# Efficient Multiple Kernel Learning

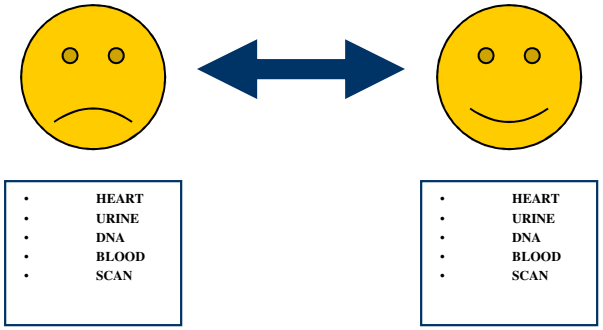
Lei Tang

# Outline

- What is Kernel Learning?
- What's the problem with existing formulation?
- Two new formulations for large scale kernel selection
  - SIL formulation (Cutting Planes)
  - More efficient MKL (Steepest Decent)

# Linear algorithm: binary classification

- **Data:** $\{(x_i, y_i)\}_{i=1...n}$
  - x $R^d$ = feature vector
  - y {-1,+1} = label




- **Question**: design a classification rule

  **y = f(x)**

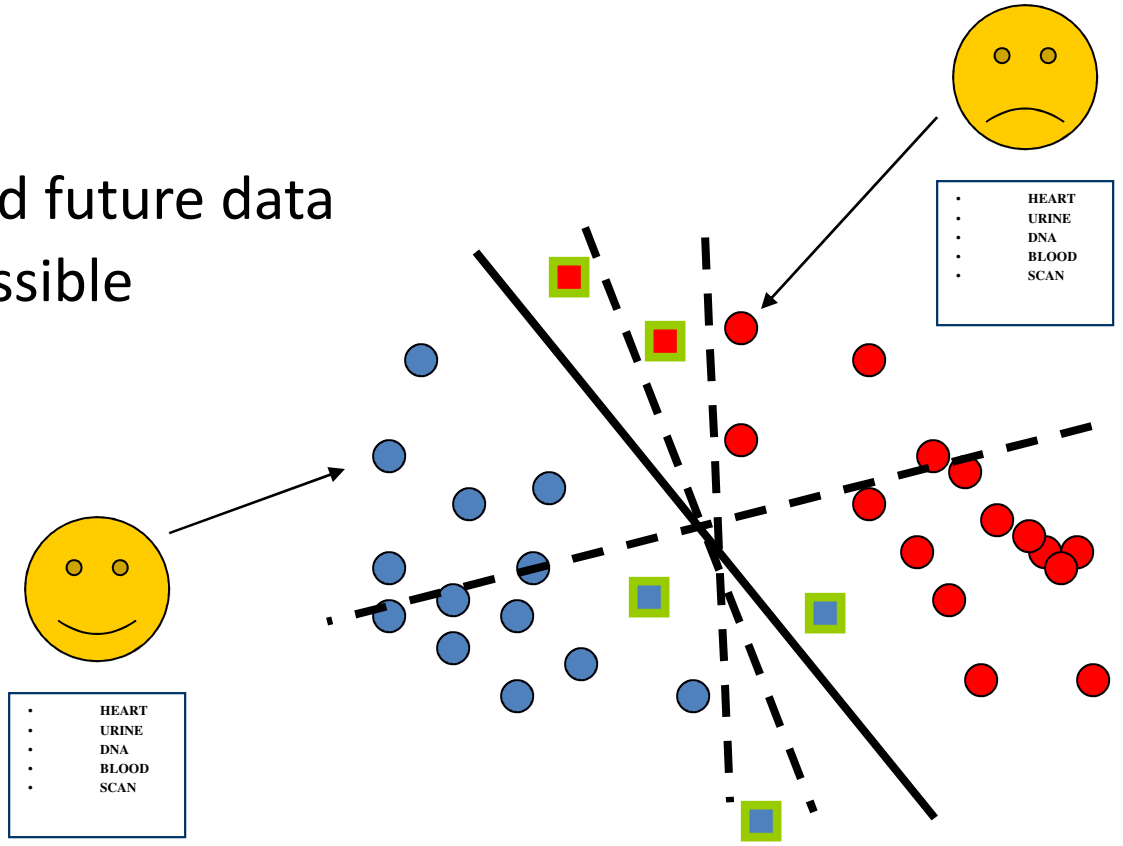
  such that, given a new x, this predicts y with minimal probability of error

