Patterns, Predictions, and Actions

STAT 499: DRP

Mentor: Ronan Perry
Mentee: Ariel Fu

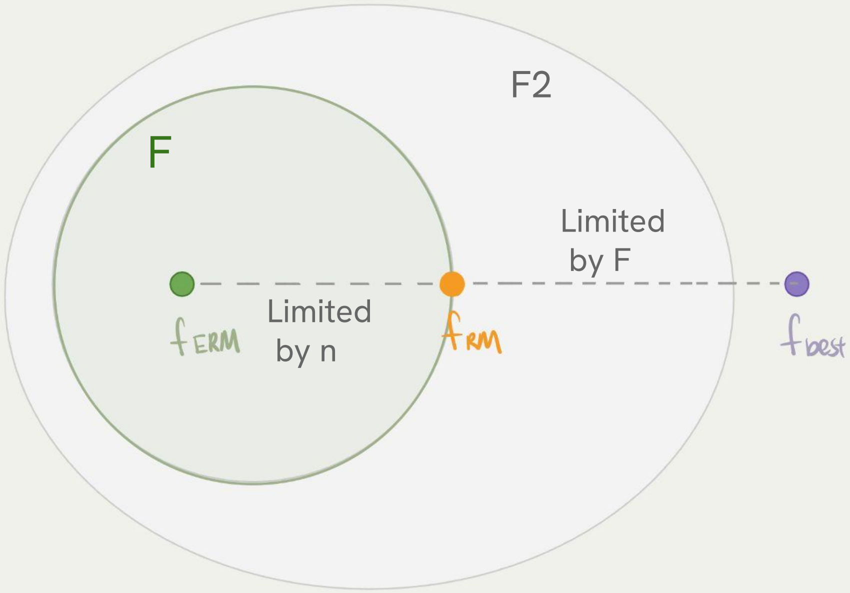
Content

1. Fundamentals of Prediction

- Attributes/covariates: $X \in \chi$
- Label/target: $Y \in Y$
- Function/predictor: $f(x)$
- Loss: $l(y, f(x))$
  - Eg. 0, 1 - loss, Brier loss
- Risk: $R(f) = E[l(y, f(x))]$
  - Eg. in regression case $R(f) = MSE = E[(Y - f(x))^2]$
- Goal: $\arg\min_f R[f]$

2. Risk Minimization

- To find risk need to choose two functions:
  1. loss function
     (e.g. $l(y, f(x)) = (Y - f(x))^2$)
  2. prediction function $f(x)$
- Risk Minimization: $\arg\min_{f \in F} R[f]$
  - need to define a function class F
  - e.g. simple, multiple linear, logistic, etc.
- Empirical risk minimization (ERM)



3. Dataset: `load_breast_cancer`

- `sklearn.datasets.load_breast_cancer`
- Breast cancer Wisconsin dataset
- Binary classification
- n = 569
- 30 features/covariates
- Loss: MSE,
- F: multiple linear regression

Approaches

1. Assume **normal distribution**
   a. Compare which class the given sample is more likely to belongs
2. OLS Analytical solution $\beta^* = (X^T X)^{-1} X^T Y$
3. Gradient descent
4. SGD and variations

```
def predict_label(x_i, mu_0, sigma_0, mu_1, sigma_1):
    p_0 = norm.pdf(x_i, loc=mu_0, scale=sigma_0)
    p_1 = norm.pdf(x_i, loc=mu_1, scale=sigma_1)
    predicted_label = np.where(p_1 > p_0, 1, 0)
    return predicted_label
 0.0s
```

