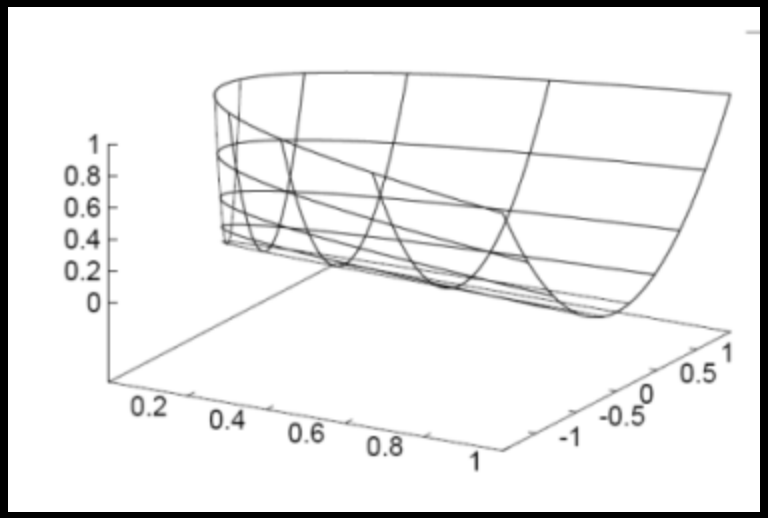
Machine Learning for Computer Vision

6 October, 2013
MVA – ENS Cachan



Lecture 3: Support Vector Machines

Iasonas Kokkinos

Iasonas.kokkinos@ecp.fr

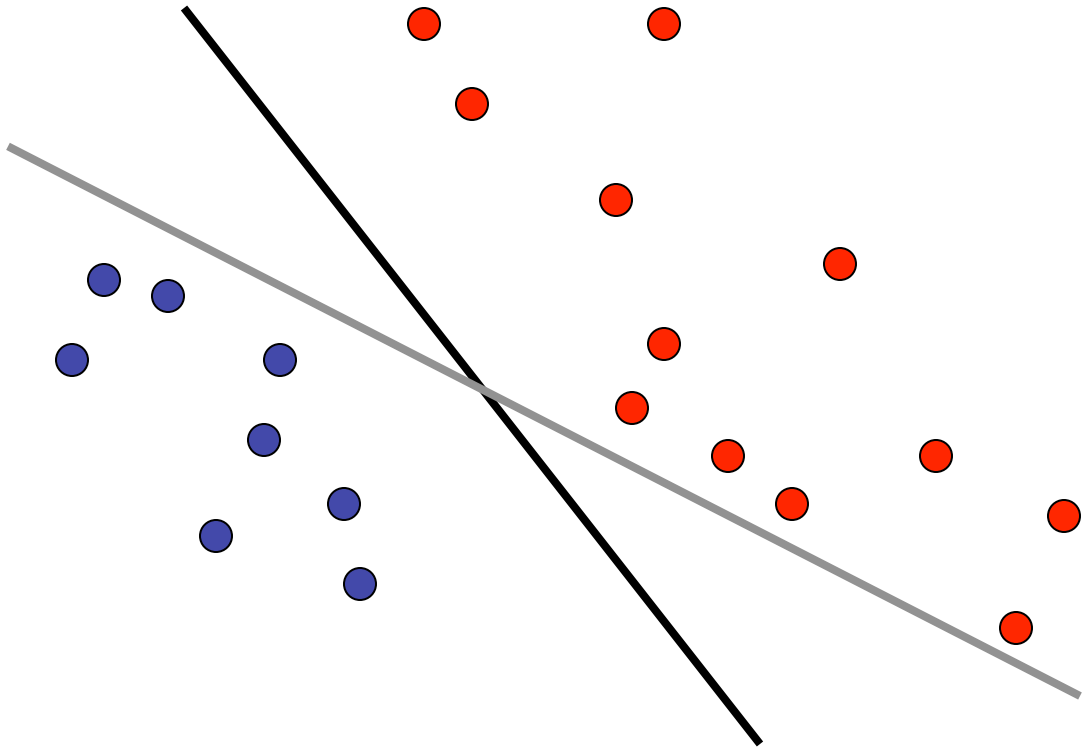
Center for Visual Computing
Ecole Centrale Paris

Galen Group
INRIA-Saclay

Lecture outline

- Introduction to Support Vector Machines
  - Geometric margins
  - Training criterion & hinge loss
- Large margins and generalization
- Optimization
- Kernels
- Applications to vision

Which classifier is best?



All points should lie clearly on the correct side of the boundary

How can we quantify this?

How can we enforce this?

Functional Margins

Consider Logistic Regression:

$$P(y=1|x;w) = g(\mathbf{w}^T x) = \frac{1}{1+\exp(-\mathbf{w}^T x)}$$

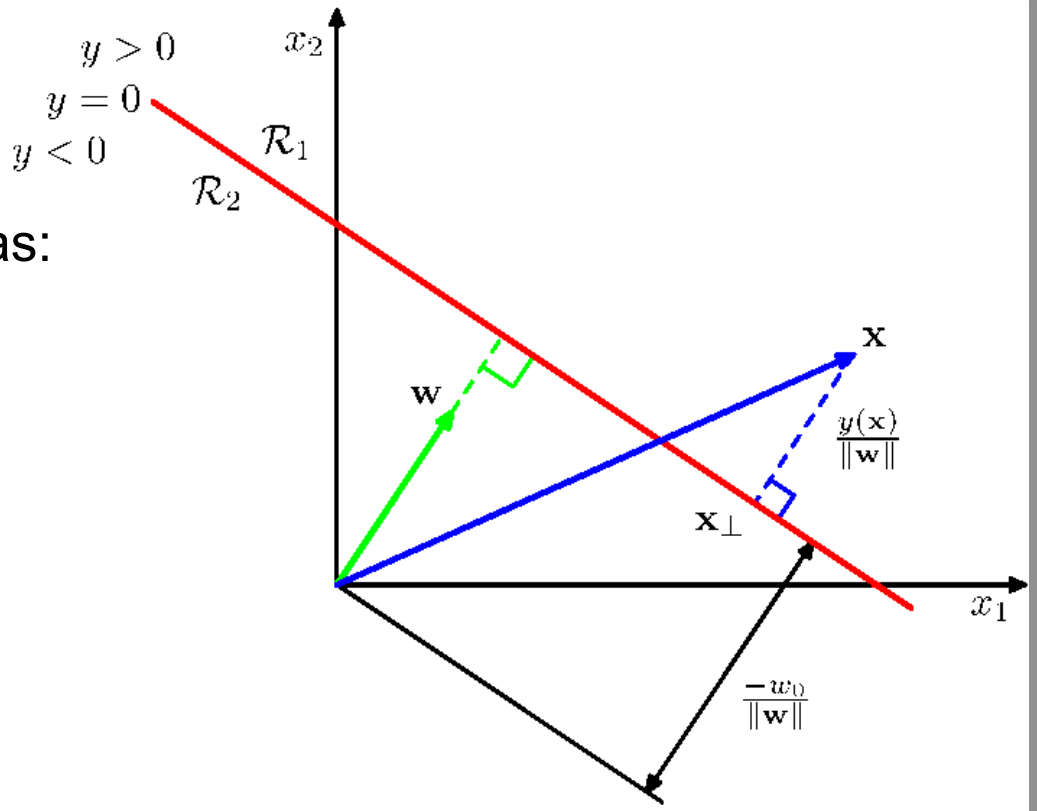
Ideally:
$$(\mathbf{w}^T x^i) \gg 0 \quad \text{if} \quad y^i = 1$$
$$(\mathbf{w}^T x^i) \ll 0 \quad \text{if} \quad y^i = -1$$

Put together: $y^i(\mathbf{w}^T x^i) \gg 0$

`functional margin'

Problem: scaling w changes functional margin, but not decision boundary

Geometric Margins



- Express point in feature space as:

$$\mathbf{x} = \mathbf{x}_\perp + \gamma \frac{\mathbf{w}}{|\mathbf{w}|}$$

- γ: distance from hyperplane

$$\mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{x}_\perp + \mathbf{w}^T\gamma\frac{\mathbf{w}}{|\mathbf{w}|}$$

$$\mathbf{w}^T\mathbf{x} = -b + \gamma|\mathbf{w}|$$

$$\gamma = \frac{\mathbf{w}^T\mathbf{x} + b}{|\mathbf{w}|} = \frac{\mathbf{w}^T}{|\mathbf{w}|}\mathbf{x} + \frac{b}{|\mathbf{w}|}$$

- Make positive also for negative class:

$$\gamma^i = y^i\left(\frac{\mathbf{w}^T}{|\mathbf{w}|}\mathbf{x}^i + \frac{b}{|\mathbf{w}|}\right)$$

Lecture outline

- Introduction to Support Vector Machines
  - Geometric margins
  - Training criterion & hinge loss
- Large margins and generalization
- Optimization
- Kernels
- Applications to vision

Margin-based Optimization

Least margin:

$$\gamma = \min_i \gamma^i$$

Optimization problem:

$$\begin{aligned} \max_{\gamma,\mathbf{w},b} \quad & \gamma \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq \gamma, \quad i = 1 \dots M \\ & |\mathbf{w}| = 1 \end{aligned}$$

Problem: non-convex constraint

Functional margin:

$$\begin{aligned} \max_{\gamma,\mathbf{w},b} \quad & \frac{\gamma'}{|\mathbf{w}|} \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq \gamma', \quad i = 1 \dots M \end{aligned}$$

Margin-Based Optimization (cont.d)

Rewrite last problem:

$$\max_{\gamma,\mathbf{w},b} \quad \frac{\gamma'}{|\mathbf{w}|}$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq \gamma', \quad i = 1 \ldots M$$

$$\max_{\mathbf{w},b} \quad \frac{1}{|\mathbf{w}|}$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1, \quad i = 1 \ldots M$$

Equivalent:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1, \quad i = 1 \ldots M$$

Quadratic programming problem

Quadratic cost

Linear constraints

Large Margin Classifiers

Goal: Maximize least margin: $\gamma = \min_i \gamma^i$

Optimization problem:

$$\begin{aligned} \max_{\gamma,\mathbf{w},b} \quad & \gamma \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq \gamma, \quad i = 1 \dots M \\ & |\mathbf{w}| = 1 \end{aligned}$$

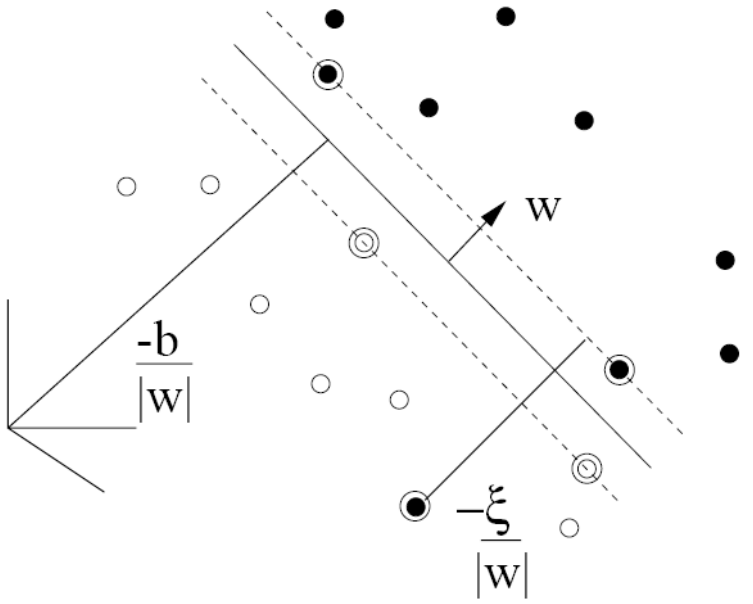
Equivalent problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}|\mathbf{w}|^2 \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq 1, \quad i = 1 \dots M \end{aligned}$$

Accounting for non-separable data

Introduce positive `slack' variables:

$$\mathbf{w}^T x^i + b \geq 1 - \xi^i, \qquad y_i = 1$$
$$\mathbf{w}^T x^i + b \leq -1 + \xi^i, \qquad y_i = -1$$



Equivalently: $y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i$

If $\xi^i > 1$ an error occurs

$\sum_i \xi^i$:upper bound on number of errors

New optimization problem

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}\xi^i$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i$$
$$\xi^i \geq 0$$

Loss function

Optimization problem:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi^i$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i$$
$$\xi^i \geq 0$$

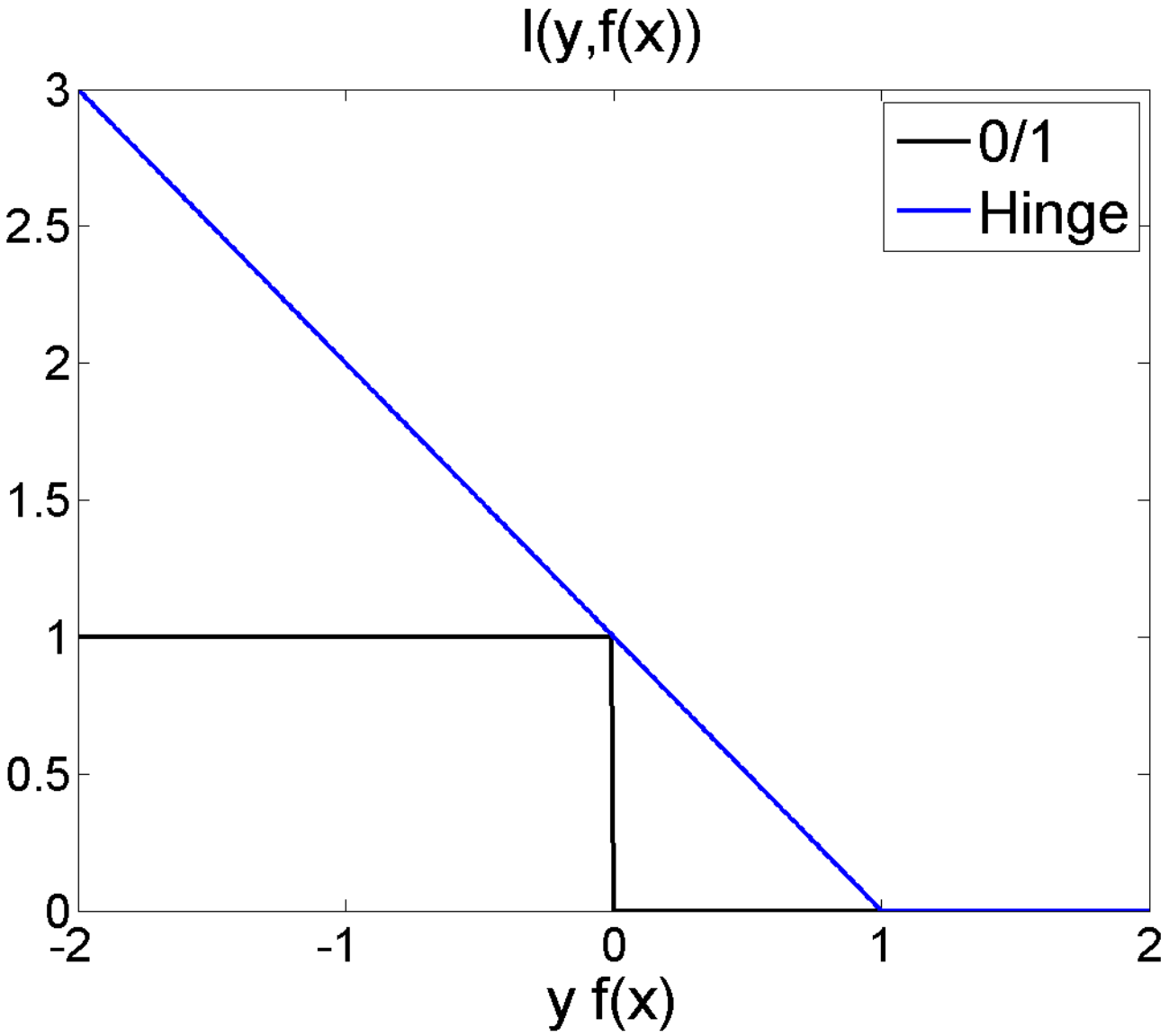
At optimum :

$$\xi^i = \max(0, 1 - y^i(\mathbf{w}^T x^i + b))$$
$$= \max(0, 1 - y^i h_{\mathbf{w},b}(x^i))$$

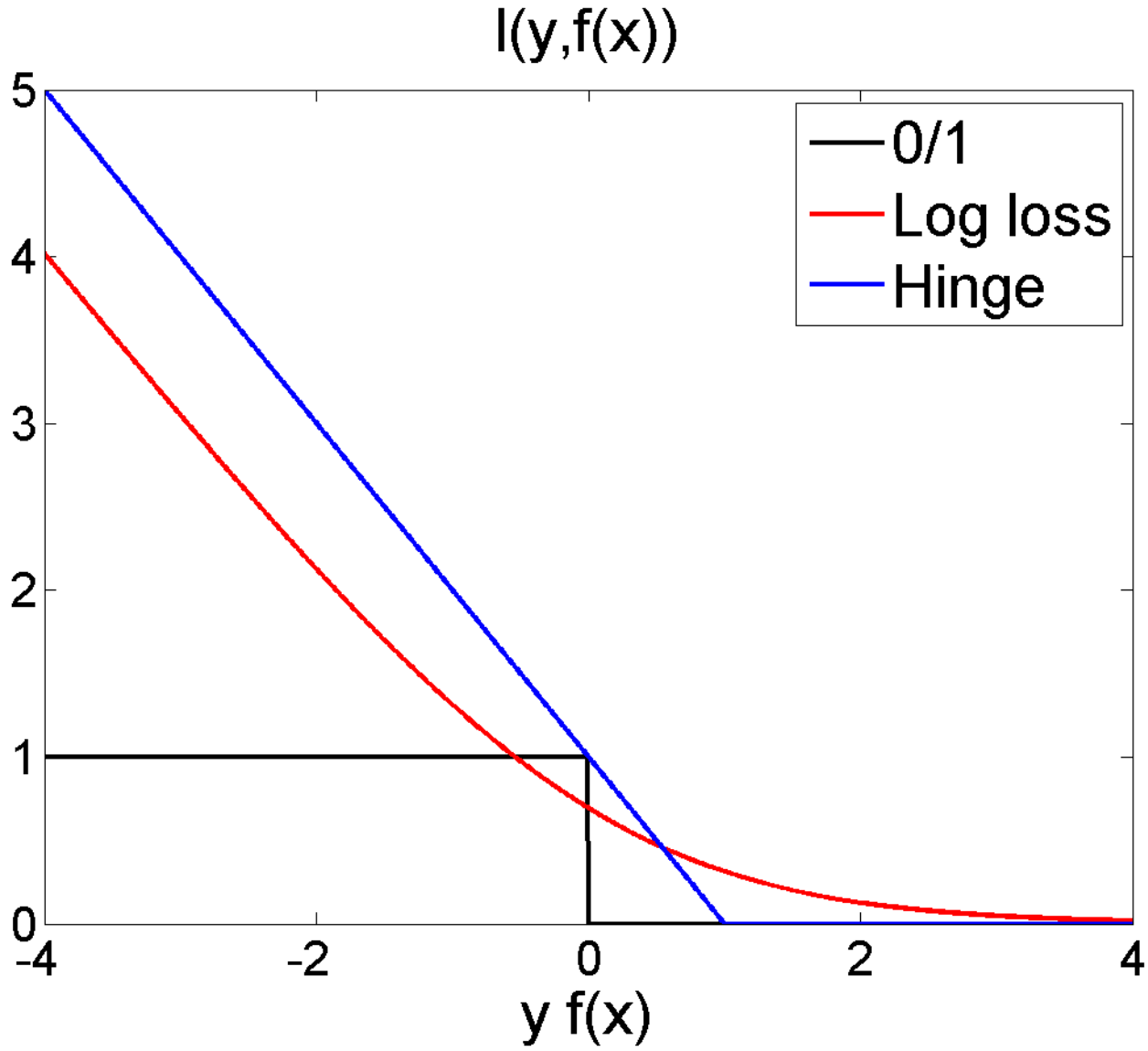
Training criterion:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\max(0, 1 - y^i h_{\mathbf{w},b}(x^i))$$
$$\propto \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\underbrace{\max(0, 1 - y^i h_{\mathbf{w},b}(x^i))}_{l(y^i, x^i)}$$

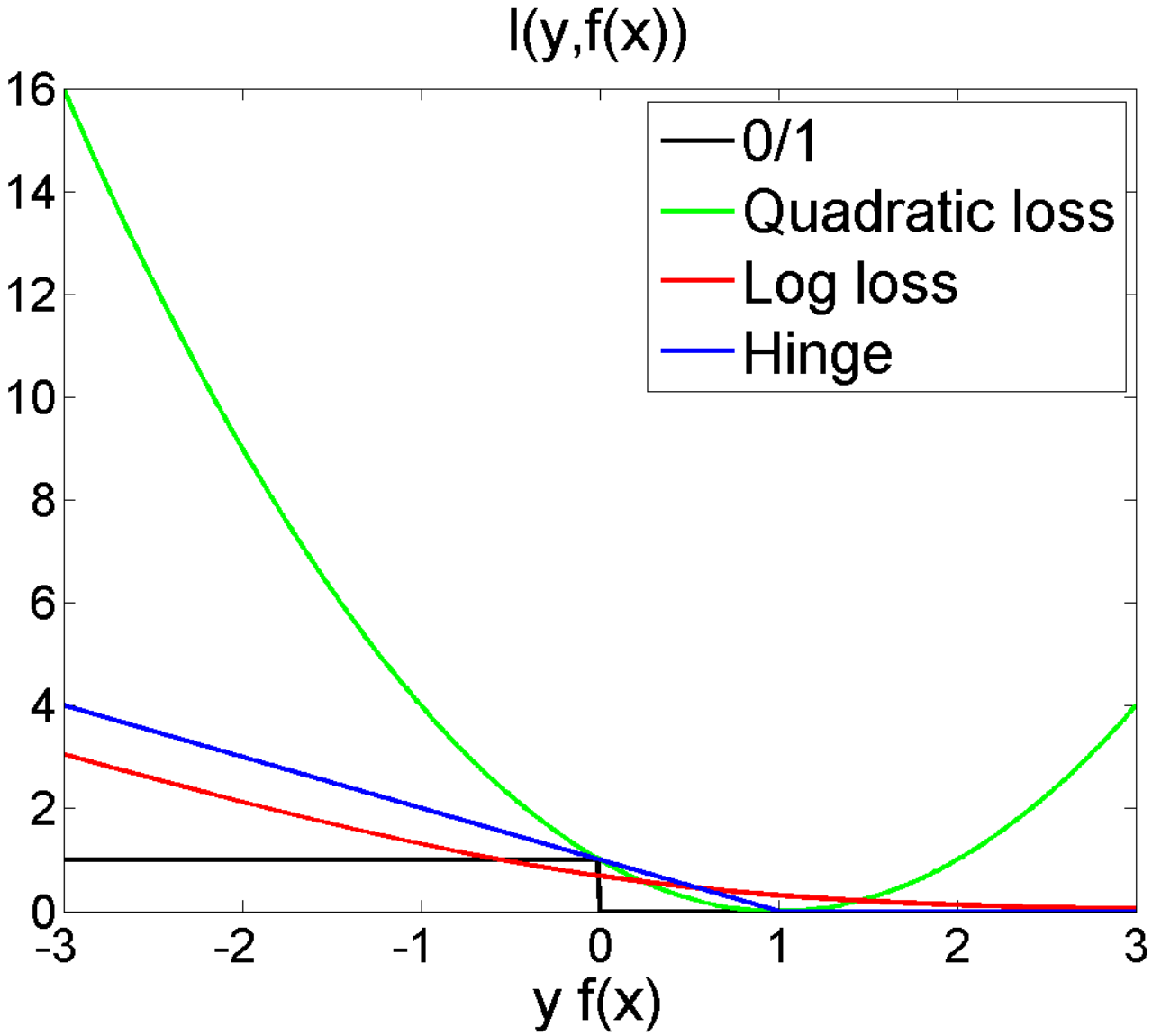
Hinge loss



Hinge loss vs log-loss



Hinge loss vs log-loss vs quadratic



Lecture outline

Recap

Large margins and generalization

Optimization

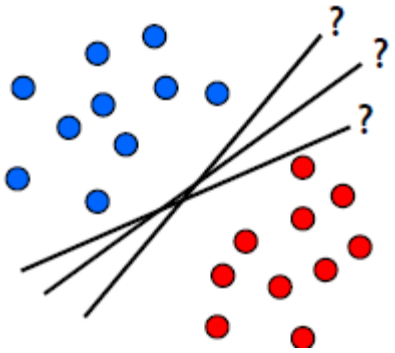
Kernels

Applications to vision

Generalization Error

- What is model complexity?
  - Number of parameters, magnitude of discriminant w?
  - Analyze complexity of hypothesis class

- Linear classifiers:
  - Different decision boundaries
    - Different generalization performance
  - Test error > training error
  - Which line gives smallest test error?



Learning Theory

- V. Vapnik, 1968
  - Mainstream Statistics: Large-sample analysis (`in the limit')
  - Pattern Recognition: Small sample properties

- Distribution-free bounds on worst performance

Empirical and Actual risk

- **Empirical risk**
  - Measured on the training/validation set

$$R_{emp}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i;\alpha))$$

- **Actual risk (= Expected risk)**
  - Expectation of the error on *all* data.

$$R(\alpha) = \int L(y_i, f(\mathbf{x};\alpha))dP_{X,Y}(\mathbf{x}, y)$$

  - $P_{X,Y}(\mathbf{x}, y)$ is the probability distribution of $(\mathbf{x},y)$. It is fixed, but typically unknown.

