Recognition

Computer Vision I
CSE252A
Lecture 20

How is this formulation used
1. It’s ignored. At each time instant, the state is estimated (perhaps a maximum likelihood estimate or something non-probabilistic)
2. The conditional distributions are represented by some convenient parametric form (e.g., Gaussian).
3. The PDF’s are represented non-parametrically, and sampling techniques are used.

Multi-variate Kalman Filter

\[
\begin{align*}
    x_{k} & \sim N(x_{k-1}, P_{k-1}) \\
    z_{k} & \sim N(h(x_{k}, u_{k}), R_{k})
\end{align*}
\]

Start Assumptions $P_{0}$ and $R_{0}$ are known

Update Equations: Prediction
\[
\begin{align*}
    x_{k|k-1} &= F_{k} x_{k-1} + B_{k} u_{k} \\
    P_{k|k-1} &= F_{k} P_{k-1} F_{k}^{T} + Q_{k}
\end{align*}
\]

Update Equations: Correction
\[
\begin{align*}
    K_{k} &= P_{k|k-1} H_{k}^{T} (H_{k} P_{k|k-1} H_{k}^{T} + R_{k})^{-1} \\
    x_{k} &= x_{k|k-1} + K_{k} (z_{k} - H_{k} x_{k|k-1}) \\
    P_{k} &= (I - K_{k} H_{k}) P_{k|k-1}
\end{align*}
\]

Tracking Modalities
(Define the features $Y$)

- Color
  - Histogram [Birchfield 1998; Brdicka 1998]
  - Volume [Wern et al., 1993; Bregler, 1997; Durrell, 1998]
- Shape
  - Deformable curve [Kass et al. 1988]
  - Template [Blake et al., 1993; Birchfield 1998]
  - Example-based [Cootes et al., 1995; Baumgart & Hogg, 1996]
- Appearance
  - Correlation [Lucas & Kanade, 1981; Shi & Tomasi, 1994]
  - Photometric variation [Hager & Belhumeur, 1998]
  - Outliers [Black et al., 1990; Hager & Belhumeur, 1998]
  - Non-rigidity [Black et al., 1990; Schnorr & Isidoro, 1998]

Color Blob tracking

- Color-based tracker gets lost on white knight: Same Color
Snakes: Active Contours

- Contour C: continuous curve on smooth surface in $\mathbb{R}^3$
- Snake S: projection of C to image
- Curve types
  - Edge between regions on surface with contrasting properties
  - Line that contrasts with surface properties on both side
  - Silhouette of surface against contrasting background
- General Algorithm:
  - Perform edge detection
  - Fit parametric or non-parametric curve to data

Snakes: Basic Approach

- Parameterize a closed contour
  \[ Q = (x^1, x^2, x^3, y^1, y^2, y^3) \]
  \[ r(s) = q'B(s) \quad \text{or} \quad r(s) = U(s)Q \]
- Given a predicted state $q$, search radially for edges
- Solve a least squares problem for new state

Tracker Composition: Only Shape (Snakes)

- Geometry-based tracker gets lost on black pawn: Same shape

Tracker Composition

Tracker Composition: Color and Shape

- Combining Trackers $\Rightarrow$ Robustness
- Trackers in video, IR and range

Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.
Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.

Object Recognition: The Problem

Given: A database D of “known” objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object

Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional
- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability (perspective, orthographic projection)
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial
- Clutter in background -> false positives
Object Recognition System Design Considerations

- How general is the problem?
  - 2D vs. 3D
  - range of viewing conditions
  - available context
  - segmentation cues

- What sort of data is best suited to the problem?
  - Whole images
  - Local 2D features (color, texture, 3D range)

- What information do we have in the database?
  - Collection of images?
  - 3D model(s)?
  - Learned representation?
  - Learned classifiers?

- How many objects are involved?
  - small: brute force search
  - large: ??

A Rough Recognition Spectrum

Appearance-based Recognition

Local Features + Spatial Relations

Shape Contexts

Geometric Invariants

Increasing Generality

3-D Model-Based Recognition

Image Abstractions/Volumetric Primitives

Function

Appearance-Based Recognition

(Eigenface, Fisherface)

Local Features + Spatial Relations

3-D Model-Based Recognition

Increasing Generality

Sketch of a Pattern Recognition Architecture

Feature Extraction

Classification

Object Identity

Image (window)

Feature Vector

Example: Face Detection

- Scan window over image.
- Classify window as either:
  - Face
  - Non-face

- So, what are the features?
- So, what is the classifier
Simplest feature:
Image as a Feature Vector

- Consider an n-pixel image to be a point in an n-dimensional space, \( x \in \mathbb{R}^n \).
- Each pixel value is a coordinate of \( x \).