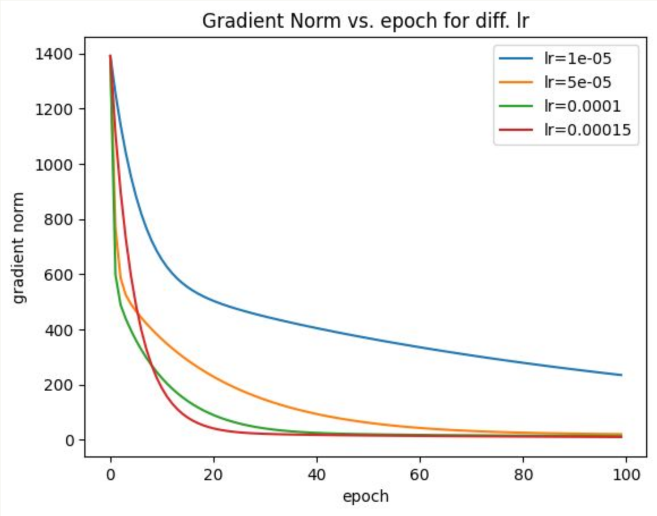
```
training accuracy: 87.25%
```

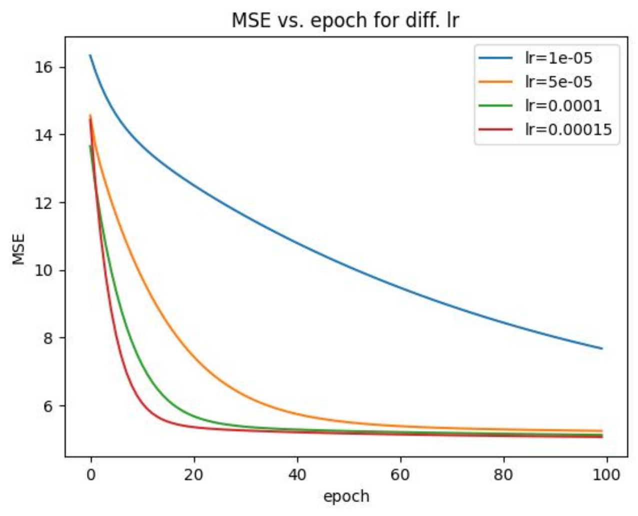
```
training accuracy: 87.47%
```

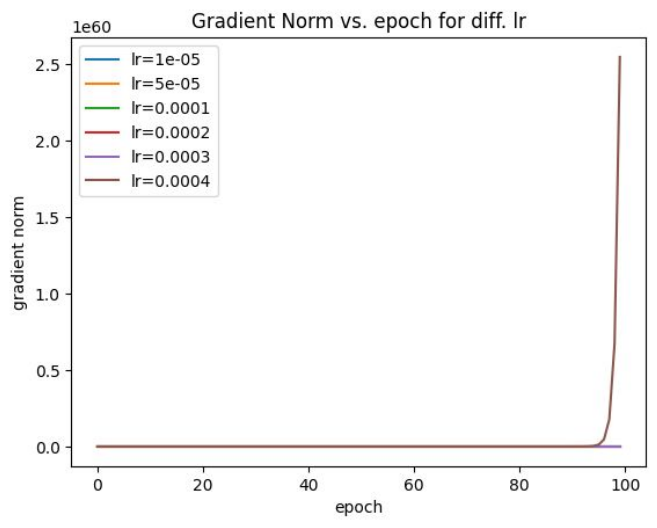
4. Gradient Descent - Full-batch

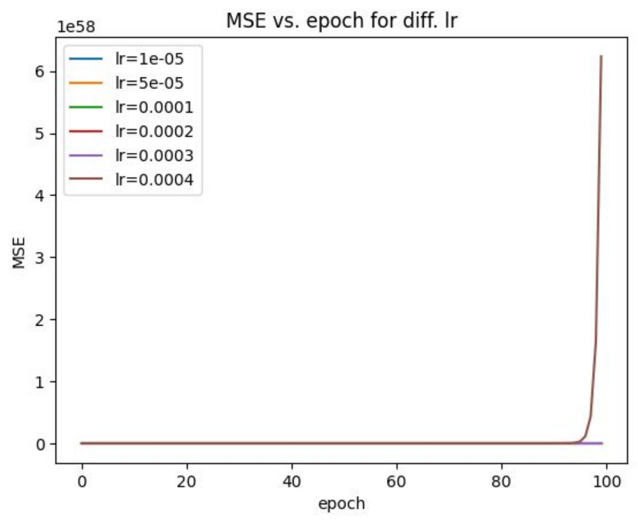
- $\min_\beta ||Y - X\beta||^2 = \min_\beta Y^T Y - 2X^T Y\beta + \beta^T X^T X\beta$
- $\Phi(w) = w^T Q w - P^T w + r$
  - where $P = 2X^T Y$ and $Q = 2X^T X$
- $\nabla\Phi(w) = Qw - P$
- Goal: find optimal $w*$ st. $\nabla\Phi(w*) = 0$
- Gradient Descent: $w_{t+1} = w_t - \alpha\nabla\Phi(w_t)$
- learning rate α

