End of Semester Review

David Carlson

May 8, 2019

Naive Bayes Classification

- Fast and simple classification algorithm (supervised)

Naive Bayes Classification

- Fast and simple classification algorithm (supervised)
- Generative model — specifies the hypothetical random process that generates data

Naive Bayes Classification

- Fast and simple classification algorithm (supervised)
- Generative model — specifies the hypothetical random process that generates data
- Gaussian naive Bayes: Data from each label is drawn from simple Gaussian distribution

Naive Bayes Classification

- Fast and simple classification algorithm (supervised)
- Generative model — specifies the hypothetical random process that generates data
- Gaussian naive Bayes: Data from each label is drawn from simple Gaussian distribution
- Find mean and standard deviation of points within a label, which defines the distribution

Naive Bayes Classification

- Fast and simple classification algorithm (supervised)
- Generative model — specifies the hypothetical random process that generates data
- Gaussian naive Bayes: Data from each label is drawn from simple Gaussian distribution
- Find mean and standard deviation of points within a label, which defines the distribution
- Allows for probabilistic classification

Naive Bayes Classification

- Fast and simple classification algorithm (supervised)
- Generative model — specifies the hypothetical random process that generates data
- Gaussian naive Bayes: Data from each label is drawn from simple Gaussian distribution
- Find mean and standard deviation of points within a label, which defines the distribution
- Allows for probabilistic classification
- Multinomial naive Bayes: Features assumed to be generated from simple multinomial distribution

Naive Bayes Classification (cont.)

- Naive Bayes is usually out-performed by more complicated models

Naive Bayes Classification (cont.)

- Naive Bayes is usually out-performed by more complicated models
- They are extremely fast for both training and prediction

Naive Bayes Classification (cont.)

- Naive Bayes is usually out-performed by more complicated models
- They are extremely fast for both training and prediction
- They provide straightforward probabilistic prediction

Naive Bayes Classification (cont.)

- Naive Bayes is usually out-performed by more complicated models
- They are extremely fast for both training and prediction
- They provide straightforward probabilistic prediction
- They are often very easily interpretable

Naive Bayes Classification (cont.)

- Naive Bayes is usually out-performed by more complicated models
- They are extremely fast for both training and prediction
- They provide straightforward probabilistic prediction
- They are often very easily interpretable
- They have very few (if any) tunable parameters

When to Use NB

- When the naive assumptions actually match the data (very rare in practice)

When to Use NB

- When the naive assumptions actually match the data (very rare in practice)
- For very well-separated categories, when model complexity is less important

When to Use NB

- When the naive assumptions actually match the data (very rare in practice)
- For very well-separated categories, when model complexity is less important
- For very high-dimensional data, when model complexity is less important

Linear Regression

- Good starting point for regression tasks

Linear Regression

- Good starting point for regression tasks
- Basis functions — transform the data through a function

Linear Regression

- Good starting point for regression tasks
- Basis functions — transform the data through a function
  - Polynomial basis functions

Linear Regression

- Good starting point for regression tasks
- Basis functions — transform the data through a function
  - Polynomial basis functions
  - Gaussian basis functions

Linear Regression

- Good starting point for regression tasks
- Basis functions — transform the data through a function
  - Polynomial basis functions
  - Gaussian basis functions
  - If use too many basis functions leads to overfitting $\rightarrow$ regularization

Linear Regression

- Good starting point for regression tasks
- Basis functions — transform the data through a function
  - Polynomial basis functions
  - Gaussian basis functions
  - If use too many basis functions leads to overfitting $\rightarrow$ regularization
    - Penalize large values of model parameters

Linear Regression

- Good starting point for regression tasks
- Basis functions — transform the data through a function
  - Polynomial basis functions
  - Gaussian basis functions
  - If use too many basis functions leads to overfitting $\rightarrow$ regularization
    - Penalize large values of model parameters
    - Ridge regression (L2 regularization)

Linear Regression

- Good starting point for regression tasks
- Basis functions — transform the data through a function
  - Polynomial basis functions
  - Gaussian basis functions
  - If use too many basis functions leads to overfitting $\rightarrow$ regularization
    - Penalize large values of model parameters
    - Ridge regression (L2 regularization)
    - Lasso regularization (L1)

Support Vector Machines

- Both classification and regression

Support Vector Machines

- Both classification and regression
- Supervised

Support Vector Machines

- Both classification and regression
- Supervised
- Discriminative classification: Rather than modeling each class, we simply find a line or curve (in two dimensions) or manifold (in multiple dimensions) that divides the classes from each other

