Tutorial 9

CPSC 340

Overview

Learning Features

- Feature Selection
- PCA and Dimensionality Reduction

Learning Probability Densities

Learning Features

[Why] Feature Selection

- Sometimes, using every feature you have is not a good idea.
- Fundamental tradeoff:
  - More features $\implies$ Lower training error.
  - More features $\implies$ Training error is worse approximation of test error.

[Why] Feature Selection

- Sometimes, using every feature you have is not a good idea.
- Fundamental tradeoff:
  - More features $\implies$ Lower training error.
  - More features $\implies$ Training error is worse approximation of test error.
- Basically,
  - More features $\implies$ overfitting.
  - Less features $\implies$ simpler model.

[How] Feature Selection

- Balance between training error and number of features.

[How] Feature Selection

- Balance between training error and number of features.
- Idea - new loss function:

$$\hat{L}(w) = L(w) + [\#\text{features}]$$

where $L(w)$ is the original loss function.

[How] Feature Selection

- Balance between training error and number of features.
- Idea - new loss function:

$$\hat{L}(w) = L(w) + [\#\text{features}]$$

  where $L(w)$ is the original loss function.
- Can't minimize $\hat{L}$ using gradient descent.
- Idea - Fix #features, then minimize L using gradient descent.

[How] Feature Selection

- Balance between training error and number of features.
- Idea - new loss function:

$$\hat{L}(w) = L(w) + [\#\text{features}]$$

  where $L(w)$ is the original loss function.
- Can't minimize $\hat{L}$ using gradient descent.
- Idea - Fix #features, then minimize L using gradient descent.
  - Repeat for every subset of the features:
    - Find $w^* = \text{argmin } L(w)$
    - Compute $\hat{L}(w^*)$
  - Return w^* that minimizes $\hat{L}$

[How] Feature Selection

- Balance between training error and number of features.
- Idea - new loss function:

$$\hat{L}(w) = L(w) + [\#\text{features}]$$

  where $L(w)$ is the original loss function.
- Can't minimize $\hat{L}$ using gradient descent.
- Idea - Fix #features, then minimize L using gradient descent.
  - Repeat for every subset of the features:
    - Find $w^* = \text{argmin } L(w)$
    - Compute $\hat{L}(w^*)$
  - Return w^* that minimizes $\hat{L}$
- With d features, this procedure requires training $2^{|d|}$ models.

[How] Feature Selection

We can approximate the previous procedure.

[How] Feature Selection

We can approximate the previous procedure.

Forward selection procedure:

- Initialize with no features: $S = \emptyset$
- Best error so far: $\text{BestErr} = \infty$
- Repeat until BestErr doesn't decrease:
  - Repeat for every feature not in S:
    - Compute min $\hat{L}$ using this feature and S.
  - If for some feature, min $\hat{L} < \text{BestErr}$
    - Add feature that minimizes $\hat{L}$ the most to S.
  - (Else exit loop.)
- Return the model trained on S.

[How] Feature Selection

We can approximate the previous procedure.

Forward selection procedure:

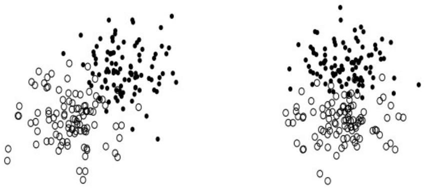
- Initialize with no features: $S = \emptyset$
- Best error so far: $\text{BestErr} = \infty$
- Repeat until BestErr doesn't decrease:
    - Repeat for every feature not in S:
        - Compute min $\hat{L}$ using this feature and S.
    - If for some feature, min $\hat{L} < \text{BestErr}$
        - Add feature that minimizes $\hat{L}$ the most to S.
    - (Else exit loop.)
- Return the model trained on S.

Number of trained models? $O(d^2)$

(There's a corresponding backward selection algorithm as well.)

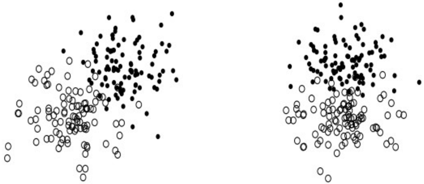
[Problem] Representation is Important.

