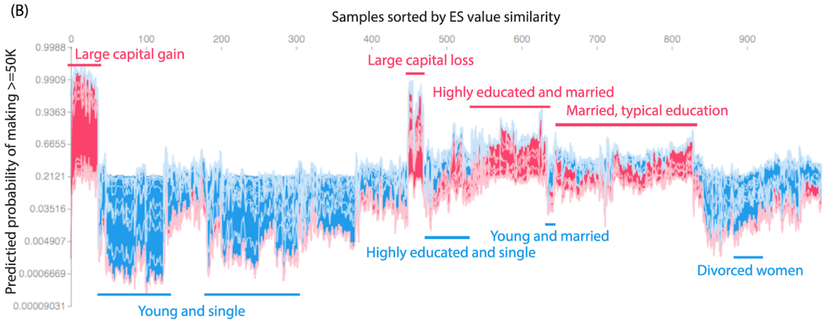
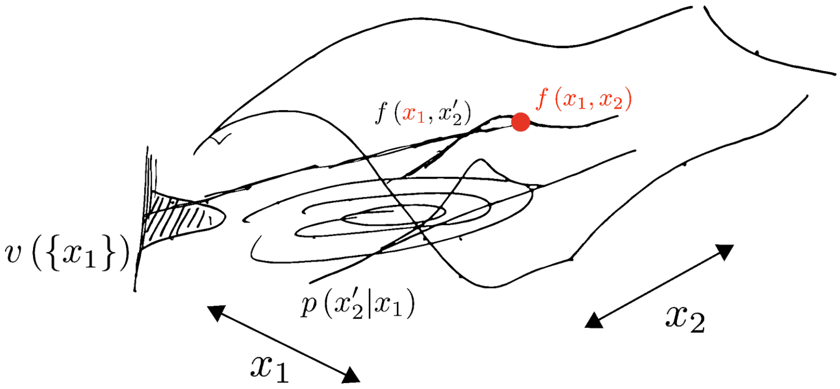


Figure 8: Example result for [Boosting Initial Steps].



1. [SHAP Visual Explanation] We gave a geometric interpretation of $v(1)$ in a toy example. Can you give the geometric interpretation of: $v(1,2), v(2)$, or $v()$?



1. [LinearSHAP Derivation] For certain choices of v, SHAP values can be computed in closed form. For example, in class we showed that if

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j \tag{1}$$

$$v(S) = f(x_S, x_{\bar{S}}^0) \tag{2}$$

then $\phi_j(f,x) = \beta_j(x_j - x_j^0)$. In this problem, we will consider the alternative

$$v(S) = \mathbb{E}[f(x_S^e, x_{\bar{S}})]$$

where the randomness in the expectation is over $x_{\bar{S}}$.

a. Describe the qualitative differences between the choices of v in equations (4b).

b. Show that with the choice of v in equation (4b), the associated SHAP values are

$$\phi_j(f,x) = \beta_j\left(x_j - \mathbf{E}\left[x_i\right]\right)$$

c. Compare and contrast the SHAP values associated with the two choices of v. On what types of data do you expect the explanations to be similar or different?

2. [KernelSHAP Code] These questions accompany the KernelSHAP code snippet below.

```
# initialize data structures
mask_matrix = np.zeros((nsamples, M))
kernel_weights = np.zeros(nsamples)
synth_data = np.tile(self.data, (nsamples, 1))

# keep values at random subsets of features
for i in range(nsamples):
    mask = np.random.choice([0, 1], size=M)
    mask_matrix[i] = mask
    synth_data[i * self.N:(i + 1) * self.N, mask == 1] = x[0, mask == 1]
    kernel_weights[i] = (M - np.sum(mask)) * np.sum(mask)

# model predictions on masked data
model_out = self.model(synth_data)
f = np.mean(model_out.reshape(nsamples, self.N, -1), axis=1) - self.f_bar

# weighted regression on adjusted f
X = mask_matrix - mask_matrix[:, -1][:, None]
y = f.flatten() - mask_matrix[:, -1] * (self.model(x) - self.f_bar)
lm = LinearRegression(fit_intercept=False).fit(X, y, sample_weight=kernel_weights)
```

a. How does the mask variable relate to the subsets S discussed in the notes?