Actual and Empirical Risk

- Idea
  - Compute an upper bound on the actual risk based on the empirical risk

$$R(\alpha) \leq R_{emp}(\alpha) + \epsilon(N, p^*, h)$$

  - where

    N: number of training examples

    p^*: probability that the bound is correct

    h: capacity of the learning machine ("VC-dimension")

- With probability $(1\text{-}\eta)$, the following bound holds

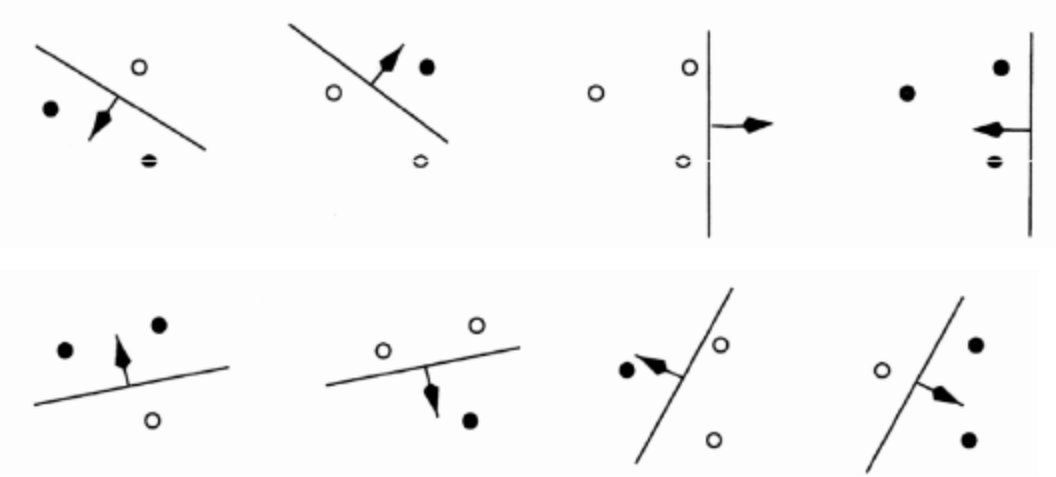
$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N}}$$

Vapnik Chervonenkis (VC) Dimension

- Shattering: *If a given set of ℓ points can be labeled in all possible 2^ℓ ways, and for each labeling, a member of the set $\{f(\alpha)\}$ can be found which correctly assigns those labels, we say that the set of points is shattered by the set of functions.*

- VC dimension *The VC dimension for the set of functions $\{f(\alpha)\}$ is defined as the maximum number of training points that can be shattered by $\{f(\alpha)\}$.*

- Example



Large Margins & VC Dimension

- Vapnik: *The class of optimal linear separators has VC dimension h bounded from above as*

$$h \leq \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$$

where ρ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples, and m_0 is the dimensionality.

- If we maximize the margins, feature dimensionality does not matter

Lecture outline

- Recap
- Large margins and generalization
- Optimization
  - Separable case
  - Non-separable case
- Kernels
- Applications to vision

Large Margin Classifiers

Separable case:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2$$

$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1, \quad i = 1 \ldots M$$

Lagrangian

- Constrained optimization problem:

$$\begin{aligned} \min_{w} \quad & f(w) \\ s.t. \quad & h_i(w) = 0, \quad i = 1 \dots l \\ & g_i(w) \leq 0, \quad i = 1 \dots m \end{aligned}$$

- Equivalent: $\min_{w} \quad f_{uc}(w) = f(w) + \sum_{i=1}^{l} I_0(h_i(w)) + \sum_{i=1}^{m} I_+(g_i(w))$

$$I_0(x) = \begin{cases} 0, & x = 0 \\ \infty, & x \neq 0 \end{cases}, \qquad I_+(x) = \begin{cases} 0, & x \leq 0 \\ \infty, & x > 0 \end{cases}$$

- `Soften' constraints:

$$L(w, \lambda, \mu) = f(w) + \sum_{i=1}^{l} \lambda_i h_i(w) + \sum_{i=1}^{m} \mu_i g_i(w), \quad \mu_i > 0 \forall i$$

- Solve: $f(w^*) = \min_{w} \max_{\lambda, \mu : \mu_i > 0} L(w, \lambda, \mu)$

Karush-Kuhn Tucker (KKT) Conditions

- At the problem solution we must have

$$
\begin{aligned}
h_i(w^*) &= 0 \\
g_i(w^*) &\leq 0 \\
\mu_i g_i(w^*) &= 0 \\
\mu_i &\geq 0 \\
\nabla f(w^*) + \sum_{i=1}^{l} \lambda_i \nabla h_i(w^*) + \sum_{i=1}^{m} \nabla g_i(w^*) &= 0
\end{aligned}
$$

Problem Lagrangian

Primal:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2$$

$$s.t. \quad -y^i(\mathbf{w}^T x^i + b) + 1 \leq 0, \quad i = 1 \ldots M$$

Lagrangian: $L(\mathbf{w}, b, \mu) = \frac{1}{2}|\mathbf{w}|^2 - \sum_{i=1}^{M} \mu_i [y^i(\mathbf{w}^T x^i + b) - 1] \qquad \mu_i \geq 0$

Optimum w.r.t. $\mathbf{w}$: $0 = \mathbf{w}^* - \sum_{i=1}^{M} \mu_i [y^i x^i] \qquad \mathbf{w}^* = \sum_{i=1}^{M} \mu_i y^i x^i$

Optimum w.r.t. b: $0 = \sum_{i=1}^{M} \mu_i y^i$

Dual for Large-Margin Classifier-I

Plug optimal values into Lagrangian:

$$
\begin{aligned}
\theta(\mu) &= L(\mathbf{w}^*, b^*, \mu) \\
&= \frac{1}{2}|\mathbf{w}^*|^2 - \sum_{i=1}^{M} \mu_i [y^i(\mathbf{w}^{*T} x^i + b) - 1] \\
&= \frac{1}{2}(\sum_{i=1}^{M} \mu_i y^i x^i)^T (\sum_{j=1}^{M} \mu_j y^j x^j) - \sum_{i=1}^{M} \mu_i [y^i((\sum_{j=1}^{M} \mu_j y^j x^j)^T x^i + b) - 1] \\
&= \sum_{i=1}^{M} \mu_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \mu_i \mu_j y^i y^j (x^i)^T (x^j) - b \sum_{i=1}^{M} \mu_i y^i
\end{aligned}
$$

Dual for Large-Margin Classifier-II

Equivalent optimization problem:

$$\max_{\mu} \quad \theta(\mu) = \sum_{i=1}^{M} \mu_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \mu_i \mu_j y^i y^j < x^i, x^j >$$

$$s.t. \quad \mu_i > 0, \quad \forall i$$

$$\sum_{i=1}^{M} \mu_i y^i = 0$$

Support Vectors

From complementary slackness (KKT) $\mu_i g_i(x) = 0$

where $g_i(x) = 1 - y^i(\mathbf{w}^T x^i + b) \quad (\leq 0)$

Therefore: $\mu_i \neq 0 \rightarrow y^i(\mathbf{w}^T x^i + b) = 1$

Interpretation: μ is nonzero only for points on the margin (hardest points)

From mimimum w.r.t. w: $\mathbf{w}^* = \sum_{i=1}^{M} \mu_i y^i x^i$

Interpretation: only points on the margin contribute to the solution

- **`Support Vectors'**

Intuitively ok: we want to maximize the margins of the hardest cases

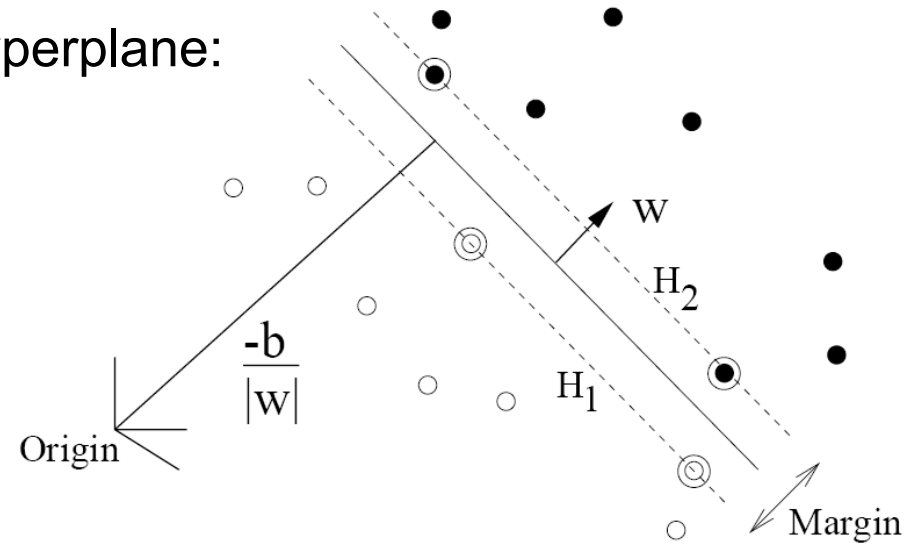
Decision Hyperplanes & Support Vectors

Use support vectors to determine b^*:

$$y^i(\mathbf{w}^T x^i + b) = 1, \quad \forall i \in S$$

$$b^* = \frac{1}{N_S} \sum_{i \in S} (y^i - \mathbf{w}^T x^i)$$

Support Vector Machine decision hyperplane:



Lecture outline

- Recap
- Large margins and generalization
- Optimization
  - Separable case
  - Non-separable case
- Kernels
- Applications to vision

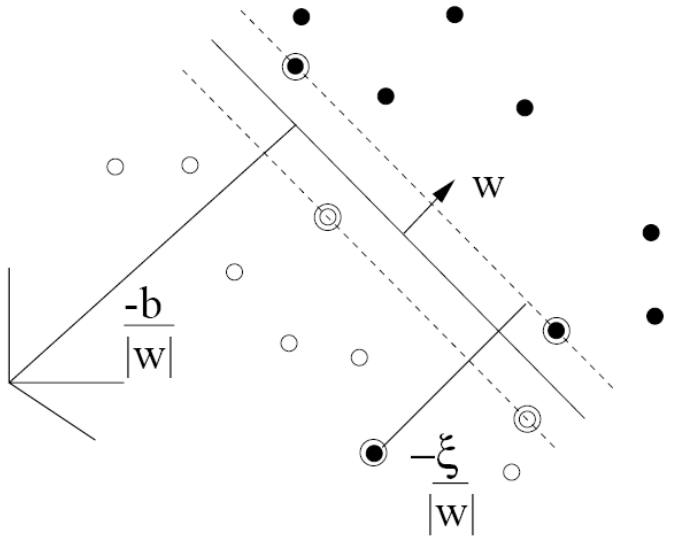
Non-separable data

Primal:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}\xi^i$$

$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i$$

$$\xi^i \geq 0$$



Lagrangian:

$$L(\mathbf{w}, b, \xi, \mu, \nu) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{M}\xi^i - \sum_{i=1}^{M}\mu_i[y^i(\mathbf{w}^T x + b) - 1 + \xi] - \sum_{i=1}^{M}\nu_i\xi_i$$

Dual:

$$\max_{\mu} \quad \sum_{i=1}^{M}\mu_i - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} y^i y^j \mu_i \mu_j < x^i, x^j >$$

$$s.t. \quad 0 \leq \mu_i \leq C$$

$$\sum_{i=1}^{M}\mu_i y^i = 0$$

$$\mu_i \geq 0, \quad \forall i$$

$$\nu_i \geq 0, \quad \forall i$$

KKT conditions – nonseparable case

$$
\begin{aligned}
C - \mu^i - \nu^i &= 0 \quad (1)\\
y^i(<\mathbf{x}^i, \mathbf{w}> + b) - 1 + \xi^i &\geq 0 \quad (2)\\
\mu^i \left[y^i(<\mathbf{x}^i, \mathbf{w}> + b) - 1 + \xi^i\right] &= 0 \quad (3)\\
\nu^i \xi^i &= 0 \quad (4)\\
\xi^i &\geq 0 \quad (5)\\
\mu^i &\geq 0 \quad (6)\\
\nu^i &\geq 0 \quad (7)
\end{aligned}
$$

Complementary slackness

Complementary slackness

Case analysis:

$$y^i(<\mathbf{x}^i, \mathbf{w}> + b) > 1 \xrightarrow{3:\xi^i \geq 0} \mu_i = 0$$

$$y^i(<\mathbf{x}^i, \mathbf{w}> + b) < 1 \xrightarrow{2:\xi^i > 0 \rightarrow 4:\nu^i = 0 \rightarrow 1:} \mu_i = C$$

$$y^i(<\mathbf{x}^i, \mathbf{w}> + b) = 1 \xrightarrow{3:\mu^i \xi^i = 0 \rightarrow 1:} \mu_i \in [0, C]$$

Interpretation: influence, μ, of any training point is bounded in [0,C]

Hinge Loss

$$
\begin{aligned}
C - \mu^i - \nu^i &= 0 \quad (1)\\
\mu^i \left[y^i (< \mathbf{x}^i, \mathbf{w} > + b) - 1 + \xi^i \right] &= 0 \quad (3)\\
\nu^i \xi^i &= 0 \quad (4)\\
\xi^i &\geq 0 \quad (5)
\end{aligned}
$$

$$\mu^i \neq 0 \overset{(3)}{\rightarrow} \xi^i = 1 - y^i (< \mathbf{x}^i, \mathbf{w} > + b)$$

$$\mu^i = 0 \overset{(1)}{\rightarrow} \nu^i = C \quad \overset{(4)}{\rightarrow} \xi^i = 0$$

$$\xi^i = \max(0, 1 - y^i (< \mathbf{x}^i, \mathbf{w} > + b))$$

$$
\begin{aligned}
\min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{M} \xi^i \\
s.t. \quad & y^i (\mathbf{w}^T x^i + b) \geq 1 - \xi^i \\
& \xi^i \geq 0
\end{aligned}
\quad \Longleftrightarrow \quad
L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y^i h_{\mathbf{w},b}(x^i))
$$

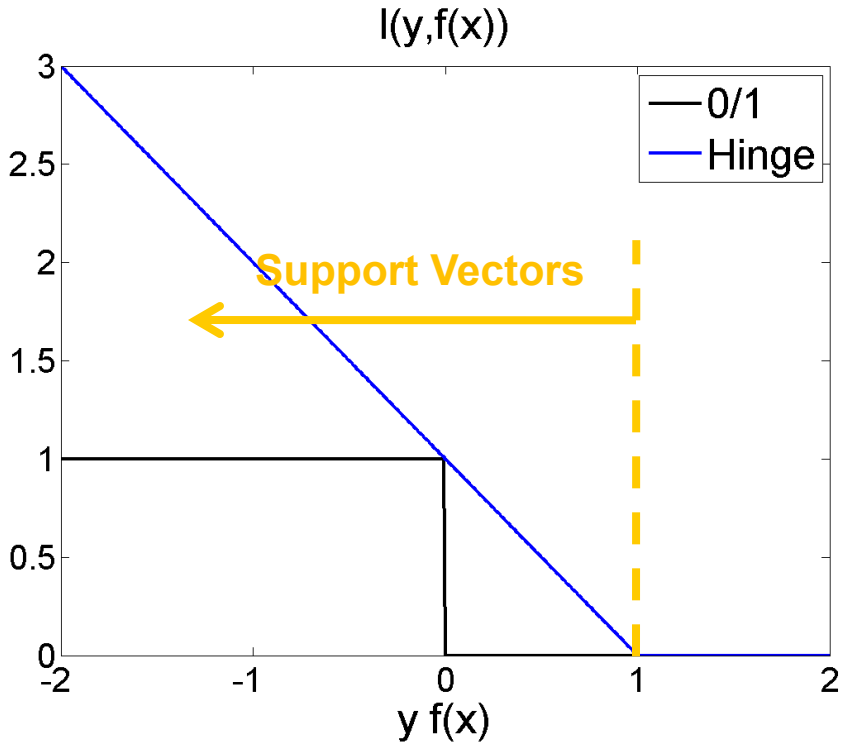
Loss function for SVM training

Optimization problem:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\max(0, 1 - y^i h_{\mathbf{w},b}(x^i))$$

$$\propto \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\underbrace{\max(0, 1 - y^i h_{\mathbf{w},b}(x^i))}_{l(y^i, x^i)}$$

Hinge loss:



Lecture outline

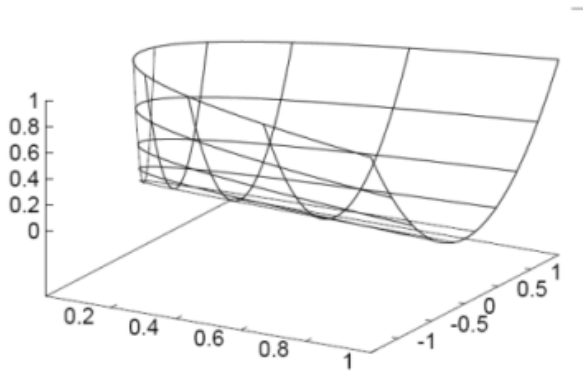
Recap

Large margins and generalization

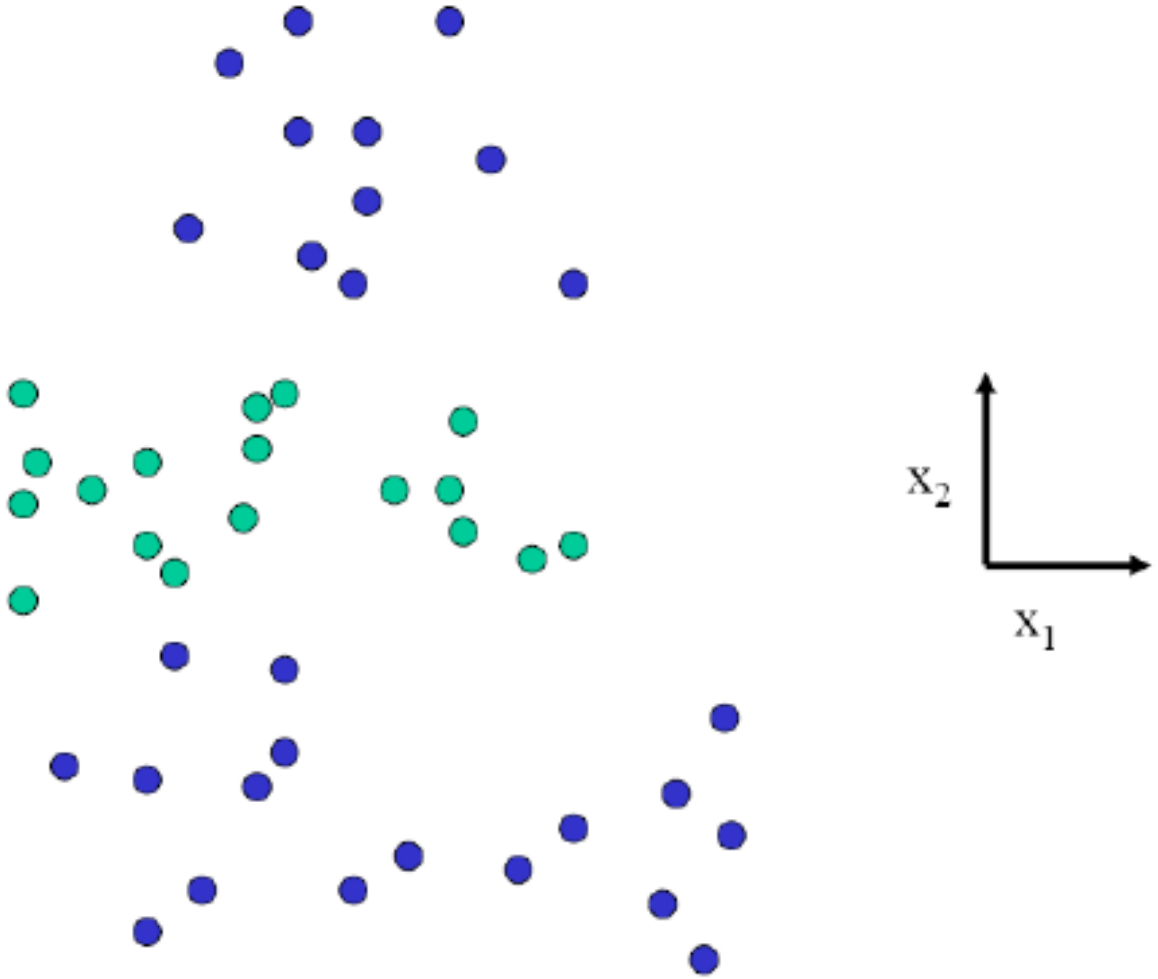
Optimization

Kernels

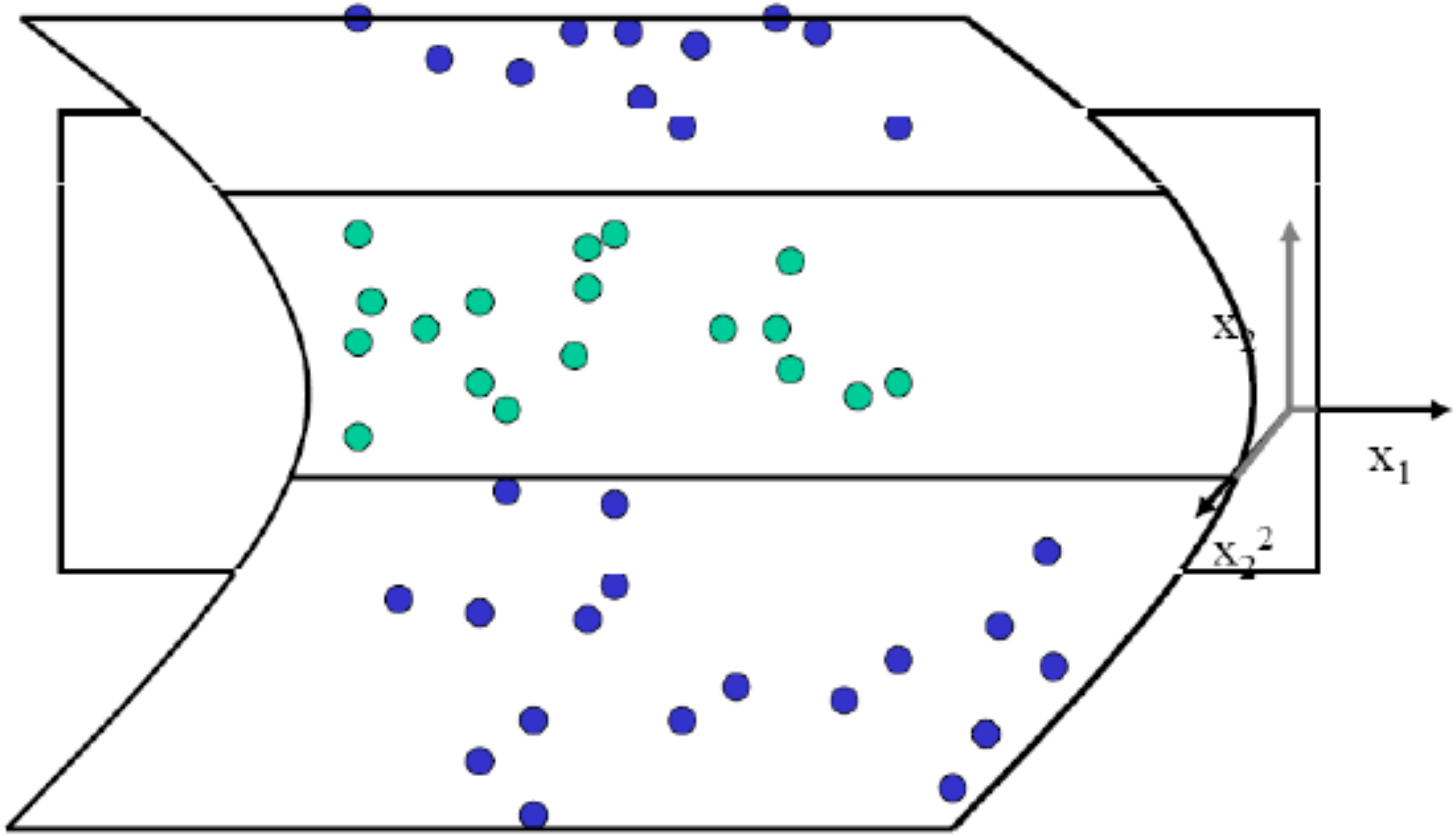
Applications to vision



Nonseparable in 2D



Separable in 3D



Non-linear Classifiers

So far, decision is based on the sign of $y = \mathbf{w}^T x = \sum_{i=1}^{N} \mathbf{w}_i x_i$

Use non-linear transformation, φ(x) of our data, x

e.g. $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ x_1 x_2 \end{bmatrix}$

Classifier: $y = \mathbf{w}^T \phi(x) = \sum_{i=1}^{K} \mathbf{w}_i \phi_k(x)$

Quadratic boundaries: $\sum_i \mathbf{w}_i \phi_i(x) = 0$

Non-linear in x, linear in φ(x)

