

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

8a - Dimensionality Reduction I

COM2004/3004

Jon Barker

Department of Computer Science
University of Sheffield

Autumn Semester

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Lecture Objectives

In this lecture we will,

- ▶ Consider the problems of using very large feature vector.
- ▶ Explain what is meant by 'the curse of dimensionality'.
- ▶ We will try to develop an intuition for high dimensional spaces by comparing 1-d and 2-d distributions.
- ▶ Introduce the idea of dimensionality reduction.

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Lecture Objectives

In this lecture we will,

- ▶ Consider the problems of using very large feature vector.
- ▶ Explain what is meant by ‘the curse of dimensionality’.
- ▶ We will try to develop an intuition for high dimensional spaces by comparing 1-d and 2-d distributions.
- ▶ Introduce the idea of dimensionality reduction.

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Lecture Objectives

In this lecture we will,

- ▶ Consider the problems of using very large feature vector.
- ▶ Explain what is meant by ‘the curse of dimensionality’.
- ▶ We will try to develop an intuition for high dimensional spaces by comparing 1-d and 2-d distributions.
- ▶ Introduce the idea of dimensionality reduction.

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Lecture Objectives

In this lecture we will,

- ▶ Consider the problems of using very large feature vector.
- ▶ Explain what is meant by ‘the curse of dimensionality’.
- ▶ We will try to develop an intuition for high dimensional spaces by comparing 1-d and 2-d distributions.
- ▶ Introduce the idea of dimensionality reduction.

Recap

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

- ▶ Last week we looked at Feature Selection.
- ▶ Using large numbers of features is problematic (more on this today)
 - ▶ May increase the number of training examples we need for good performance.
 - ▶ For some classifiers adds to the memory costs.
 - ▶ Nearly always increases computational cost.
- ▶ Feature Selection is the process of reducing the feature vector size by choosing a small subset of the raw feature set.

Recap

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

- ▶ Last week we looked at Feature Selection.
- ▶ Using large numbers of features is problematic (more on this today)
 - ▶ May increase the number of training examples we need for good performance.
 - ▶ For some classifiers adds to the memory costs.
 - ▶ Nearly always increases computational cost.
- ▶ Feature Selection is the process of reducing the feature vector size by choosing a small subset of the raw feature set.

Recap

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

- ▶ Last week we looked at Feature Selection.
- ▶ Using large numbers of features is problematic (more on this today)
 - ▶ May increase the number of training examples we need for good performance.
 - ▶ For some classifiers adds to the memory costs.
 - ▶ Nearly always increases computational cost.
- ▶ Feature Selection is the process of reducing the feature vector size by choosing a small subset of the raw feature set.

Recap: a feature selection system

- ▶ We want to select a set of features that maximises classification performance (i.e. produces few classification errors).
- ▶ There are typically two components to a feature selection system: a **class separability measure**, and the **feature selection algorithm**.
- ▶ The separability measure is a measure of how separated the classes are given a particular set of features (e.g. divergence).
- ▶ We use a separability measure because we can't usually find the set of features by trying them on the test data directly.
- ▶ For a well-designed separability measure a high separability will be a good predictor of low classification error rate.
- ▶ Designing a good separability measure is not easy.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: a feature selection system

- ▶ We want to select a set of features that maximises classification performance (i.e. produces few classification errors).
- ▶ There are typically two components to a feature selection system: a **class separability measure**, and the **feature selection algorithm**.
- ▶ The separability measure is a measure of how separated the classes are given a particular set of features (e.g. divergence).
- ▶ We use a separability measure because we can't usually find the set of features by trying them on the test data directly.
- ▶ For a well-designed separability measure a high separability will be a good predictor of low classification error rate.
- ▶ Designing a good separability measure is not easy.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: a feature selection system

- ▶ We want to select a set of features that maximises classification performance (i.e. produces few classification errors).
- ▶ There are typically two components to a feature selection system: a **class separability measure**, and the **feature selection algorithm**.
- ▶ The separability measure is a measure of how separated the classes are given a particular set of features (e.g. divergence).
- ▶ We use a separability measure because we can't usually find the set of features by trying them on the test data directly.
- ▶ For a well-designed separability measure a high separability will be a good predictor of low classification error rate.
- ▶ Designing a good separability measure is not easy.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: a feature selection system

- ▶ We want to select a set of features that maximises classification performance (i.e. produces few classification errors).
- ▶ There are typically two components to a feature selection system: a **class separability measure**, and the **feature selection algorithm**.
- ▶ The separability measure is a measure of how separated the classes are given a particular set of features (e.g. divergence).
- ▶ We use a separability measure because we can't usually find the set of features by trying them on the test data directly.
- ▶ For a well-designed separability measure a high separability will be a good predictor of low classification error rate.
- ▶ Designing a good separability measure is not easy.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: a feature selection system

- ▶ We want to select a set of features that maximises classification performance (i.e. produces few classification errors).
- ▶ There are typically two components to a feature selection system: a **class separability measure**, and the **feature selection algorithm**.
- ▶ The separability measure is a measure of how separated the classes are given a particular set of features (e.g. divergence).
- ▶ We use a separability measure because we can't usually find the set of features by trying them on the test data directly.
- ▶ For a well-designed separability measure a high separability will be a good predictor of low classification error rate.
- ▶ Designing a good separability measure is not easy.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: a feature selection system

- ▶ We want to select a set of features that maximises classification performance (i.e. produces few classification errors).
- ▶ There are typically two components to a feature selection system: a **class separability measure**, and the **feature selection algorithm**.
- ▶ The separability measure is a measure of how separated the classes are given a particular set of features (e.g. divergence).
- ▶ We use a separability measure because we can't usually find the set of features by trying them on the test data directly.
- ▶ For a well-designed separability measure a high separability will be a good predictor of low classification error rate.
- ▶ Designing a good separability measure is not easy.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: feature selection algorithms

- ▶ **Rank features** according to the class separability that each provides and take the best N .
- ▶ **Forward sequential selection**: build the vector up starting with 1 feature and picking the vector that maximises separability at each size.
- ▶ **Backward sequential selection**: prune the vector down starting with all the features and picking the vector the maximises separability at each size.
- ▶ **Brute force**: try all (or large proportion of all) possible feature vectors. Only really possible for toy problems.
- ▶ Many other more complicated variants.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: feature selection algorithms

- ▶ **Rank features** according to the class separability that each provides and take the best N .
- ▶ **Forward sequential selection**: build the vector up starting with 1 feature and picking the vector that maximises separability at each size.
- ▶ **Backward sequential selection**: prune the vector down starting with all the features and picking the vector the maximises separability at each size.
- ▶ **Brute force**: try all (or large proportion of all) possible feature vectors. Only really possible for toy problems.
- ▶ Many other more complicated variants.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: feature selection algorithms

- ▶ **Rank features** according to the class separability that each provides and take the best N .
- ▶ **Forward sequential selection**: build the vector up starting with 1 feature and picking the vector that maximises separability at each size.
- ▶ **Backward sequential selection**: prune the vector down starting with all the features and picking the vector the maximises separability at each size.
- ▶ **Brute force**: try all (or large proportion of all) possible feature vectors. Only really possible for toy problems.
- ▶ Many other more complicated variants.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: feature selection algorithms

- ▶ **Rank features** according to the class separability that each provides and take the best N .
- ▶ **Forward sequential selection**: build the vector up starting with 1 feature and picking the vector that maximises separability at each size.
- ▶ **Backward sequential selection**: prune the vector down starting with all the features and picking the vector the maximises separability at each size.
- ▶ **Brute force**: try all (or large proportion of all) possible feature vectors. Only really possible for toy problems.
- ▶ Many other more complicated variants.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Recap: feature selection algorithms

- ▶ **Rank features** according to the class separability that each provides and take the best N .
- ▶ **Forward sequential selection**: build the vector up starting with 1 feature and picking the vector that maximises separability at each size.
- ▶ **Backward sequential selection**: prune the vector down starting with all the features and picking the vector the maximises separability at each size.
- ▶ **Brute force**: try all (or large proportion of all) possible feature vectors. Only really possible for toy problems.
- ▶ Many other more complicated variants.