b. How many linear regressions are run to compute SHAP values across all samples? How large are the input datasets for each of those regressions, and how is it related to nsamples?

c. What is the relationship between the returned attributions φ and the coefficients of our weighted linear regression?

d. `synth_data` can be viewed as `nsamples` blocks of N rows each. How do the blocks differ from one another?

e. If we ignore the terms involving the last column of mask_matrix (i.e., `mask_matrix[:, -1]`), then how can we interpret the response of the weighted linear regression?

3. [marginal vs. conditional shap]. In a dataset with highly colinear features, see how conditional SHAP can give high attributions even when coefficients are small. Explain why this discrepancy arises.

4. [importance of baselines] Two data scientists get different explanations for the same model. They chose different baselines. Explain why they get the different results, and explain the

5. [KernelSHAP variance] Compute KernelSHAP across different sample sizes and for many replicates. What is the dependence on sample size? What would you recommend?

6. [SHAP Stability]. Fix a model. Evaluate overall SHAP value when applied to different subsamples.

7. [Comparing models with SHAP]

   - data splitting, then EDA on training set. Report one interesting finding.
   - fit RF + Lasso
   - compare prediction accuracy
   - compute SHAP values for a single run
   - Run bootstrap on training set. How do explanations change?

8. [Compare and Contrast] Compare and contrast the explainability techniques {A} and {B} (A and B given in class). Comment on:

   - What are the final outputs of the method?
   - How do the mathematical formulations relate to one another?
   - Who is the assumed audience for the method's final output?

Deep Learning

1. [Two moons]

2. [width vs. depth] Models with same total number of parameters, but change the

3. [Layernorm from Scratch]

4. [Normalization Ablations]

5. [Hotel Review Embeddings] This exercise explores how large language models capture meaning from a corpus of hotel reviews. We will use the llama3.2:1b model. (give details on tokenizing the reviews and loading the models using either R or python)

   a. How many heads are used in the MHA mechanism in the first transformer layer?

   b. Extract embeddings at layers 2 and 9. Store the results in arrays with shape $N \times D_l$.

   c. Compute the pairwise distances $\|h_i^l - h_{i'}^l\|$ between all review embeddings at layers $l = 2$ and 9. Make a scatterplot where each point corresponds to a single pair ii', the x-axis corresponds to $l = 2$, and the y-axis corresponds to $l = 9$.

d. Find a pair of reviews that lies far from the identity line in the scatterplot from part (c). Comment on how this relates to property that deeper layers learn more abstract representations of the input.

some helper code:

```
def extract_embeddings(text, model, tokenizer, layer=-1):
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=False)
    with torch.no_grad():
        outputs = model(**inputs, output_hidden_states=True)
    return outputs.hidden_states[layer].mean(axis=(0, 1))


# Load pre-trained model and tokenizer
tokenizer = AutoTokenizer.from_pretrained("EleutherAI/gpt-neo-125m")
model = GPTNeoModel.from_pretrained("EleutherAI/gpt-neo-125m")
model = model.eval()
```

6. [Integrated Gradients Loop]1. [Self-Attention from Scratch]

   - implementation with matrix multiplications
   - implementation with einsum
   - comparison with pytorch implementation

7. [Sequence Length Scaling] For attention mechanism, see how computational cost grows as sequence length increases.

8. [Sanity Checks Review] The reviews for Adebayo et al. 2018 are publicly available. Reviewer 1 has some excellent critiques. Discuss the points below. Why do you think the reviewer raised them, and what types of follow-up analysis do they suggest?

   - "Role of and relation to human judgement: Visual explanations are useless if humans do not interpret them correctly (see framework in [1])… Do the proposed sanity checks help identify explanation methods which are more human friendly? Even if the answer to the last question is no, it would be useful to discuss."

   - What if the layers are randomized in the other direction (from input to output)? Is it still the classifier layer that matters most?