Inner products & SVMs

We know that: $\mathbf{w}^* = \sum_{i=1}^{M} \mu_i y^i x^i$

Classifier form: $y = \text{sign}(< \mathbf{w}, \mathbf{x} > +b) \overset{\mu_i' = \mu_i y^i}{=} \text{sign}(\sum_{i=1}^{M} \mu_i' < \mathbf{x}^i, \mathbf{x} > +b)$

If instead of x we use φ(x), classifier becomes:

$$y = \text{sign}(\sum_{i=1}^{M} \mu_i' < \phi(\mathbf{x}^i), \phi(\mathbf{x}) > +b)$$

Dual:

$$\max_{\mu} \quad \sum_{i=1}^{M} \mu_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} y^i y^j \mu_i \mu_j < \phi(x^i), \phi(x^j) >$$

$$s.t. \quad 0 \leq \mu_i \leq C$$

$$\sum_{i=1}^{M} \mu_i y^i = 0$$

`Kernelization'

Define Kernel: $K(\mathbf{x}, \mathbf{y}) = < \phi(\mathbf{x}), \phi(\mathbf{y}) >$

Rewrite everything with kernels: same optimization in the dual!

$$y = \text{sign}(\sum_{i=1}^{M} \mu_i y^i K(\mathbf{x}^i, \mathbf{x}) + b)$$

$$\max_{\mu} \quad \sum_{i=1}^{M} \mu_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} y^i y^j \mu_i \mu_j K(\mathbf{x}^i, \mathbf{x}^j)$$

$$s.t. \quad 0 \leq \mu_i \leq C$$

$$\sum_{i=1}^{M} \mu_i y^i = 0$$

`Kernel trick'

Kernel: $K(\mathbf{x}, \mathbf{y}) = < \phi(\mathbf{x}), \phi(\mathbf{y}) >$

Consider: $\phi(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{bmatrix}$

We then have:

$$\begin{aligned} < \phi(x), \phi(y) > &= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\ &= (x_1y_1 + x_2y_2)^2 \\ &= (x^Ty)^2 \\ &= K(x, y) \end{aligned}$$

Using polynomial Kernel $K(x, y) = (x^Ty + c)^d$

amounts to using a vector of dimensionality $\begin{pmatrix} N+d \\ d \end{pmatrix}$

Kernel Examples

- Polynomial kernel

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{\mathrm{T}}\mathbf{y} + 1)^p$$

- Radial Basis Function kernel

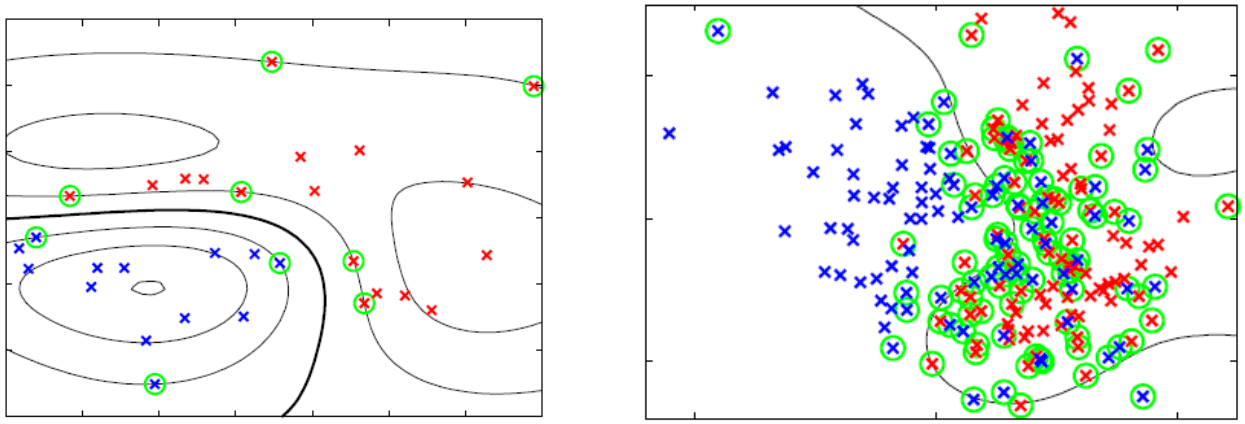
$$k(\mathbf{x}, \mathbf{y}) = \exp\left\{-\frac{(\mathbf{x} - \mathbf{y})^2}{2\sigma^2}\right\}$$

- Hyperbolic tangent kernel

$$k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^{\mathrm{T}}\mathbf{y} + \delta)$$

Nonlinear SVMs

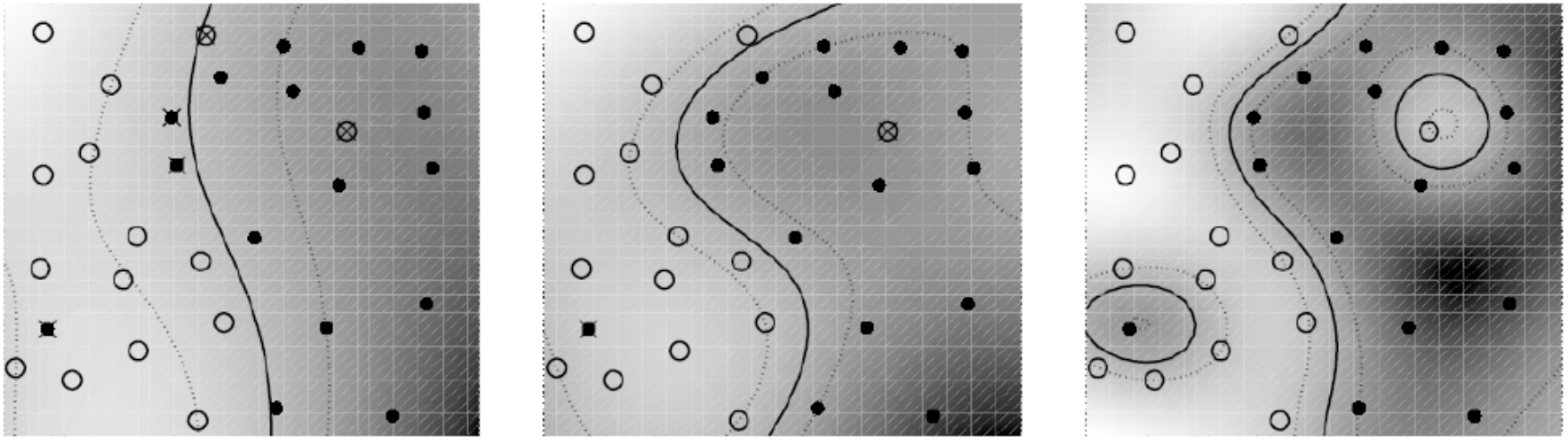
Decision function and boundaries with Gaussian kernel:



Similar flexibility with RBF networks, 1-layer perceptrons, etc.

But: global optimum + good generalization

Large margins for nonlinear classifiers



RBF Kernel width (σ)

Margin size: determined by both σ and regularizer

Lecture outline

Recap

Large margins and generalization

Optimization

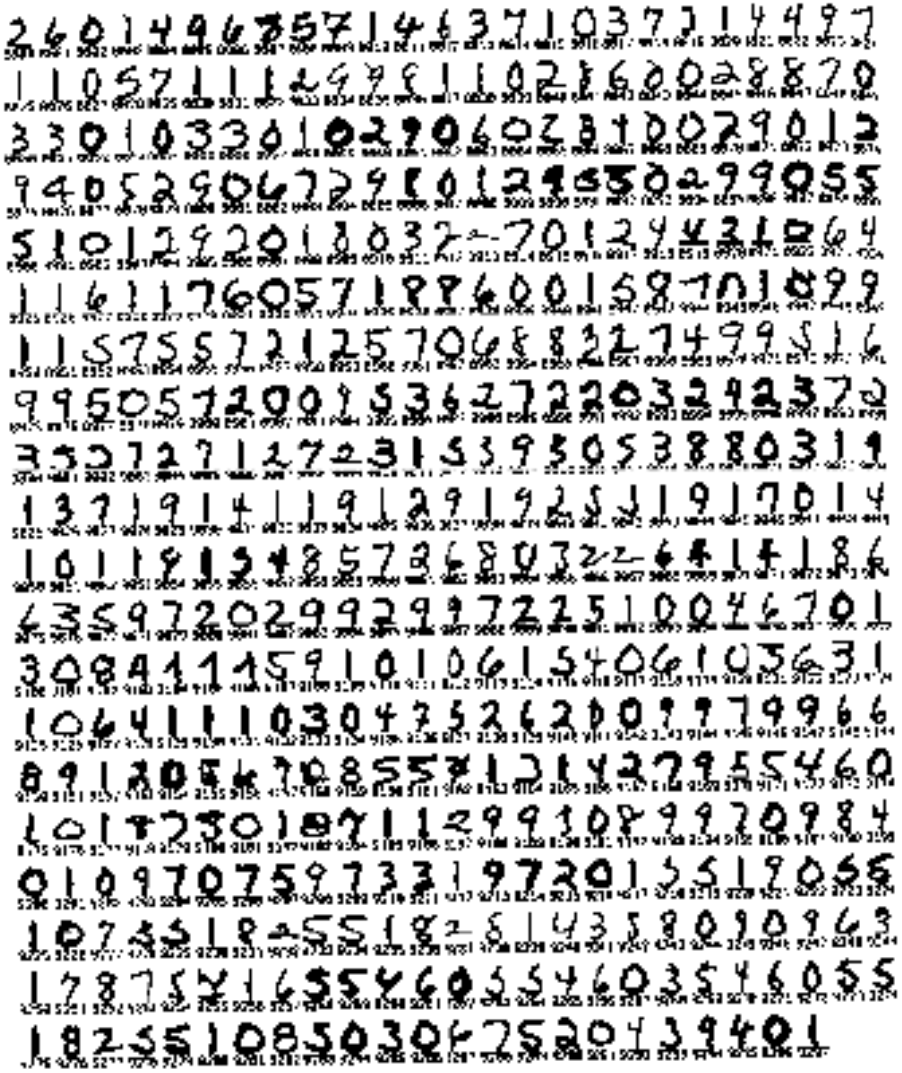
Kernels

Applications to vision

- Strong models
- Loose models

Guyon & Vapnik, 1995

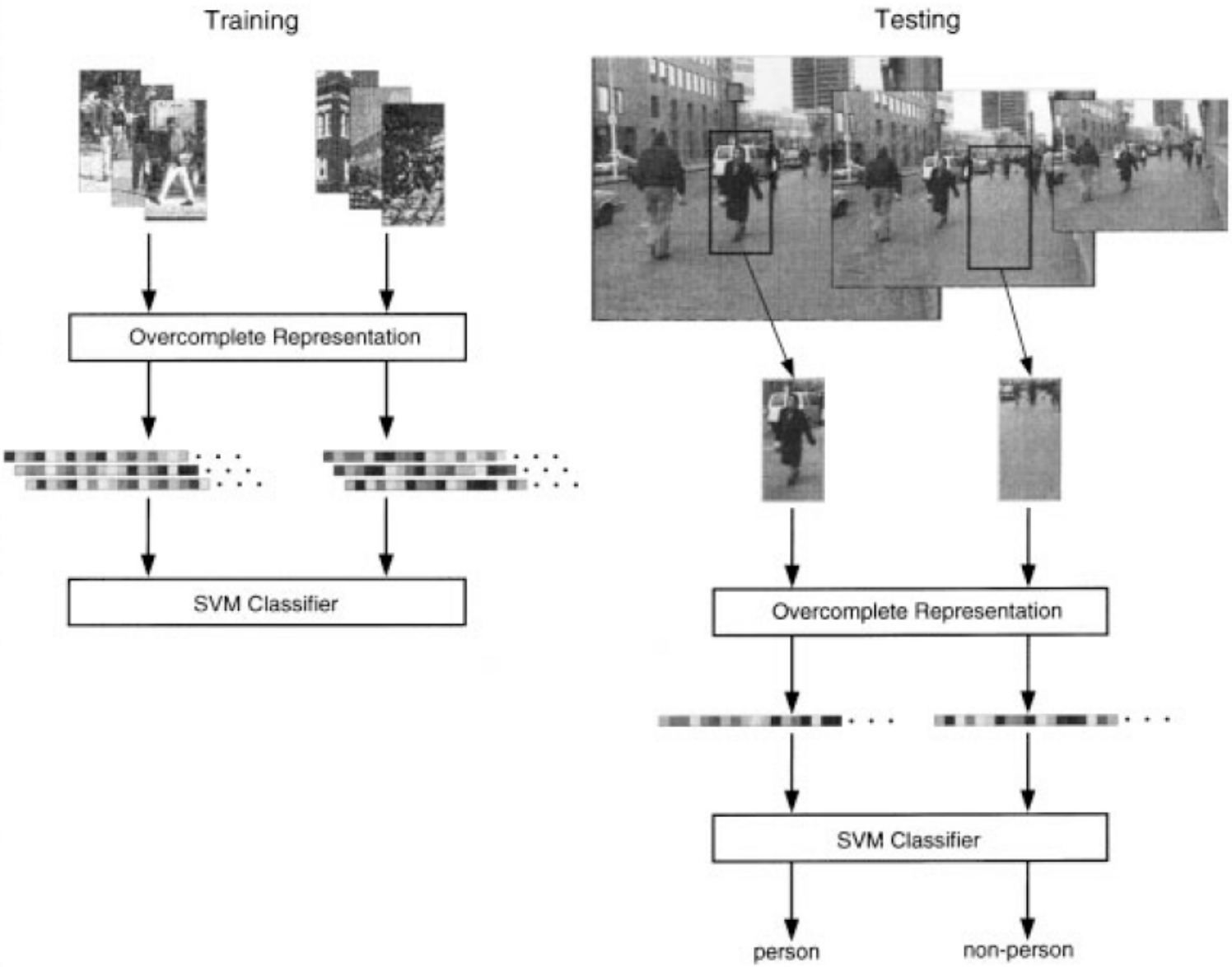
- Handwritten digit recognition
  - US Postal Service Database
  - Standard benchmark task for many learning algorithms



Guyon & Vapnik 1995

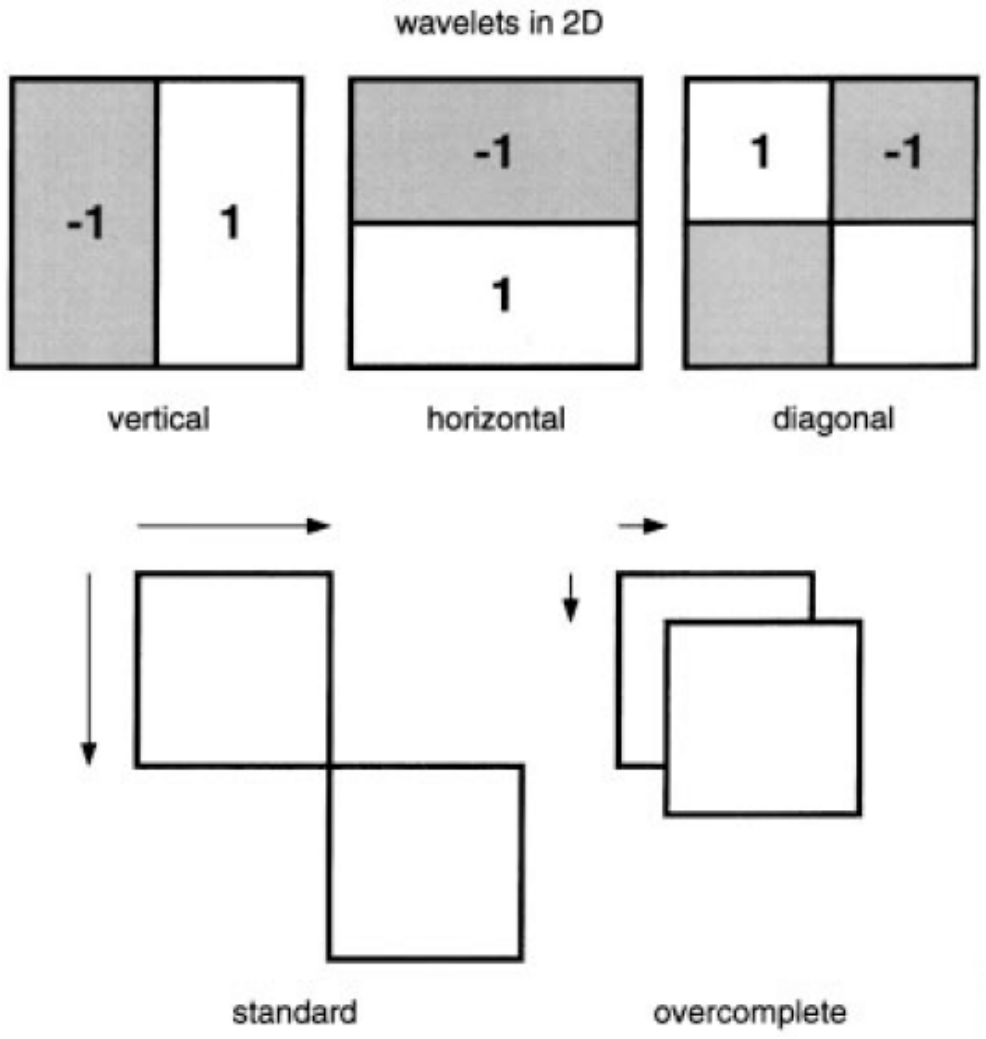
- USPS benchmark
  - 2.5% error: human performance
- Different learning algorithms
  - 16.2% error: Decision tree (C4.5)
  - 5.9% error: (best) 2-layer Neural Network
  - 5.1% error: LeNet 1 - (massively hand-tuned) 5-layer network
- Different SVMs
  - 4.0% error: Polynomial kernel (p=3, 274 support vectors)
  - 4.1% error: Gaussian kernel (σ=0.3, 291 support vectors)