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

overview

► Dimensionality Reduction

- ▶ Feature Selection is one particular type of *dimensionality reduction*, i.e. we just throw away the least useful dimensions.
- ▶ This week we will be looking at a more general way of reducing dimensionality.
- ▶ **Today:** Further discussion of problems caused by high dimensionality.
- ▶ **Next Lecture:** Principle Components Analysis; Linear Discriminant Analysis

Face Recognition

- ▶ Attach a name label to an unidentified face image (classification task).
- ▶ Classifier is trained using a number of labelled examples for each individual.



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Many security applications:

Example 1

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

Example 2

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Representation of an image

- ▶ Assume all face images have been rescaled to a fixed size, e.g., 17 by 17 pixel.
- ▶ For a grey scale image, each pixel is represented by an 8-bit number (i.e., 0-255) corresponding to the grey level.
 - ▶ typically 0 = black, 255 = white



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Representation of an image

- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

Representation of an image

- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

Representation of an image

- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Representation of an image

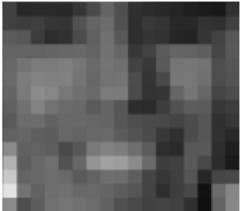
- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Representation of an image

- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Representation of an image

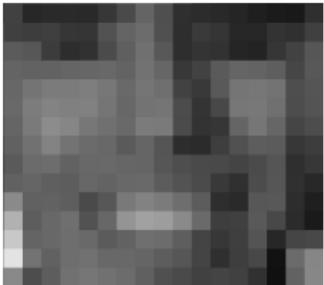
- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

Representation of an image

- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

Representation of an image

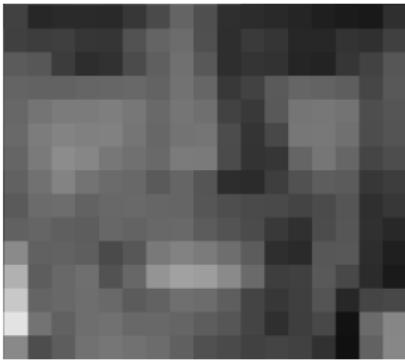
- ▶ If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

Representation of an image

- If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

Representation of an image

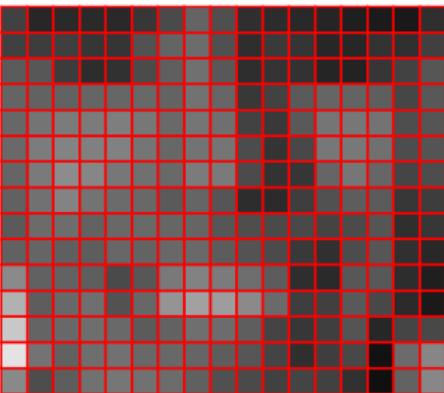
- If we zoom in we can see the individual pixels.



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Representation of an image

- If we zoom in we can see the individual pixels.



- The n by n grid of grey levels can be laid out into one long feature vector containing n^2 elements.
- For a 17 by 17 image the feature vector would contain 289 features, $\mathbf{x} = \{x_1, x_2, \dots, x_{289}\}$.

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ **Characters = 1 of 26; Faces = 1 of 100's or 1000's.**
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ Characters = 1 of 26; Faces = 1 of 100's or 1000's.
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ Characters = 1 of 26; Faces = 1 of 100's or 1000's.
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ Characters = 1 of 26; Faces = 1 of 100's or 1000's.
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ Characters = 1 of 26; Faces = 1 of 100's or 1000's.
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ Characters = 1 of 26; Faces = 1 of 100's or 1000's.
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ Characters = 1 of 26; Faces = 1 of 100's or 1000's.
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Aside: Comparing face recognition and character recognition

- ▶ Characters = 1 of 26; Faces = 1 of 100's or 1000's.
- ▶ Characters, 2-d images; Faces 2-d projections of a 3-d form.
- ▶ Characters, pixels essentially black/white; Faces, grey level is important.
- ▶ Characters, low resolution may suffice; Faces, probably need higher resolution (i.e. more pixels)
- ▶ Characters, separable by local features; Faces, holistic, c.f. Gestalt perception
- ▶ Characters, within-class variability probably low; Face, within-class variability very high.
- ▶ Characters, class clearly separable; Faces, classes possibly less separable.
- ▶ Conclusion: Face recognition is a harder problem. Would our simple feature selection strategy be effective?

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Jon Barker

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Jon Barker

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Jon Barker

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Jon Barker

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Jon Barker

Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



Recap

Overview

Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

Curse of dimensionality

- ▶ Large feature vectors are bad news...
- ▶ Remember, classification depends on comparing $p(\mathbf{x})$ for each class of data.
- ▶ $p(\mathbf{x})$ – the true distribution – isn't ever known precisely. It is estimated from example training data.
- ▶ Now, if \mathbf{x} has n elements then $p(\mathbf{x})$ is an n dimensional distribution, e.g. $p(\mathbf{x})$ for the raw face images would be a $17 \times 17 = 289$ dimensional distribution.
- ▶ Let's consider the implications of this.

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

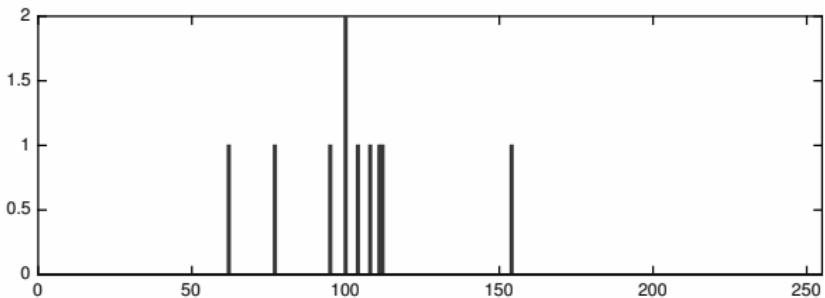
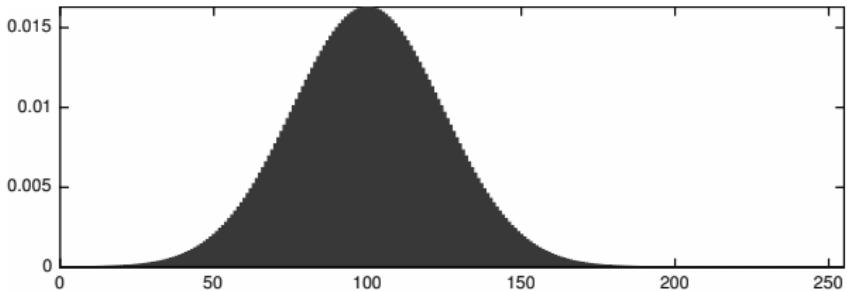
Estimating a 1-d distribution

