Statistical Paradigm

- Many problems can be posed as pattern recognition
  - Image classification: indoor vs. outdoor? Face?
  - Shot boundary detection, story segmentation
    - Is the current point a boundary?
- Statistical models to handle uncertainty and provide flexibility
- Image processing tools available
  - E.g., homework #1
- Rich tools for learning and prediction
  - See course web site
- Increasing data available
  - NIST TREC Video: 300+ hours
  - Consumer and youtube videos

Important issues (1)

- Image/video processing
  - What’s the adequate quality, resolution, etc?
- Feature extraction
  - Color, texture, motion, region, shape, interest points, audio, speech, text, etc
- Feature representation
  - Histogram, bag, graph etc
  - Invariance to scale, rotation, translation, view, illumination, ...
- How to reduce dimensions?
Important issues (2)

- Distance measurement
  - How to measure similarity between images/videos?
  - L1, L2, Mahalanobis, Earth Mover's distance, vector/graph matching
- Classification models
  - Generative vs. discriminative
  - Multi-modal fusion, early fusion vs. late fusion
    - E.g., how to use joint audio-visual features to detect events (dancing, wedding...)
- Efficiency issues
  - How to speed up training and testing processes?
  - How to rapidly build a model for new domains
- Validation and evaluation
  - How to measure performance?
  - Are models generalizable to new domains?

Three related problems

- Retrieval, Ranking
  - Given a query image, find relevant ones
  - May apply rank threshold to decide relevance
- Classification, categorization, detection
  - Given an image x, predict class label y
- Clustering, grouping
  - Group images/videos into clusters of distinct attributes
An example

- News story segmentation using multi-modal, multi-scale features

First Understand Data Types and Explore Unique Characteristics

- Percentage of content:
  - (a): regular anchor segment (32.0%)
  - (b): different anchor (15.0%)
  - (c): multi-story in an anchor seg. (21.5%)
  - (d): conti. sports briefings (8.8%)
  - (e): cont. short briefings
  - (f): sep. by music or anim.
  - (g): weather report
  - (h): anchor lead-in before comm.
  - (i): comm. after sports

- Types of segments:
  - : story
  - : commercial
  - : weather
  - : misc./animation
  - : visual anchors
News Story Segmentation

- Objective: a story boundary at time $\tau_k$?
  - $\tau_k = \{\text{shot boundaries or significant pauses}\}$

Need to decide how to formulate features

<table>
<thead>
<tr>
<th>Modality</th>
<th>Raw Features</th>
<th>Data Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>motion</td>
<td>segment</td>
<td>continuous</td>
</tr>
<tr>
<td></td>
<td>shot boundary</td>
<td>point</td>
<td>binary</td>
</tr>
<tr>
<td></td>
<td>face</td>
<td>segment</td>
<td>continuous</td>
</tr>
<tr>
<td></td>
<td>commercial</td>
<td>segment</td>
<td>continuous</td>
</tr>
<tr>
<td>Audio</td>
<td>pause</td>
<td>point</td>
<td>continuous</td>
</tr>
<tr>
<td></td>
<td>pitch jump</td>
<td>point</td>
<td>continuous</td>
</tr>
<tr>
<td></td>
<td>significant pause</td>
<td>point</td>
<td>continuous</td>
</tr>
<tr>
<td></td>
<td>musc./spch. disc.</td>
<td>segment</td>
<td>binary</td>
</tr>
<tr>
<td></td>
<td>spch seg./rapidity</td>
<td>segment</td>
<td>continuous</td>
</tr>
<tr>
<td>Text</td>
<td>ASR cue terms</td>
<td>point</td>
<td>binary</td>
</tr>
<tr>
<td></td>
<td>V-OCR cue terms</td>
<td>point</td>
<td>binary</td>
</tr>
<tr>
<td></td>
<td>text seg. score</td>
<td>point</td>
<td>continuous</td>
</tr>
<tr>
<td>Misc.</td>
<td>combinatorial</td>
<td>point</td>
<td>binary</td>
</tr>
<tr>
<td></td>
<td>sports</td>
<td>segment</td>
<td>binary</td>
</tr>
</tbody>
</table>