# Linear algorithm: binary classification

- Find **good hyperplane**

  **(w,b)** $\mathbf{R^{d+1}}$

  that classifies this and future data points as good as possible

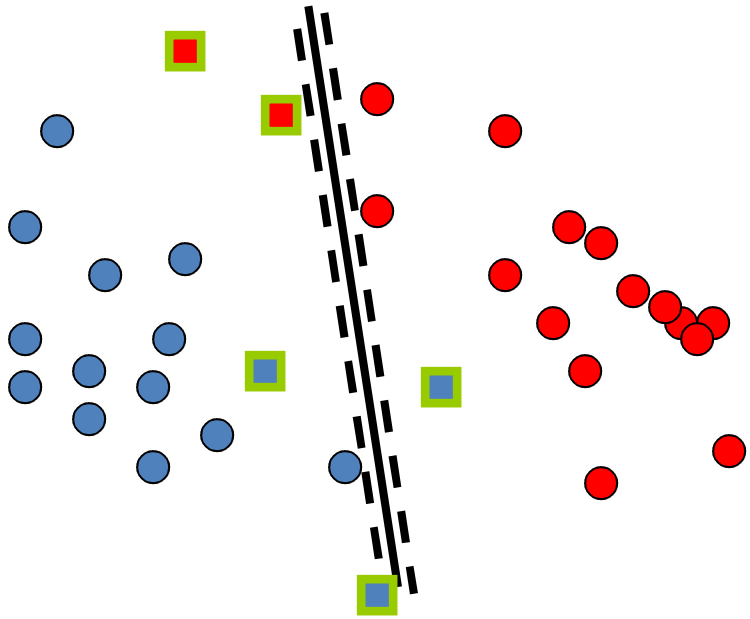



**Classification Rule:**

$$y = f(x) = \text{sign}(w^T x + b)$$

# Linear algorithm: binary classification

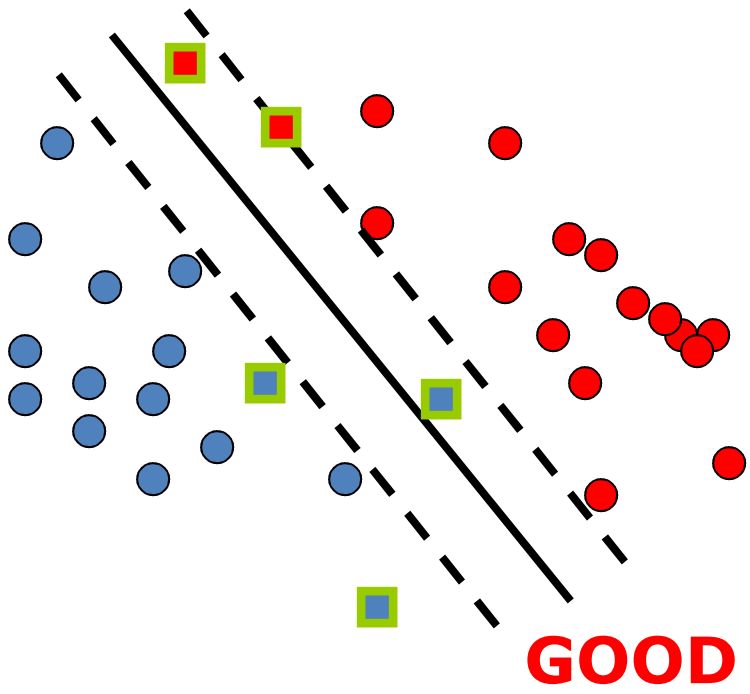
- **Intuition** (Vapnik, 1965) if linearly separable:
  - Separate the data
  - Place hyerplane "far" from the data: **large margin**



**BAD**

# Linear algorithm: binary classification

- **Intuition** (Vapnik, 1965) if linearly separable:
  - Separate the data
  - Place hyerplane “far” from the data: **large margin**