- ▶ Consider the distribution of gray levels for a single pixel, $p(x)$
- ▶ x can have any one of 256 values, so the true $p(x)$ can be represented exactly by a histogram with 256 bins.
- ▶ We estimate the true $p(x)$ by constructing a histogram from training examples
- ▶ $p(x = n) = \text{number of times } x \text{ has the value } n \text{ in the training data} / \text{total number of training examples}$
- ▶ The more training examples that are used the more accurate the histogram becomes, i.e. the closer the estimated $p(x)$ will be to the true $p(x)$.

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 1-d distribution

- ▶ Estimating the true histogram using progressively more samples

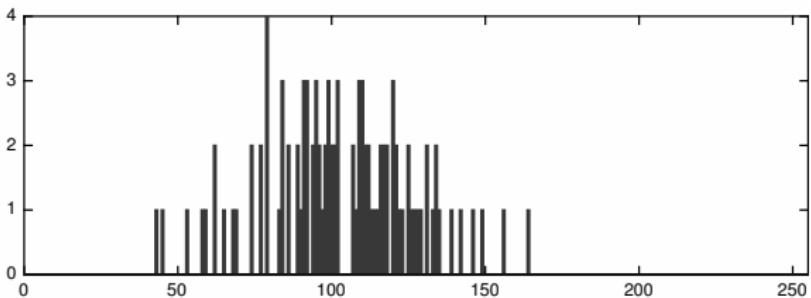
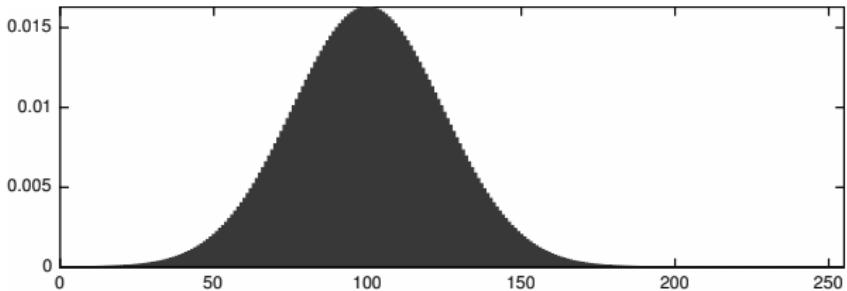


- ▶ 10 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 1-d distribution

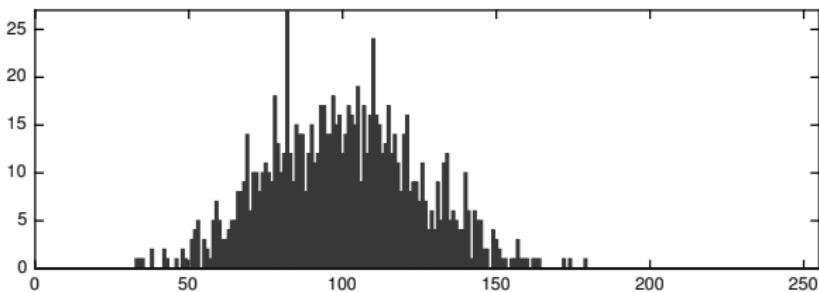
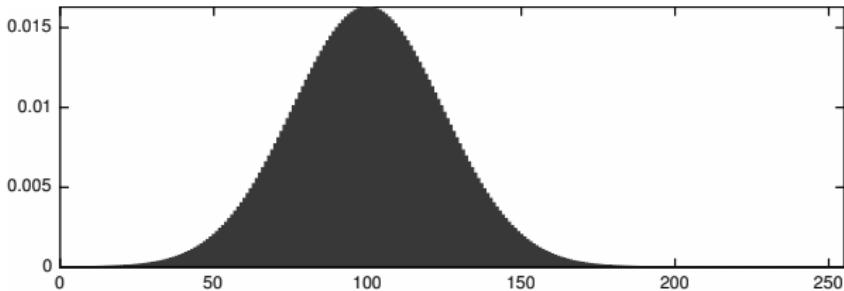
- ▶ Estimating the true histogram using progressively more samples



- ▶ 100 samples

Estimating a 1-d distribution

- ▶ Estimating the true histogram using progressively more samples



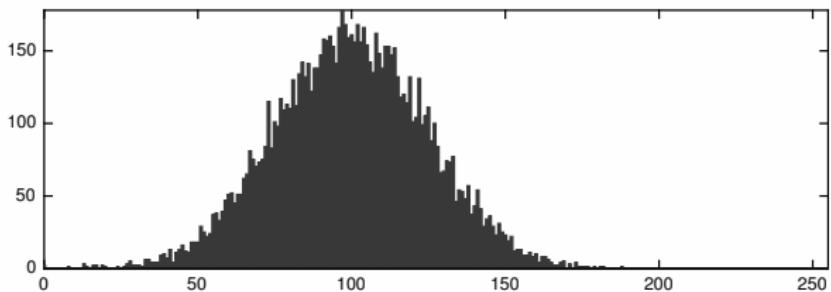
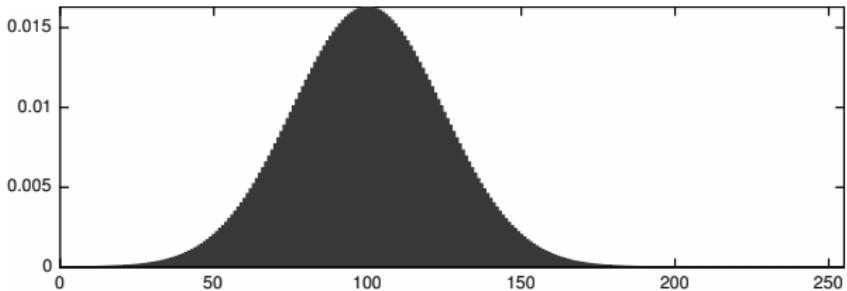
- ▶ 1,000 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 1-d distribution