Challenge: diverse features

One way is to use binary predicate:

If $x > \text{threshold}$, then predict segment boundary ($b=1$)
Example Predicates

<table>
<thead>
<tr>
<th>no</th>
<th>raw feature set</th>
<th>Predicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Anchor Face</td>
<td>An anchor face segment just starts after the candidate point.</td>
</tr>
<tr>
<td>2</td>
<td>Significant pause &amp; non-commercial</td>
<td>A significant pause within the non-commercial section appears in the surrounding observation window.</td>
</tr>
<tr>
<td>3</td>
<td>Pause</td>
<td>An audio pause with the duration larger than 2.0 second appears after the boundary point.</td>
</tr>
<tr>
<td>4</td>
<td>Significant pause</td>
<td>The surrounding observation window has a significant pause with the pitch jump intensity larger than the normalized pitch threshold 1.0 and the pause duration larger than 0.5 second.</td>
</tr>
<tr>
<td>5</td>
<td>Speech segment</td>
<td>A speech segment before the candidate point</td>
</tr>
<tr>
<td>6</td>
<td>Speech segment</td>
<td>A speech segment starts in the surrounding observation window</td>
</tr>
<tr>
<td>7</td>
<td>Commercial</td>
<td>A commercial starts in 15 to 20 seconds after the candidate point.</td>
</tr>
<tr>
<td>8</td>
<td>Speech segment</td>
<td>A speech segment ends after the candidate point</td>
</tr>
<tr>
<td>9</td>
<td>Anchor face</td>
<td>An anchor face segment occupies at least 10% of next window</td>
</tr>
<tr>
<td>10</td>
<td>Pause</td>
<td>The surrounding observation window has a pause with the duration larger than 0.25 second.</td>
</tr>
</tbody>
</table>

Collect Features from Training Samples

Each row represents one predicate $f_i$
- Face
- Motion
- Significant Pause
- Speech segment
- Commercial
- Text segmentation score
- ASR cue terms

One training sample
Choose Model

**Maximum entropy model**

\[ q(x | b) = \frac{1}{Z(x)} e^{\sum_i \lambda_i f_i(x, b)} \]

where \( f_i(x, b), b \in \{0, 1\} \)

For example

- predicate \( f_1 = \text{"anchor face"} \), \( f_2 = \text{"significant pause"} \)
- if current observation: face = YES pause = NO
- \( q(b = YES | x) = e^{\lambda_{1}} / (e^{\lambda_{1}} + e^{\lambda_{2}}) \)
- \( q(b = NO | x) = e^{\lambda_{2}} / (e^{\lambda_{1}} + e^{\lambda_{2}}) \)

Classification: if \( q(b = YES | x) > 0.5 \), then predict YES.

---

Background: Entropy

- **Entropy (bits)**

\[ H = -\sum_{i=1}^{m} p_i \log_2 p_i \]

- **Kullback-Leibler (K-L) Distance**
  - A measure of ‘distance’ between 2 distributions
  
  \[ D_{KL}(p(x), q(x)) = \sum_{x} q(x)\log \frac{q(x)}{p(x)} \]
  
  or \[ \int_{-\infty}^{\infty} q(x)\log \frac{q(x)}{p(x)}dx \]

  \( D_{KL} \geq 0 \), and \( = 0 \iff p(\cdot) = q(\cdot) \)

- Not necessarily symmetric, may not satisfy triangular inequality
How to Determine the Weights in the Model?

- Estimate $q_x(b \mid x)$ from training data $T = \{(x_i, b_i)\}$ by minimizing Kullback-Leibler divergence, defined as

$$D(\tilde{p} \parallel q_x) = \sum_b \tilde{p}(b, x) \log \frac{\tilde{p}(b \mid x)}{q_x(b \mid x)} = -\sum_b \tilde{p}(x, b) \log q_x(b \mid x)$$

- Find $\lambda_i$ to maximize the ‘entropy’

$$L_p(q_x) = \sum_b \tilde{p}(x, b) \log q_x(b \mid x)$$

- Iteratively find $\lambda_i$

$$\lambda_i' = \lambda_i + \Delta \lambda_i, \quad \Delta \lambda_i = \frac{1}{M} \log \left( \frac{\sum_{x, b} \tilde{p}(x, b) f_i(x, b)}{\sum_{x, b} \tilde{p}(x) q_x(b \mid x) f_i(x, b)} \right)$$

* The objective function is convex. So the iterative process can reach the optimum.