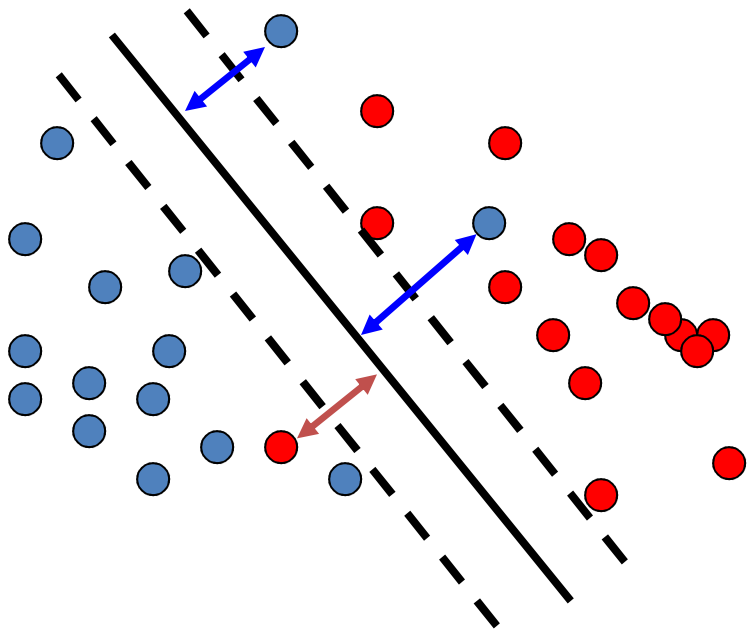



→ **Maximal Margin Classifier**

# Linear algorithm: binary classification

If **not linearly separable**:

- **Allow** some **errors**
- Still, try to place hyerplane “far” from each class



# SVM: Primal & Dual

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

Primal:

$$\min_{w,b} \quad \frac{1}{2} \| w \|_2^2 + C \sum_{i=1}^{N} \xi_i$$

$$\text{subject to} \quad y_i (w \cdot x_i + b) \geq 1 - \xi_i \qquad \forall i$$

$$\xi_i \geq 0$$

Dual:

$$\max_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\text{subject to} \quad \sum_i \alpha_i y_i = 0, \quad C \geq \alpha_i \geq 0$$

# Linear algorithm: binary classification

- **Training = convex optimization problem (QP):**

$$\max_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

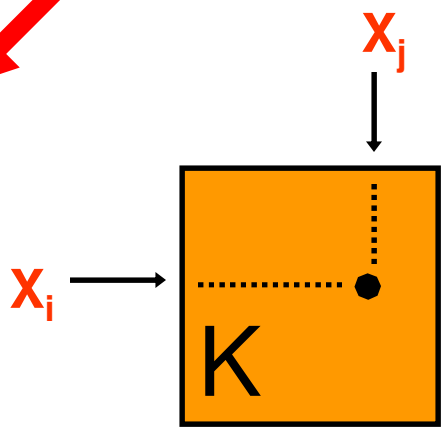
$$\text{subject to} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

implicit embedding

$$K_{ij} = k(x_i, x_j)$$

$$\mathbf{max}_{\alpha} \quad e^T \alpha - \frac{1}{2}\alpha^T D_y K D_y \alpha$$

$$\mathbf{subject\ to} \quad y^T \alpha = 0, \quad \alpha \geq 0$$



# Kernel algorithm: Support Vector Machine (SVM)

- **Training = convex optimization problem (QP)**:

$$\max_{\alpha} \quad e^T\alpha - \frac{1}{2}\alpha^T D_y K D_y \alpha$$

$$\text{subject to} \quad y^T\alpha = 0, \quad C \geq \alpha \geq 0$$

- **Classification rule: classify new** data point **x**:

$$f(x) = sign(w^T x + b) = sign\left(\sum_{i=1}^{n_{SV}} \alpha_i y_i x_i^T x + b\right)$$

**Kernel algorithm !**

# Support Vector Machines (SVM)

- Hand-writing recognition (e.g., USPS)
- Computational biology (e.g., micro-array data)
- Text classification
- Face detection
- Face expression recognition
- Time series prediction (regression)
- Drug discovery (novelty detection)