- ▶ Estimating the true histogram using progressively more samples

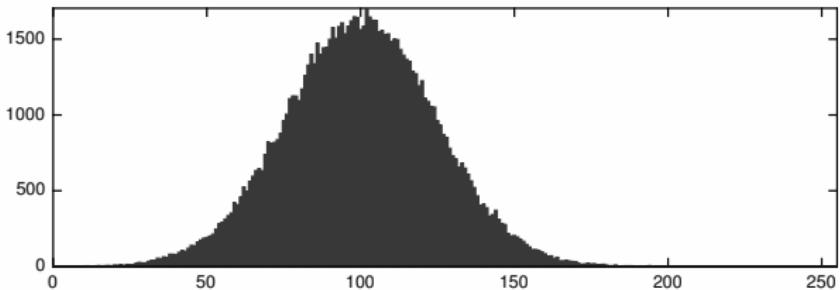
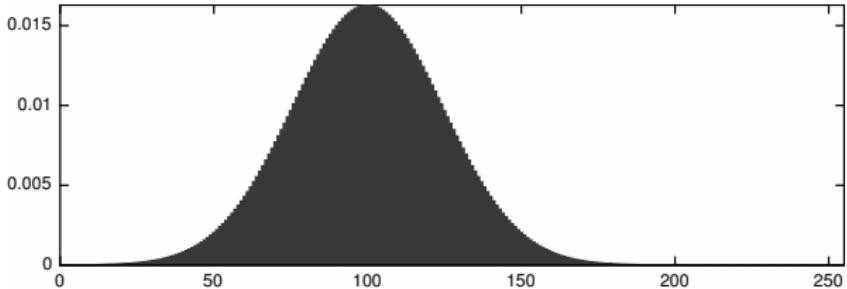


- ▶ 10,000 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 1-d distribution

- ▶ Estimating the true histogram using progressively more samples



- ▶ 100,000 samples

Recap

Overview

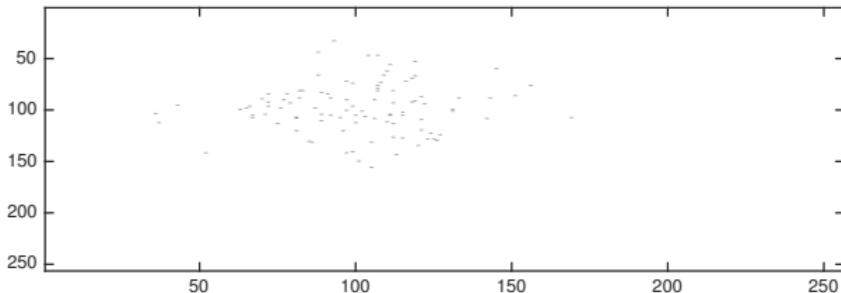
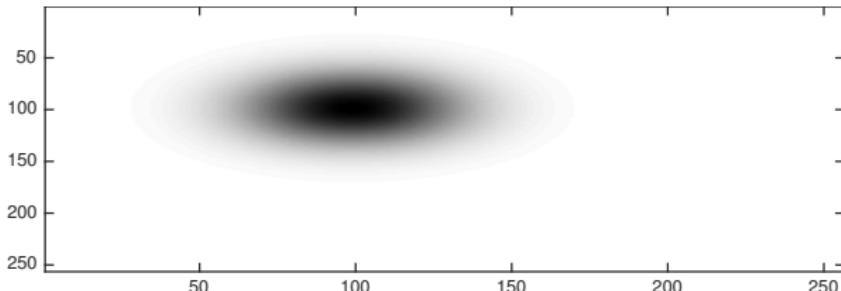
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Estimating a 2-d distribution

- ▶ 2-D histogram represented as an image



- ▶ 100 samples
- ▶ With the extra dimension, more samples are needed to make sure enough fall in each histogram bin.

Recap

Overview

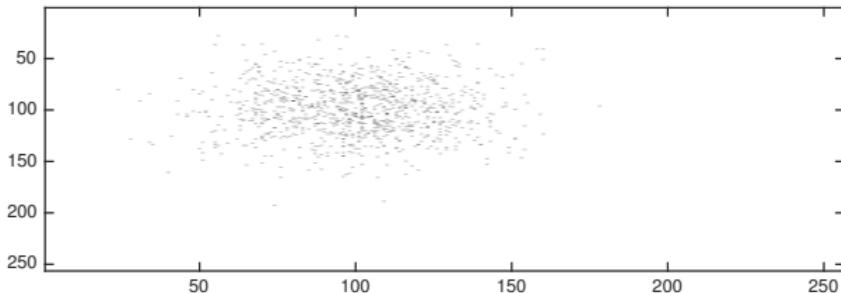
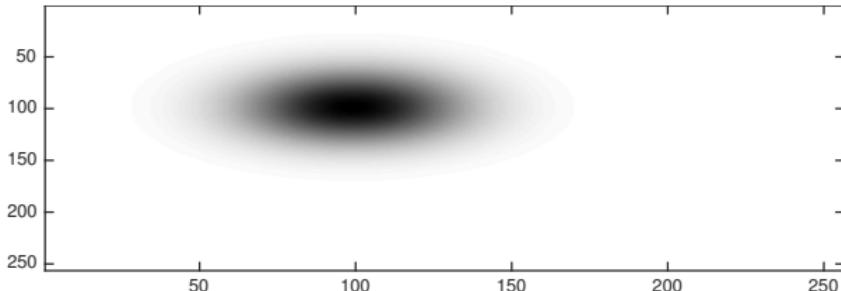
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Estimating a 2-d distribution

- ▶ 2-D histogram represented as an image



- ▶ 1,000 samples
- ▶ With the extra dimension, more samples are needed to make sure enough fall in each histogram bin.

Recap

Overview

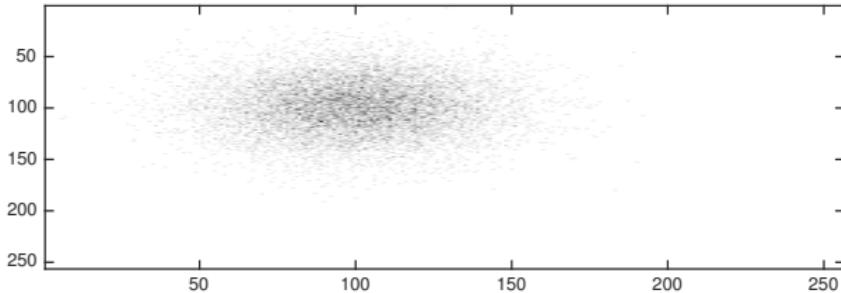
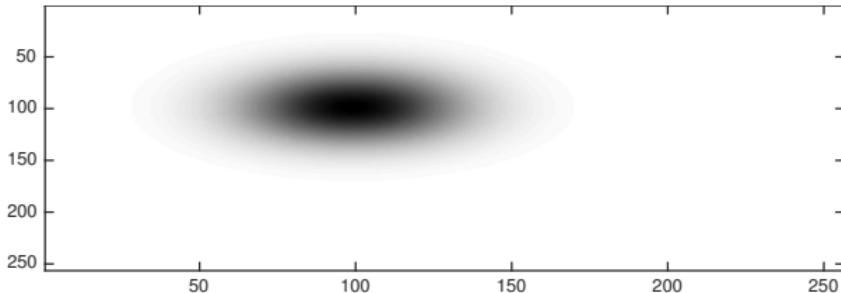
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Estimating a 2-d distribution

- ▶ 2-D histogram represented as an image



- ▶ 10,000 samples
- ▶ With the extra dimension, more samples are needed to make sure enough fall in each histogram bin.