Consider training decision trees on the follow two datasets:
(Binary classification)



[Problem] Representation is Important.

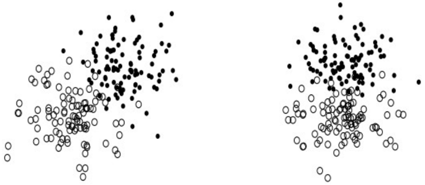
Consider training decision trees on the follow two datasets:
(Binary classification)



- The decision tree for the left dataset requires more depth to get a good error, and will likely overfit.

[Problem] Representation is Important.

Consider training decision trees on the follow two datasets:
(Binary classification)



- The decision tree for the left dataset requires more depth to get a good error, and will likely overfit.
- The problem? The right dataset is the exact same data, just rotated 45 degrees.

[PCA] Principal Component Analysis

Objective of PCA:

- Represent the same data points using different features.
- Features are ordered by variance. Oftentimes, features with more variance will give more information about the data.

[PCA] Principal Component Analysis

Objective of PCA:

- Represent the same data points using different features.
- Features are ordered by variance. Oftentimes, features with more variance will give more information about the data.

Result of PCA:

- $X = ZW$
- X is $n \times d$, Z is $n \times d$, W is $d \times d$
- W is interpreted as a new basis for X. Typically orthogonal.

[PCA] Principal Component Analysis

Objective of PCA:

- Represent the same data points using different features.
- Features are ordered by variance. Oftentimes, features with more variance will give more information about the data.

Result of PCA:

- $X = ZW$
- X is $n \times d$, Z is $n \times d$, W is $d \times d$
- W is interpreted as a new basis for X. Typically orthogonal.

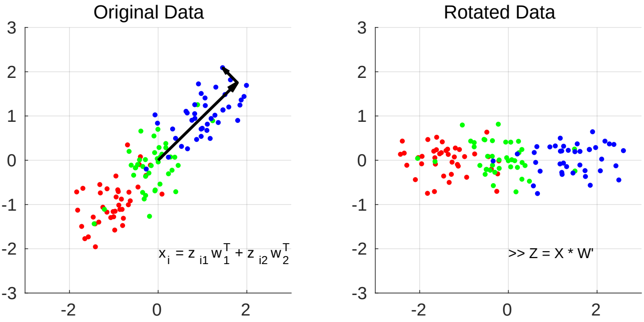
$$\underbrace{\begin{bmatrix} -x_1^T- \\ \vdots \\ -x_n^T- \end{bmatrix} = \begin{bmatrix} -z_1^T- \\ \vdots \\ -z_n^T- \end{bmatrix} \begin{bmatrix} | & & | \\ (w^T)_1 & \cdots & (w^T)_n \\ | & & | \end{bmatrix}}_{X=ZW}$$

(x_i is a linear combination of the columns of W, with z_i as coefficients.)

PCA Visualized

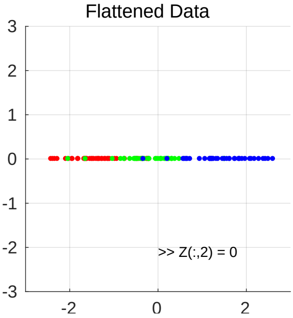
Result of PCA:

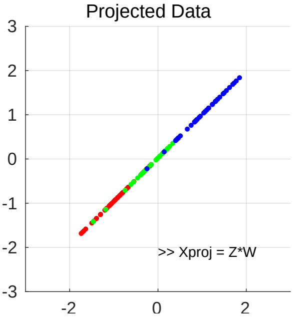
- $X = ZW$
- X is $n \times d$, Z is $n \times d$, W is $d \times d$
- W is interpreted as a new basis for X. Typically orthogonal.



Dimensionality Reduction

- The columns of W are the principal components.
- These components are ordered.
- The first principal component contains the most amount of information relevant to describing the data X.