Papageorgiou & Poggio, CVPR 1998

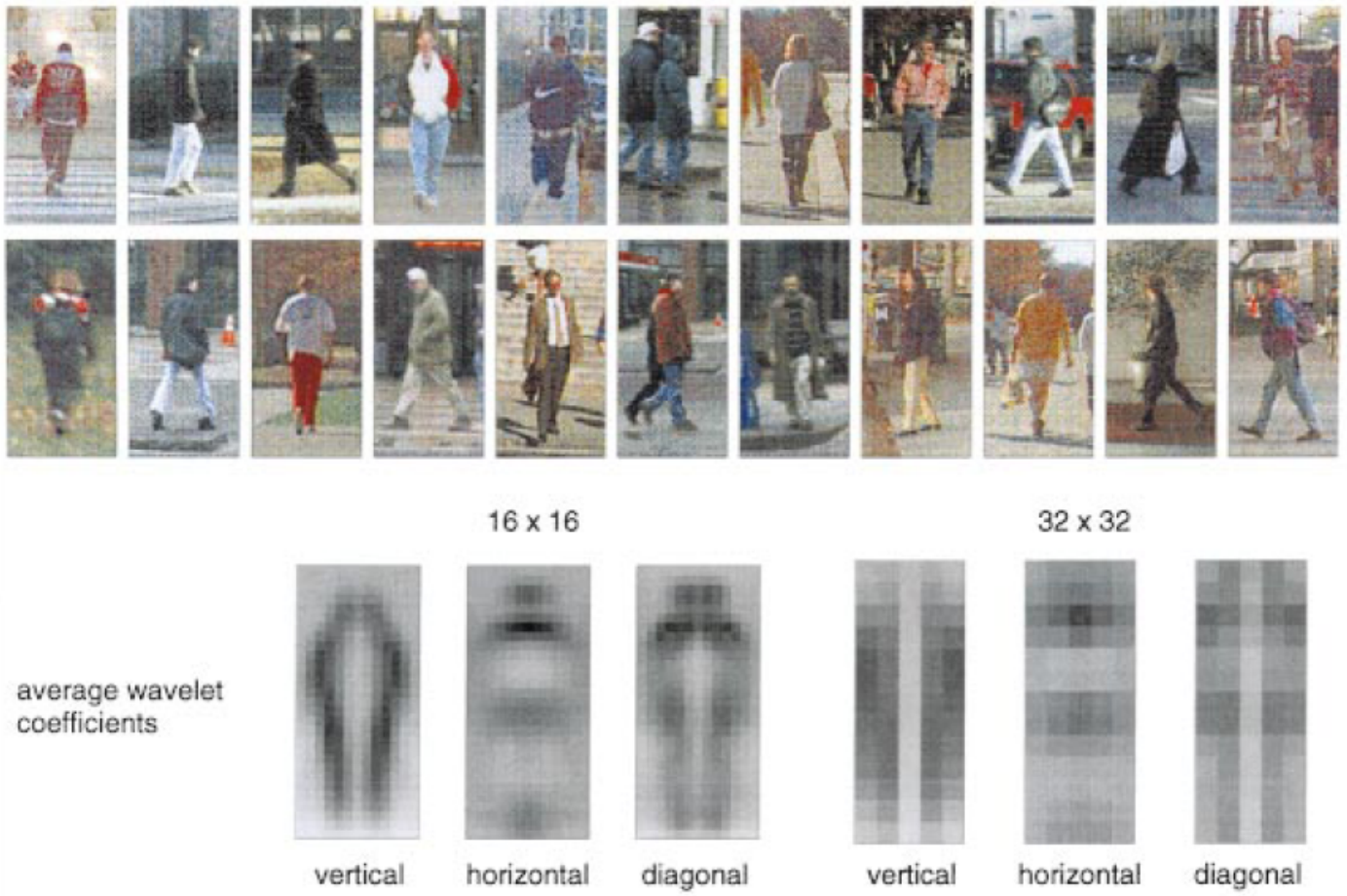


Papageorgiou & Poggio: Image Representation

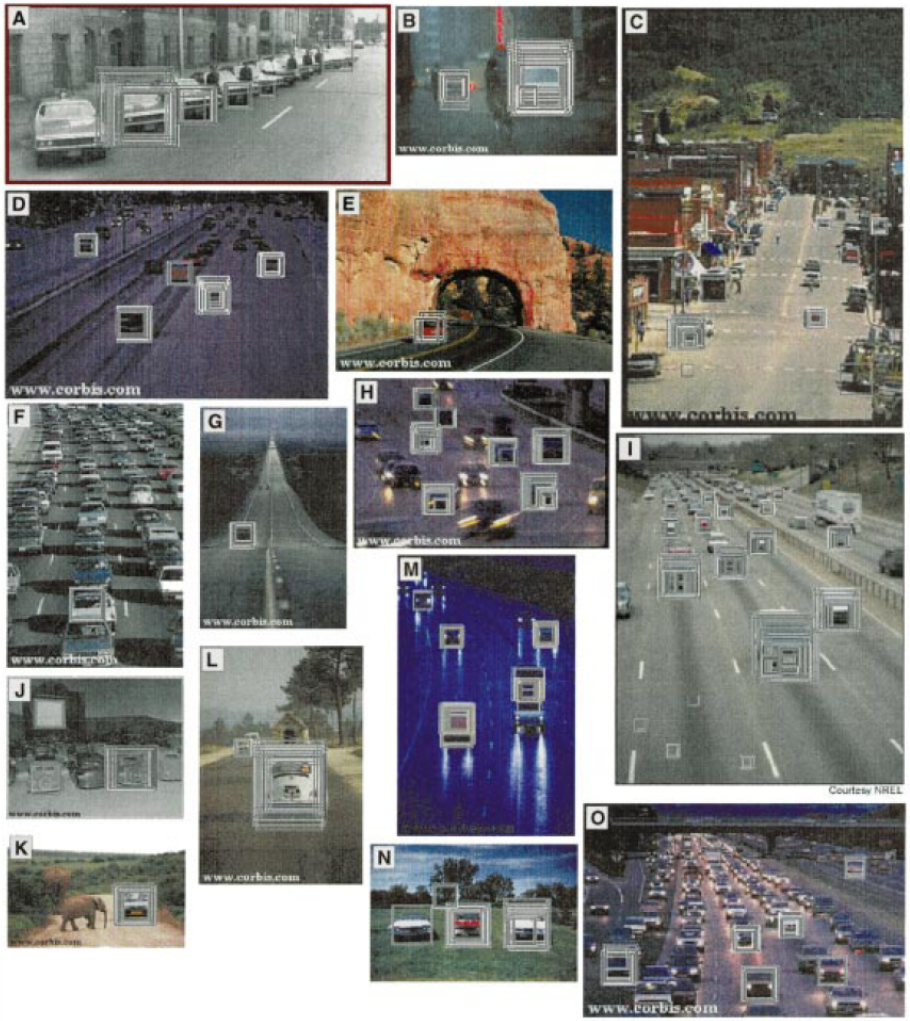
- Haar features



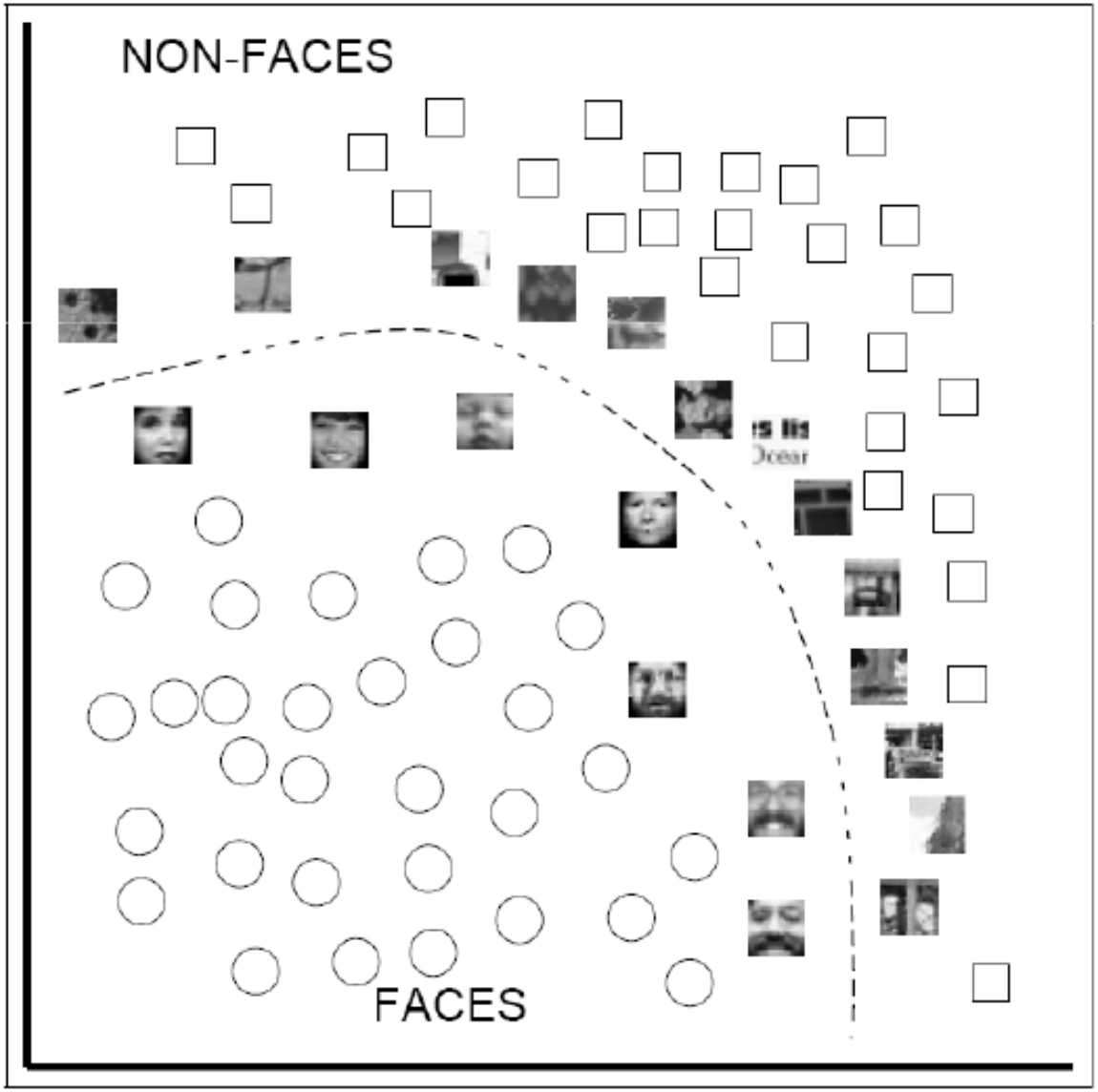
Papageorgiou & Poggio: training set



Sample Detections

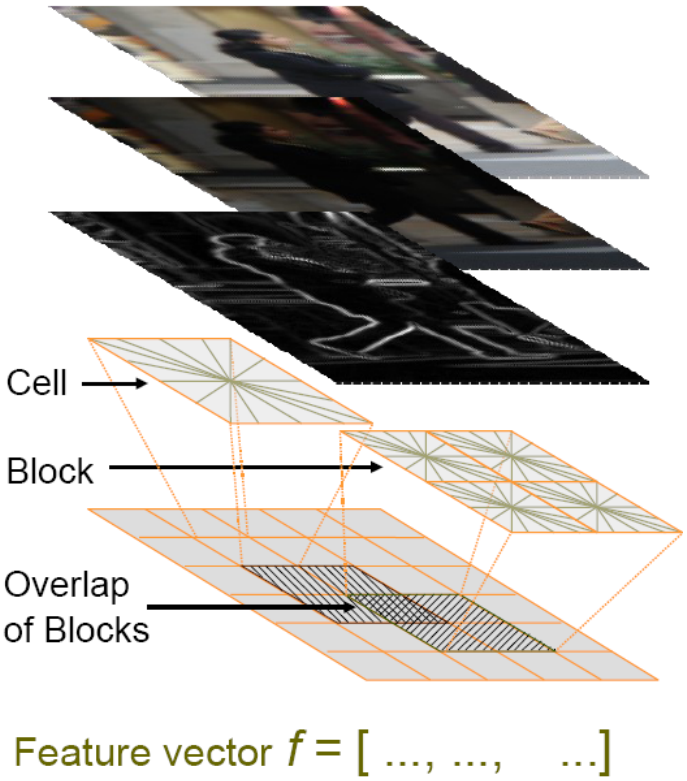


Support vectors for Faces (P&P 98)



Dalal and Triggs, ICCV 2005

- Histogram of Oriented Gradient (HOG) features
- Highly accurate detection using linear SVM



Lecture outline

- Recap
- Large margins and generalization
- Optimization
- Kernels
- Applications to vision
  - Strong models
  - Loose models

Bag-of-word models

Object → Bag of 'words'

Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that [illegible] brain from our eyes. For [illegible] ought that the re[illegible] point by [illegible] brain; t[illegible] screen [illegible] image [illegible] the dis[illegible] now kn[illegible] visual pe[illegible] considerab[illegible] of events. By fo[illegible] ses along their path [illegible] vers of the optical cortex, Hubel and [illegible] have been able to demonstrate th[illegible] *message about the image falling o[illegible] retina undergoes a step-wise analys[illegible] system of nerve cells stored in colum[illegible] In this system each cell has its specific function and is responsible for a specific detail in the pattern of the retinal image.*

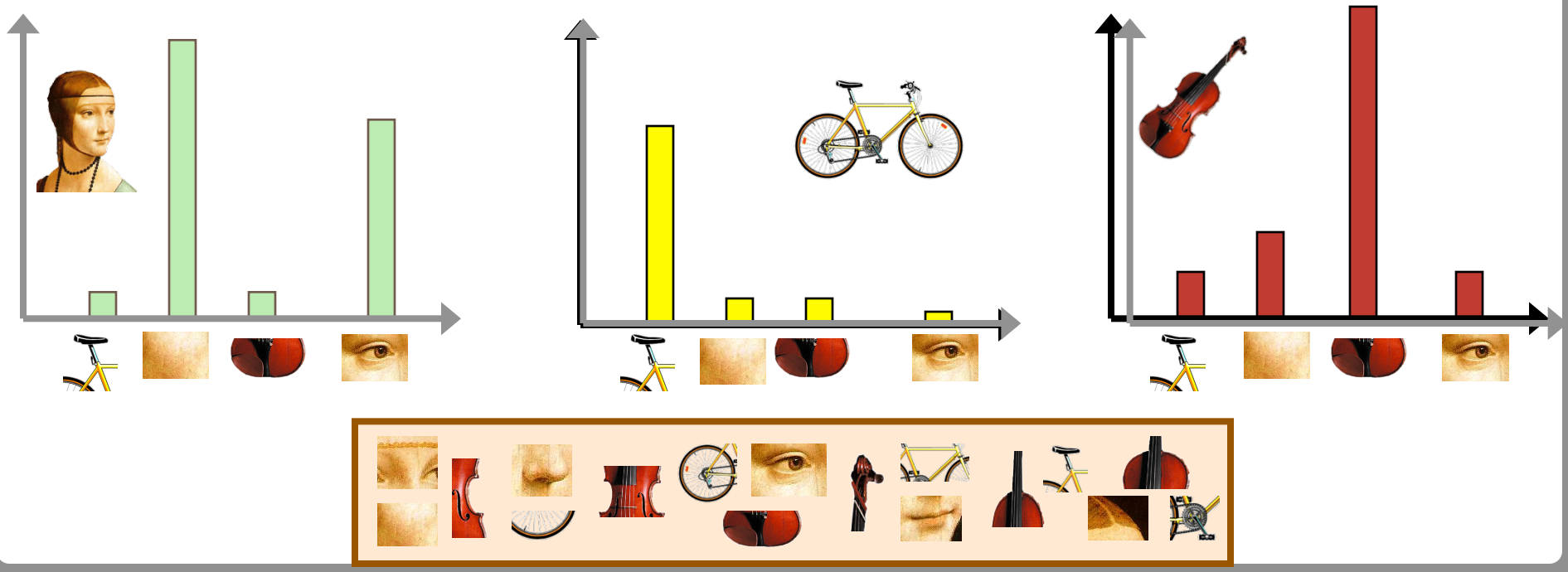
sensory, brain, visual, perception, retinal, cerebral cortex, eye, cell, optical nerve, image Hubel, Wiesel

China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump in exports to $[illegible] with a 18% rise in [illegible] figures are likely [illegible] ch has lon[illegible] e unfairl[illegible] under[illegible] surplu[illegible] only o[illegible] Zhou X[illegible] needed t[illegible] demand so[illegible] the country. China[illegible] f the yuan against the d[illegible] ly and permitted it to trade within a[illegible] w band, but the US wants the yuan t[illegible] allowed to trade freely. However, Be[illegible] has made it clear that it will take its t[illegible] and tread carefully before allowing the yuan to rise further in value.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

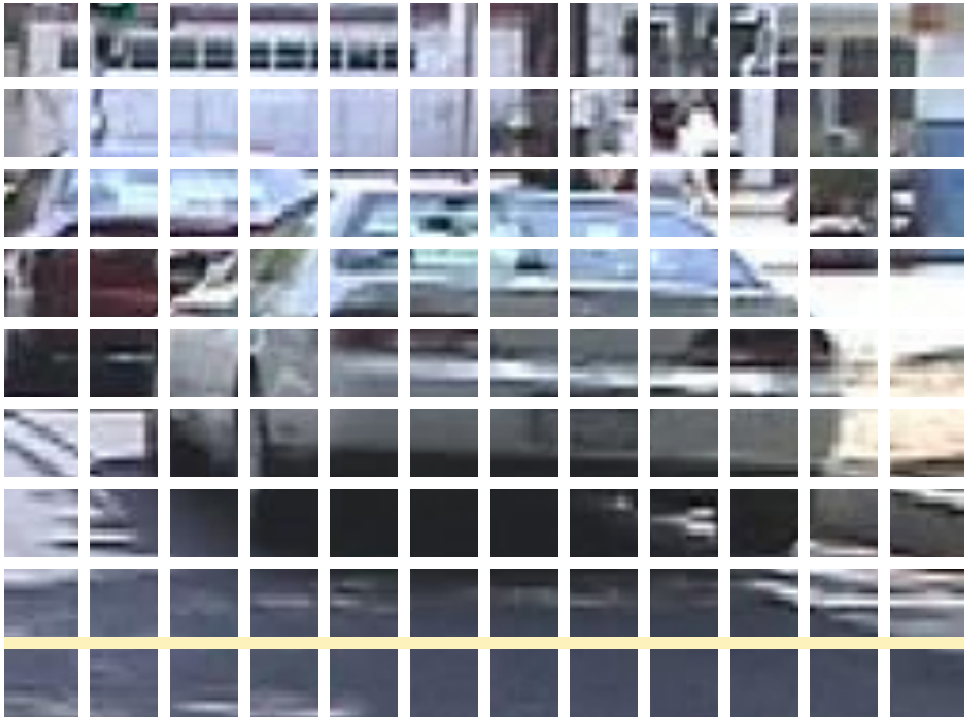
Bag of words model

- Independent features
- histogram representation



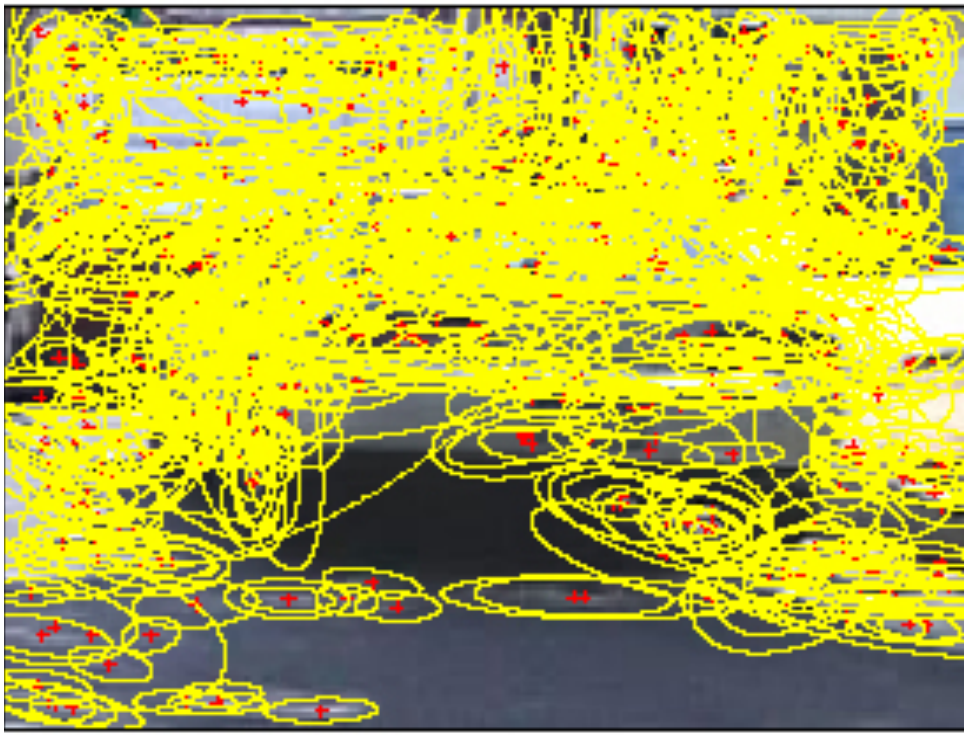
Feature Detection

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005



Feature Detection

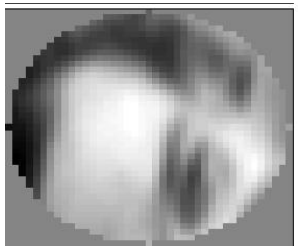
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic, et al. 2005



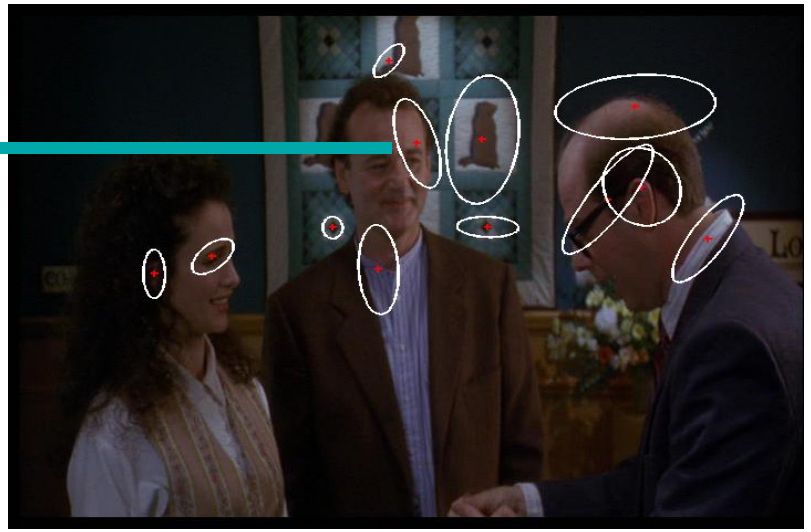
Feature Representation

Compute SIFT descriptor

[Lowe'99]



Normalize patch



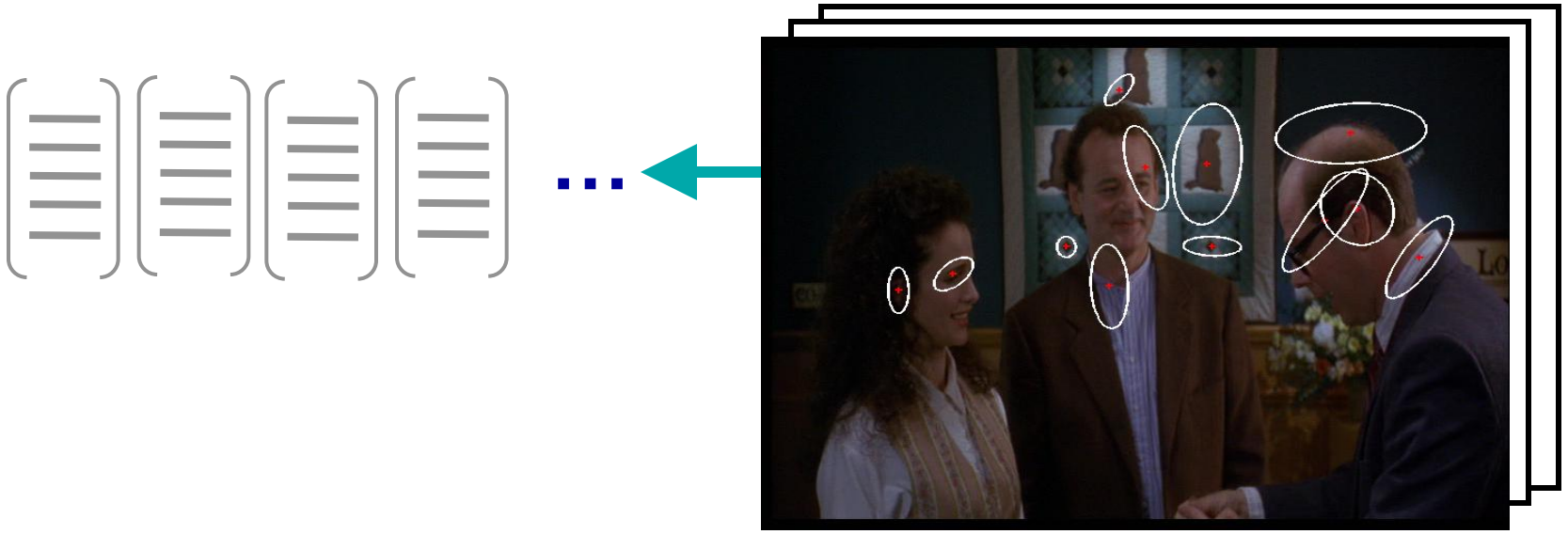
Detect patches

[Mikojaczyk and Schmid '02]

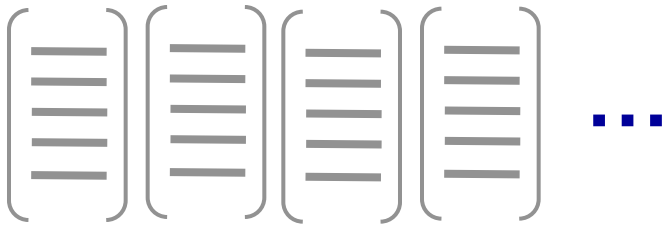
[Mata, Chum, Urban & Pajdla, '02]

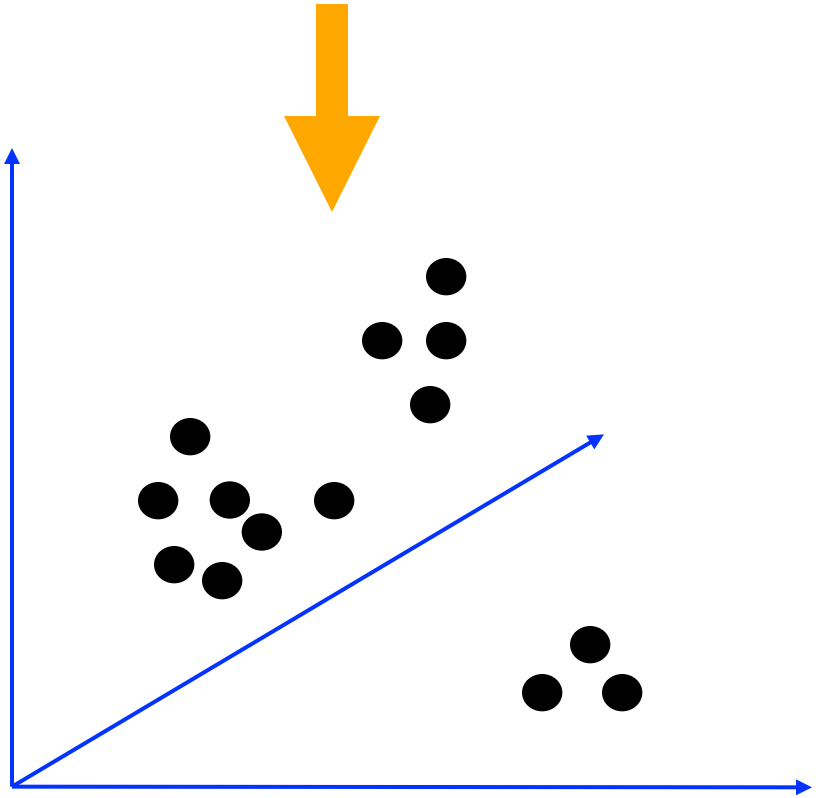
[Sivic & Zisserman, '03]