Recap

Overview

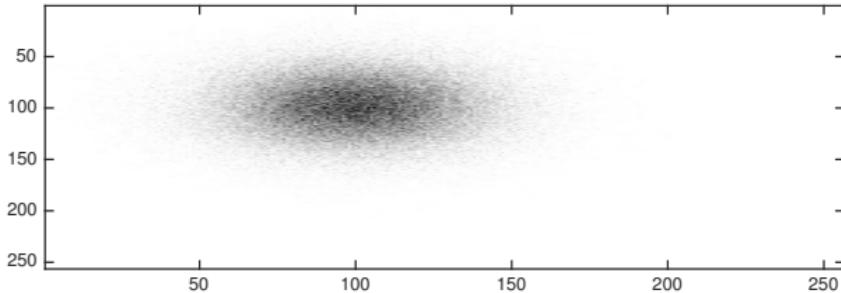
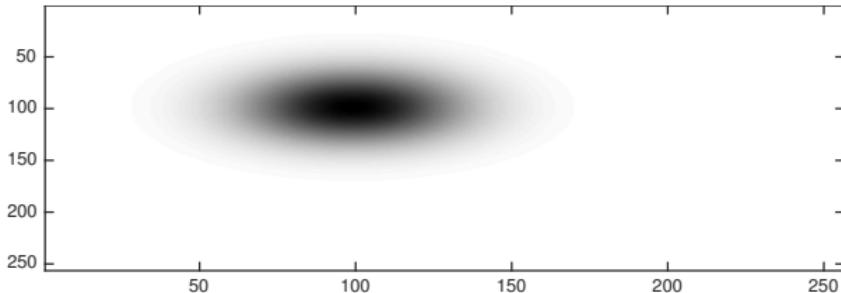
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Estimating a 2-d distribution

- ▶ 2-D histogram represented as an image



- ▶ 100,000 samples
- ▶ With the extra dimension, more samples are needed to make sure enough fall in each histogram bin.

Recap

Overview

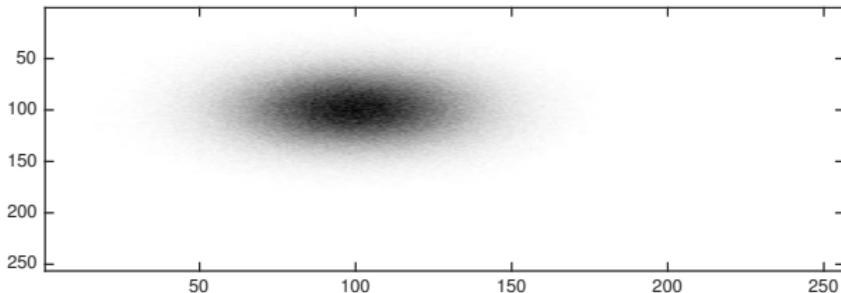
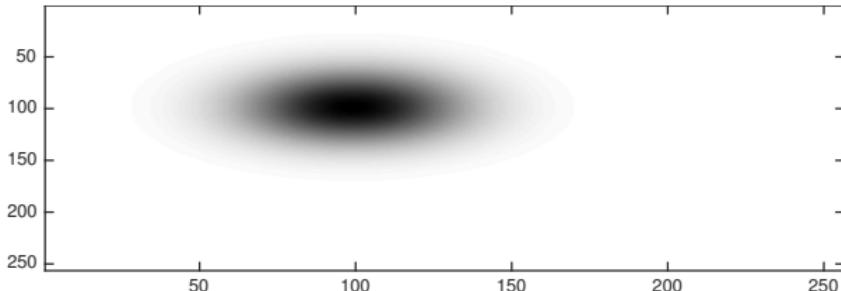
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Estimating a 2-d distribution

- ▶ 2-D histogram represented as an image



- ▶ 1,000,000 samples
- ▶ With the extra dimension, more samples are needed to make sure enough fall in each histogram bin.

Recap

Overview

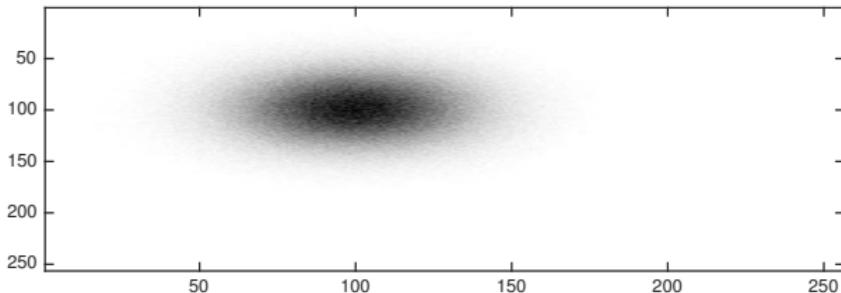
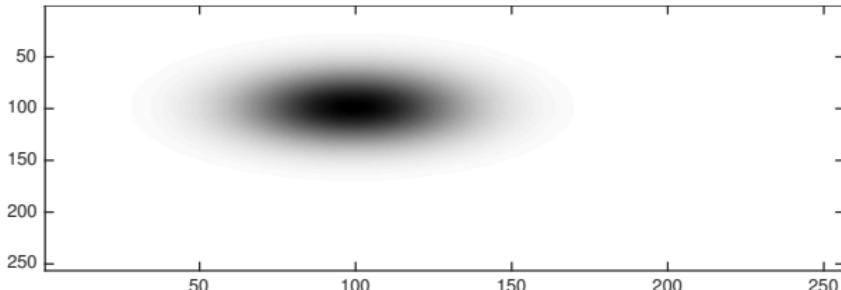
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Estimating a 2-d distribution

- ▶ 2-D histogram represented as an image



- ▶ 1,000,000 samples
- ▶ With the extra dimension, more samples are needed to make sure enough fall in each histogram bin.

1-d slice through 2-d histogram

8a - Dimensionality

Reduction I

Jon Barker

Recap

Overview

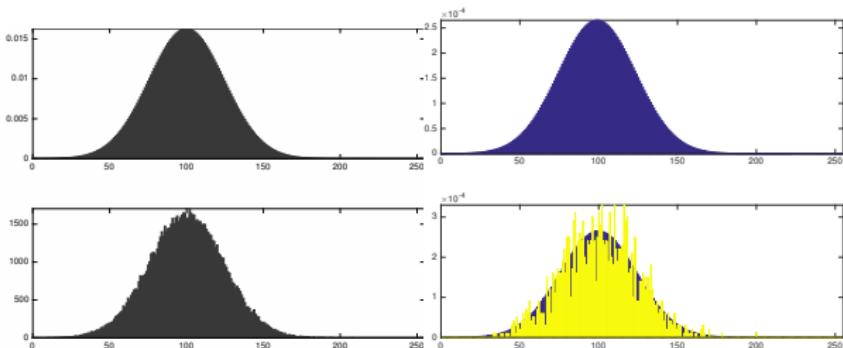
Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

- ▶ Whereas 100,000 samples produced a good approximation to the 1-d distribution, it is far less good for the 2-d distribution.
- ▶ Left: 1d; Right: slice through 2d