More features

- Filtered image
- Filter with multiple filters (bank of filters)
- Histogram of colors
- Histogram of Gradients (HOG)
- Haar wavelets
- Scale Invariant Feature Transform (SIFT)
- Speeded Up Robust Feature (SURF)

Pattern Classification Summary

- Supervised vs. Unsupervised: Do we have labels?
- Supervised
  - Nearest Neighbor
  - Bayesian
    - Plug in classifier
    - Distribution-based
    - Projection Methods (Fisher’s, LDA)
  - Neural Network
  - Support Vector Machine
  - Kernel methods
- Unsupervised
  - Clustering
  - Reinforcement learning

Nearest Neighbor Classifier

\[ \{ R_i \} \text{ are set of training images.} \]

\[ ID = \arg \min \text{dist}(R_i, I) \]

Comments

- Sometimes called “Template Matching”
- Variations on distance function (e.g. \( L_1 \), robust distances)
- Multiple templates per class—perhaps many training images per class.
- Expensive to compute \( k \) distances, especially when each image is big (\( N \) dimensional).
- May not generalize well to unseen examples of class.
- Some solutions:
  - Bayesian classification
  - Dimensionality reduction

An Example

“Sorting incoming Fish on a conveyor according to species using optical sensing”
• Adopt the lightness and add the width of the fish

\[
\begin{align*}
\text{Fish} & \quad x^T = [x_1, x_2] \\
\text{Lightness} & \\
\text{Width} & 
\end{align*}
\]

Bayesian Decision Theory
Continuous Features
Introduction

- The sea bass/salmon example
  - State of nature is a random variable, $\omega_I$
    - $\omega_1$: the fish is a salmon
    - $\omega_2$: the fish is a sea bass
  - Prior Probabilities
    - $P(\omega_1), P(\omega_2)$
    - $P(\omega_1) > 0$
    - $P(\omega_1) + P(\omega_2) = 1$ (exclusivity and exhaustivity)
  - Example prior: bass & salmon are equally likely
    - $P(\omega_1) = P(\omega_2) = \frac{1}{2}$ (uniform priors)

- Decision rule with only the prior information
  - Decide $\omega_1$ if $P(\omega_1) > P(\omega_2)$ otherwise decide $\omega_2$

- Use of the class-conditional information
  - $P(x | \omega_i)$ and $P(x | \omega_j)$ describe the difference in lightness between populations of sea-bass and salmon

- Posterior, likelihood, evidence
  - (BAYES RULE)
  - In words, this can be said as:
    - Posterior = (Likelihood * Prior) / Evidence
  - Where in case of two categories
    - $P(\omega_j | x) = \frac{P(x | \omega_j)P(\omega_j)}{P(x)}$

- Intuitive decision rule given the posterior probabilities:
  - Given $x$:
    - if $P(\omega_1 | x) > P(\omega_2 | x)$ \(\Rightarrow\) True state of nature = $\omega_1$
    - if $P(\omega_1 | x) < P(\omega_2 | x)$ \(\Rightarrow\) True state of nature = $\omega_2$

Why do this?: Whenever we observe a particular $x$, the probability of error is:
- $P(error | x) = P(\omega_1 | x)$ if we decide $\omega_2$
- $P(error | x) = P(\omega_2 | x)$ if we decide $\omega_1$
Plug-in classifiers

- Assume that class conditional distributions $P(x|\omega_i)$ have some parametric form - now estimate the parameters from the data.
- Common:
  - assume a normal distribution with shared covariance, different means; use usual estimates
  - Normal distribution but with different covariances;

Support Vector Machines

- Bayes classifiers and generative approaches in general try to model of the posterior, $p(\omega|x)$
- Instead, try to obtain the decision boundary directly
  - potentially easier, because we need to encode only the geometry of the boundary, not any irrelevant wiggles in the posterior.
  - Not all points affect the decision boundary

Support Vector Machines

- Set $S$ of points $x_i \in \mathbb{R}^n$, each $x_i$ belongs to one of two classes $y_i \in \{-1,1\}$
- The goal is to find a hyperplane that divides $S$ in these two classes

$$w \cdot x + b = 0$$

Support Vector Machines

- Optimal separating hyperplane maximizes $\frac{1}{w}$

$$\text{Maximize} \quad \frac{1}{w}$$

$$\text{Subject to} \quad y_i (w \cdot x_i + b) \geq 1, \quad i = 1,2,\ldots, n$$

Final Exam

- Closed book
- One cheat sheet
  - Single piece of paper any size, handwritten, no photocopying, no physical cut & paste.
- What to study
  - Basically material presented in class, and supporting material from text
  - If it was in text, but NEVER mentioned in class, it is very unlikely to be on the exam
- Question style:
  - Short answer
  - Some longer problems to be worked out.