Model Selection and Regularization

CS534

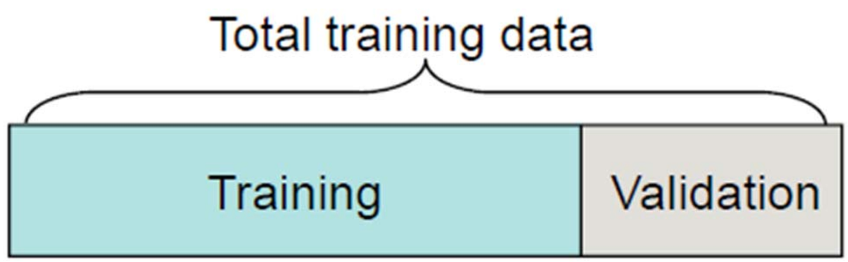
General Model Selection Problem

- Assume that we have a set of models M=$\{M_1, M_2, \cdots, M_d\}$ that we are trying to select from. Some examples include:
- **Feature Selection:** each M_i corresponds to using a different feature subset from a large set of potential features
- **Algorithm Selection:** each M_i corresponds to an algorithm, e.g., Naïve Bayes, Logistic Regression, DT …
- **Parameter selection:** each M_i corresponds to a particular parameter choice, e.g., the choice of kernel and C for SVM

Approaches to Model Selection

- Holdout and Cross-validation methods
  - Experimentally determine when overfitting occurs
- Penalty methods
  - MAP Penalty
  - Minimum Description Length
  - Many others
- Ensembles
  - Instead of choosing, consider many possibilities and let them vote

Simple Holdout Method

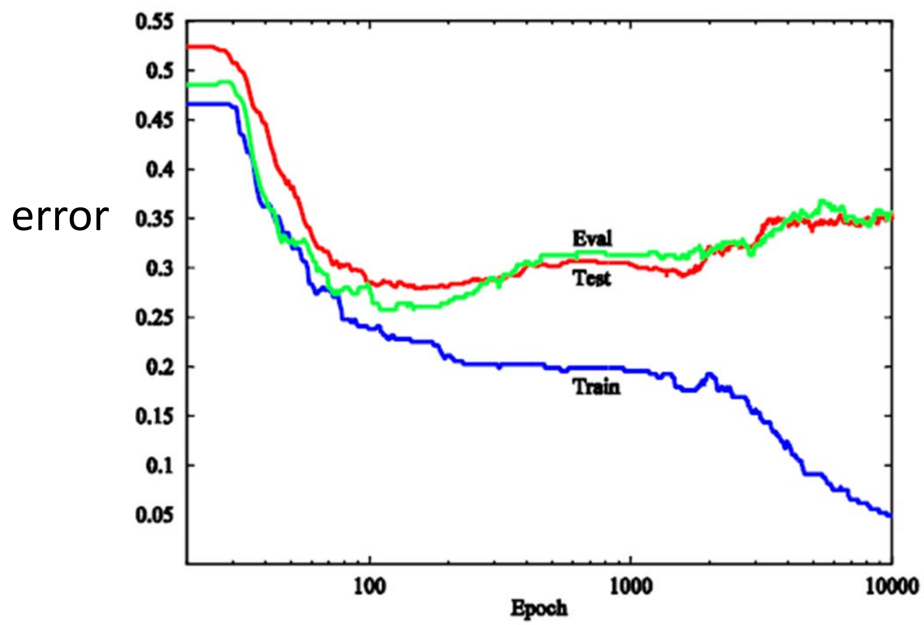


1. Divide training set S into S_{train} and S_{valid}
2. Train each model M_i on S_{train} to get a hypothesis h_i
3. Choose and output h_i with the smallest error rate on S_{valid}

Could retrain the selected model on the whole dataset to get the final hypothesis h - this will improve the original h_i because of more training data

Notes on hold-out methods

- Hold-out method often used for choosing among nested hypotheses:
  - Deciding # of training epochs for Neural net
  - Deciding when to stop growing or pruning a decision tree
  - Deciding when to stop growing an ensemble



Example:

Selecting # of epochs for neural net

Issues

- It wastes part of the data
  - The model selection choice is still made using only part of the data
  - Still possible to overfit the validation data since it is a relatively small set of data

- To address these problems, we can use a method called **Cross-Validation**

K-fold Cross-validation

- Partition (randomly) S into K disjoint subsets $S_1, \cdots, S_K$
- To evaluate model M_j:

> *for i=1:K*
>
> 1. *Train* M_j *on* $S \backslash S_i$ *(*S *removing* S_i*)* $\rightarrow h_{ji}$
> 2. *Test* h_{ji} *on* $S_i \rightarrow \epsilon_j(i)$
>
> *End for*
>
> $$\epsilon_j = \frac{1}{K}\sum_i \epsilon_j(i)$$

- Select model that minimizes the error:

$$M^* = \underset{M_j}{\operatorname{argmin}}\, \epsilon_j$$

- Train M^* on S and output resulting hypothesis

Comments on k-fold Cross-Validation

- Computationally more expensive than simple hold-out method but better use of data
- If the data is really scarce, we can use the extreme choice of $k = |S|$
  - Each validation set contains only one data point
  - leave-one-out (LOO) cross-validation