- ▶ 100,000 samples versus 100,000 samples

1-d slice through 2-d histogram

8a - Dimensionality

Reduction I

Jon Barker

Recap

Overview

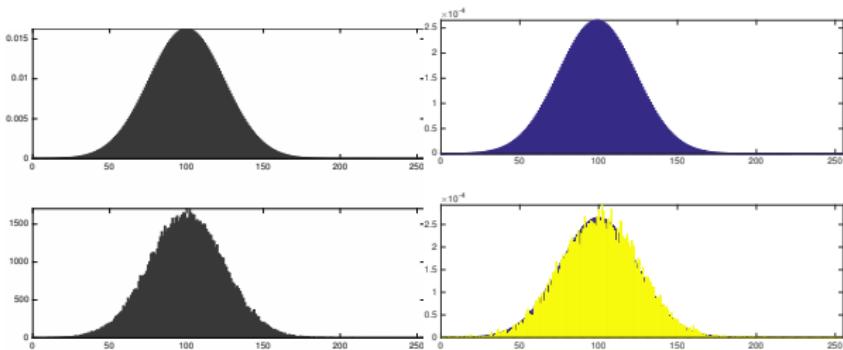
Large feature vectors

Example: Face
Recognition

Curse of
dimensionality

Summary

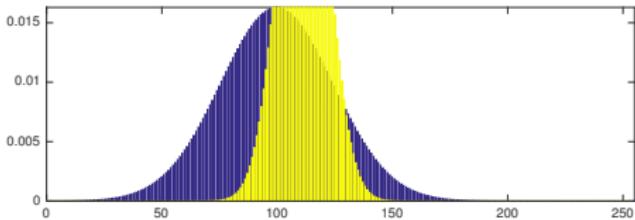
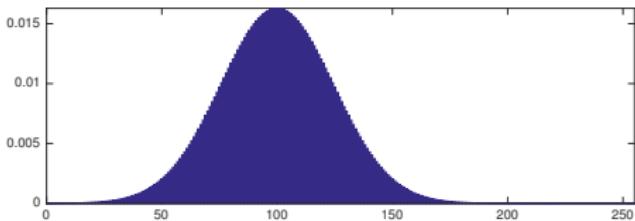
- ▶ Whereas 100,000 samples produced a good approximation to the 1-d distribution, it is far less good for the 2-d distribution.
- ▶ Left: 1d; Right: slice through 2d



- ▶ 100,000 samples versus 1,000,000 samples

Using a parametric model

- ▶ In practice, we would rarely estimate the histogram directly.
- ▶ Instead we would use a parametric model: here we make an assumption about the parametric form of the true distribution (e.g. Gaussian, Laplacian etc) and then attempt to estimate the model parameters (e.g. mean and variance for a Gaussian)

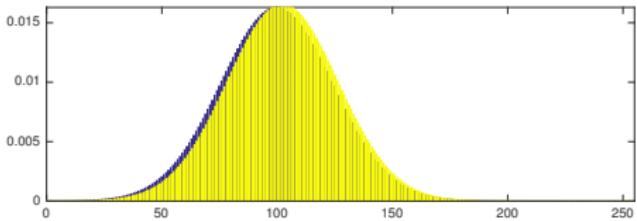
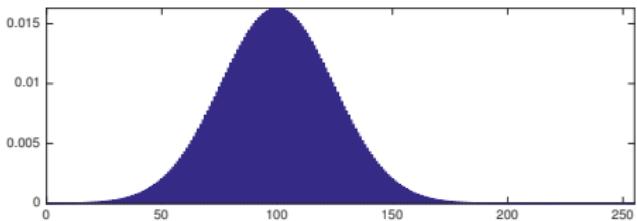


- ▶ 3 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Using a parametric model

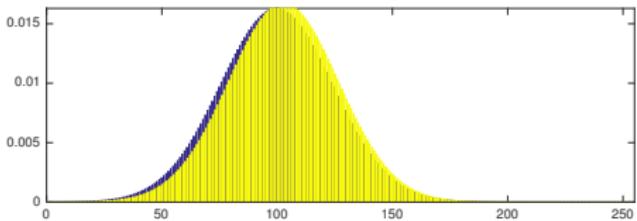
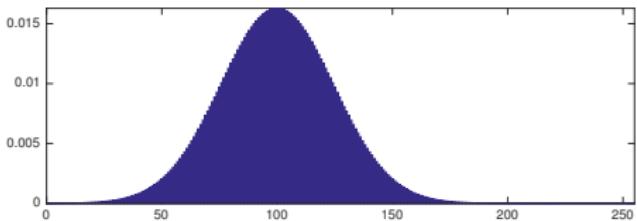
- ▶ In practice, we would rarely estimate the histogram directly.
- ▶ Instead we would use a parametric model: here we make an assumption about the parametric form of the true distribution (e.g. Gaussian, Laplacian etc) and then attempt to estimate the model parameters (e.g. mean and variance for a Gaussian)



- ▶ 10 samples

Using a parametric model

- ▶ In practice, we would rarely estimate the histogram directly.
- ▶ Instead we would use a parametric model: here we make an assumption about the parametric form of the true distribution (e.g. Gaussian, Laplacian etc) and then attempt to estimate the model parameters (e.g. mean and variance for a Gaussian)



- ▶ 100 samples

Recap

Overview

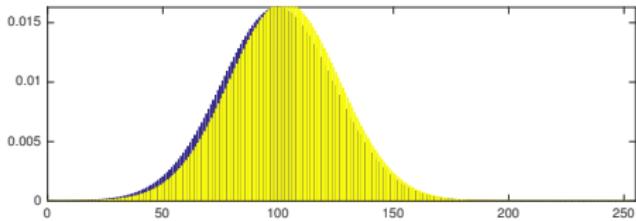
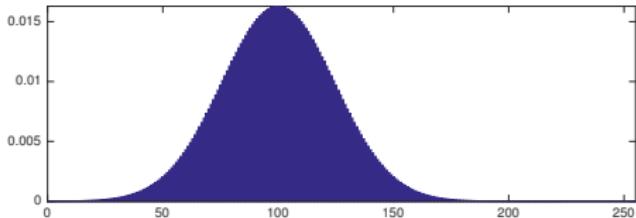
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Using a parametric model

- ▶ In practice, we would rarely estimate the histogram directly.
- ▶ Instead we would use a parametric model: here we make an assumption about the parametric form of the true distribution (e.g. Gaussian, Laplacian etc) and then attempt to estimate the model parameters (e.g. mean and variance for a Gaussian)



- ▶ 100 samples
- ▶ Note, far fewer training samples are needed. But the modelling assumption must be correct.