The same model used to select features

- **Input:** collection of candidate features, training samples, and the desired model size
- **Output:** optimal subset of features and their corresponding exponential weights
- **Current model** $q$ augmented with feature $h$ with weight $\beta$, $q_{\alpha, h}(b \mid x) = \frac{e^{\alpha h(x, b)} q(b \mid x)}{Z_\alpha(x)}$

Select the candidate which improves current model $q$ the most, in each iteration;

$$h' = \arg \max_{h \in C} \left\{ \sup_a \left\{ D(\tilde{p} \parallel q) - D(\tilde{p} || q_{\alpha, h}) \right\} \right\}$$

$$= \arg \max_{h \in C} \left\{ \sup_a \left\{ L_p(q_{\alpha, h}) - L_p(q) \right\} \right\}$$

Reduction of divergence

Increase of log-likelihood
Optimal Features (from CNN news video)

* The first 10 “A+V” features automatically discovered for the CNN channel

<table>
<thead>
<tr>
<th>no</th>
<th>raw feature set</th>
<th>gain</th>
<th>$\lambda$</th>
<th>interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Anchor Face</td>
<td>0.3879</td>
<td>0.4771</td>
<td>An anchor face segment just starts after the candidate point</td>
</tr>
<tr>
<td>2</td>
<td>Significant pause &amp; non-commercial</td>
<td>0.0160</td>
<td>0.7471</td>
<td>A significant pause within the non-commercial section appears in the surrounding observation window</td>
</tr>
<tr>
<td>3</td>
<td>Pause</td>
<td>0.0058</td>
<td>0.2434</td>
<td>An audio pause with the duration larger than 0.2 second appears after the boundary point</td>
</tr>
<tr>
<td>4</td>
<td>Significant pause</td>
<td>0.0024</td>
<td>0.7947</td>
<td>The surrounding observation window has a significant pause with the pitch jump intensity larger than the normalized pitch threshold 1.0 and the pause duration larger than 0.5 second</td>
</tr>
<tr>
<td>5</td>
<td>Speech segment</td>
<td>0.0019</td>
<td>-0.3566</td>
<td>A speech segment before the candidate point</td>
</tr>
<tr>
<td>6</td>
<td>Speech segment</td>
<td>0.0015</td>
<td>0.3734</td>
<td>A speech segment starts in the surrounding observation window</td>
</tr>
<tr>
<td>7</td>
<td>Commercial</td>
<td>0.0015</td>
<td>1.0782</td>
<td>A commercial starts in 15 to 20 seconds after the candidate point</td>
</tr>
<tr>
<td>8</td>
<td>Speech segment</td>
<td>0.0022</td>
<td>-0.4127</td>
<td>A speech segment ends after the candidate point</td>
</tr>
<tr>
<td>9</td>
<td>Anchor face</td>
<td>0.0016</td>
<td>0.7251</td>
<td>An anchor face segment occupies at least 10% of next window</td>
</tr>
<tr>
<td>10</td>
<td>Pause</td>
<td>0.0008</td>
<td>0.0939</td>
<td>The surrounding observation window has a pause with the duration larger than 0.25 second</td>
</tr>
</tbody>
</table>

Every modality helps: anchor face, prosody, and speech segment

Issues of this model (Discussion)

- **Features**
  - Binary predicates reasonable?
  - Does it capture the unique characteristics?
    - Binary predicate: 
      \[
      q_b(x) = \frac{1}{Z_b(x)} \sum_{s} e^{\lambda_{s,b}}
      \]
  - Models
    - Exponential models with linear weights adequate?
    - How about the learning algorithm?
      - Enough data to learn the probability models?
      - Speed and complexity
A Broader Perspective: Classification Paradigms

Which one does the previous model fall into?