Penalty (Regularization) Methods

- Basic idea: include a penalty term in the objective function to penalize complex hypothesis
- We have seen examples of this:
  - Regularized linear regression

$$J(w) = \sum_i (y_i - \mathbf{w}^T \mathbf{x}_i)^2 + \lambda |\mathbf{w}|^2$$

Regularization term to control model complexity

  - Regularized logistic regression

$$J(w) = L(w) + \lambda |\mathbf{w}|^2$$

Log-likelihood

- A common approach for deriving such regularization method is Maximum A Posteria (MAP) estimation

Bayesian VS Frequentist

- When it comes to parameter estimation, there are two different statistical views
  - Frequentist: parameter is deterministic, it takes an unknown value
  - Bayesian: parameter is a random variable with a unknown distribution
    - We can express our belief about the parameter using priors
    - After observing the data, we can update our belief to obtain the posterior distribution of the parameter

Prior distribution of θ

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} = \frac{p(\theta)p(D|\theta)}{\int p(D|\theta)p(\theta)d\theta}$$

Posterior distribution of θ

Conjugate prior

- How should we specify the prior?

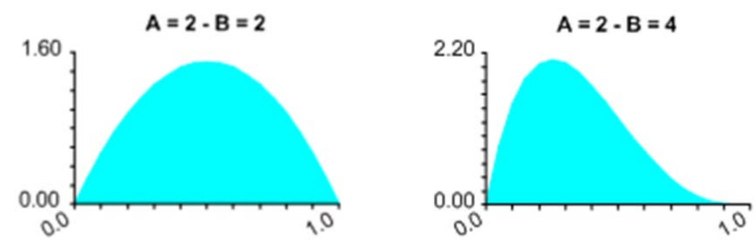
If the posterior distribution $p(\theta|D)$ is in the same family as the prior distribution $p(\theta)$, then $p(\theta)$ is called a ***conjugate prior***

- ***conjugate prior*** is an algebraic convenience, giving a closed-form expression for the posterior

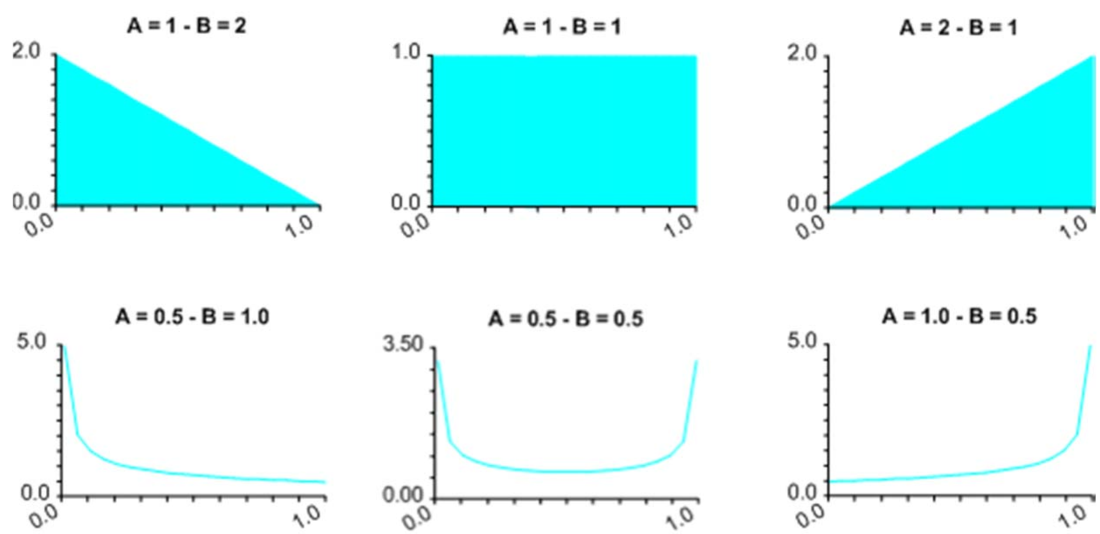
Example: Bernoulli

- $z \sim Ber(\theta)$
- What is the conjugate prior for Bernoulli?
- Beta distribution

$$p(\theta; A, B) = \frac{\theta^{A-1}(1-\theta)^{B-1}}{\mathrm{beta}(A, B)}$$



- A distribution over a continuous variable $p \in [0,1]$
- Two parameters: A>0, B>0
- For A=B=1, reduce to a uniform distribution
- A and B can be viewed as the effective prior number of observations of z=1 and z=0.



MAP Estimation for Bernoulli

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)}$$

$$p(\theta) = \frac{\theta^{A-1}(1-\theta)^{B-1}}{\mathrm{beta}(A,B)}$$

$$p(D|\theta) = \theta^{n_1}(1-\theta)^{n_0}$$

$$p(\theta|D) = \frac{\theta^{n_1+A-1}(1-\theta)^{n_0+B-1}}{?} = \frac{\theta^{n_1+A-1}(1-\theta)^{n_0+B-1}}{beta(A+n_1, B+n_0)}$$

$$\theta|D \sim Beta(\theta; A+n_1, B+n_0)$$

Maximum A Posterior estimation:

$$\hat{\theta}_{map} = \mathrm{argmax}_{\theta}\, p(\theta|D) = \frac{n_1+A}{n+A+B}$$

MAP as a penalty method

$$\hat{\theta}_{map} = \underset{\theta}{\operatorname{argmax}}\, p(\theta|D)$$

$$= \underset{\theta}{\operatorname{argmax}}\, p(D|\theta)p(\theta)$$

$$= \underset{\theta}{\operatorname{argmax}}\; \log p(D|\theta) + \log p(\theta)$$

penalty