# Different Kernels

- Various kinds of Kernel
  - Linear kernel

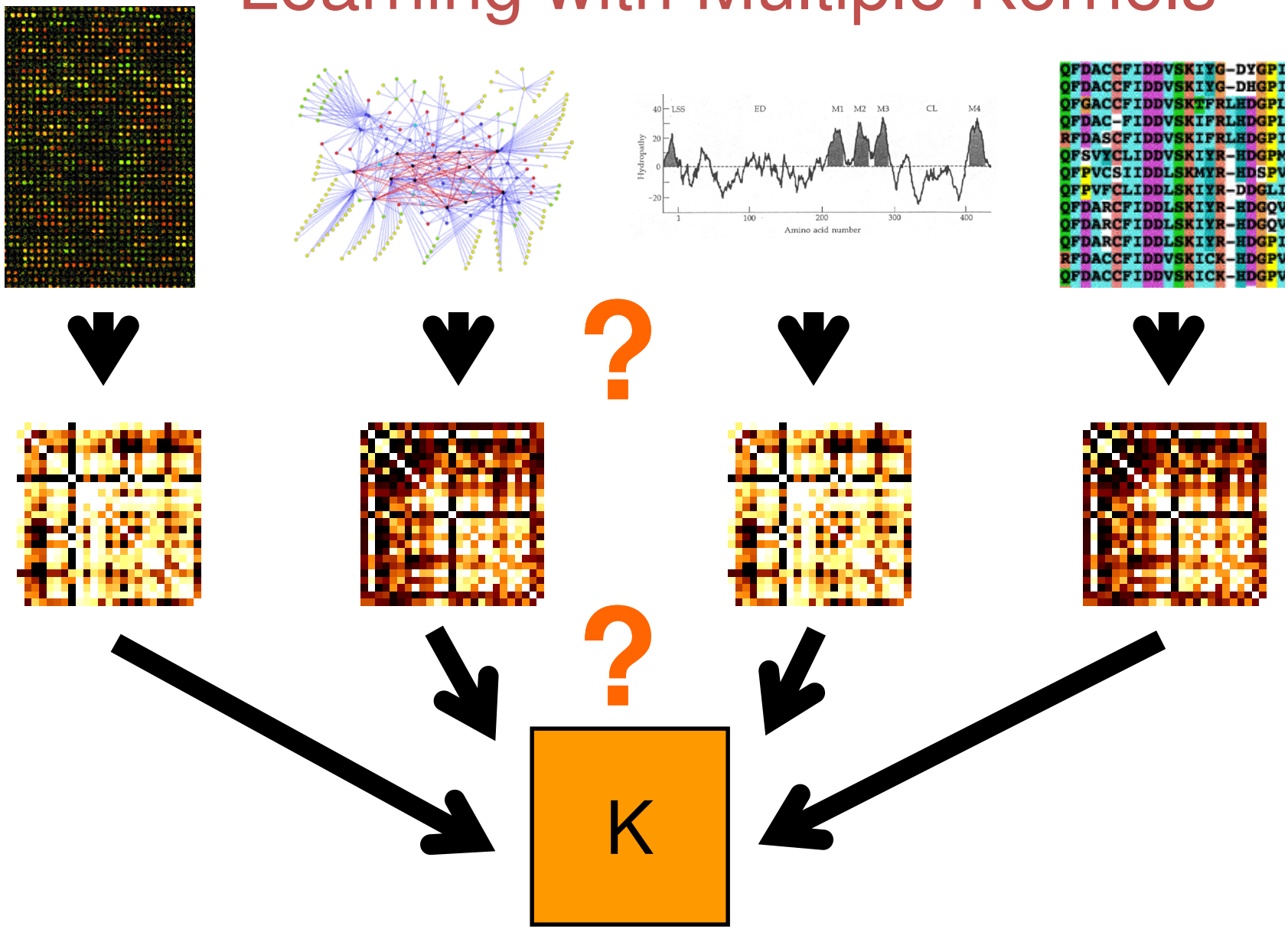
$$K(X,Y) = \frac{\sum_i X_i Y_i}{\sqrt{\sum_i X_i X_i}\sqrt{\sum_i Y_i Y_i}}$$

  - Gaussian kernel

$$K(X,Y) = \exp\left(\frac{-\|X-Y\|^2}{2\sigma^2}\right)$$

  - Diffusion kernel
  - String Kernel
  - ……

# Learning with Multiple Kernels



# Learning the optimal Kernel

***Overview of SVM with single kernel :***

Binary classification of data $x_i \in \mathbb{R}^d$, labels $y_i \in \{-1, 1\}$ $(i = 1, \ldots, n)$ using $f(x) = w^\top \phi(x) + b$.