- Generative Models
  - Gaussian Mixture Model
One common issue is to learn probability models

- Gaussian distribution
  \[ p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \]

- Given the same mean and variance, Gaussian has the max entropy

- Sum of a large number of small, independent random variables approaches Gaussian

\[ \text{Pr}[|x - \mu| \leq \sigma] \approx 0.68 \]
\[ \text{Pr}[|x - \mu| \leq 2\sigma] \approx 0.95 \]
\[ \text{Pr}[|x - \mu| \leq 3\sigma] \approx 0.997 \]

Mahalanobis distance from \( x \) to \( \mu \)
\[ r = \frac{|x - \mu|}{\sigma} \]

**Entropy of Gaussian:**
\[ H_{\text{Gaussian}} = 0.5 + \log_2(\sqrt{2\pi\sigma}) \]

Comparison w. uniform

---

Multi-variate Gaussian

- Multivariate Gaussian, \( N(\mu, \Sigma) \)
\[
p(x) = \frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu)}
\]

where \( x, \mu \) are \( D \)-dimensional vectors

\( \Sigma \): \( D \times D \) matrix

\(|\Sigma|\) is the determinant of \( \Sigma \)

\[ (\sigma_{ij})^2 = \Sigma(i, j) = \text{cov}(x(i), x(j)) \]
\[ = E[(x(i) - \mu(i))(x(j) - \mu(j))] \]
Effect of Linear Transformation

- Linear transformation of Gaussian
  \[ y = A'x \quad y : k \times 1, \quad A : d \times k, \quad x : d \times 1 \]
  \[ y \sim N(A'\mu, A'\Sigma A) \]

- Whitening transform
  \[ \Sigma = \Phi A \Phi' \text{ (SVD, Eigenvectors)} \]
  \[ \Phi : [\phi_1 | \phi_2 | \cdots | \phi_k] \text{ columns are orthogonal ev} \]
  \[ A : \text{diag. matrix of eigenvalues} \]
  \[ A'\Sigma A = A'\Phi A \Phi' A = I \]

  Whitening Trans. \[ A_w = \Phi \Lambda^{-1/2} \]
  also PCA Transform

  \[ y = A_w'x \quad \sim \quad N(A_w'\mu, I) \]

Mahalanobis Distance

- Mahalanobis dist in 1-D
  \[ p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x - \mu)^2} \]

- Multi-Dimensional case
  \[ p(x) = \frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma|}} \exp(-\frac{1}{2} (x - \mu)' \Sigma^{-1} (x - \mu)) \]
  \[ = \frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma|}} \exp(-\frac{1}{2} (x - \mu)' \Phi \Lambda^{-1} \Phi' (x - \mu)) \]
  \[ = \frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma|}} \exp(-\frac{1}{2} (A_w' (x - \mu))' (A_w' (x - \mu))) \]

  \[ r \text{ is the Mahalanobis distance} \]
  \[ \text{is also the Euclidean dist in the PCA space} \]

\[ r^2 \]
- Malalanobis distance from point x to the mean of a Gaussian

\[ x \rightarrow C_j, \text{ if } p(C_j | x) \geq p(C_i | x), \text{ when } i \neq j \]

MAP classifier

\[
p(C_j | x) = \frac{\text{likelihood}}{p(x | C_j)} \frac{\text{prior}}{p(C_j)} \frac{\text{evidence}}{p(x)}
\]

ML classification:

- if uniform \( p(C_j) \Rightarrow C_j = \arg \max_C p(x | C) \)

\( p(x | C) \) can be modeled by Gaussians
How to Estimate Gaussian Model Parameters?