Summary

- We can choose to use less features to combat overfitting.
- Sometimes the features we have just aren't useful for the model we choose.
- So we perform some operation (eg. PCA) to obtain more meaningful features.
- Along with PCA, we can do dimensionality reduction.
- Reducing the number of features achieves similar effects to feature selection.
- However, PCA + dimensionality reduction ensures we reduce the number of features in a meaningful way.

Learning Probability Densities

Parametric Probability Densities

- Let's place ourselves in a probabilistic setting.
- We have some data $X = (x_1, x_2, \dots, x_n)$
- We assume the samples x_i are i.i.d. from some density p

Parametric Probability Densities

- Let's place ourselves in a probabilistic setting.
- We have some data $X = (x_1, x_2, \dots, x_n)$
- We assume the samples x_i are i.i.d. from some density p
- The density p is parameterized by w. Notation $p(x; w)$.

Parametric Probability Densities

- Let's place ourselves in a probabilistic setting.
- We have some data $X = (x_1, x_2, \ldots, x_n)$
- We assume the samples x_i are i.i.d. from some density p
- The density p is parameterized by w. Notation $p(x; w)$.

Examples:

- $x_i \sim \text{Bernoulli}(w)$

$$p(x; w) = w^x (1 - w)^{1-x}$$

Parametric Probability Densities

- Let's place ourselves in a probabilistic setting.
- We have some data $X = (x_1, x_2, \ldots, x_n)$
- We assume the samples x_i are i.i.d. from some density p
- The density p is parameterized by w. Notation $p(x; w)$.

Examples:

- $x_i \sim \text{Bernoulli}(w)$

$$p(x; w) = w^x (1 - w)^{1-x}$$

- $x_i \sim \text{Normal}(\mu, \sigma^2)$

$$p(x; w) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

$$w = \{\mu, \sigma^2\}$$

Parametric Probability Densities

- Let's place ourselves in a probabilistic setting.
- We have some data $X = (x_1, x_2, \ldots, x_n)$
- We assume the samples x_i are i.i.d. from some density p
- The density p is parameterized by w. Notation $p(x; w)$.

Examples:

- $x_i \sim \text{Bernoulli}(w)$

$$p(x; w) = w^x (1-w)^{1-x}$$

- $x_i \sim \text{Normal}(\mu, \sigma^2)$

$$p(x; w) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$
$$w = \{\mu, \sigma^2\}$$

We interpret $p(x; w)$ as a density function parametrized by w.
Oftentimes people write $p(x|w)$ as well.

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.
We don't know w and would like to learn it.

- Frequentist Approach[1]:

- Bayesian Approach:

[1]Maximum likelihood estimation (MLE)
[2]Maximum a posteriori (MAP)

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.
We don't know w and would like to learn it.

- Frequentist Approach[1]:
    - We assume there exist a true distribution $p(x; w_0)$.
- Bayesian Approach:

[1] Maximum likelihood estimation (MLE)
[2] Maximum a posteriori (MAP)

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.

We don't know w and would like to learn it.

- Frequentist Approach[1]:
  - We assume there exist a true distribution $p(x; w_0)$.
  - We want to learn an approximation to w_0

$$w^* = \arg\max_w p(X|w)$$

- Bayesian Approach:

[1] Maximum likelihood estimation (MLE)

[2] Maximum a posteriori (MAP)

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.
We don't know w and would like to learn it.

- Frequentist Approach[1]:
  - We assume there exist a true distribution $p(x; w_0)$.
  - We want to learn an approximation to w_0
    $$w^* = \arg\max_w p(X|w)$$
  - Result: w^* is the parameter that maximizes the likelihood of observing/sampling X.
- Bayesian Approach:

[1] Maximum likelihood estimation (MLE)
[2] Maximum a posteriori (MAP)

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.
We don't know w and would like to learn it.