Support Vector Machines

- Both classification and regression
- Supervised
- Discriminative classification: Rather than modeling each class, we simply find a line or curve (in two dimensions) or manifold (in multiple dimensions) that divides the classes from each other
- Beyond linear boundaries: Kernel SVM

Support Vector Machines

- Both classification and regression
- Supervised
- Discriminative classification: Rather than modeling each class, we simply find a line or curve (in two dimensions) or manifold (in multiple dimensions) that divides the classes from each other
- Beyond linear boundaries: Kernel SVM
- Hardness of the margin is controlled by a tuning parameter, most often known as C

Support Vector Machines

- Both classification and regression
- Supervised
- Discriminative classification: Rather than modeling each class, we simply find a line or curve (in two dimensions) or manifold (in multiple dimensions) that divides the classes from each other
- Beyond linear boundaries: Kernel SVM
- Hardness of the margin is controlled by a tuning parameter, most often known as C
- Optimum value of C needs to be tuned through cross-validation

Benefits of SVM

- Their dependence on relatively few support vectors means that they are very compact models, and take up very little memory

Benefits of SVM

- Their dependence on relatively few support vectors means that they are very compact models, and take up very little memory
- Once the model is trained, the prediction phase is very fast

Benefits of SVM

- Their dependence on relatively few support vectors means that they are very compact models, and take up very little memory
- Once the model is trained, the prediction phase is very fast
- Because they are affected only by points near the margin, they work well with high-dimensional data — even data with more dimensions than samples

Benefits of SVM

- Their dependence on relatively few support vectors means that they are very compact models, and take up very little memory
- Once the model is trained, the prediction phase is very fast
- Because they are affected only by points near the margin, they work well with high-dimensional data — even data with more dimensions than samples
- Their integration with kernel methods makes them very versatile, able to adapt to many types of data

Disadvantages of SVM

- For large numbers of training samples, computational cost can be prohibitive

Disadvantages of SVM

- For large numbers of training samples, computational cost can be prohibitive
- C must be carefully chosen via cross-validation, which can be expensive as datasets grow in size

Disadvantages of SVM

- For large numbers of training samples, computational cost can be prohibitive
- C must be carefully chosen via cross-validation, which can be expensive as datasets grow in size
- Results do not have a direct probabilistic interpretation

Decision Trees and Random Forests

- Random forests are an example of an ensemble method built on decision trees

Decision Trees and Random Forests

- Random forests are an example of an ensemble method built on decision trees
- Decision trees are a series of questions designed to zero in on the classification

Decision Trees and Random Forests

- Random forests are an example of an ensemble method built on decision trees
- Decision trees are a series of questions designed to zero in on the classification
- We can fit multiple trees on subsets of the data and aggregate the results $\rightarrow$ random forests

Decision Trees and Random Forests

- Random forests are an example of an ensemble method built on decision trees
- Decision trees are a series of questions designed to zero in on the classification
- We can fit multiple trees on subsets of the data and aggregate the results $\rightarrow$ random forests
- Bagging makes use of an ensemble of parallel estimators, each of which overfits the data, and averages the results to find a better classification

Decision Trees and Random Forests

- Random forests are an example of an ensemble method built on decision trees
- Decision trees are a series of questions designed to zero in on the classification
- We can fit multiple trees on subsets of the data and aggregate the results $\rightarrow$ random forests
- Bagging makes use of an ensemble of parallel estimators, each of which overfits the data, and averages the results to find a better classification
- Random forests can also be used for regression tasks

Advantages and Disadvantages of Random Forests

- Both training and prediction are very fast, because of the simplicity of the underlying decision trees

Advantages and Disadvantages of Random Forests

- Both training and prediction are very fast, because of the simplicity of the underlying decision trees
- Both tasks can be straightforwardly parallelized, because the individual trees are entirely independent entities

Advantages and Disadvantages of Random Forests

- Both training and prediction are very fast, because of the simplicity of the underlying decision trees
- Both tasks can be straightforwardly parallelized, because the individual trees are entirely independent entities
- The multiple trees allow for a probabilistic classification: a majority vote among estimators gives an estimate of the probability