- \( \theta = (\mu, \sigma^2) \)
- \( l = \ln P(x_k | \theta) = -\frac{1}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(x_k - \mu)^2 \)
- \( \nabla_{\theta} l = \begin{bmatrix} \frac{\partial}{\partial \mu} (\ln P(x_k | \theta)) \\ \frac{\partial}{\partial \sigma^2} (\ln P(x_k | \theta)) \end{bmatrix} = \begin{bmatrix} \frac{1}{\theta^2} (x_k - \mu) \\ -\frac{1}{2\sigma^2} \frac{(x_k - \mu)^2}{2(\sigma^2)^2} \end{bmatrix} \)
- \( \sum_k \nabla_{\theta} \ln P(x_k | \theta) = 0 \quad \Rightarrow \quad \mu = \frac{1}{n} \sum_k x_k \quad \sigma^2 = \frac{1}{n} \sum_k (x_k - \mu)^2 \)

- Multi-Dimensional \( \theta = (\mu, \Sigma) \)
- \( \Rightarrow \mu = (1/n) \sum_k \tilde{x}_k \quad \Sigma = (1/n) \sum_k (\tilde{x}_k - \mu)(\tilde{x}_k - \mu)' \)

ML estimator: mean -> sample mean, variance -> biased sample variance

Mixture Of Gaussians

- Real distributions seldom follow a single Gaussian \( \rightarrow \) mixture of Gaussians
- \( p(x) = \sum_z p(x, z) = \sum_z p(z)p(x | z) = \sum_z \pi_z N(x | \mu_z, \Sigma_z) = \sum_z \pi_z \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x - \mu_z)^2} \)
- Given data \( x_1, ..., x_N \), define log-likelihood:
- \( l = \log(\prod_n p(x_n)) = \sum_{n=1}^{N} \log(\pi_0 N(x_n | \mu_0, \Sigma_0) + \pi_1 N(x_n | \mu_1, \Sigma_1)) \)
- Posterior probability of \( x \) being generated by a specific component
- \( \text{Posteriors} = \tau^i = p(z = i | x, \theta), \quad \theta = \{\mu_0, \Sigma_0, \mu_1, \Sigma_1\} \)
E-M Optimization Method

**log likelihood**

\[ l(\theta) = \sum_{n=1}^{N} \log \sum_{z} p(x_n, z | \theta) \]

- Maximization of \( l(\theta) \) directly is hard due to log_of_sum
- Instead, look at
  \[ \Delta l(\theta) = l(\theta) - l(\theta'), \theta' : current \ estimation \ of \ \theta \]

- **Jensen’s Inequality**

  If \( f \) is concave, \( f(E[x_i]) \geq E(f[x_i]) \)
  
  \[ f(E[g(x)]) \geq E(f[g(x)]) \]

  e.g., \( f(x) = \log(x) \)

  \[ \log \left( \sum_{i} p_i x_i \right) \geq \sum_{i} p_i \log(x_i), \text{where} \ \sum_{i} p_i = 1 \]

  If \( f \) is convex, \( f(E[x_i]) \leq E(f[x_i]) \)

---

**Auxiliary Function in E-M**

\[ \Delta l(\theta) = l(\theta) - l(\theta') = \sum_{n=1}^{N} \log p(x_n | \theta) - \sum_{n=1}^{N} \log p(x_n | \theta') \]

\[ = \sum_{n=1}^{N} \log \frac{p(x_n | \theta)}{p(x_n | \theta')} = \sum_{n=1}^{N} \log \sum_{z} \frac{p(x_n, z | \theta)}{p(x_n, z | \theta')} \]

- Marginalization

\[ = \sum_{n=1}^{N} \log \sum_{z} \frac{p(x_n, z | \theta) p(x_n, z | \theta')}{p(x_n, z | \theta')} = \sum_{n=1}^{N} \log \sum_{z} p(z | x_n, \theta') \frac{p(x_n, z | \theta)}{p(x_n, z | \theta')} \]

\[ \geq \sum_{n=1}^{N} \sum_{z} p(z | x_n, \theta') \log \frac{p(x_n, z | \theta)}{p(x_n, z | \theta')} \]

\[ = Q(\theta | \theta') \]

- Note there is no log_of_sum.
  So taking derivative is easier
E-M improves likelihood

- Auxiliary function derived based on Jensen’s Inequality, 
  \[ Q(\theta | \theta_t) = \sum_{n=1}^{N} \sum_{z} p(z | x_n, \theta_t) \log p(x_n, z | \theta) + \text{const} \]
  - expectation over \( z \) with current \( \theta_t \)
  - joint likelihood of observed & hidden