Gradient Norm & MSE





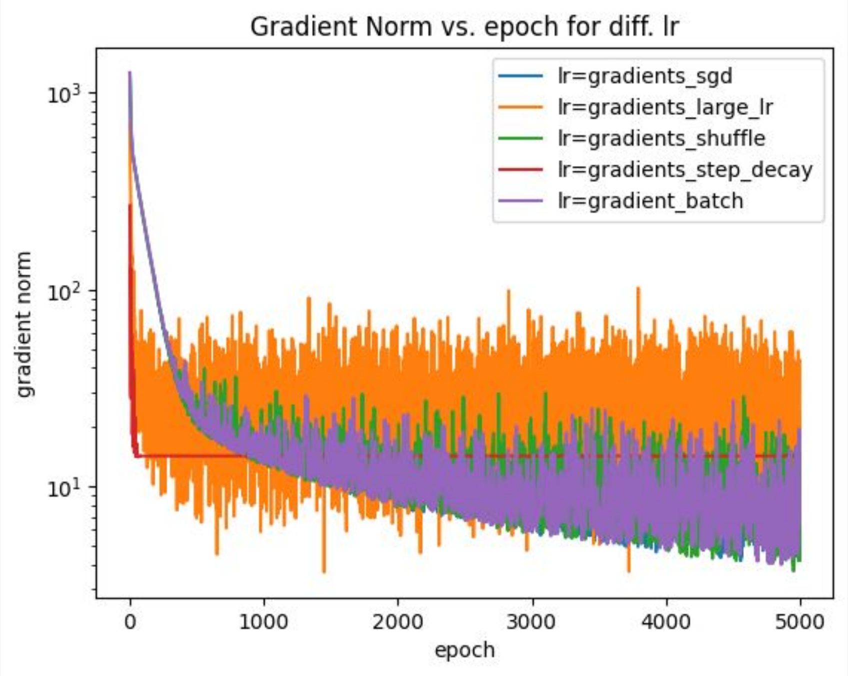


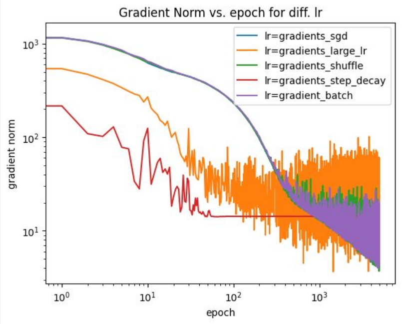


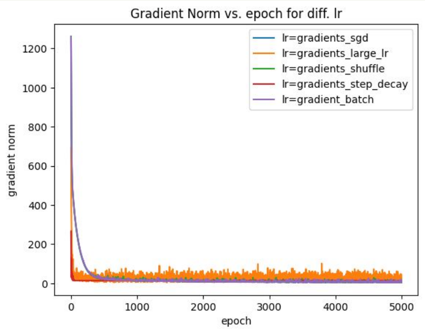
5. SGD

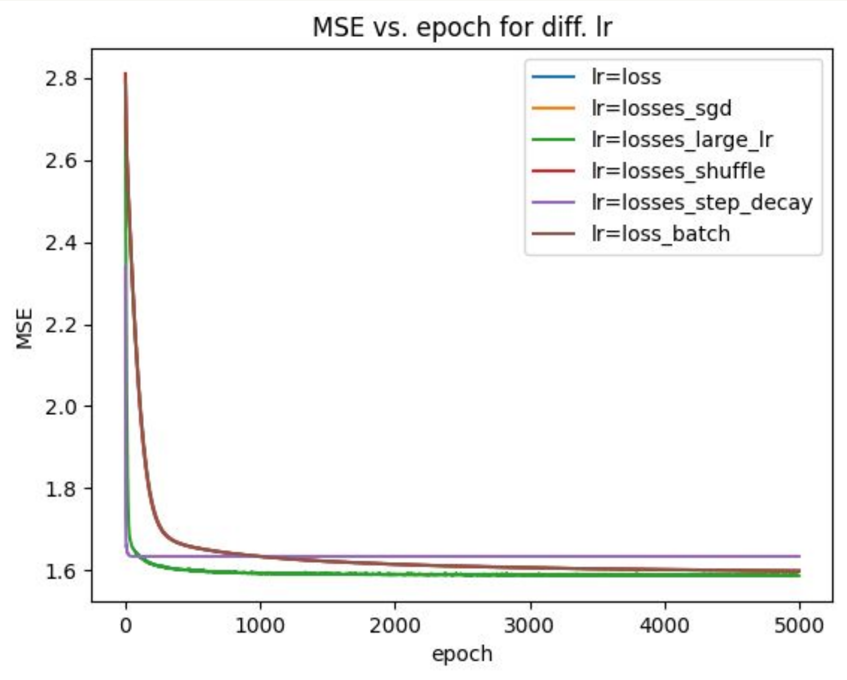
- $w_{t+1} = w_t - \alpha_t \nabla l_{w_t}(f(x_i), y_i)$
  - i is random, i.e. a random sample select from the dataset
  - update weight after each point, n * epoch updates
- Mini-batch: update the weight after m examples, $\frac{n}{m} * $ epoch updates
  - $w_{k+1} = w_k - \alpha_k \frac{1}{m} \sum_{j \in batch_k} \nabla l_{w_k}(f(x_j, w_k), y_i)$
- Shuffling: sampling each gradient with replacement
- Step decay: suppose $\alpha_0 = 0.001$, $\alpha_t = \alpha_0 \gamma^t$, $\gamma = 0.9$

Gradient Norm & MSE









6. Generalization

- $\hat{f} = \arg\min_{f \in F} R_s(f)$ via optimization
- Goal: $f^* = \arg\min_{f \in F} R(f)$ population
- $R(\hat{f}) = R_s(\hat{f}) + \boxed{(R(\hat{f}) - R_s(\hat{f}))}$

Generalization Gap

Thank you!

Q&A