Advantages and Disadvantages of Random Forests

- Both training and prediction are very fast, because of the simplicity of the underlying decision trees
- Both tasks can be straightforwardly parallelized, because the individual trees are entirely independent entities
- The multiple trees allow for a probabilistic classification: a majority vote among estimators gives an estimate of the probability
- The nonparametric model is extremely flexible, and can thus perform well on tasks that are underfit by other estimators

Advantages and Disadvantages of Random Forests

- Both training and prediction are very fast, because of the simplicity of the underlying decision trees
- Both tasks can be straightforwardly parallelized, because the individual trees are entirely independent entities
- The multiple trees allow for a probabilistic classification: a majority vote among estimators gives an estimate of the probability
- The nonparametric model is extremely flexible, and can thus perform well on tasks that are underfit by other estimators
- Results are not easily interpretable

Principal Component Analysis

- Unsupervised estimators can highlight interesting aspects of the data without reference to any known labels

Principal Component Analysis

- Unsupervised estimators can highlight interesting aspects of the data without reference to any known labels
- PCA — dimensionality reduction

Principal Component Analysis

- Unsupervised estimators can highlight interesting aspects of the data without reference to any known labels
- PCA — dimensionality reduction
- Finding a list of the principal axes in the data, and using those axes to describe the dataset

Principal Component Analysis

- Unsupervised estimators can highlight interesting aspects of the data without reference to any known labels
- PCA — dimensionality reduction
- Finding a list of the principal axes in the data, and using those axes to describe the dataset
- Using PCA for dimensionality reduction involves zeroing out one or more of the smallest principal components, resulting in a lower-dimensional projection of the data that preserves the maximal data variance

Principal Component Analysis

- Unsupervised estimators can highlight interesting aspects of the data without reference to any known labels
- PCA — dimensionality reduction
- Finding a list of the principal axes in the data, and using those axes to describe the dataset
- Using PCA for dimensionality reduction involves zeroing out one or more of the smallest principal components, resulting in a lower-dimensional projection of the data that preserves the maximal data variance
- PCA can also be used for noise filtering

Principal Component Analysis

- Unsupervised estimators can highlight interesting aspects of the data without reference to any known labels
- PCA — dimensionality reduction
- Finding a list of the principal axes in the data, and using those axes to describe the dataset
- Using PCA for dimensionality reduction involves zeroing out one or more of the smallest principal components, resulting in a lower-dimensional projection of the data that preserves the maximal data variance
- PCA can also be used for noise filtering
- PCA's main weakness is that it tends to be highly affected by outliers in the data

Principal Component Analysis

- Unsupervised estimators can highlight interesting aspects of the data without reference to any known labels
- PCA — dimensionality reduction
- Finding a list of the principal axes in the data, and using those axes to describe the dataset
- Using PCA for dimensionality reduction involves zeroing out one or more of the smallest principal components, resulting in a lower-dimensional projection of the data that preserves the maximal data variance
- PCA can also be used for noise filtering
- PCA's main weakness is that it tends to be highly affected by outliers in the data
- It does not perform so well when there are nonlinear relationships within the data

Manifold Learning

- In practice manifold learning techniques tend to be finicky enough that they are rarely used for anything more than simple qualitative visualization of high-dimensional data — PCA tends to be better

Manifold Learning

- In practice manifold learning techniques tend to be finicky enough that they are rarely used for anything more than simple qualitative visualization of high-dimensional data — PCA tends to be better
- In manifold learning, there is no good framework for handling missing data. In contrast, there are straightforward iterative approaches for missing data in PCA

Manifold Learning

- In practice manifold learning techniques tend to be finicky enough that they are rarely used for anything more than simple qualitative visualization of high-dimensional data — PCA tends to be better
- In manifold learning, there is no good framework for handling missing data. In contrast, there are straightforward iterative approaches for missing data in PCA
- In manifold learning, the presence of noise in the data can short-circuit the manifold and drastically change the embedding. In contrast, PCA naturally filters noise from the most important components

Manifold Learning

- In practice manifold learning techniques tend to be finicky enough that they are rarely used for anything more than simple qualitative visualization of high-dimensional data — PCA tends to be better
- In manifold learning, there is no good framework for handling missing data. In contrast, there are straightforward iterative approaches for missing data in PCA
- In manifold learning, the presence of noise in the data can short-circuit the manifold and drastically change the embedding. In contrast, PCA naturally filters noise from the most important components
- The manifold embedding result is generally highly dependent on the number of neighbors chosen, and there is generally no solid quantitative way to choose an optimal number of neighbors. In contrast, PCA does not involve such a choice

Mainfold Learning (cont.)