   - The analysis of a conv layer is rather hand wavy. It is not clear to me that edges should appear in the produced saliency mask as claimed at l241. The evidence in figure 6 helps, but it is not completely convincing and the visualizations do not (strictly speaking) immitate an edge detector (e.g., look at the vegitation in front of the lighthouse).

9. [Saliency Map Context] In week 1, we introduced the following vocabulary: inherently interpretable models, post-hoc explanations, local explanations, global explanations. How do saliency maps relate to each of these terms?

10. [Saliency-based Model Critique]

11. [Model Developers vs. Users] Linear probes were proposed with model developers in mind. How would you explain the outputs of either approach to an audience of model users? First, explain what the main outputs are, and then relate them to outputs they may be familiar from more basic statistics or machine learning courses.

12. [Linear Probe Pseudocode]

13. [Probe Applications] Linear probes were presented in the context of natural image classification problems. To what extent can these ideas be adapted to other types of data? Choose one of the following dataset types below, and discuss which elements of the two techniques do/do not seem generalizable to that context:

    - Node classification in a network dataset
    - Text generation in a natural language dataset
    - Loan default prediction in a financial dataset
    - Cell type annotation in a single-cell dataset
    - Galaxy type classification from a telescope images dataset
    - Cancer tumor segmentation in a histopathology dataset

14. [FastCAV Visual Explanation] Develop a visual explanation for the following excerpt from our reading (equations 6 in Schmalwasser et al. 2025). Make sure to explain how the mathematical symbols relate to the parts of your visual explanation. Precisely describe any simplifying assumptions to make the figure clearer.

    > In Equation (3), we again apply the respective maximum likelihood estimator $\hat{\mu}_{D_c}$ meaning $v_l^c \propto \hat{\mu}_{D_c} - \hat{\mu}_{D_c \cup D_r}$. Under the assumptions specified above and again drawing an expectation over the sample of random inputs and concept examples, we get $\mathbb{E}[v_l^c] \propto \mu_c - \frac{\mu_c+\mu_r}{2} = \frac{\mu_c-\mu_r}{2}$.

15. [FastCAV inputs/outputs] Review the concept-based explanation experiments discussed in section 4.3 of Schmalwasser et al. (2025).

    - What are the inputs required for CAV extraction.
    - How do you read Figure 3 and what are the main claims the authors make based on it?
    - In your view, how strong is the evidence for the authors' interpretation?

16. [Agency vs. Automation] The excerpt below comes from Agency plus automation: Designing artificial intelligence into interactive systems (Heer 2018). Consider how the techniques discussed this week relate to the intelligence augmentation research agenda discussed here.

> Much contemporary rhetoric regards the prospects and pitfalls of using artificial intelligence techniques to automate an increasing range of tasks, especially those once considered the purview of people alone....This long-standing focus on purely automated methods unnecessarily cedes a promising design space: one in which computational assistance augments and enriches, rather than replaces, people's intellectual work. This tension between human agency and machine automation poses vital challenges for design and engineering...To improve outcomes and support learning by both people and machines, we describe the use of shared representations of tasks augmented with predictive models of human capabilities and actions. We conclude with a discussion of future prospects and scientific frontiers for intelligence augmentation research.

17. [Implement FISTA]
18. [SAE Applications] Can Sparse Autoencoders be applied to these problems? If so, explain how, if not, explain why not.

    - Updating all parameters when training an LLM is expensive. You would like to develop a new training strategy that pauses every 100,000 steps and freezes weights from layers that have not changed much since the last check.
    - You have extracted activations from a medical image classification model across two different datasets, one related to a vulnerable patient population and another from a healthy control. You would like to see what aspects of the learned representations differ across these groups.
    - A new deep learning model architecture has been introduced in your research area. A critic argues that the learned representations are not fundamentally different, despite the improved performance. You would like to resolve the dispute by objectively comparing the representations.

19. [SAE vs. Concepts] Compare and contrast testing concept activation vectors with feature surveys in sparse autoencoders.

    - How do the methods compare in the way that they extract interpretable directions from model embedding spaces?
    - How do the methods compare with respect to the types of problems they apply their methods to?