- Now estimate \( \theta_{t+1} \) by maximizing \( Q \)
  \[ \theta_{t+1} = \arg \max_{\theta} Q(\theta | \theta_t) \]

- So in the expectation step, compute \( \tau_n^z \), the ‘responsibility’ of component \( z \) for sample \( x_n \)

- In the maximization step, take derivative of \( Q \) over \( \theta \), and find the new estimate for \( \theta \) (Note only sum_of_log is involved)

EM Always Improves Likelihood

- Why does EM always improve \( l(\theta) \)?
  \[ \Delta l(\theta_{t+1}) = l(\theta_{t+1}) - l(\theta_t) \geq Q(\theta_{t+1} | \theta_t) \]
  \[ Q(\theta_{t+1} | \theta_t) = \max_{\theta} Q(\theta | \theta_t) \geq Q(\theta_t | \theta_t) = 0 \quad \therefore \quad \Delta l(\theta_{t+1}) \geq 0 \]

- General steps of EM:
  - Define likelihood model with parameters \( \theta \)
  - Identify hidden variables \( z \)
  - Derive the auxiliary function and the E and M equations
  - In each iteration, estimate the posteriors of hidden variables
  - Re-estimate the model parameters. Repeat until stop
Expectation-Maximization (E-M) Solution of GMM

\[
Q(\theta | \hat{\theta}) = \sum_{n=1}^{N} \sum_{z_n} p(z_n | x_n, \hat{\theta}) \log p(x_n, z_n | \theta) + \text{const}
\]

- EM for estimating \( \theta \) and \( \tau_i \).
  - Follow ‘divide and conquer’ principle. In iteration step t:
    
    **Expectation:** \( \tau_n^{(t)} = \frac{\pi_i^{(t)} N(x_n | \mu_i^{(t)}, \Sigma_i^{(t)})}{\sum_j \pi_j^{(t)} N(x_n | \mu_j^{(t)}, \Sigma_j^{(t)})} \)  
      Weight from component \( i \)
    
    **Maximation:** \( \mu_i^{(t+1)} = \frac{\sum_n \tau_n^{(t)} x_n}{\sum_n \tau_n^{(t)}} \)  
      \( \Sigma_i^{(t+1)} = \frac{\sum_n \tau_n^{(t)} (x_n - \mu_i^{(t)}) (x_n - \mu_i^{(t)})^T}{\sum_n \tau_n^{(t)}} \)  
      \( \pi_i^{(t+1)} = \frac{\sum_n \tau_n^{(t)}}{N} \)

Divide data to each group, Compute mean and variance from each group

- Discriminative Models
### Simple Discriminative Classifier

- Find most opportunistic dimension in each step
- Selection criterion
  - Entropy
  - Variance before / after
- Stop criterion
  - Avoid overfitting

### Parametric Discriminant Analysis

- Example of discriminant function
  - Linear and quadratic discriminant

\[
g(\bar{x}) = ax_1 + bx_2
\]

\[
g(\bar{x}) = ax_1^2 + bx_2^2 + cx_1x_2
\]
Linear Discriminant Classifiers

\[ g(x) = w^T x + w_0 \Rightarrow \text{find weight } w \text{ and bias } w_0 \]

- Augmented Vector
  \[ y = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \quad a = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \Rightarrow g(x) = g(y) = a^T y \]

map \( y \) to class \( \omega_1 \) if \( g(y) > 0 \), otherwise class \( \omega_2 \)

- Design Objective
  \[ a^T y > b, \text{ if class } \omega_1, \]
  \[ a^T y < -b, \text{ if class } \omega_2, \forall y_i, b > 0 \]

- Each \( y \) defines a half plane in the weight space (a).
- Note we search weight solutions in the a-space.