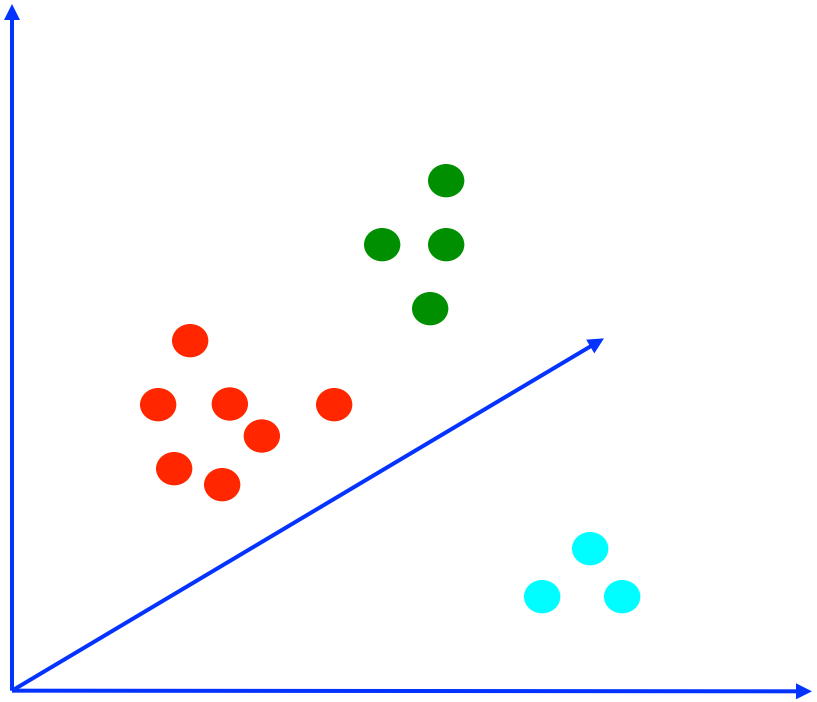
Feature Representation



Codebook Formation

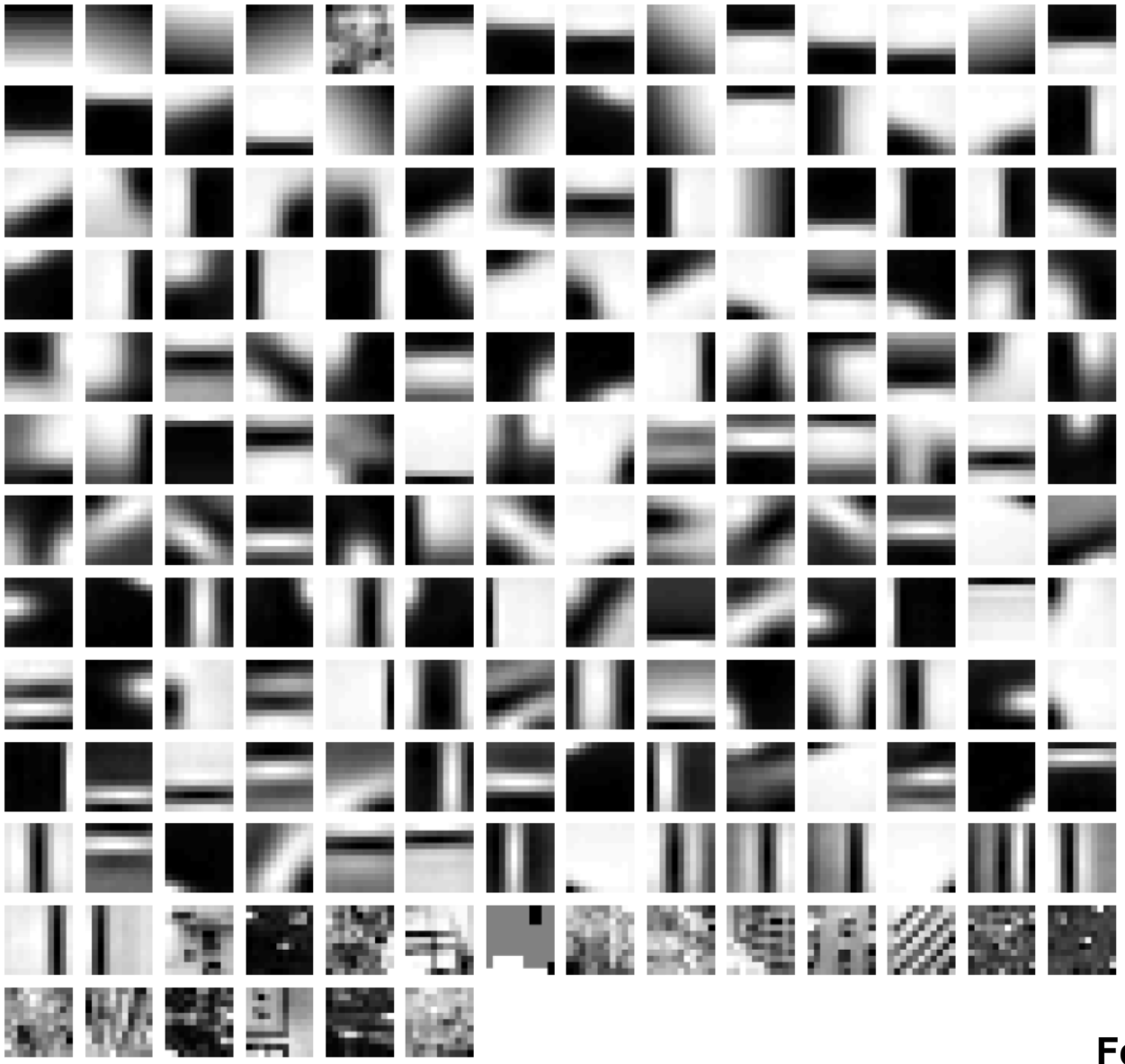






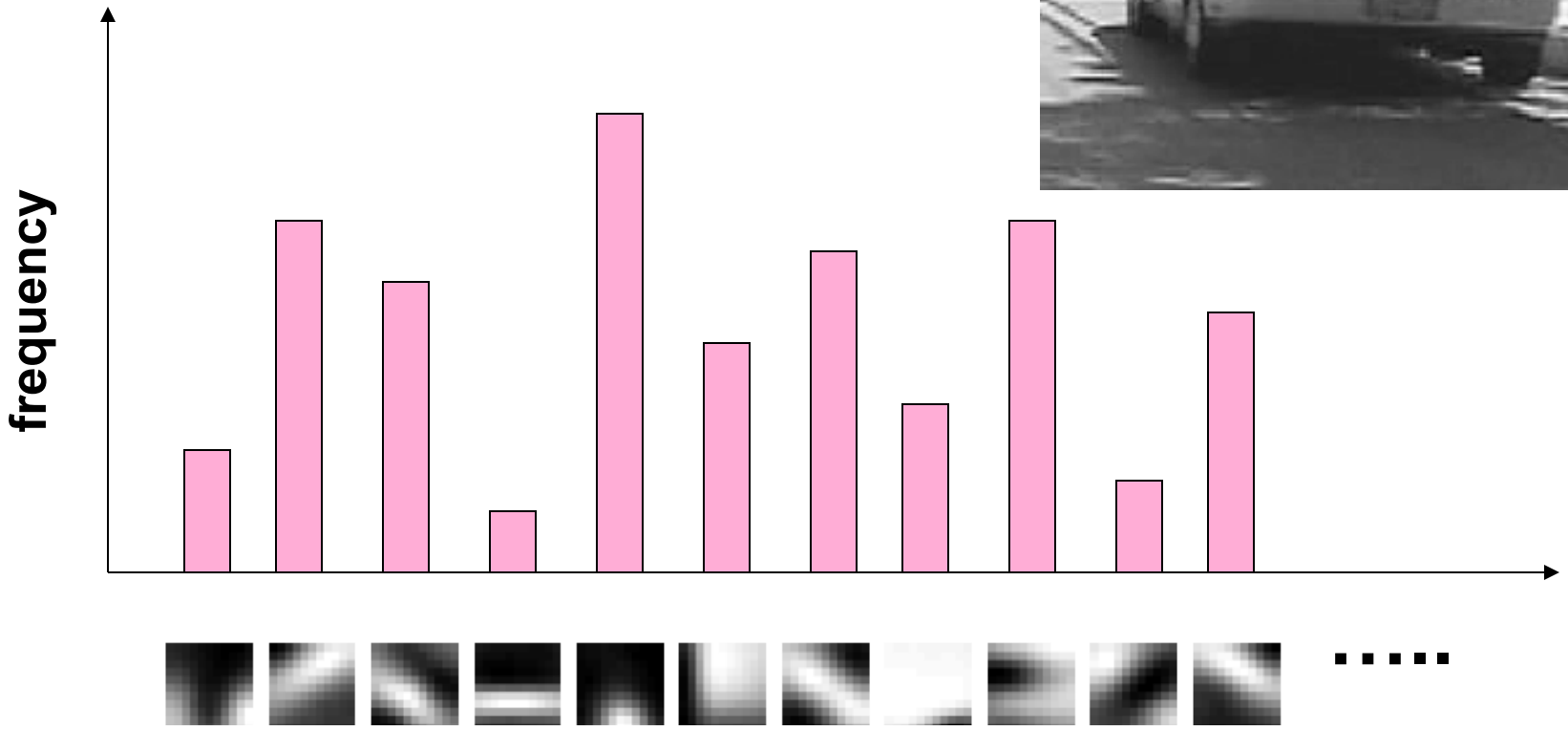
Clustering (e.g. k-means)

Codebook



Fei-Fei et al. 2005

Image Representation (Feature Vector)



Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories

- Svetlana Lazebnik, Cordelia Schmid, Jean Ponce

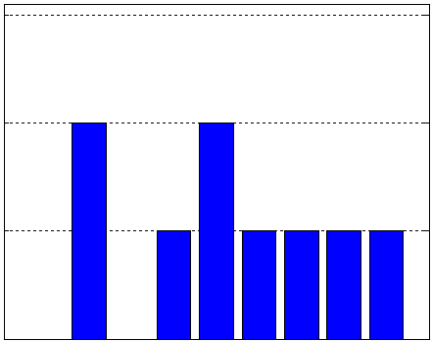
The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features

- Kristen Grauman, Trevor Darrell

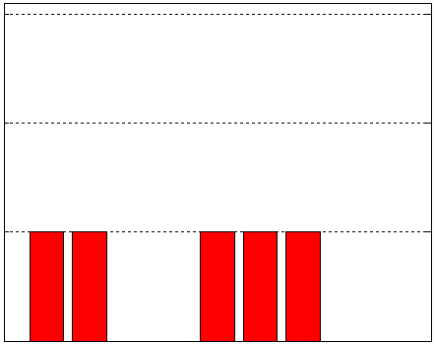
Intersection Kernel

Histogram intersection

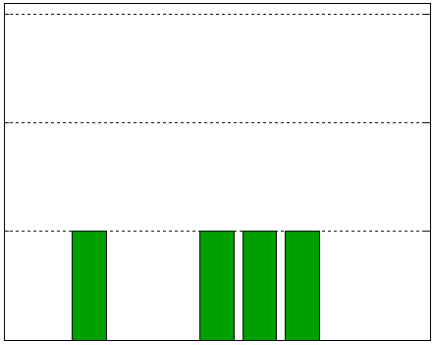
$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$



$H(\mathbf{X})$



$H(\mathbf{Y})$



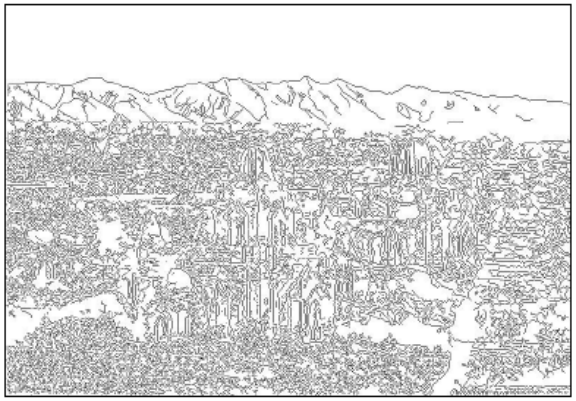
$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = 4$

Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Create intersection kernels
5. Train an SVM

Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Create intersection kernels
5. Train an SVM



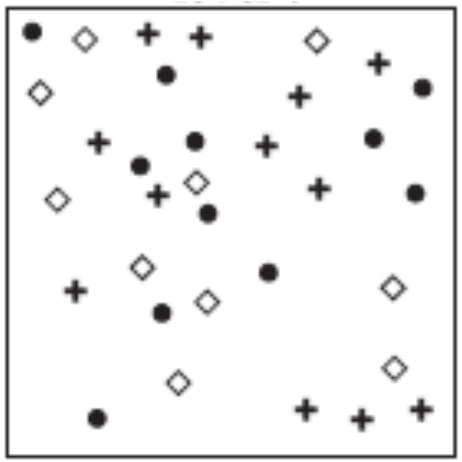
Weak (edge orientations)

OR

Strong (SIFT)

Algorithm

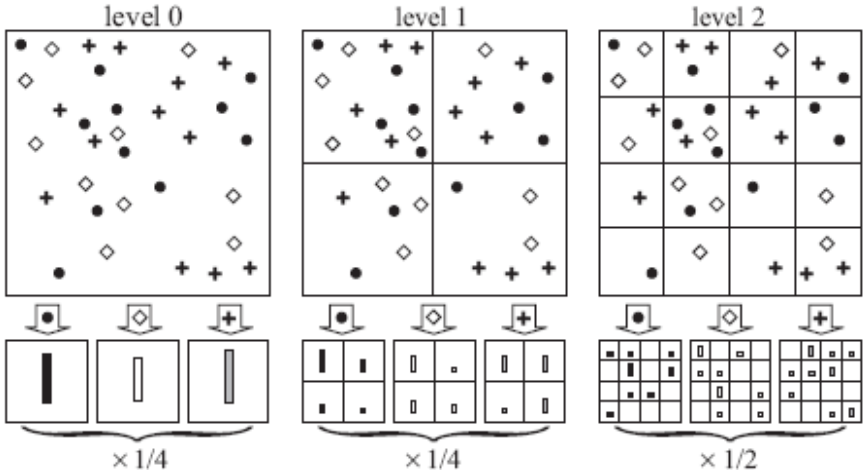
1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Create intersection kernels
5. Train an SVM



- **Vector quantization**
- **Usually K-means clustering**
- **Vocabulary size (16 to 400)**

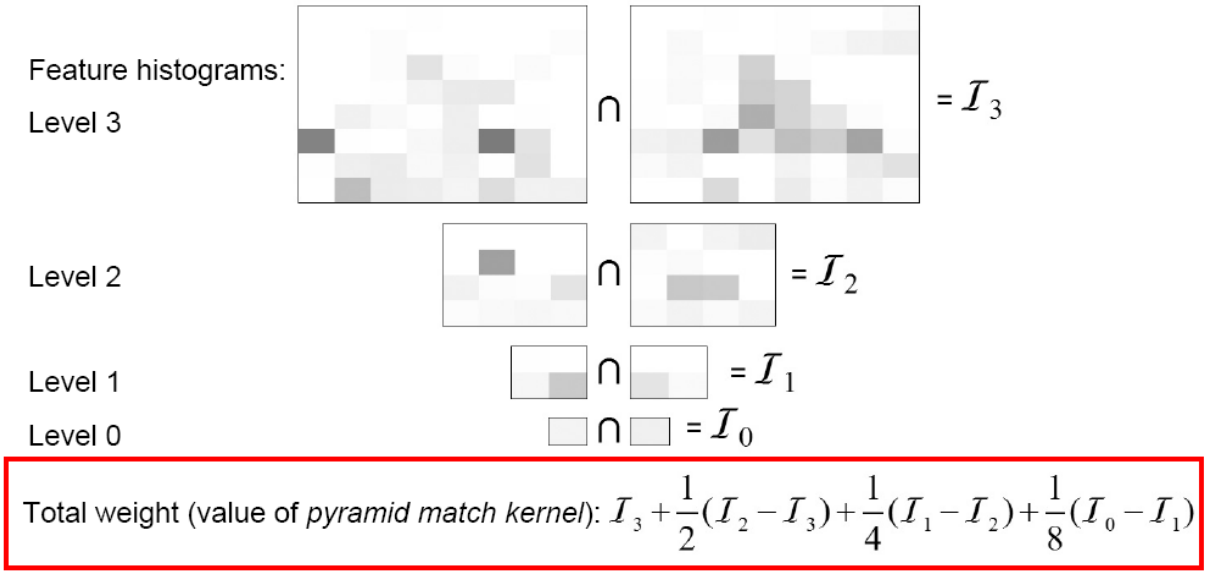
Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Create intersection kernels
5. Train an SVM



Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Create intersection kernels
5. Train an SVM



Algorithm

1. Extract interest point descriptors (dense scan)
2. Construct visual word dictionary
3. Build spatial histograms
4. Create intersection kernels
5. Train an SVM

Scene category dataset

Fei-Fei & Perona (2005), Oliva & Torralba (2001)

http://www-cvr.ai.uiuc.edu/ponce_grp/data

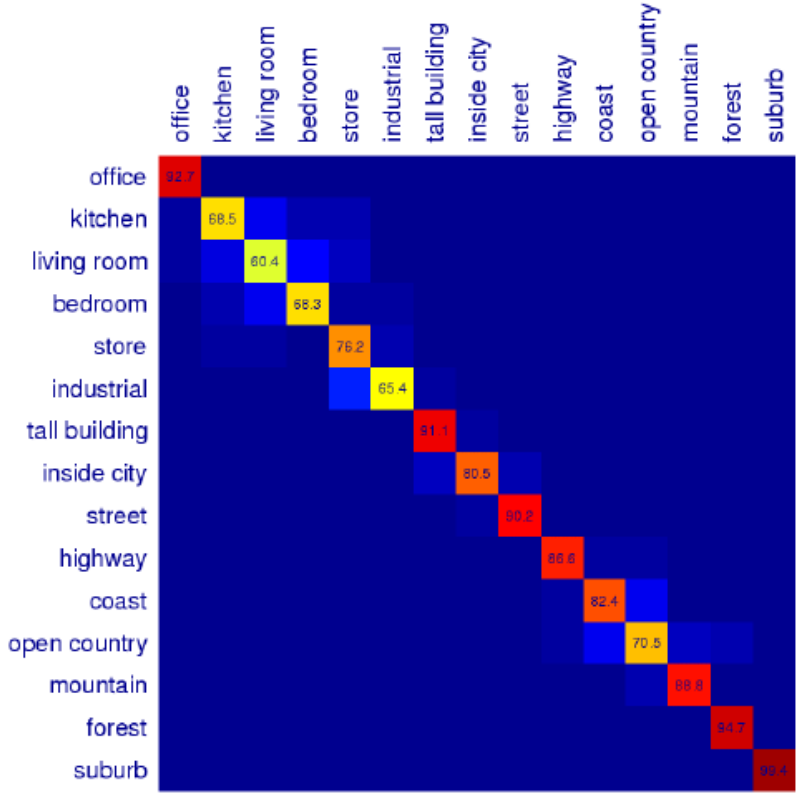
Multi-class classification results (100 training images per class)

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1 × 1)	45.3 ±0.5		72.2 ±0.6	
1 (2 × 2)	53.6 ±0.3	56.2 ±0.6	77.9 ±0.6	79.0 ±0.5
2 (4 × 4)	61.7 ±0.6	64.7 ±0.7	79.4 ±0.3	**81.1** ±0.3
3 (8 × 8)	63.3 ±0.8	**66.8** ±0.6	77.2 ±0.4	80.7 ±0.3

Fei-Fei & Perona: 65.2%

Scene category retrieval

Scene category confusions



Difficult indoor images

kitchen

living room

bedroom

Beyond Sliding Windows: Object Localization by *Efficient Subwindow Search*

Christoph H. Lampert[†], Matthew B. Blaschko[†], & Thomas Hofmann[‡]



CVPR 2008 best paper award

Sliding window classifiers

0.1

Sliding window classifiers

-0.2

Sliding window classifiers

. . .
1.5
. . .

Sliding window classifiers

0.5

Sliding window classifiers

0.3

Sliding window classifiers

0.1
-0.2
-0.1
0.1
. . .
1.5
. . .
0.5
0.4
0.3

Sliding window classifiers

approach: sliding window classifier

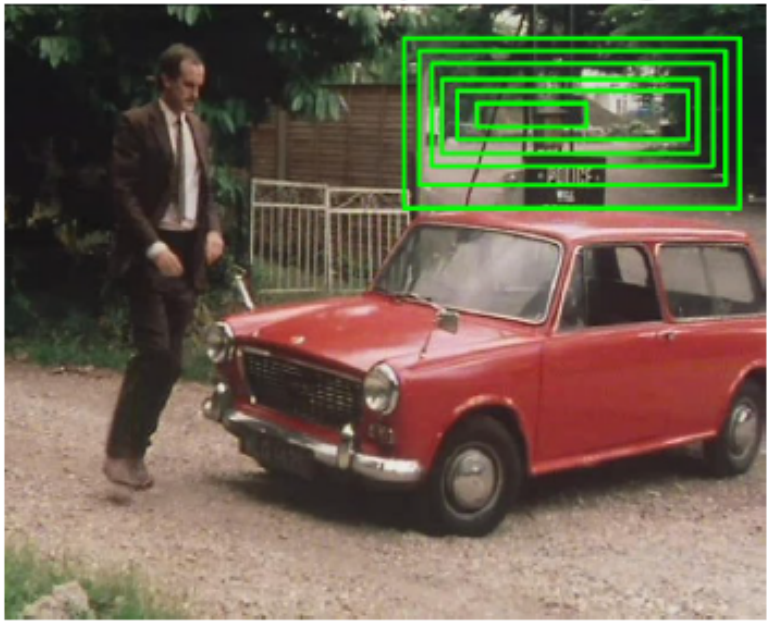
- evaluate classifier at candidate regions in an image - $\text{argmax}_{B \in \mathcal{B}} f_I(B)$
- for a 640×480 pixel image, there are over *10 billion* possible regions to evaluate

sample a subset of regions to evaluate

- scale
- aspect ratio
- grid size

Beyond sliding windows

Problem: Exhaustive evaluation of $\text{argmax}_{B \in \mathcal{B}} f_I(B)$ is too slow.
Solution: Use the problem's *geometric structure*.

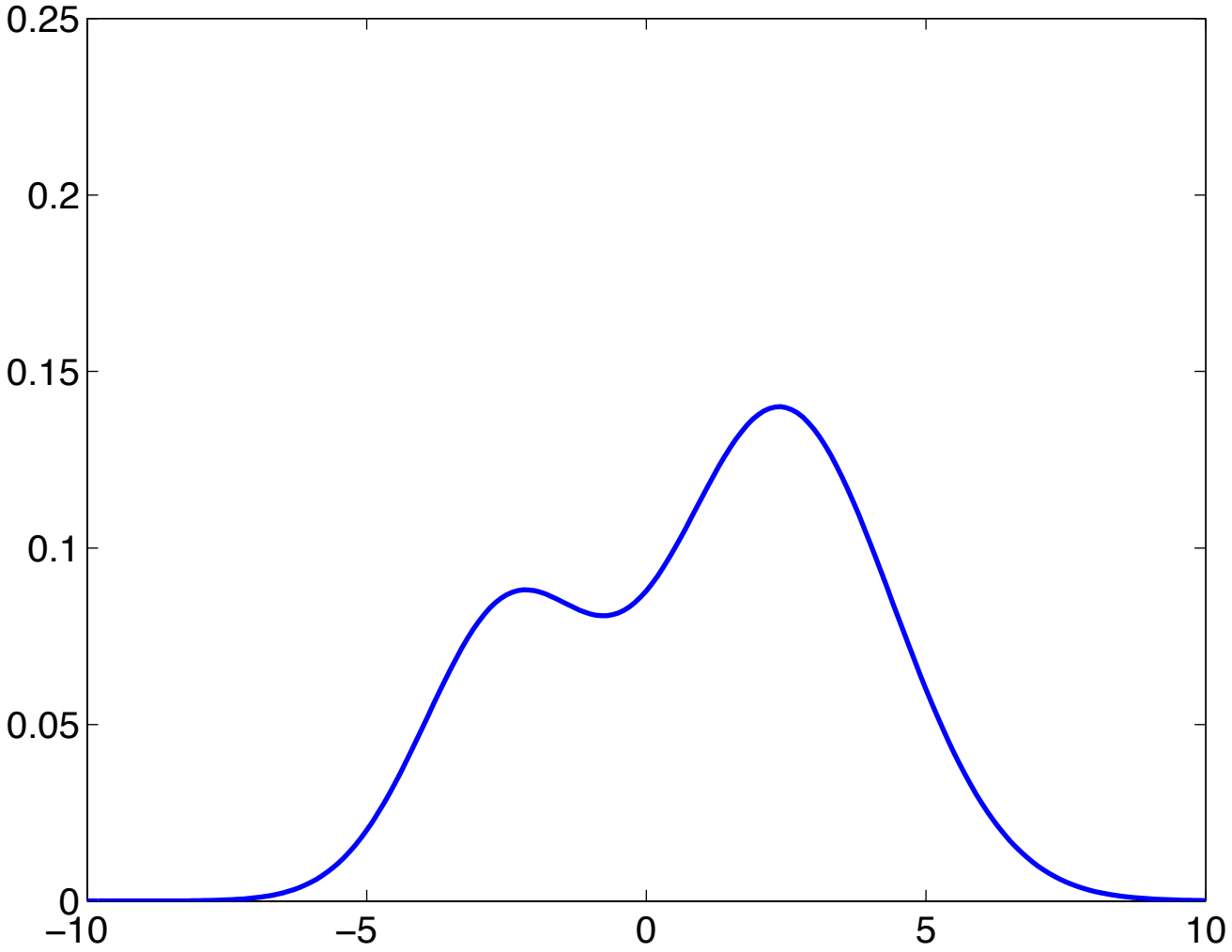


- Similar boxes have similar scores.
- Calculate scores for *sets of boxes* jointly (upper bound).
- If no element can contain the object, discard the set.
- Else, split the set into smaller parts and re-check, etc.