- In manifold learning, the globally optimal number of output dimensions is difficult to determine. In contrast, PCA lets you find the output dimension based on the explained variance

Mainfold Learning (cont.)

- In manifold learning, the globally optimal number of output dimensions is difficult to determine. In contrast, PCA lets you find the output dimension based on the explained variance
- In manifold learning, the meaning of the embedded dimensions is not always clear. In PCA, the principal components have a very clear meaning

Mainfold Learning (cont.)

- In manifold learning, the globally optimal number of output dimensions is difficult to determine. In contrast, PCA lets you find the output dimension based on the explained variance
- In manifold learning, the meaning of the embedded dimensions is not always clear. In PCA, the principal components have a very clear meaning
- Can preserve nonlinear relationships

Mainfold Learning (cont.)

- In manifold learning, the globally optimal number of output dimensions is difficult to determine. In contrast, PCA lets you find the output dimension based on the explained variance
- In manifold learning, the meaning of the embedded dimensions is not always clear. In PCA, the principal components have a very clear meaning
- Can preserve nonlinear relationships
- Explore manifold only after PCA

Mainfold Learning (cont.)

- In manifold learning, the globally optimal number of output dimensions is difficult to determine. In contrast, PCA lets you find the output dimension based on the explained variance
- In manifold learning, the meaning of the embedded dimensions is not always clear. In PCA, the principal components have a very clear meaning
- Can preserve nonlinear relationships
- Explore manifold only after PCA
- For high-dimensional data from real-world sources, locally linear embedding often produces poor results, and isometric mapping (Isomap) seems to generally lead to more meaningful embeddings

k-Means Clustering

- Unsupervised; clustering instead of dimensionality reduction

k-Means Clustering

- Unsupervised; clustering instead of dimensionality reduction
- k-means algorithm searches for a predetermined number of clusters within an unlabeled multidimensional dataset

k-Means Clustering

- Unsupervised; clustering instead of dimensionality reduction
- k-means algorithm searches for a predetermined number of clusters within an unlabeled multidimensional dataset
- Simple conception of what the optimal clustering looks like:

k-Means Clustering

- Unsupervised; clustering instead of dimensionality reduction
- k-means algorithm searches for a predetermined number of clusters within an unlabeled multidimensional dataset
- Simple conception of what the optimal clustering looks like:
    - "Cluster center" is the arithmetic mean of all the points belonging to the cluster

k-Means Clustering

- Unsupervised; clustering instead of dimensionality reduction
- k-means algorithm searches for a predetermined number of clusters within an unlabeled multidimensional dataset
- Simple conception of what the optimal clustering looks like:
    - "Cluster center" is the arithmetic mean of all the points belonging to the cluster
    - Each point is closer to its own cluster center than to other cluster centers

Disadvantages of k-Means

- No assurance that algorithm will lead to global best solution

Disadvantages of k-Means

- No assurance that algorithm will lead to global best solution
- Number of clusters must be determined beforehand

Disadvantages of k-Means

- No assurance that algorithm will lead to global best solution
- Number of clusters must be determined beforehand
- k-means is limited to linear cluster boundaries

Disadvantages of k-Means

- No assurance that algorithm will lead to global best solution
- Number of clusters must be determined beforehand
- k-means is limited to linear cluster boundaries
- The fundamental model assumptions of k-means (points will be closer to their own cluster center than to others) means that the algorithm will often be ineffective if the clusters have complicated geometries

Disadvantages of k-Means

- No assurance that algorithm will lead to global best solution
- Number of clusters must be determined beforehand
- k-means is limited to linear cluster boundaries
- The fundamental model assumptions of k-means (points will be closer to their own cluster center than to others) means that the algorithm will often be ineffective if the clusters have complicated geometries
- t-distributed stochastic neighbor embedding (t-SNE) is a nonlinear embedding algorithm that is particularly adept at preserving points within clusters

Gaussian Mixture Models

- Extension of k-means

Gaussian Mixture Models

- Extension of k-means
- Can be used for estimation beyond simple clustering

Gaussian Mixture Models

- Extension of k-means
- Can be used for estimation beyond simple clustering
- Two disadvantages of k-means - its lack of flexibility in cluster shape and lack of probabilistic cluster assignment - mean that for many datasets (especially low-dimensional datasets) it may not perform as well as you might hope