Support Vector Machine (tutorial by Burges '98)

- Look for separation plane with the highest margin
  Decision boundary
  \[ H_0: \quad w^T x + b = 0 \]

- Linearly separable
  \[ w^T x_i + b \geq +1, \forall x_i \text{ in class } \omega_1 \text{ i.e. } y_i = +1 \]
  \[ w^T x_i + b \leq -1, \forall x_i \text{ in class } \omega_2 \text{ i.e. } y_i = -1 \]

Inequality constraints: \( (\text{label}) \cdot (w^T x_i + b) - 1 \geq 0, \forall i \)

- Two parallel hyperplanes defining the margin
  hyperplane \( H_1(H_+): w^T x_i + b = +1 \)
  hyperplane \( H_1(H_-): w^T x_i + b = -1 \)

- Margin: sum of distances of the closest points to the separation plane
  \[ \text{margin} = \frac{2}{||w||} \]
  - Best plane defined by \( w \) and \( b \)
Finding the maximal margin

minimize \( \frac{1}{2} ||\mathbf{w}||^2 \) subject to inequality constraints
\[ y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1, \ldots, l \]
\( y_i \) is label

- Use the Lagrange multiplier technique for the constrained opt. problem

Primal Problem

\[ L_p = \frac{1}{2} ||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1) \]
\( \alpha_i \geq 0 \)
\[ \frac{dL_p}{d\mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \]
\[ \frac{dL_p}{db} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \]

Dual Problem

\[ L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \]
\( \alpha_i \geq 0 \)
\( \sum_{i=1}^{l} \alpha_i y_i = 0 \)

Prime and Dual have the same solutions of \( \mathbf{w} \) and \( b \)

KKT conditions

\[ \frac{\partial}{\partial \nu} L_P = \nu \sum_{i=1}^{l} \alpha_i y_i x_i = 0 \quad \nu = 1, \ldots, d \]
\[ \mathbf{w}^* = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \]

- Weight sum from positive class = Weight sum from negative class
- Direction of \( \mathbf{w} \):
  roughly from negative support vectors to positive ones

if \( \alpha_i > 0 \), \( \mathbf{x}_i \) is on \( H_+ \) or \( H_- \) and is a support vector
Non-separable: not every sample can be correctly classified

- Add slack variables $\xi_i$
  
  \[ x_i \cdot w + b \geq +1 - \xi_i \quad \text{for } y_i = +1 \]
  
  \[ x_i \cdot w + b \leq -1 + \xi_i \quad \text{for } y_i = -1 \]

  \[ \xi_i \geq 0 \quad \forall i \]

  If $\xi_i > 1$, then $x_i$ is misclassified (i.e. training error)

Lagrange multiplier: minimize

\[
L_P = \frac{1}{2}||w||^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum \mu_i \xi_i
\]

New objective function

Ensure positivity

KKT Conditions for non-separable Solutions

\[
\frac{\partial L_P}{\partial w} = w - \sum \alpha_i y_i x_i = 0
\]

\[
\frac{\partial L_P}{\partial \alpha_i} = C - \alpha_i - \mu_i = 0
\]

\[
\frac{\partial L_P}{\partial \xi_i} = y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0
\]

\[
\xi_i \geq 0
\]

\[
\mu_i \geq 0
\]

If $0 < \alpha_i < C$, then $\xi_i = 0$ : $x_i$ is on $H_1$ or $H_2$

If $\alpha_i = C$,

then $\xi_i > 0$ : $x_i$ is inside the margin or on the wrong side

or $\xi_i = 0$ : $x_i$ is on $H_1$ or $H_2$
All the points located in the margin gap or the wrong side will get $\alpha_i = C$

What if $C$ increases?

$0 < \alpha_i < C$

after $C$ increases

When $C$ increases, incorrect samples get more weights
- try to minimize incorrect samples
- better training accuracy, but smaller margin
- less generalization performance

Generalized Linear Discriminant Functions

- Include more than just the linear terms
  
  $g(x) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{j=1}^{d} \sum_{i=j}^{d} w_{ij} x_i x_j = w_0 + \mathbf{w}^T \mathbf{x} + \mathbf{x}^T \mathbf{W} \mathbf{x}$

- Shape of decision boundary
  - ellipsoid, hyperhyperboloid, lines etc

- In general
  
  $g(x) = \sum_{j=1}^{d} a_j \phi_j(x) = \mathbf{a}^T \Phi$

- Example
  
  $g(x) = a_1 + a_2 x + a_3 x^2$
  $g(x) = a_1 x_1 + a_2 x_2 + a_3 x_3$

  
  $\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 1 & x & x^2 \end{bmatrix}^T = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_1 x_2 \end{bmatrix}$