$\Rightarrow$ efficient branch & bound algorithm

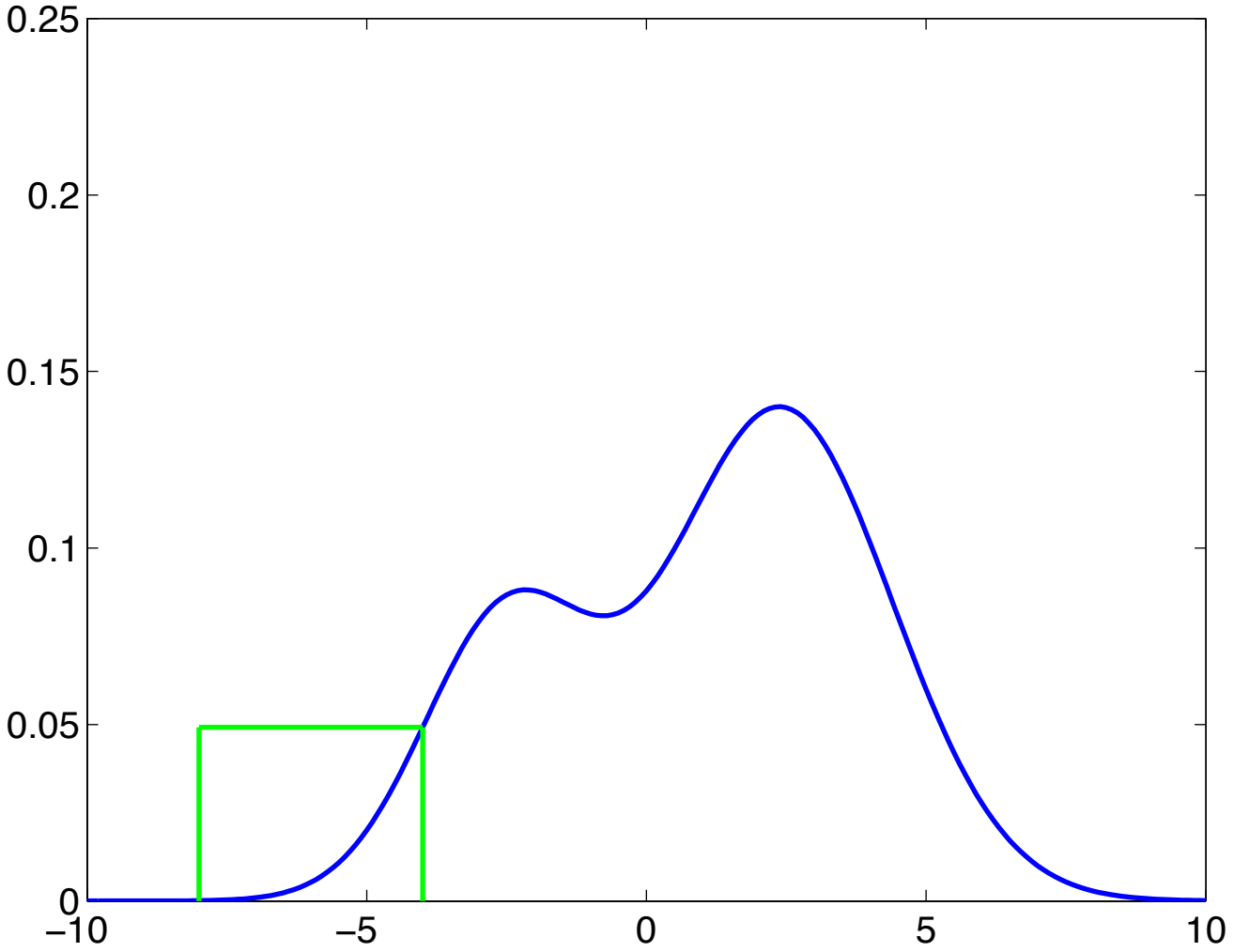
Branch and bound

$$\max_{x \in X} f(x)$$



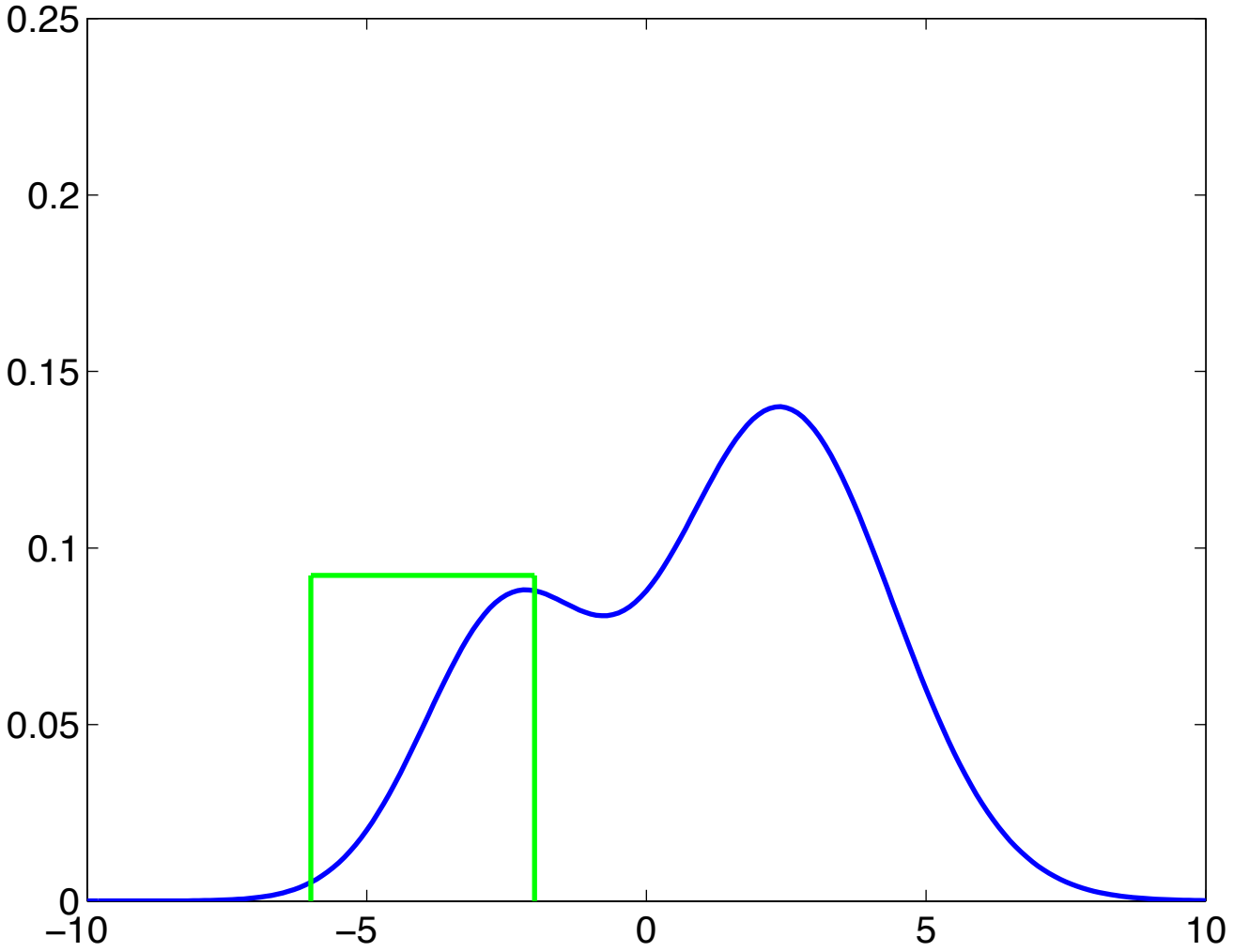
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



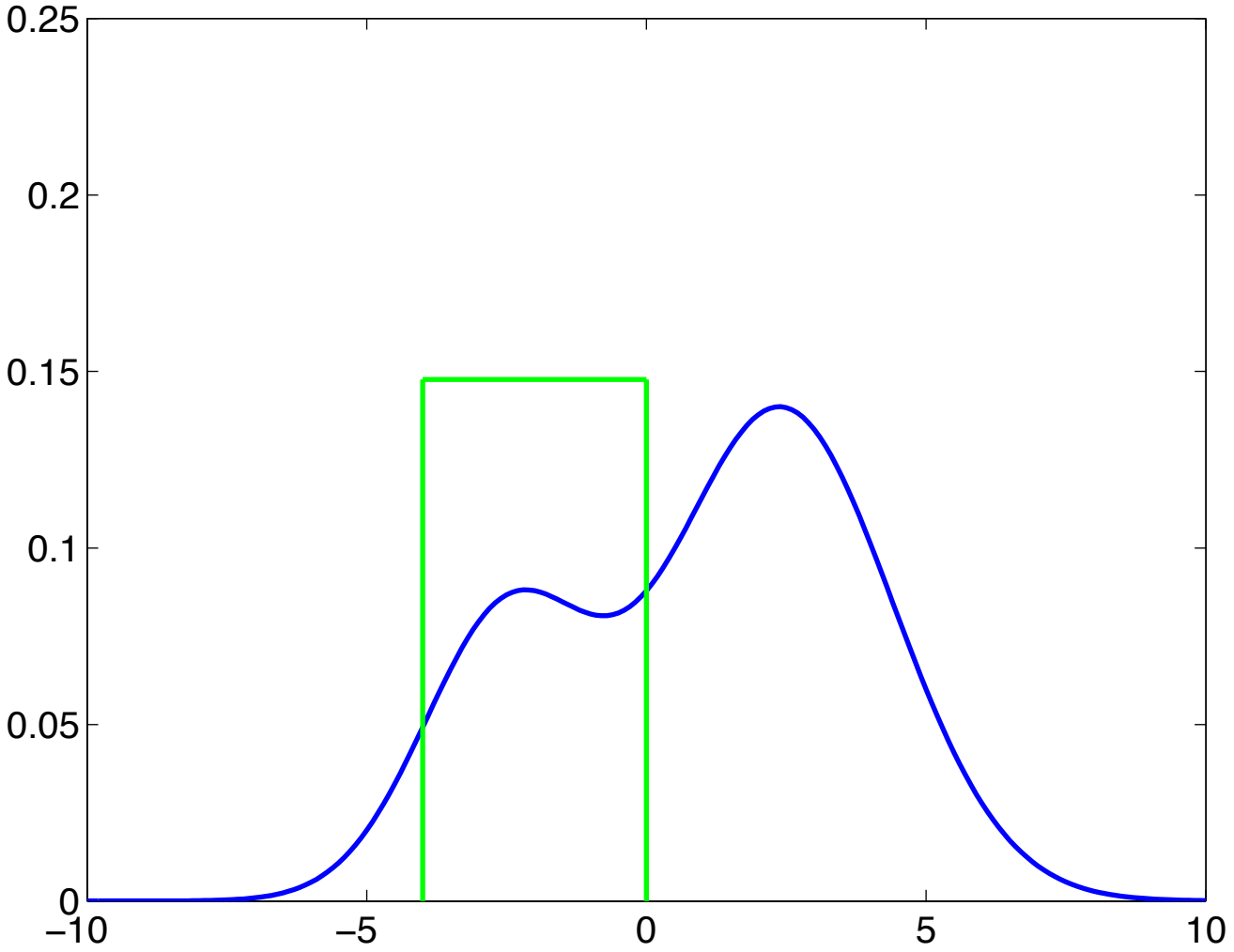
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



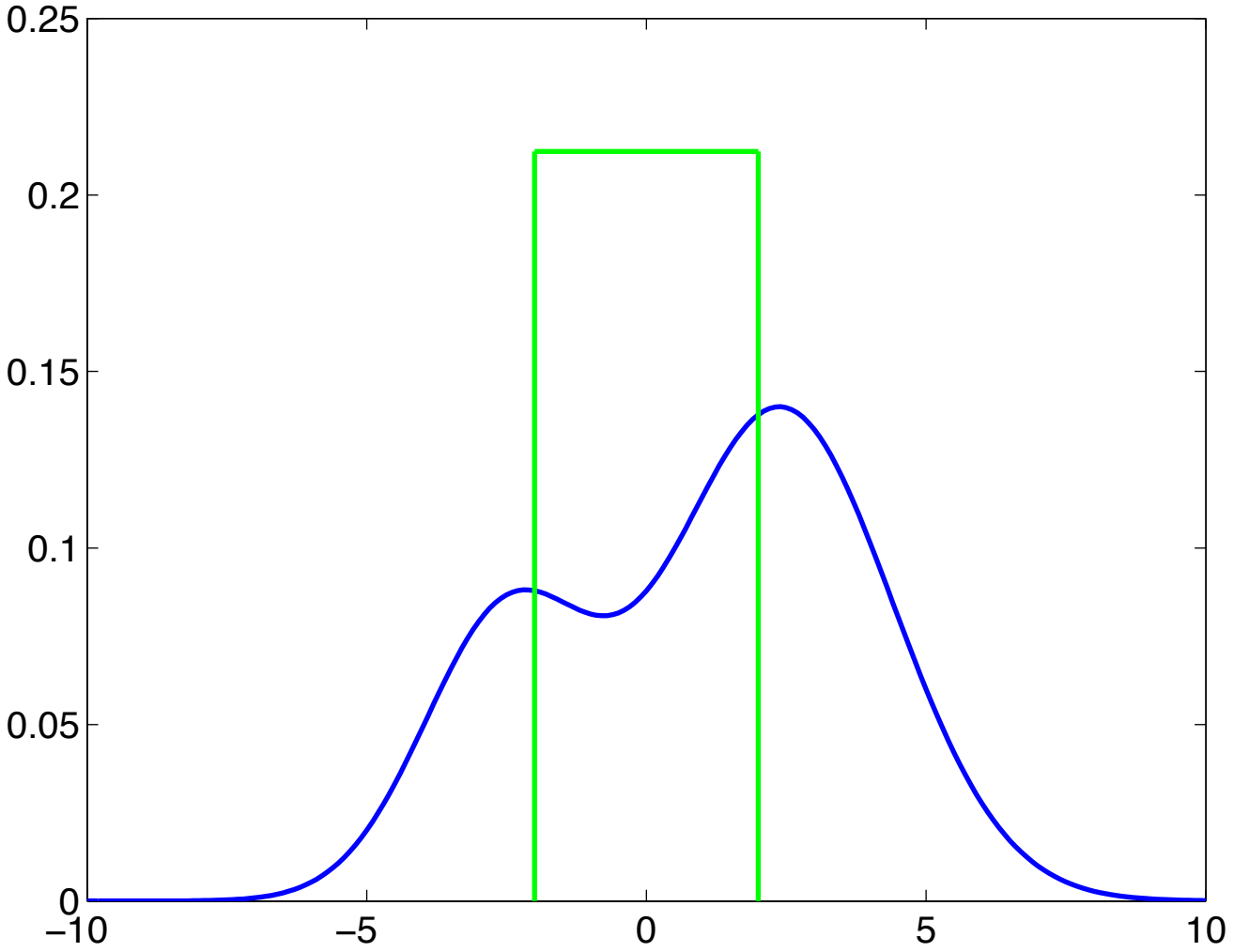
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



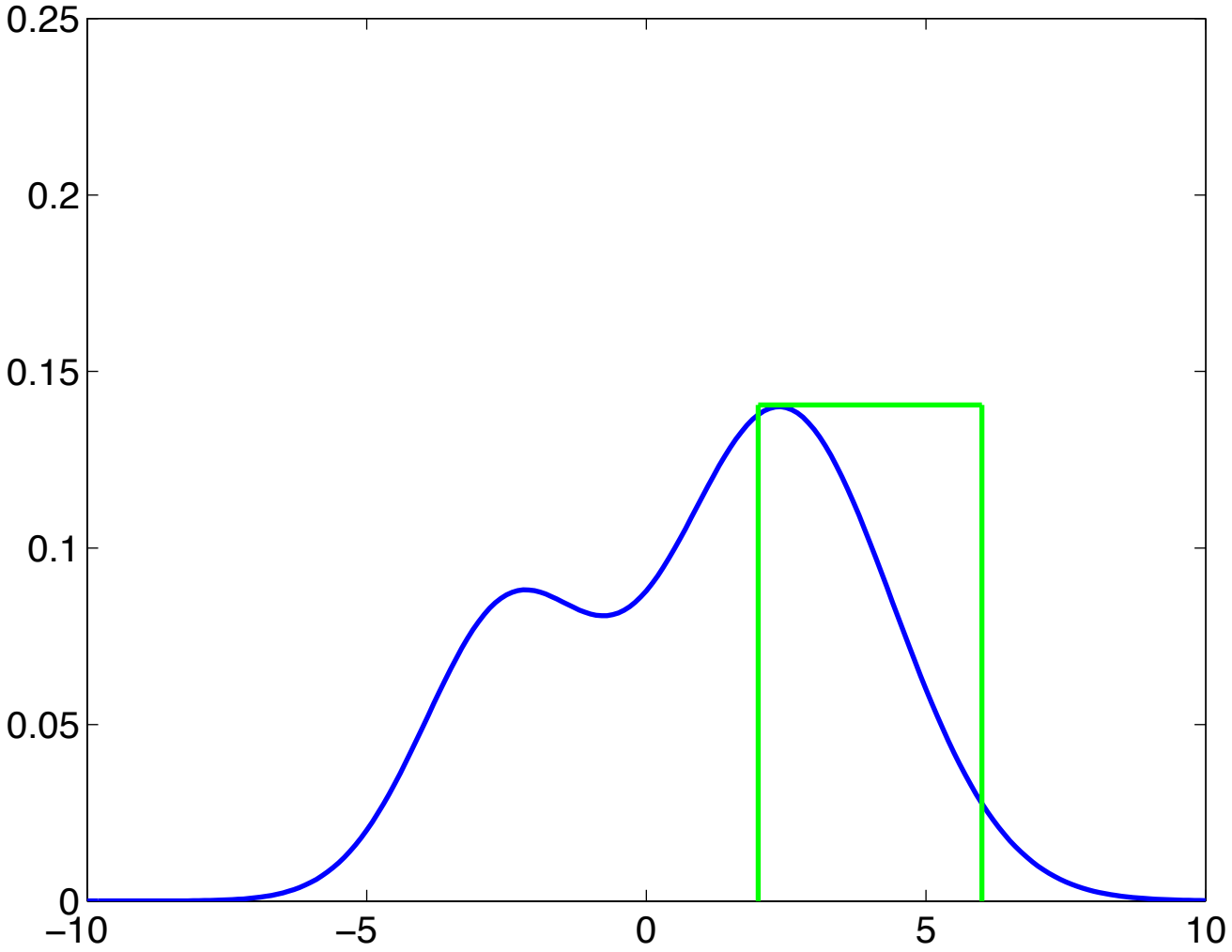
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



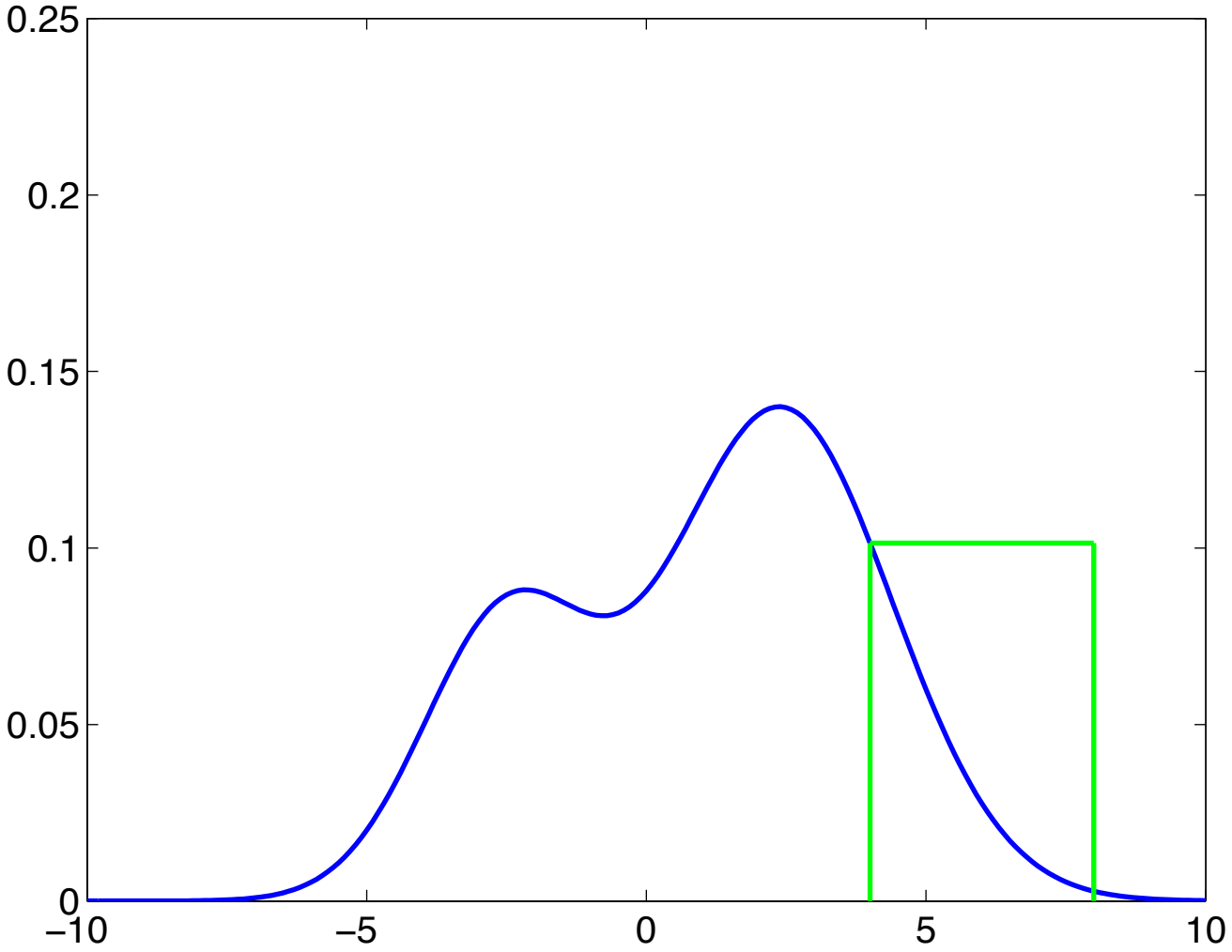
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



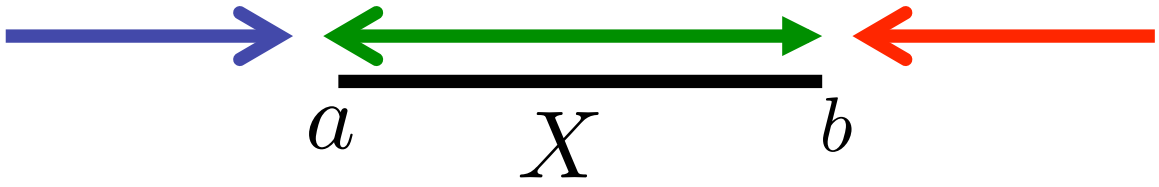
Bounding a mixture-of-gaussians

- Property: $\max_{x \in X} h(x) + g(x) \leq \max_{x \in X} h(x) + \max_{x \in X} g(x)$
- Function: $f(x) = \pi_1 N(x; \mu_1, \sigma_1) + \pi_2 N(x; \mu_2, \sigma_2)$

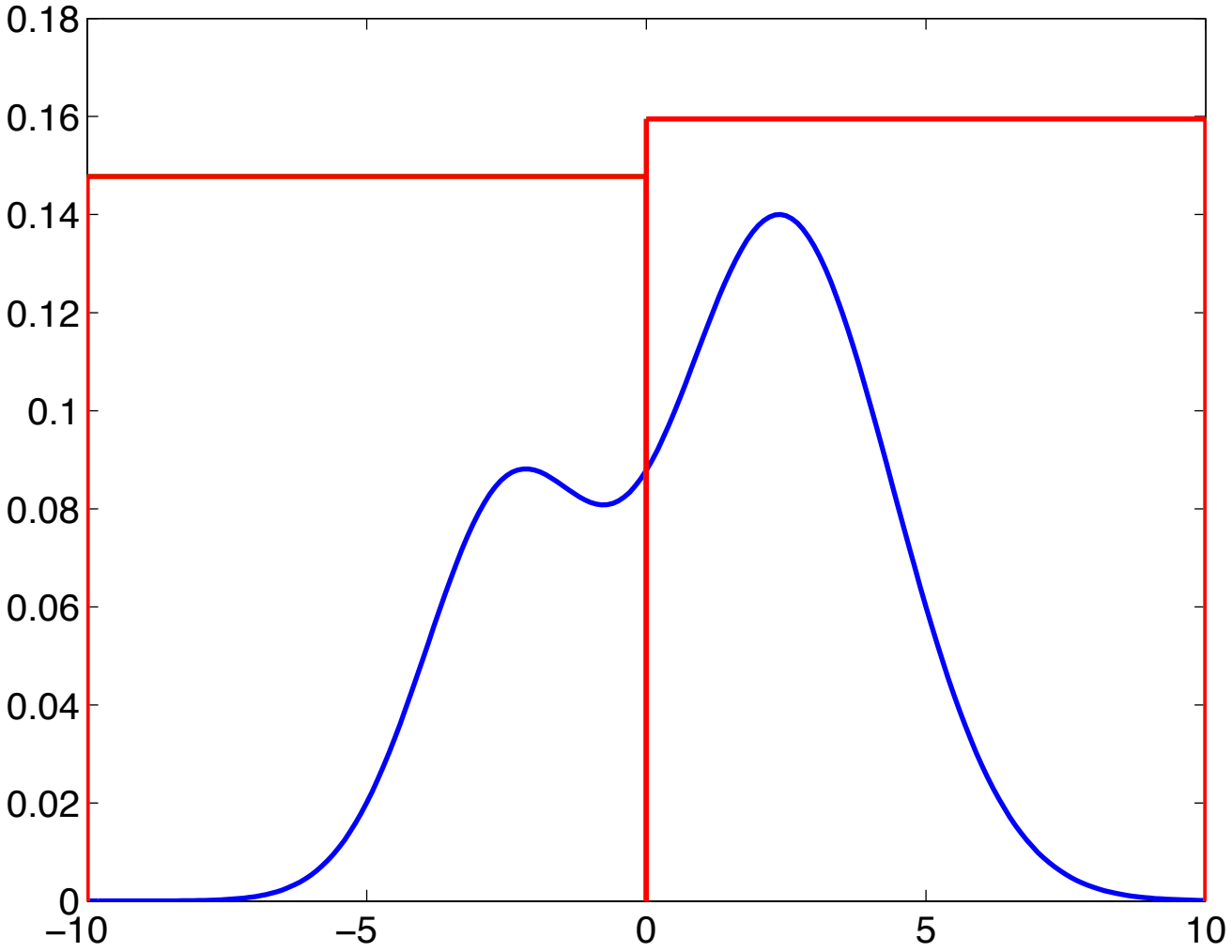
$$\max_{x \in X} f(x) \leq \max_{x \in X} [\pi_1 N(x; \mu_1, \sigma_1)] + \max_{x \in X} [\pi_2 N(x; \mu_2, \sigma_2)]$$

$$= \pi_1 N(d(X, \mu_1, \sigma_1); 0, 1) + \pi_2 N(d(X, \mu_2, \sigma_2); 0, 1)$$