Gaussian Mixture Models

- Extension of k-means
- Can be used for estimation beyond simple clustering
- Two disadvantages of k-means - its lack of flexibility in cluster shape and lack of probabilistic cluster assignment - mean that for many datasets (especially low-dimensional datasets) it may not perform as well as you might hope
- GMM attempts to find a mixture of multidimensional Gaussian probability distributions that best model any input dataset

GMM as Density Estimation

- Result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data

GMM as Density Estimation

- Result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data
- Use many more components and ignore the cluster labels

GMM as Density Estimation

- Result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data
- Use many more components and ignore the cluster labels
- Mixture of many Gaussians serves not to find separated clusters of data, but rather to model the overall distribution of the input data

GMM as Density Estimation

- Result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data
- Use many more components and ignore the cluster labels
- Mixture of many Gaussians serves not to find separated clusters of data, but rather to model the overall distribution of the input data
- GMM gives us the recipe to generate new random data distributed similarly to our input

GMM as Density Estimation

- Result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data
- Use many more components and ignore the cluster labels
- Mixture of many Gaussians serves not to find separated clusters of data, but rather to model the overall distribution of the input data
- GMM gives us the recipe to generate new random data distributed similarly to our input
- Cross-validation, AIC, BIC to avoid over-fitting

GMM as Density Estimation

- Result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data
- Use many more components and ignore the cluster labels
- Mixture of many Gaussians serves not to find separated clusters of data, but rather to model the overall distribution of the input data
- GMM gives us the recipe to generate new random data distributed similarly to our input
- Cross-validation, AIC, BIC to avoid over-fitting
- Better to think of GMM as a density estimator rather than a clustering algorithm unless the dataset is simple

GMM as Density Estimation

- Result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data
- Use many more components and ignore the cluster labels
- Mixture of many Gaussians serves not to find separated clusters of data, but rather to model the overall distribution of the input data
- GMM gives us the recipe to generate new random data distributed similarly to our input
- Cross-validation, AIC, BIC to avoid over-fitting
- Better to think of GMM as a density estimator rather than a clustering algorithm unless the dataset is simple
- Kernel density estimation (KDE) is in some senses an algorithm that takes the mixture-of-Gaussians idea to its logical extreme: it uses a mixture consisting of one Gaussian component per point, resulting in a nonparametric estimator of density

Gaussian Processes

- Supervised (generally)

Gaussian Processes

- Supervised (generally)
- Regression and classification

Gaussian Processes

- Supervised (generally)
- Regression and classification
- Unlike other learning models, easy to parameterize for standard inferences

Gaussian Processes

- Supervised (generally)
- Regression and classification
- Unlike other learning models, easy to parameterize for standard inferences
- Relaxes assumptions of conditional independence (spatial, temporal, any variable)

Gaussian Processes

- Supervised (generally)
- Regression and classification
- Unlike other learning models, easy to parameterize for standard inferences
- Relaxes assumptions of conditional independence (spatial, temporal, any variable)
- Bayesian GPs are most robust, but relatively slow

Gaussian Processes

- Supervised (generally)
- Regression and classification
- Unlike other learning models, easy to parameterize for standard inferences
- Relaxes assumptions of conditional independence (spatial, temporal, any variable)
- Bayesian GPs are most robust, but relatively slow
- Many different kernels for different data

Gaussian Processes

- Supervised (generally)
- Regression and classification
- Unlike other learning models, easy to parameterize for standard inferences
- Relaxes assumptions of conditional independence (spatial, temporal, any variable)
- Bayesian GPs are most robust, but relatively slow
- Many different kernels for different data
- TSCS analyses, measurement, text analysis, preference learning, prediction, forecasting, and many more applications

Neural Networks

- Supervised (generally)

Neural Networks

- Supervised (generally)
- Classification and regression (can also be used as a generative model)

Neural Networks

- Supervised (generally)
- Classification and regression (can also be used as a generative model)
- Difficult to make standard inferences

Neural Networks

- Supervised (generally)
- Classification and regression (can also be used as a generative model)
- Difficult to make standard inferences
- Generally best when n is large

Neural Networks

- Supervised (generally)
- Classification and regression (can also be used as a generative model)
- Difficult to make standard inferences
- Generally best when n is large
- Fantastic for image analysis

Neural Networks

- Supervised (generally)
- Classification and regression (can also be used as a generative model)
- Difficult to make standard inferences
- Generally best when n is large
- Fantastic for image analysis
- Other uses include text analysis, prediction, measurement, and more