- Data become separable in higher-dimensional space
  - learning parameters in high dimension is hard (curse of dimensionality)
  - instead, try to maximize margins $\rightarrow$ SVM
Non-Linear Space

\( \Phi : \mathbb{R}^d \mapsto \mathcal{H} \). Map to a high dimensional space, to make the data separable

- Find the SVM in the high-dim space
  
  \[
  g(x) = \sum_{i=1}^{N_y} \alpha_i y_i \Phi(s_i) \cdot \Phi(x) + b
  \]

- Luckily, we don’t have to find \( \Phi(s_i) \) nor \( \sum_{i=1}^{N_y} \alpha_i y_i \Phi(s_i) \)

- Instead, we define kernel
  
  \[
  K(s_i, x) = \Phi(s_i) \cdot \Phi(x)
  \]

  \[
  \Rightarrow g(x) = \sum_{i=1}^{N_y} \alpha_i y_i K(s_i, x) + b
  \]

- We can use the same method to maximize \( L_D \) to find \( \alpha_i \)
  
  \[
  L_D = \sum_{i=1}^{N_S} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_S} \sum_{j=1}^{N_S} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j)
  \]

  \[
  = \sum_{i=1}^{N_S} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_S} \sum_{j=1}^{N_S} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
  \]

Some popular kernels

\[
K(x, y) = (x \cdot y + 1)^p \quad \text{polynomial}
\]

\[
K(x, y) = e^{-||x-y||^2/2\sigma^2} \quad \text{Gaussian Radial Basis Function (RBF)}
\]

\[
K(x, y) = \tanh(\kappa x \cdot y - \delta) \quad \text{sigmoidal neural network}
\]

separable

Cubic polynomial

non-separable
See the SVM demos (Eric)

Evaluation

- N Test Image
- K detected results

Detection
- $A = \sum_{n=0}^{K-1} V_n$
- $B = \sum_{n=0}^{K-1} (1 - V_n)$
- $C = (\sum_{n=0}^{N-1} V_n) - A$
- $D = (\sum_{n=0}^{N-1} (1 - V_n)) - B$

False Alarms

Misses

Correct Dismissals

"Returned" "Relevant Ground Truth"

Recall
- $R = A/(A + C)$

Precision
- $P = A/(A + B)$

Fallout
- $F = B/(B + D)$

Combined
- $F_i = \frac{P \cdot R}{(P + R)/2}$
Evaluation Measures

1. Precision Recall Curve

2. Receiver Operating Characteristic (ROC Curve)

Evaluation Metric: Average Precision

- Ranked list of data in response to a query
- Ground truth
- Precision

**Average precision:** 
\[ AP = \frac{1}{R} \sum_{j=1}^{R} \frac{R_j}{j} \]

**AP** measures the average of precision values at R relevant data points.
Evaluation Metric: Average Precision (2)

- AP depends on the rankings of relevant data and the size of the relevant data set. E.g., R=10

Case I:

| + + + + + + + + + - - - - - | Pre: 1 1 1 1 1 1 1 1 1 1 1 | 0 0 0 0 0 0 0 0 0 | AP = 1

Case II:

| - + + + + + + + + + + + + + + + | Pre: 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 | AP = 1/2

Case III:

| + + + + + + + + + + + + + + + + | Pre: 1/11 2/12 10/20 | AP ~ 0.3

Example: SVM for News Story Segmentation

- SVM with 195 binary features performs the best
  - SVM has excellent feature fusion capability
  - Predicate binarization shields noise in the feature

Precision vs. recall on CNN news of NE/Boosting/SVM approaches

- SVM-based
- Maximum Entropy
- BST

\[ \text{Precision} = \frac{TP}{TP + FP} \]
\[ \text{Recall} = \frac{TP}{TP + FN} \]
Appropriate if the same distributions are followed over different sets

Cross validation, leave-one-out

Rotate the choice of the test set and average the performance over runs
Curse of Dimensionality and Overtraining

Very rough rule of thumb –

# of training samples per class feature dimension

> 10

A case of overfitting

Test performance