Primal problem

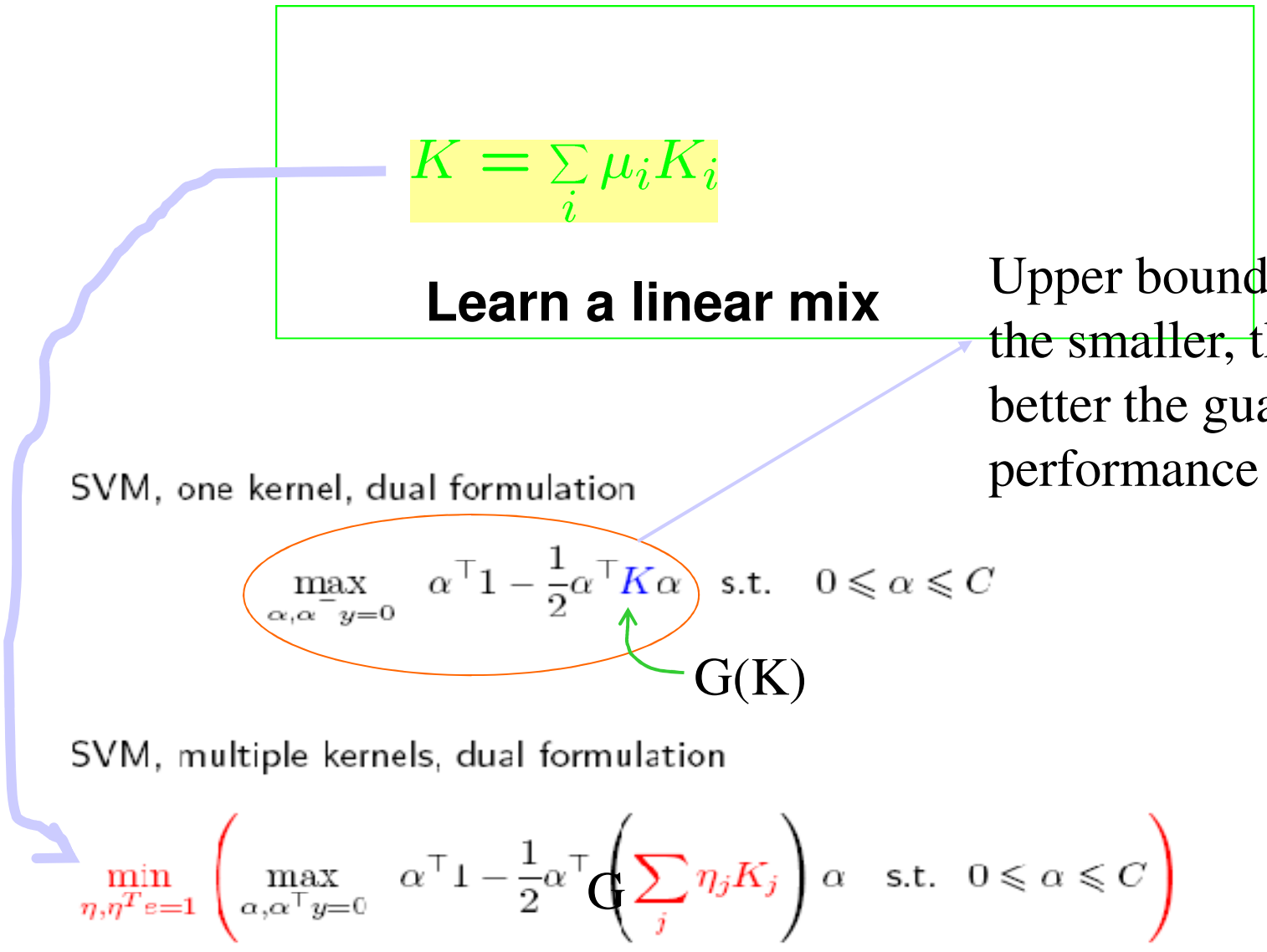
$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_i \xi_i \\ s.t. \quad & y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{aligned}$$

Dual problem

G(K)

$$\max_{\alpha} \quad \alpha^\top 1 - \frac{1}{2}\alpha^\top K \alpha \quad \text{s.t.} \quad \alpha^\top y = 0, \; 0 \leqslant \alpha \leqslant C$$

# Learning the optimal Kernel



To be Continued