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Using a parametric model

- ▶ Even with a parametric model we need more samples to estimate the model parameters as the dimensionality increases
- ▶ Number of free model parameters increases as dimensionality increases
 - ▶ 1-d gaussian has 2 parameters (mean, variance)
 - ▶ 2-d gaussian has 5 parameters (2 means, 2 variances, 1 covariance)
 - ▶ 3-d gaussian has 9 parameters (3 means, 3 variances, 3 covariances)
 - ▶ n -d has $\frac{1}{2}(n + 3)$ i.e., $O(n^2)$

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Using a parametric model

- ▶ Even with a parametric model we need more samples to estimate the model parameters as the dimensionality increases
- ▶ Number of free model parameters increases as dimensionality increases
 - ▶ 1-d gaussian has 2 parameters (mean, variance)
 - ▶ 2-d gaussian has 5 parameters (2 means, 2 variances, 1 covariance)
 - ▶ 3-d gaussian has 9 parameters (3 means, 3 variances, 3 covariances)
 - ▶ n -d has $\frac{n}{2}(n + 3)$ i.e., $O(n^2)$

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Using a parametric model

- ▶ Even with a parametric model we need more samples to estimate the model parameters as the dimensionality increases
- ▶ Number of free model parameters increases as dimensionality increases
 - ▶ 1-d gaussian has 2 parameters (mean, variance)
 - ▶ 2-d gaussian has 5 parameters (2 means, 2 variances, 1 covariance)
 - ▶ 3-d gaussian has 9 parameters (3 means, 3 variances, 3 covariances)
 - ▶ n -d has $\frac{n}{2}(n + 3)$ i.e., $O(n^2)$

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Using a parametric model

- ▶ Even with a parametric model we need more samples to estimate the model parameters as the dimensionality increases
- ▶ Number of free model parameters increases as dimensionality increases
 - ▶ 1-d gaussian has 2 parameters (mean, variance)
 - ▶ 2-d gaussian has 5 parameters (2 means, 2 variances, 1 covariance)
 - ▶ 3-d gaussian has 9 parameters (3 means, 3 variances, 3 covariances)
 - ▶ n -d has $\frac{n}{2}(n + 3)$ i.e., $O(n^2)$

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

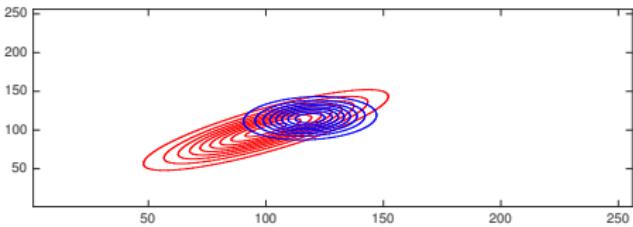
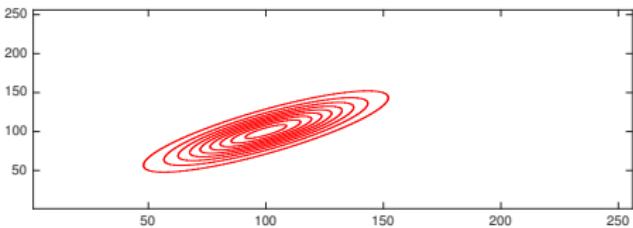
Using a parametric model

- ▶ Even with a parametric model we need more samples to estimate the model parameters as the dimensionality increases
- ▶ Number of free model parameters increases as dimensionality increases
 - ▶ 1-d gaussian has 2 parameters (mean, variance)
 - ▶ 2-d gaussian has 5 parameters (2 means, 2 variances, 1 covariance)
 - ▶ 3-d gaussian has 9 parameters (3 means, 3 variances, 3 covariances)
 - ▶ n -d has $\frac{n}{2}(n + 3)$ i.e., $O(n^2)$

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 2 D gaussian

- ▶ Contours of true distribution versus estimate using increasing number of samples

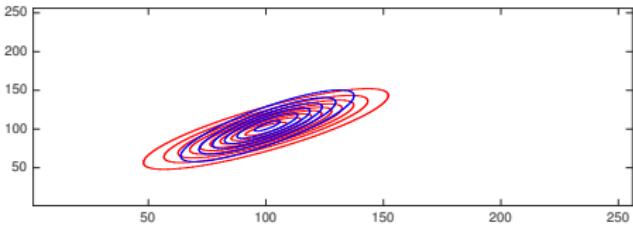
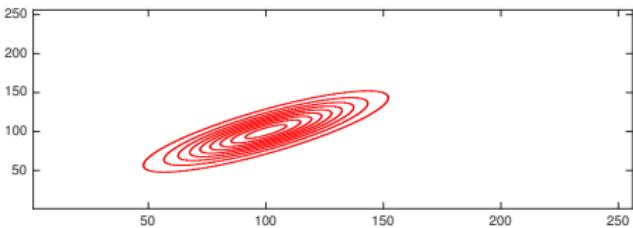


- ▶ 3 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 2 D gaussian

- ▶ Contours of true distribution versus estimate using increasing number of samples

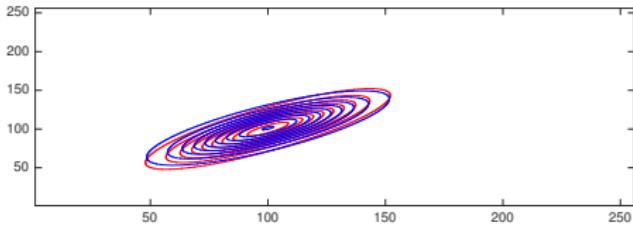
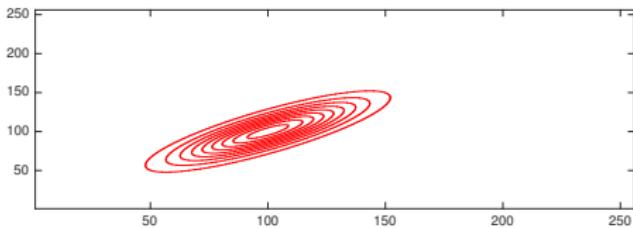


- ▶ 10 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 2 D gaussian

- ▶ Contours of true distribution versus estimate using increasing number of samples

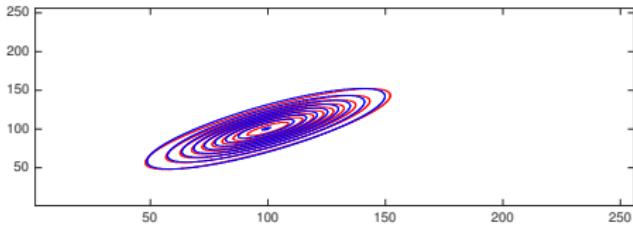
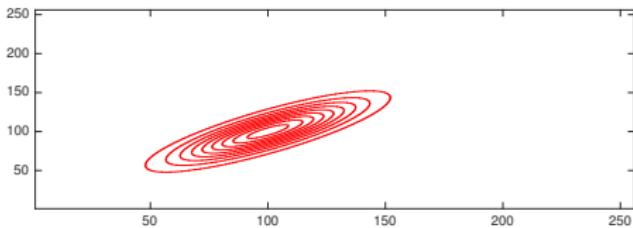


- ▶ 100 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

Estimating a 2 D gaussian

- ▶ Contours of true distribution versus estimate using increasing number of samples

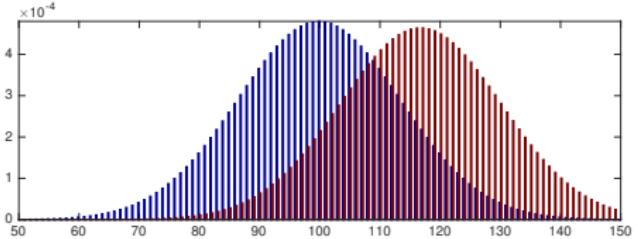
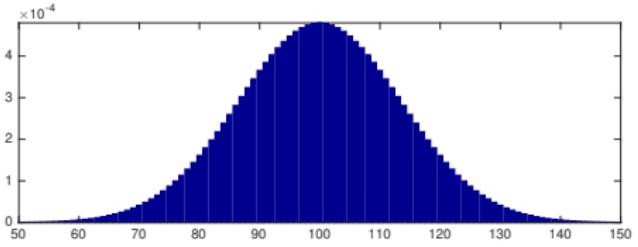