- Frequentist Approach[1]:
  - We assume there exist a true distribution $p(x; w_0)$.
  - We want to learn an approximation to w_0
    $$w^* = \arg\max_w p(X|w)$$
  - Result: w^* is the parameter that maximizes the likelihood of observing/sampling X.
- Bayesian Approach:
  - We assume w is a random variable from some distribution $p(w)$. Now the density function $p(x; w)$ turns into a conditional density $p(x|w)$.

[1] Maximum likelihood estimation (MLE)
[2] Maximum a posteriori (MAP)

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.
We don't know w and would like to learn it.

- Frequentist Approach[1]:
  - We assume there exist a true distribution $p(x; w_0)$.
  - We want to learn an approximation to w_0
  $$w^* = \arg\max_w p(X|w)$$
  - Result: w^* is the parameter that maximizes the likelihood of observing/sampling X.
- Bayesian Approach:
  - We assume w is a random variable from some distribution $p(w)$. Now the density function $p(x; w)$ turns into a conditional density $p(x|w)$.
  - After observing X, we update this distribution to reflect our data samples: $p(w|X)$.

[1]Maximum likelihood estimation (MLE)
[2]Maximum a posteriori (MAP)

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.
We don't know w and would like to learn it.

- Frequentist Approach[1]:
  - We assume there exist a true distribution $p(x; w_0)$.
  - We want to learn an approximation to w_0
  $$w^* = \arg\max_w p(X|w)$$
  - Result: w^* is the parameter that maximizes the likelihood of observing/sampling X.
- Bayesian Approach:
  - We assume w is a random variable from some distribution $p(w)$. Now the density function $p(x; w)$ turns into a conditional density $p(x|w)$.
  - After observing X, we update this distribution to reflect our data samples: $p(w|X)$.
  - Result: We have now learned a distribution over w. $p(w) \rightarrow_{learn} p(w|X)$

[1]Maximum likelihood estimation (MLE)
[2]Maximum a posteriori (MAP)

Frequentist vs. Bayesian Modeling

Assume we have some samples iid from a distribution $p(x; w)$.
We don't know w and would like to learn it.

- Frequentist Approach[1]:
  - We assume there exist a true distribution $p(x; w_0)$.
  - We want to learn an approximation to w_0
$$w^* = \arg\max_w p(X|w)$$
  - Result: w^* is the parameter that maximizes the likelihood of observing/sampling X.
- Bayesian Approach:
  - We assume w is a random variable from some distribution $p(w)$. Now the density function $p(x; w)$ turns into a conditional density $p(x|w)$.
  - After observing X, we update this distribution to reflect our data samples: $p(w|X)$.
  - Result: We have now learned a distribution over w. $p(w) \rightarrow_{learn} p(w|X)$
  - Sometimes people still want a single estimate:
    $w^* = \arg\max_w p(w|X)$ (mode)[2] or $w^* = \mathbb{E}_{p(w|X)}[w|X]$ (mean)

[1] Maximum likelihood estimation (MLE)
[2] Maximum a posteriori (MAP)

[How] Maximum Likelihood Estimation

Remember: We want $w^* = \arg\max_w p(X|w)$

[How] Maximum Likelihood Estimation

Remember: We want $w^* = \arg\max_w p(X|w)$

Procedure for MLE:

- Step 1. Given n samples $\{x_1, x_2, \ldots, x_n\}$, write down the joint distribution of the data: $p(X; w)$
- Step 2. Compute the log-likelihood: $\log p(X; w)$.
- Step 3. Differentiate and equate to zero to find w^*.

(Oftentimes $\log p$ is easier to work with than p itself.)

[How] Maximum Likelihood Estimation

Remember: We want $w^* = \arg\max_w p(X|w)$
Procedure for MLE:

- Step 1. Given n samples $\{x_1, x_2, \dots, x_n\}$, write down the joint distribution of the data: $p(X; w)$
- Step 2. Compute the log-likelihood: $\log p(X; w)$.
- Step 3. Differentiate and equate to zero to find w^*.

(Oftentimes $\log p$ is easier to work with than p itself.)

Exercise:
Assume $x_i \sim \text{Bernoulli}(w)$.

$$p(x; w) = w^x (1 - w)^{1-x}$$

Write down w^* in terms of x_i and n.