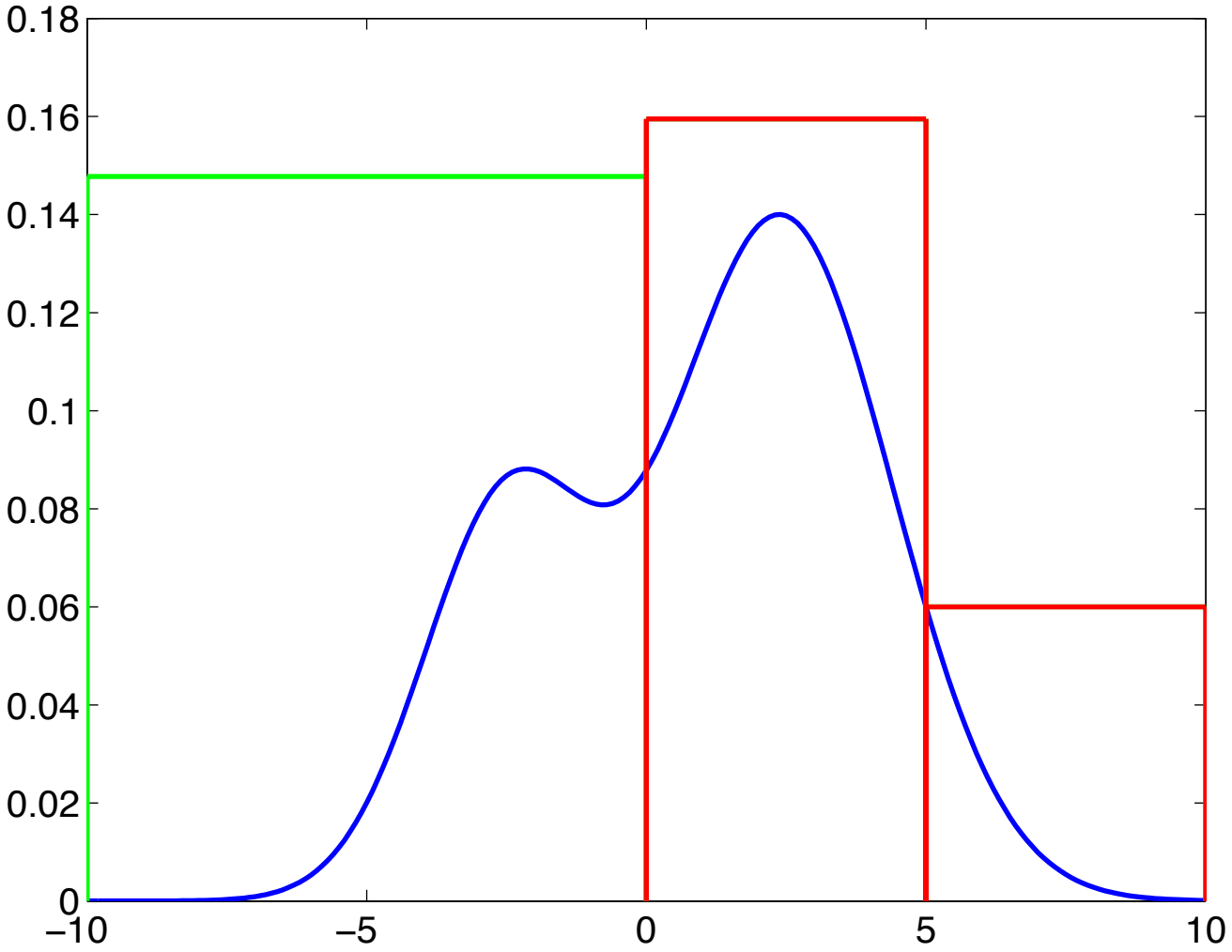
$$d(X, \mu, \sigma) = \begin{cases} 0 & a \leq \mu \leq b \\ \frac{1}{\sigma^2}(\mu - a)^2 & \mu \leq a \\ \frac{1}{\sigma^2}(\mu - b)^2 & b < \mu \end{cases}$$



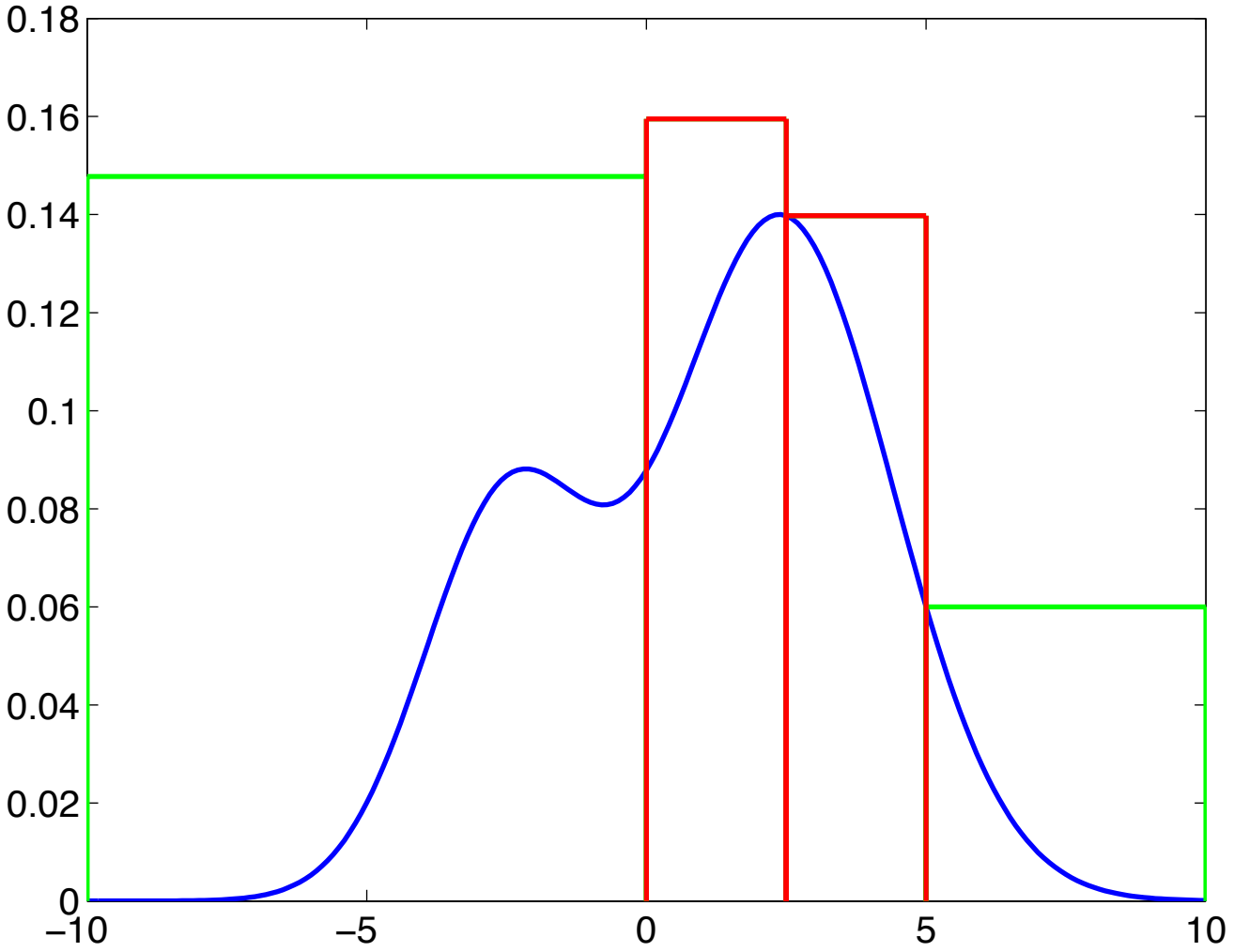
Branch-and-bound: all you need is bounds



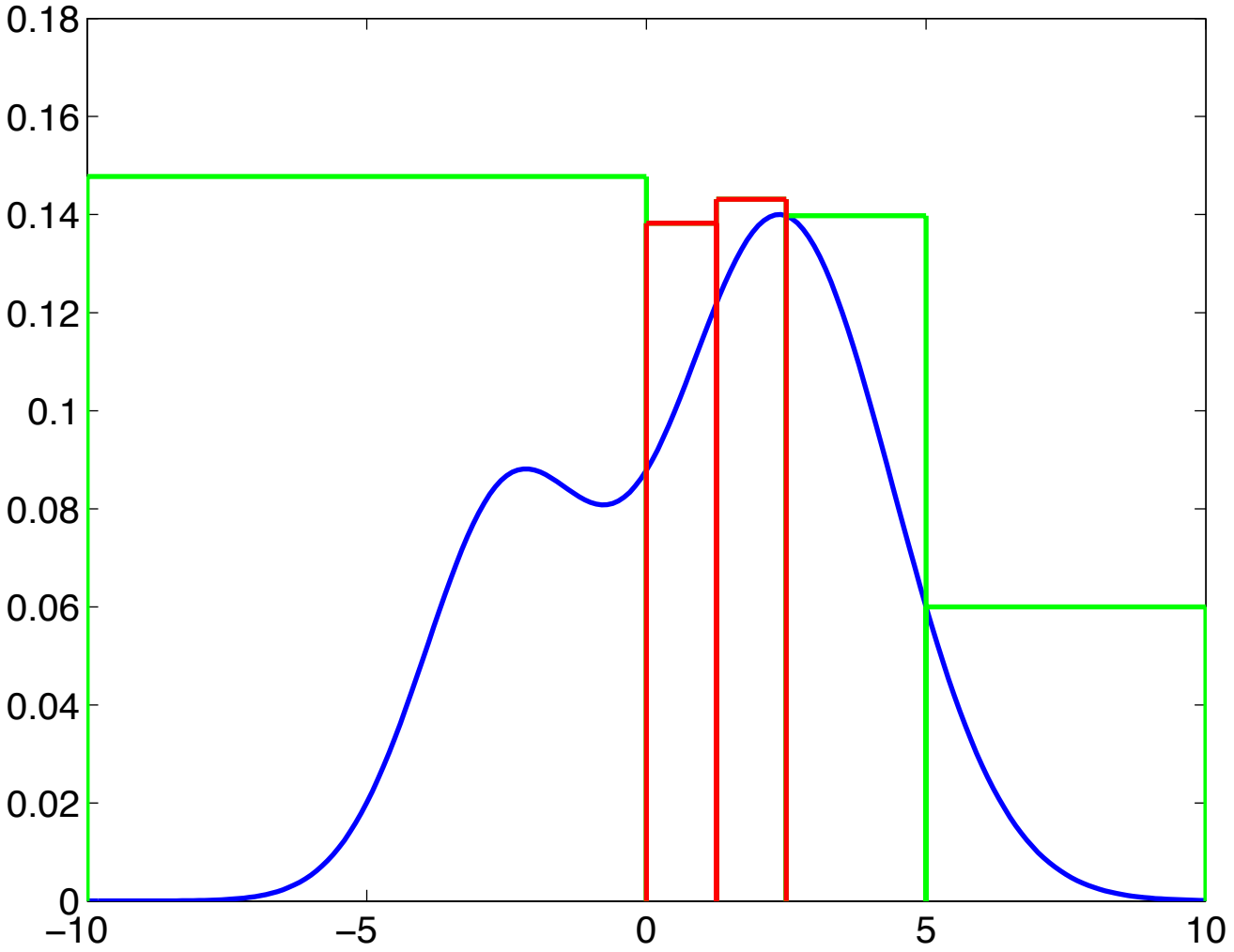
Branch-and-bound: all you need is bounds



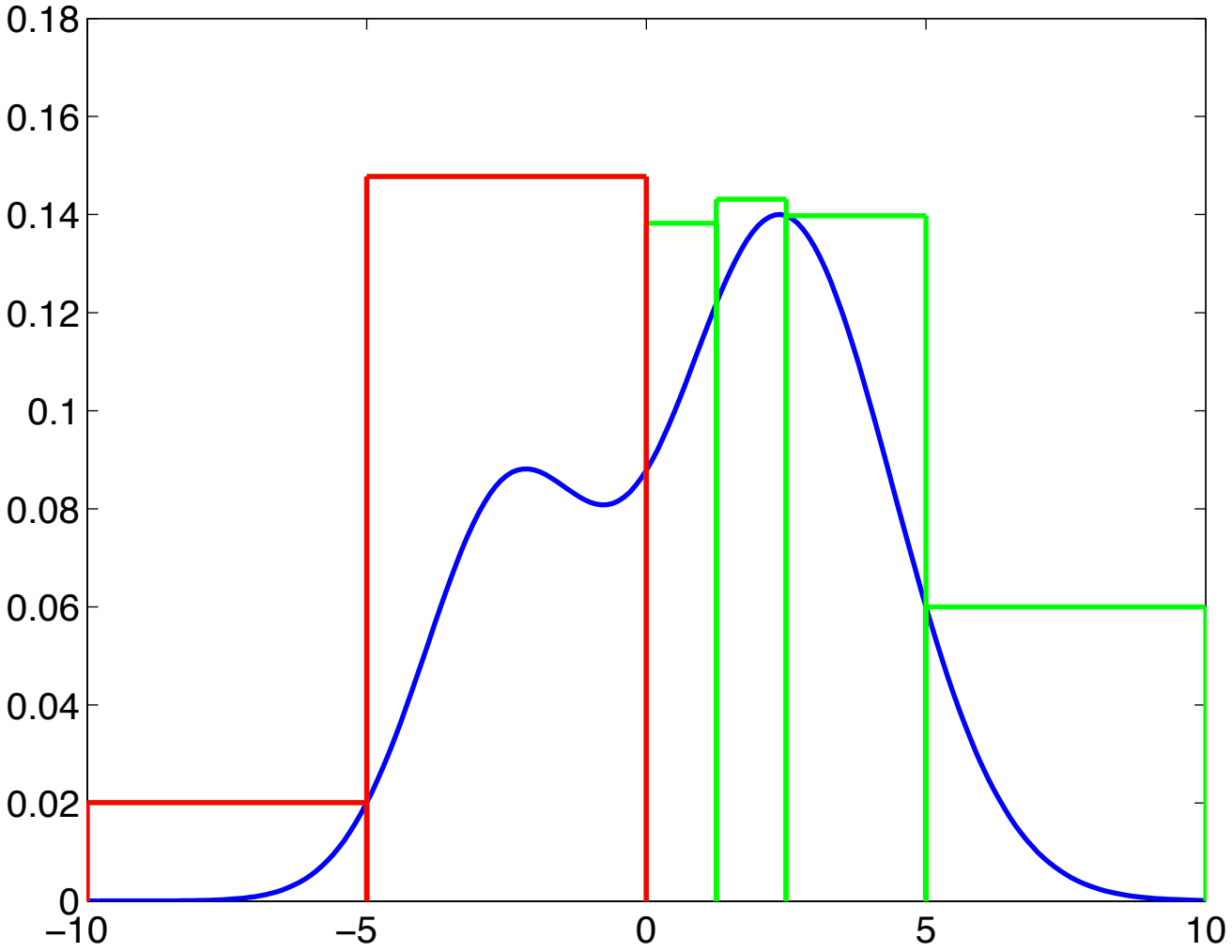
Branch-and-bound: all you need is bounds



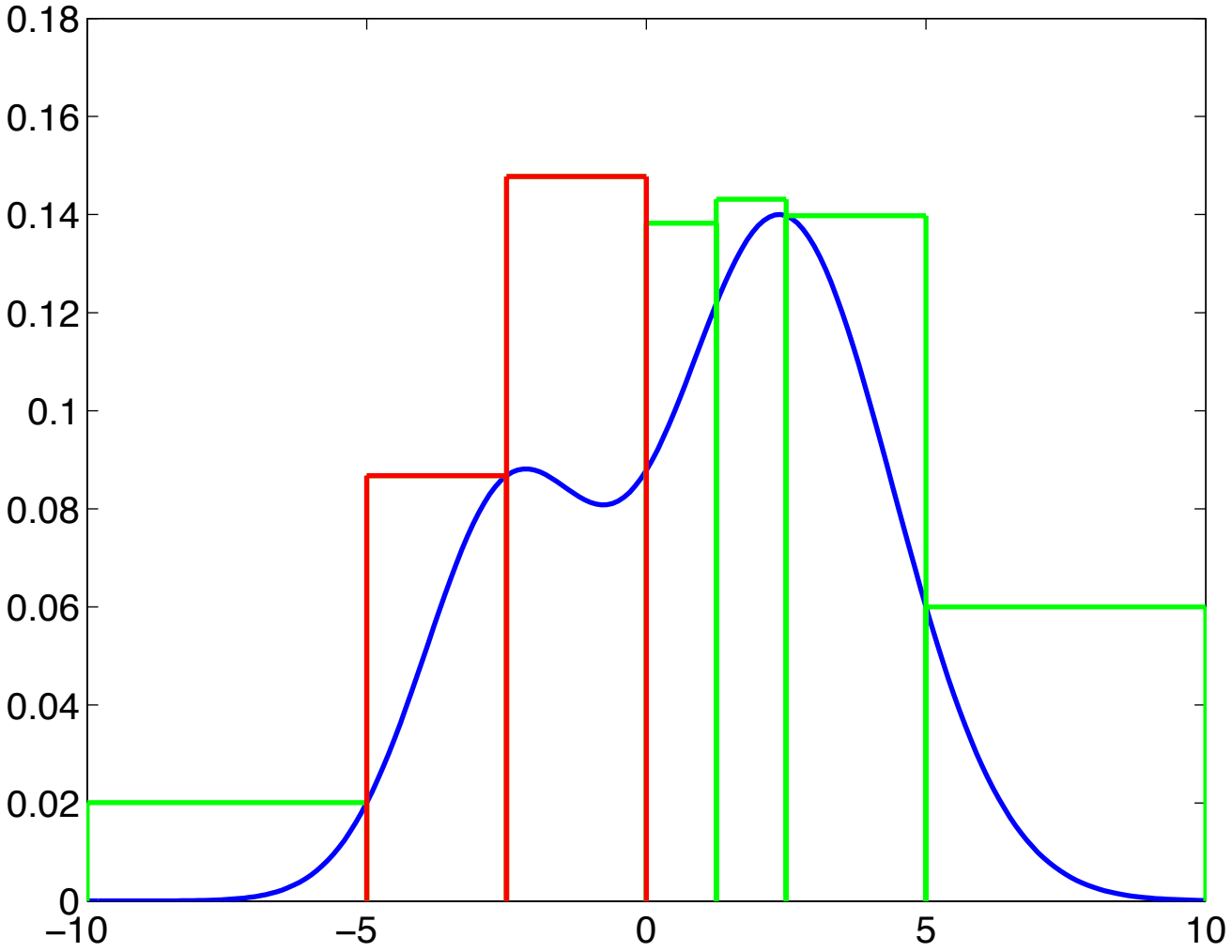
Branch-and-bound: all you need is bounds



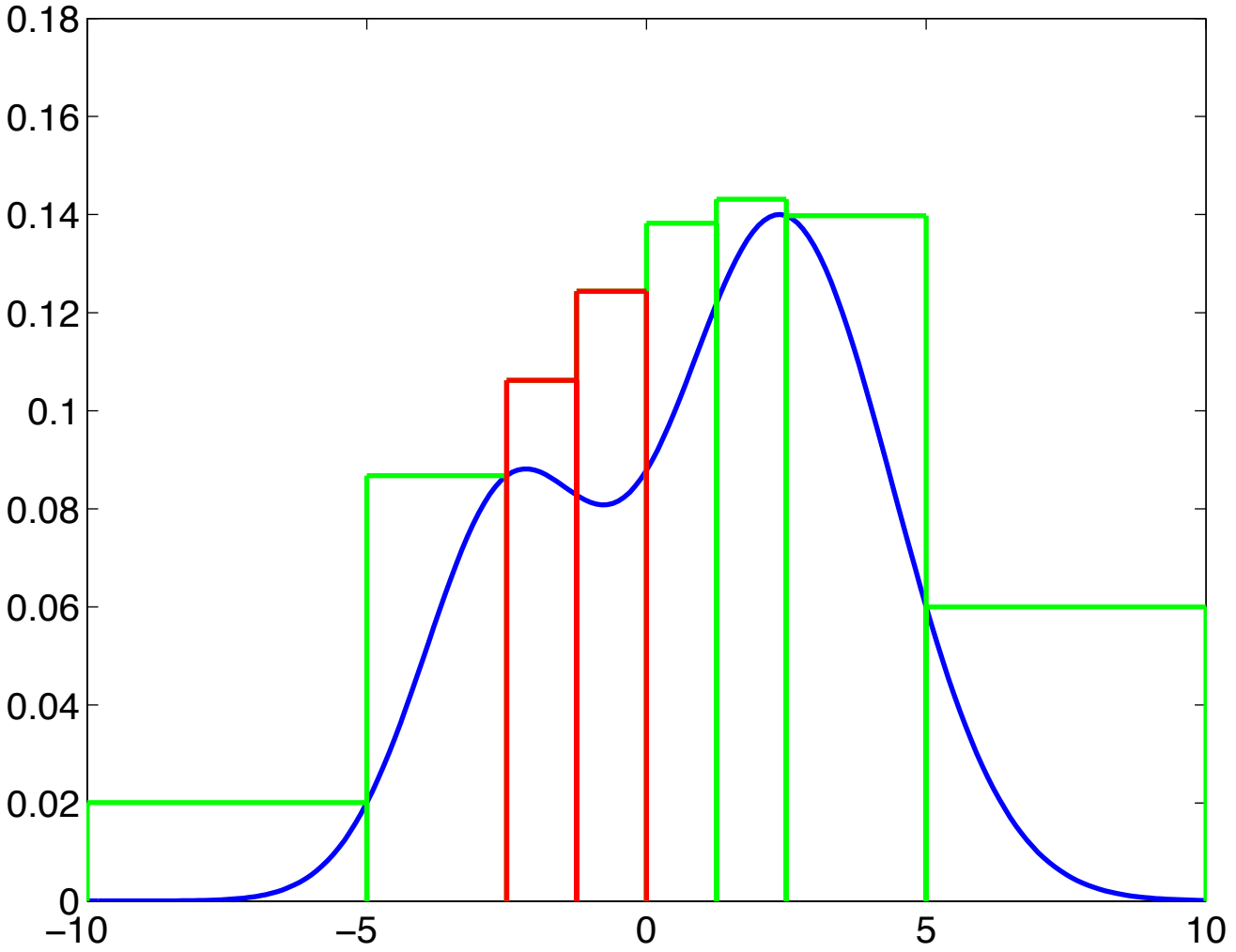
Branch-and-bound: all you need is bounds



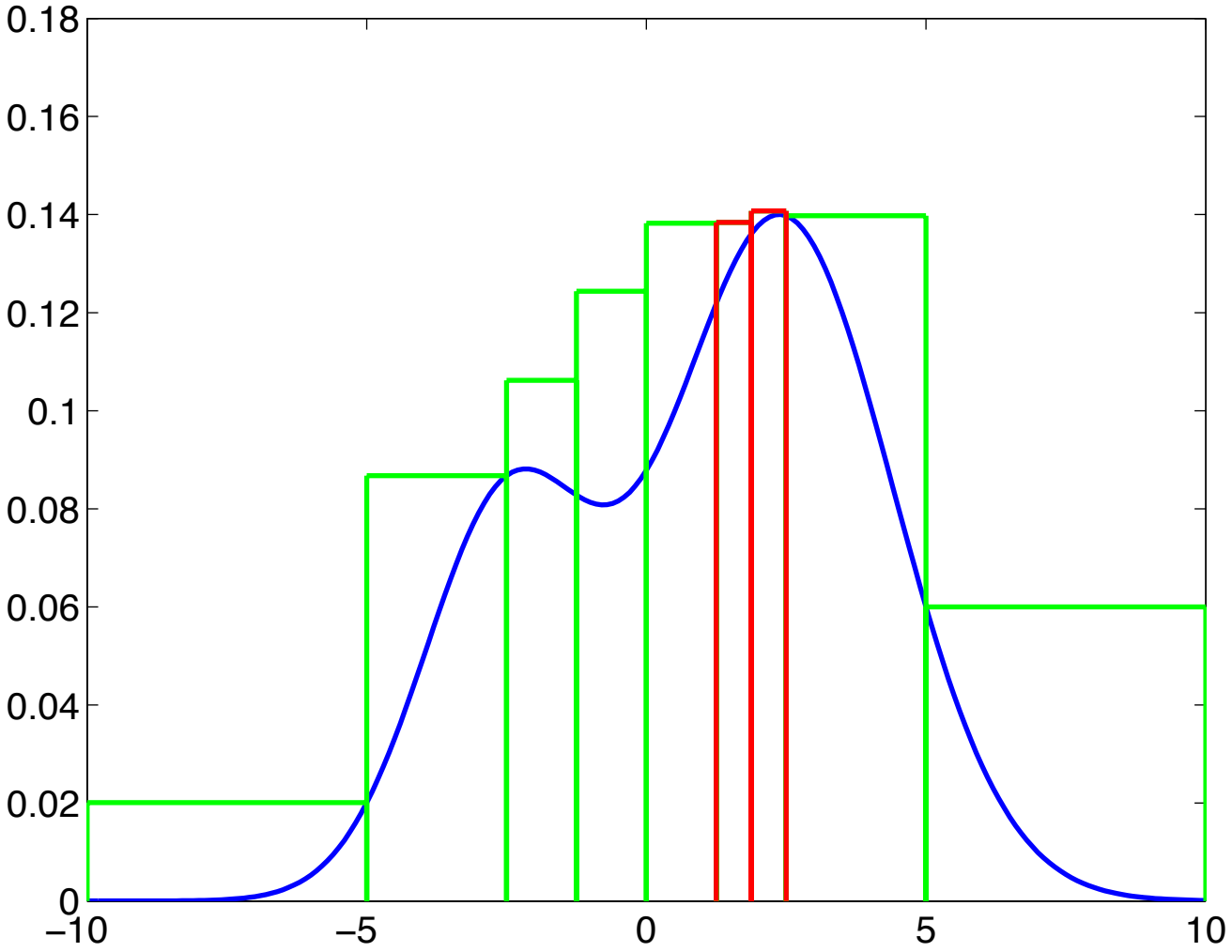
Branch-and-bound: all you need is bounds