- ▶ 500 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

1-D slice through 2-D gaussian

- ▶ Estimate converges on true distribution as number of samples increases

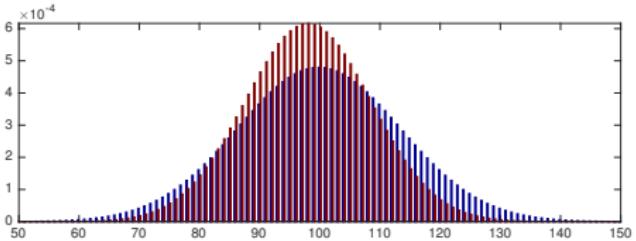
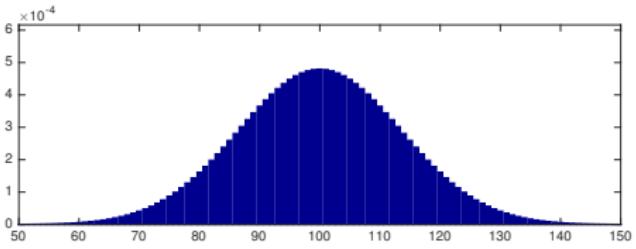


- ▶ 3 samples
- ▶ Match becomes close at around 500 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

1-D slice through 2-D gaussian

- ▶ Estimate converges on true distribution as number of samples increases

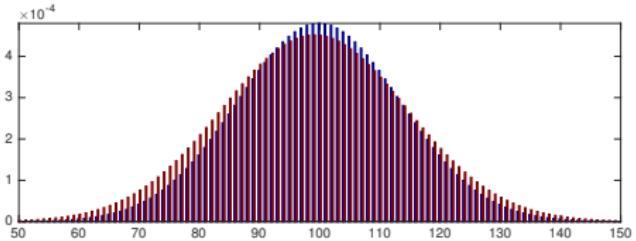
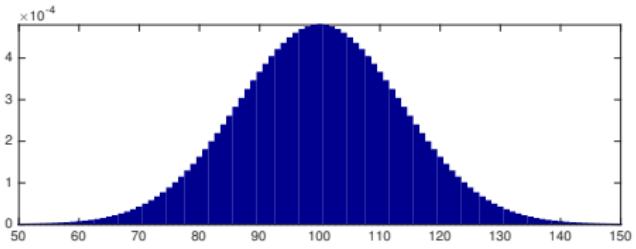


- ▶ 10 samples
- ▶ Match becomes close at around 500 samples

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

1-D slice through 2-D gaussian

- ▶ Estimate converges on true distribution as number of samples increases



- ▶ 100 samples
- ▶ Match becomes close at around 500 samples

Recap

Overview

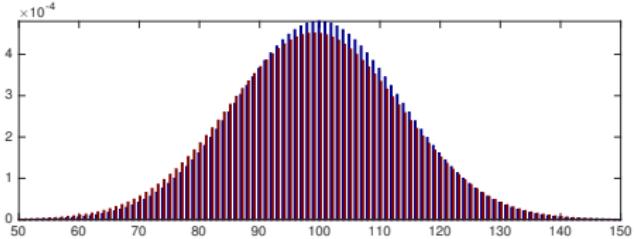
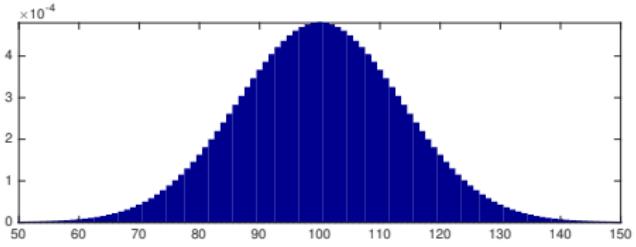
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

1-D slice through 2-D gaussian

- ▶ Estimate converges on true distribution as number of samples increases



- ▶ 500 samples
- ▶ Match becomes close at around 500 samples

Recap

Overview

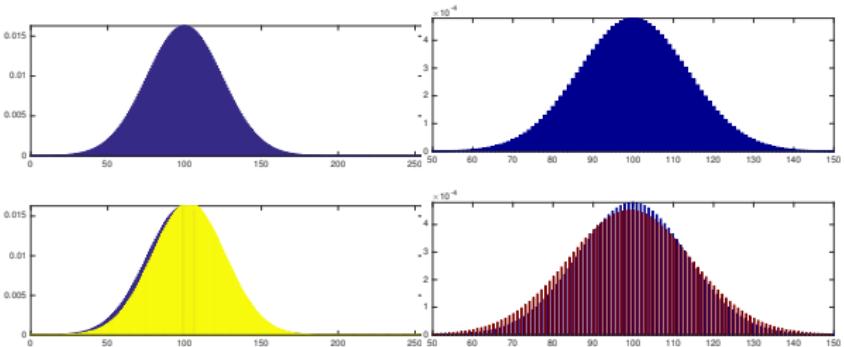
Large feature vectors

Example: Face
RecognitionCurse of
dimensionality

Summary

Comparing fit for 1-D and 2-D distributions

- ▶ Using 100 samples parameters of 1-D distribution well-estimated, whereas those of 2-D distribution are not.
- ▶ Left: 1d; Right: slice through 2d



[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face
Recognition](#)[Curse of
dimensionality](#)[Summary](#)

How can we reduce dimensionality of x

Consider our face data.

- ▶ Select some subset of elements, e.g. keep just a line of pixels down the center of the image
 - ▶ Will loose information.
 - ▶ How do we select which pixels to keep... ?
- ▶ Use feature selection techniques like those discussed last week
 - ▶ But are any individual pixels likely to discriminate between classes?
 - ▶ People can't be identified by looking at individual pixels.
 - ▶ Need to find features that are less 'local'.
- ▶ We are going to exploit the correlation between the features
 - ▶ Note, adjacent pixels tend to have similar values – they are correlated
 - ▶ A pair of correlated features hold less information than a pair of independent features
 - ▶ Intuitively, the 'effective' dimensionality of the face images is less than 17×17

[Recap](#)[Overview](#)[Large feature vectors](#)[Example: Face Recognition](#)[Curse of dimensionality](#)[Summary](#)

- ▶ If \mathbf{x} has a small number of elements, then $p(\mathbf{x})$ can be modelled with a smaller number of parameters
 - ▶ the parameters will require less storage
 - ▶ the computation of $p(\mathbf{x})$ will be faster
 - ▶ the parameters can then be robustly estimated with a smaller number of training examples
- ▶ Feature selection is a form of dimensionality reduction.
- ▶ However, it is not always appropriate: e.g. faces probably can't be robustly classified on the basis of a small number of pixel values.