[How] Bayesian Learning

Remember: We want $p(w|X)$, assuming we have some $p(w)$

[How] Bayesian Learning

Remember: We want $p(w|X)$, assuming we have some $p(w)$

Procedure for Bayesian learning:

- Step 1. Given n samples $\{x_1, x_2, \ldots, x_n\}$, write down the joint distribution of the data conditioned on w: $p(X|w)$
- Step 2. Specify a prior: $p(w)$
- Step 3. Compute the posterior: $p(w|X)$

$$p(w|X) = \frac{p(X|w)p(w)}{p(X)}$$

[How] Bayesian Learning

Remember: We want $p(w|X)$, assuming we have some $p(w)$

Procedure for Bayesian learning:

- Step 1. Given n samples $\{x_1, x_2, \ldots, x_n\}$, write down the joint distribution of the data conditioned on w: $p(X|w)$
- Step 2. Specify a prior: $p(w)$
- Step 3. Compute the posterior: $p(w|X)$

$$p(w|X) = \frac{p(X|w)p(w)}{p(X)}$$

Exercise:

Assume $x_i \sim \text{Bernoulli}(w)$.

$$p(x|w) = w^x(1-w)^{1-x}$$

Assume $p(w) \sim \text{Beta}(a, b)$.

$$p(w) \propto w^{a-1}(1-w)^{b-1}$$

Derive $p(w|X)$. Optionally, derive the MAP estimate.

Deriving Ridge Regression

Using MLE/MAP estimation is often equivalent to minimizing some loss function.

Deriving Ridge Regression

Using MLE/MAP estimation is often equivalent to minimizing some loss function.

Assignment 4 Question 2.2:

- The likelihood is given by $p(y_i|x_i, w) \sim \text{Normal}(w^T x_i, 1)$.
- The prior for each variable j is given by $p(w_j) \sim \text{Normal}(0, \lambda^{-1})$.

Deriving Ridge Regression

Using MLE/MAP estimation is often equivalent to minimizing some loss function.

Assignment 4 Question 2.2:

- The likelihood is given by $p(y_i|x_i, w) \sim \text{Normal}(w^T x_i, 1)$.
- The prior for each variable j is given by $p(w_j) \sim \text{Normal}(0, \lambda^{-1})$.

Then the MAP estimate is given by:

$$w^* = \underbrace{\arg\max_w p(w|X)}_{\text{MAP}} = \underbrace{\arg\min_w \frac{1}{2}||Xw - y||^2 + \frac{\lambda}{2}||w||^2}_{\text{L2-Regularized Least Squares}}$$

Deriving Ridge Regression

Using MLE/MAP estimation is often equivalent to minimizing some loss function.

Assignment 4 Question 2.2:

- The likelihood is given by $p(y_i|x_i, w) \sim \text{Normal}(w^T x_i, 1)$.
- The prior for each variable j is given by $p(w_j) \sim \text{Normal}(0, \lambda^{-1})$.

Then the MAP estimate is given by:

$$w^* = \underbrace{\arg\max_w p(w|X)}_{\text{MAP}} = \underbrace{\arg\min_w \frac{1}{2}||Xw - y||^2 + \frac{\lambda}{2}||w||^2}_{\text{L2-Regularized Least Squares}}$$

Exercise:

Show the above. You may find this useful:

$$N(x; \mu, \sigma^2) \propto \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

Summary

- Two approaches: Frequentist vs. Bayesian.
- Maximum likelihood estimate:

$$w^* = \arg\max_w p(X; w)$$

  Requires knowing/assuming $p(X; w)$[3].
- Maximum a posterior:

$$w^* = \arg\max_w p(w|X)$$

  Requires knowing/assuming $p(X|w)$ and $p(w)$. (Apply Bayes' Rule.)
- General rule for converting density functions to loss functions:

$$\arg\max_w f(w) = \arg\min_w - \log f(w)$$

  Oftentimes $\log f$ is easier to work with than f itself.

[3] Or written as $p(X|w)$.