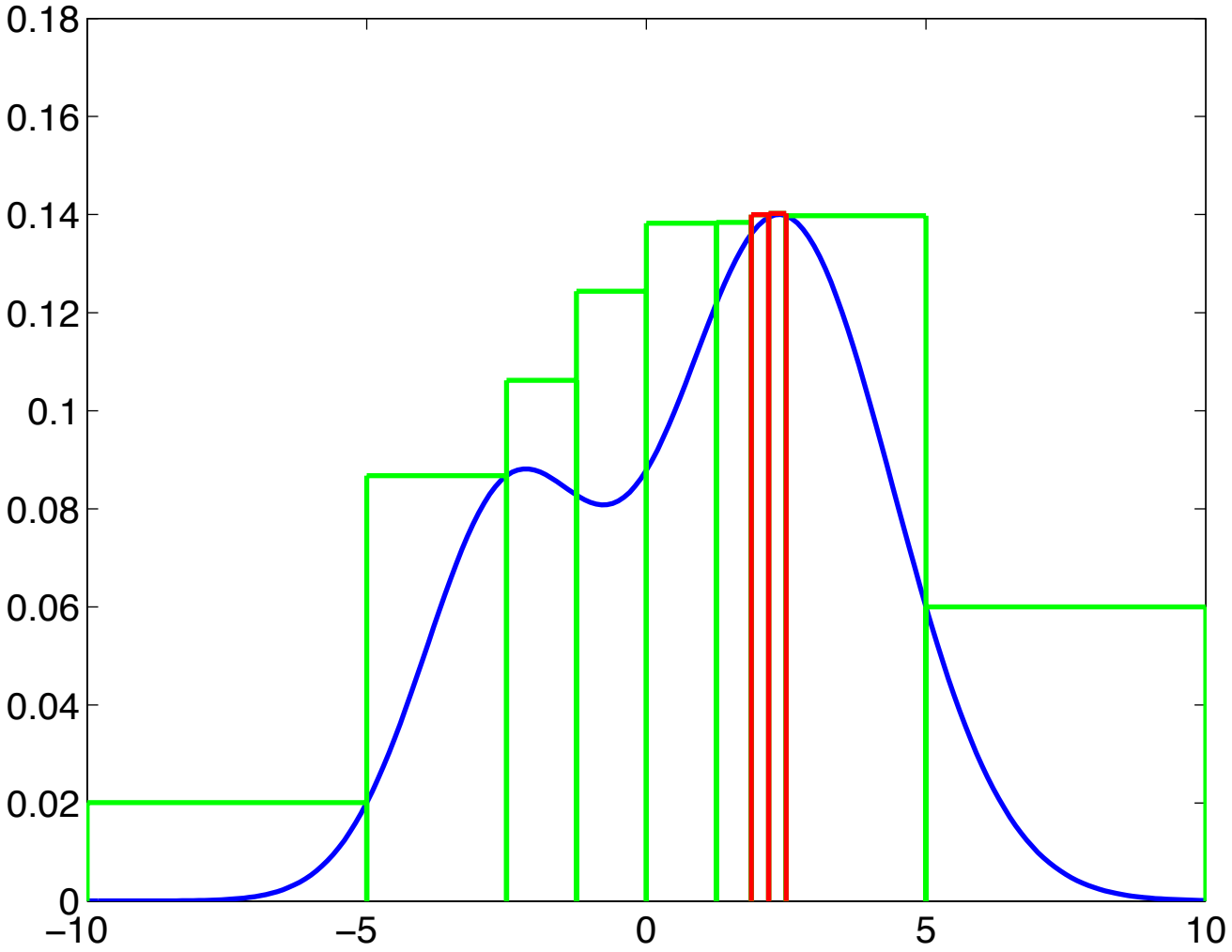
Branch-and-bound: all you need is bounds



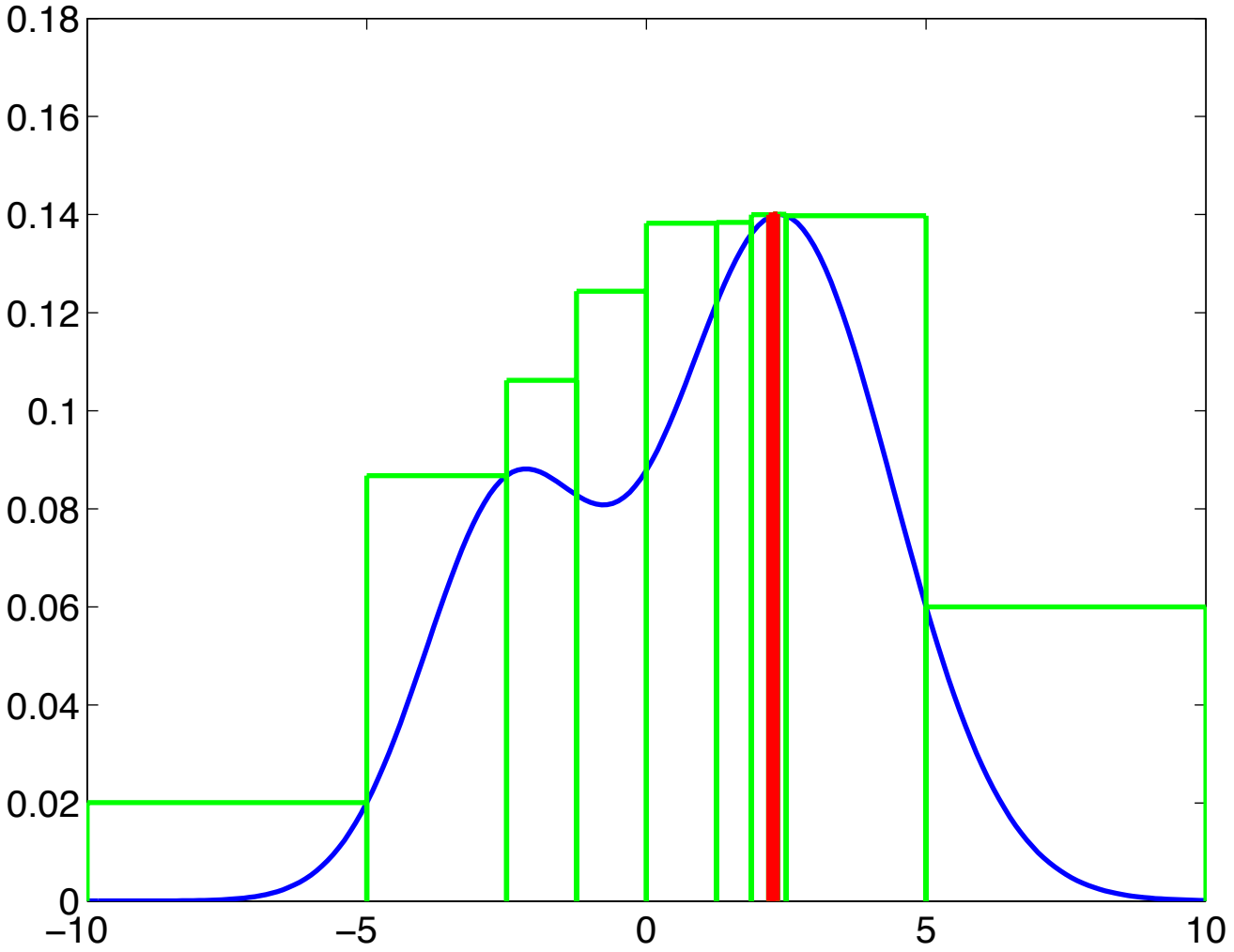
Branch-and-bound: all you need is bounds



Branch-and-bound: all you need is bounds

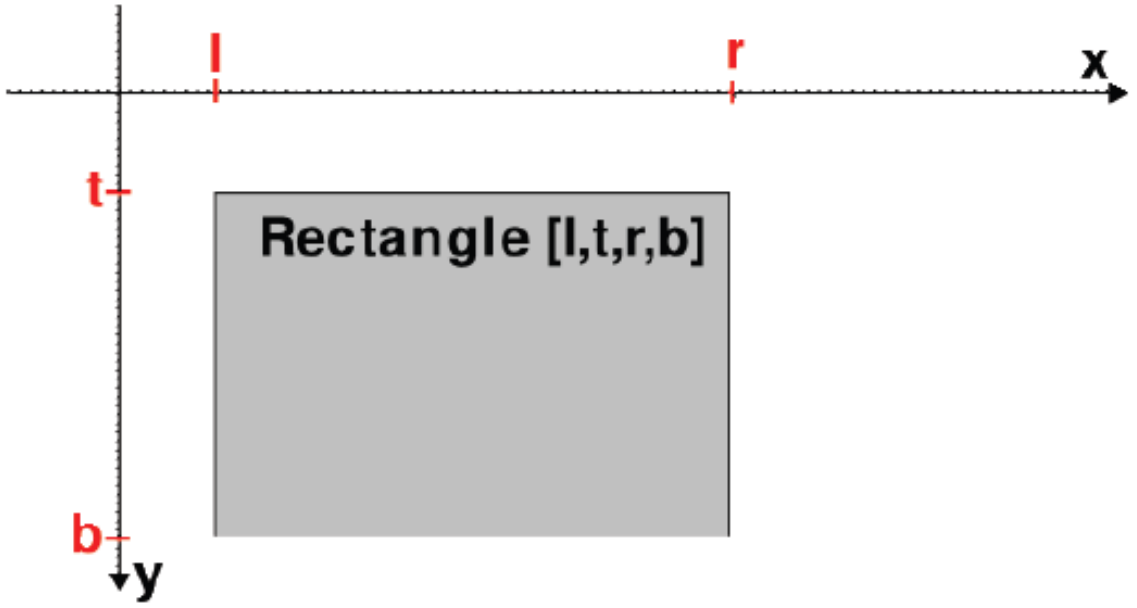


Branch-and-bound: all you need is bounds



Parameterization of solution

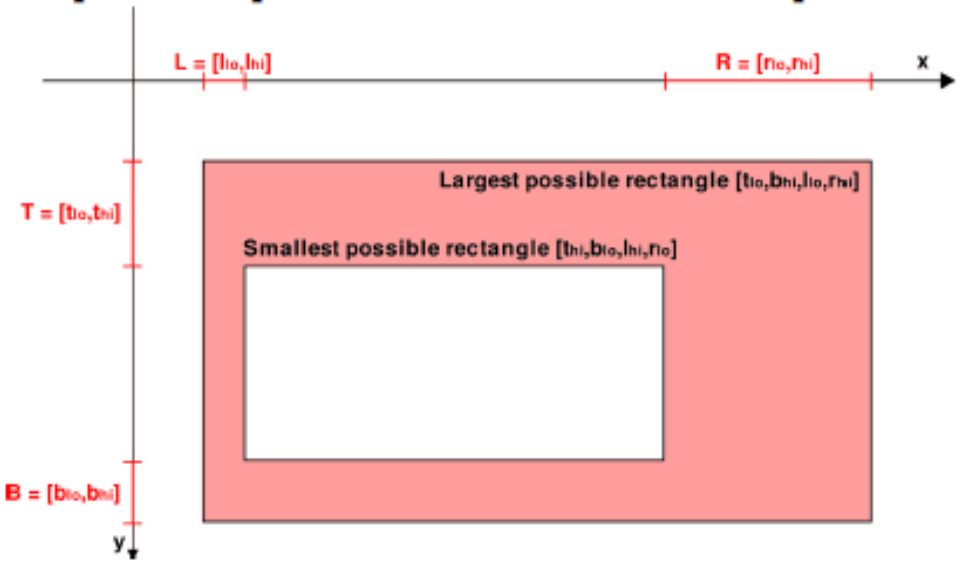
- low dimensional parametrization of bounding box (left, top, right, bottom)



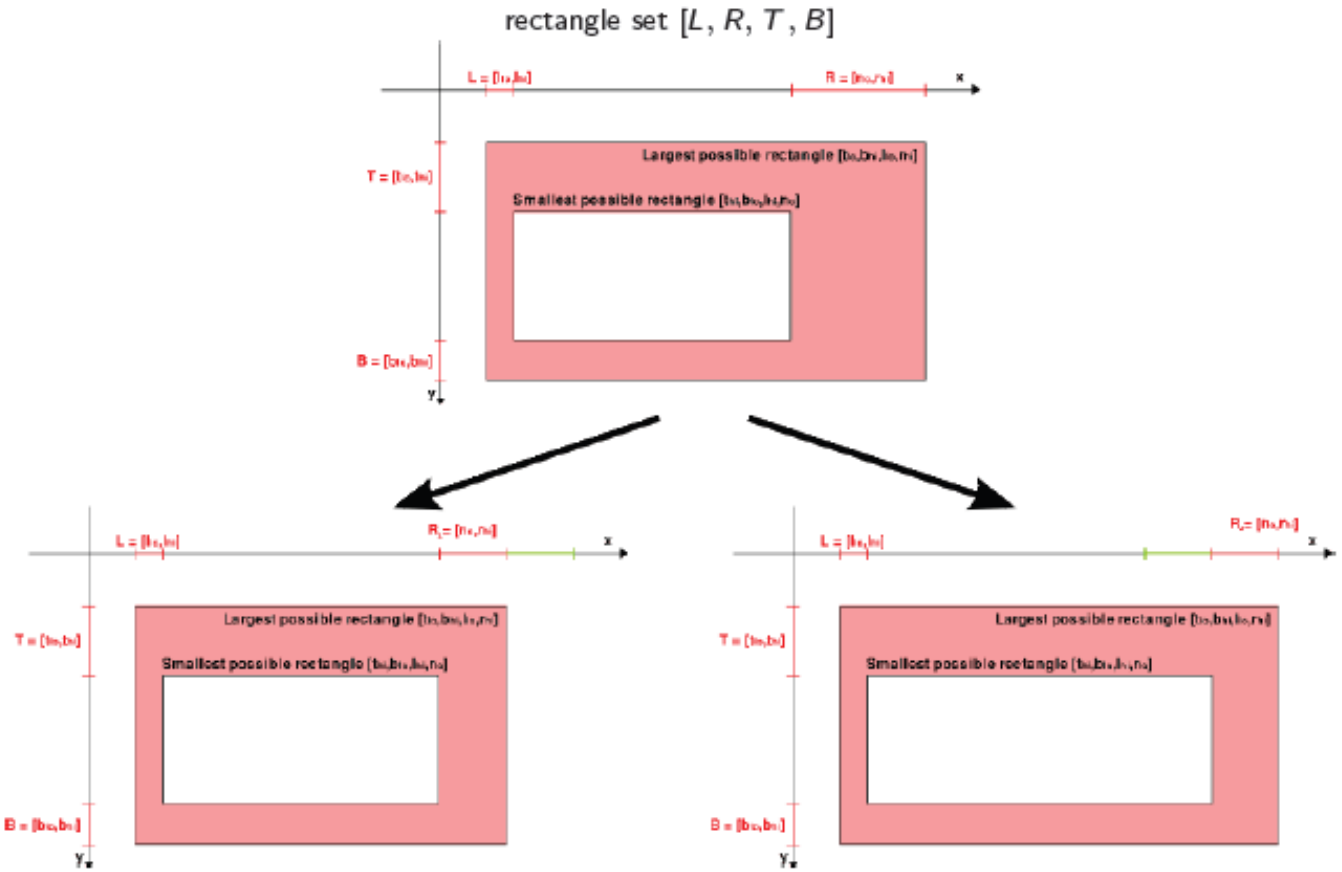
Parameterization of solution interval

- Instead of four numbers $[l, t, r, b]$, store four intervals $[L, T, R, B]$:

$$L = [l_{lo}, l_{hi}]$$
$$T = [t_{lo}, t_{hi}]$$
$$R = [r_{lo}, r_{hi}]$$
$$B = [b_{lo}, b_{hi}]$$



Branching of solution interval



$[L, R_1, T, B]$ with $R_1 := [r_{lo}, \lfloor \frac{r_{lo}+r_{hi}}{2} \rfloor]$

$[L, R_2, T, B]$ with $R_2 := [\lfloor \frac{r_{lo}+r_{hi}}{2} \rfloor + 1, r_{hi}]$

Bounding a solution interval

We have to construct $f^{upper} : \{ \text{ set of boxes } \} \rightarrow \mathbb{R}$ such that

i) $f^{upper}(\mathcal{B}) \geq \max_{B \in \mathcal{B}} f(B)$,

ii) $f^{upper}(\mathcal{B}) = f(B)$, if $\mathcal{B} = \{B\}$.

$$
\begin{aligned}
f(B) &= \sum_j \alpha_j \langle h^B, h^j \rangle && h^B \text{ the histogram of the box } B. \\
&= \sum_j \alpha_j \sum_k h^B_k h^j_k = \sum_k h^B_k w_k, && \text{for } w_k = \sum_j \alpha_j h^j_k \\
&= \sum_{x_i \in B} w_{c_i}, && c_i \text{ the cluster ID of the feature } x_i
\end{aligned}
$$

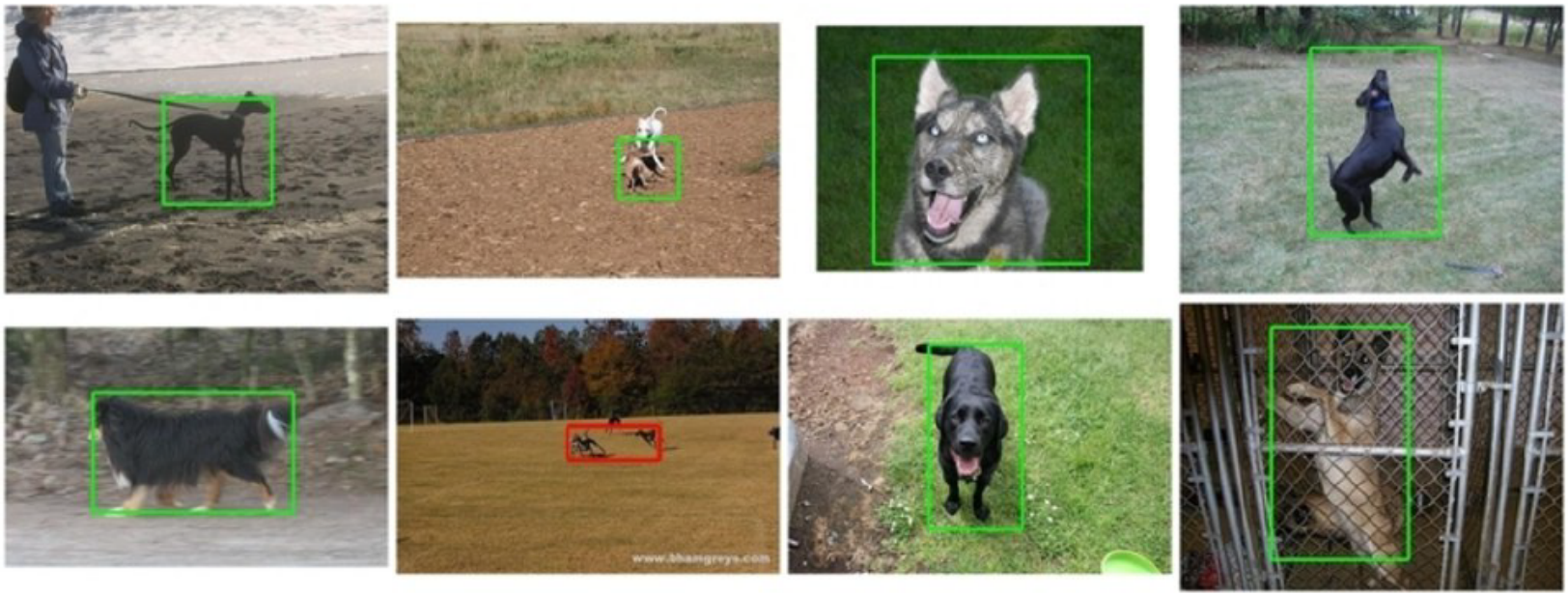
Set $f^+(B) = \sum_{x_i \in B} [w_i]_+$, $f^-(B) = \sum_{x_i \in B} [w_i]_-$.

Set $B^{max} :=$ *largest box in* $\mathcal{B}$, $B^{min} :=$ *smallest box in* $\mathcal{B}$.

$f^{upper}(\mathcal{B}) := f^+(B^{max}) + f^-(B^{min})$ fulfills *i*) and *ii*).

Evaluating $f^{upper}(\mathcal{B})$ has same complexity as $f(B)$!

- High localization quality: first place in 5 of 20 categories.
- High speed: $\approx 40ms$ per image (excl. feature extraction)



Example detections on VOC 2007 dog.

Branch-and-bound localization allows efficient extensions:

- Multi-Class Object Localization:

$$(B, C)^{\text{opt}} = \underset{B \in \mathcal{B},\, C \in \mathcal{C}}{\text{argmax}}\; f_I^C(B)$$

finds best object class $C \in \mathcal{C}$.

- Localized retrieval from image databases or videos

$$(I, B)^{\text{opt}} = \underset{B \in \mathcal{B},\, I \in \mathcal{D}}{\text{argmax}}\; f_I(B)$$

find best image I in database $\mathcal{D}$.

Runtime is *sublinear* in $|\mathcal{C}|$ and $|\mathcal{D}|$.

Nearest Neighbor query for *Red Wings* Logo in 10,000 video keyframes in "Ferris Buellers Day Off"

Appendix: Duality

- Constrained optimization problem:

$$\begin{aligned} \min_{w} \quad & f(w) \\ s.t. \quad & h_i(w) = 0, \quad i = 1 \dots l \\ & g_i(w) \leq 0, \quad i = 1 \dots m \end{aligned}$$

- Equivalent to unconstrained problem:

$$\min_{w} \quad f_{uc}(w) = f(w) + \sum_{i=1}^{l} I_0(h_i(w)) + \sum_{i=1}^{m} I_+(g_i(w))$$

$$I_0(x) = \begin{cases} 0, & x = 0 \\ \infty, & x \neq 0 \end{cases}, \qquad I_+(x) = \begin{cases} 0, & x \leq 0 \\ \infty, & x > 0 \end{cases}$$

- Soften constraint terms $I_0(x_i) \rightarrow \lambda_i x_i \qquad I_+(x_i) \rightarrow \mu_i x_i, \mu > 0$

Lagrangian

- Replace hard constraints with soft ones

$$\min_w \quad f_{uc}(w) = f(w) + \sum_{i=1}^{l} I_0(h_i(w)) + \sum_{i=1}^{m} I_+(g_i(w))$$

$$L(w, \lambda, \mu) = f(w) + \sum_{i=1}^{l} \lambda_i h_i(w) + \sum_{i=1}^{m} \mu_i g_i(w), \quad \mu_i > 0 \forall i$$

- Observe that

$$f_{uc}(w) = \max_{\lambda, \mu : \mu_i > 0} L(w, \lambda, \mu)$$

- At an optimum:

$$f(w^*) = \min_w \max_{\lambda, \mu : \mu_i > 0} L(w, \lambda, \mu)$$

Lagrange Dual Function

- Form $$\theta(\lambda, \mu) = \inf_{w} L(w, \lambda, \mu)$$

- θ : lower bound on optimal value of the original problem

$$\begin{aligned} L(w^*, \lambda, \mu) &= f(w^*) + \sum_{i=1}^{l} \lambda_i h_i(w^*) + \sum_{i=1}^{m} \mu_i g_i(w^*) = \\ &\overset{w^*:feasible}{=} f(w^*) + \sum_{i=1}^{l} \lambda_i 0 + \sum_{i=1}^{m} \mu_i \underbrace{g_i(w^*)}_{<0} = \\ &\overset{\mu_i \geq 0}{\leq} f(w^*) \end{aligned}$$

- Therefore: $$\theta(\lambda, \mu) = \inf_{w} L(w, \lambda, \mu) \leq L(w^*, \lambda, \mu) \leq f(w^*)$$

Dual Problem

- Maximize the lower bound on the cost of the primal

$$\max_{\lambda,\mu} \quad \theta(\lambda,\mu)$$
$$s.t. \quad \mu_i > 0 \quad \forall i$$

- In general:

$$\begin{aligned} d^* &= \max_{\lambda,\mu} \theta(\lambda,\mu) \\ &= \max_{\lambda,\mu:\mu_i>0} \min_{w} L(w,\lambda,\mu) \\ &\leq \min_{w} \max_{\lambda,\mu:\mu_i>0} L(w,\lambda,\mu) \\ &= \min_{w} f_{uc}(w) = p^* \end{aligned}$$

- For convex cost and convex constraints (SVM case): $d^* = p^*$

Complementary Slackness

- Assume $d^* = p^*$
- There exists a feasible solution w^*, λ^*, μ^* to the primal and dual problems, such that $f(w^*) = \theta(\lambda^*, \mu^*)$
- We will have

$$\begin{aligned} f(w^*) &= \theta(\lambda^*, \mu^*) \\ &= \inf_w f(w) + \sum_{i=1}^{l} \lambda_i^* h_i(w) + \sum_{i=1}^{m} \mu_i^* f_i(w) \\ &\leq f(w^*) + \sum_{i=1}^{M} \lambda_i^* h_i(w^*) + \sum_{i=1}^{m} \mu_i^* f_i(w^*) \\ &\leq f(w^*) \end{aligned}$$

- This means $\mu_i^* f_i(w^*) = 0, \quad \forall i$

Karush-Kuhn Tucker (KKT) Conditions

- Solution of the primal problem:
  - minimum of the Lagrangan w.r.t. the primal variables
  - therefore $$\nabla f(w^*) + \sum_{i=1}^{l} \lambda_i \nabla h_i(w^*) + \sum_{i=1}^{m} \nabla f_i(w^*) = 0$$

- Putting all constraints together: KKT conditions

$$\begin{aligned} h_i(w^*) &= 0 \\ f_i(w^*) &\leq 0 \\ \mu_i f_i(w^*) &= 0 \\ \mu_i &\geq 0 \\ \nabla f(w^*) + \sum_{i=1}^{l} \lambda_i \nabla h_i(w^*) + \sum_{i=1}^{m} \nabla f_i(w^*) &= 0 \end{aligned}$$