# SVM Incremental Learning, Adaptation and Optimization


Chris Diehl
Applied Physics Laboratory
Johns Hopkins University
Chris.Diehl@jhuapl.edu

Gert Cauwenberghs
ECE Department
Johns Hopkins University
gert@jhu.edu



July 23, 2003

IJCNN – Incremental Learning Workshop


RTDC
RESEARCH & TECHNOLOGY DEVELOPMENT CENTER

# Motivation

- **What is the objective of machine learning?**
  - To identify a model from the data that generalizes well
- **When learning incrementally, the search for a good model involves repeatedly**
  - Selecting a hypothesis class (HC)
  - Searching the HC by minimizing an objective function over the model parameter space
  - Evaluating the resulting model's generalization performance

# Contributions

- **Unified framework for incremental learning and adaptation of SVM classifiers that supports**
  - Learning and unlearning of individual or multiple examples
  - Adaptation of the current SVM to changes in regularization and kernel parameters
  - Generalization performance assessment through exact and approximate leave-one-out error estimation
- **Generalization of incremental learning algorithm presented in**

  G. Cauwenberghs and T. Poggio, "Incremental and Decremental SVM Learning," *Advances in Neural Information Processing Systems (NIPS 2000)*, vol. 13, 2001.

# SVM Learning for Classification

- **Quadratic Programming Problem (Dual Form)**

$$\min_{0 \le a_i \le C} \mathrm{W} = \tfrac{1}{2} \sum_{i,j=1}^{N} a_i Q_{ij} a_j - \sum_{i=1}^{N} a_i + b \sum_{i=1}^{N} y_i a_i$$

$$Q_{ij} = y_i y_j K\left(x_i, x_j \middle| q\right) \qquad f(x) = \sum_{i=1}^{N} y_i a_i K\left(x_i, x \middle| q\right) + b$$

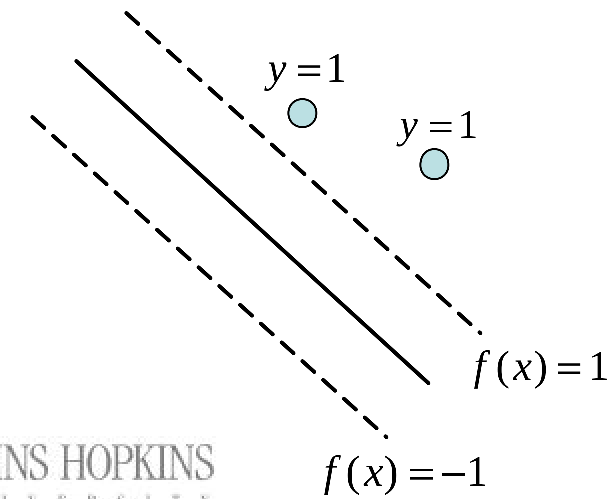
- **Model Selection**
  - Repeatedly select $(C, q)$
  - Solve the quadratic program
  - Perform cross-validation (approximate or exact)

# Karush-Kuhn-Tucker (KKT) Conditions

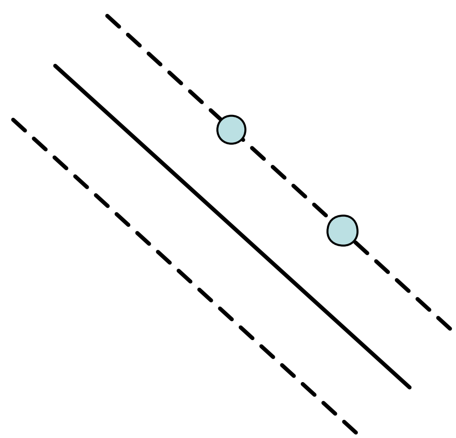
$$g_i = \frac{\partial W}{\partial a_i} = y_i f(x_i) - 1 = \begin{cases} > 0 & \forall i \in R \\ = 0 & \forall i \in S \\ < 0 & \forall i \in E \end{cases}$$

$$h = \frac{\partial W}{\partial b} = \sum_{j=1}^{N} y_j a_j = 0$$

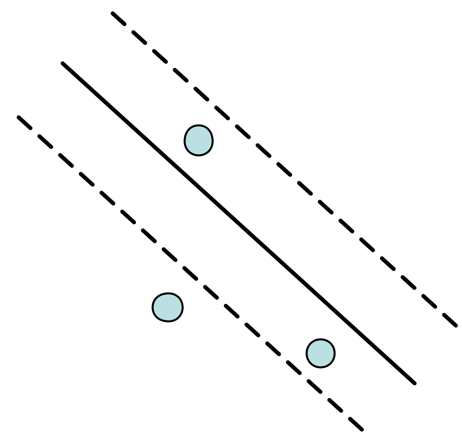
$R$: Reserve Vectors $(a = 0)$




$S$: Margin Vectors $(0 \le a \le C)$



$E$: Error Vectors $(a = C)$



# Incrementing Additional Examples into the Solution

$$f(x) = \sum_{i \in E} y_i a_i K\left(x_i, x \middle| q\right) + \sum_{j \in S} y_j a_j K\left(x_j, x \middle| q\right) + b$$

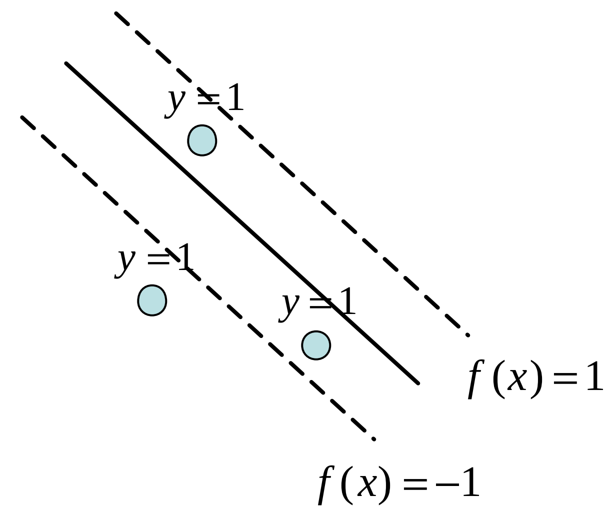
$$\Downarrow$$

$$f\,'(x) = \sum_{i \in E} y_i a_i K\left(x_i, x \middle| q\right) + \sum_{j \in S} y_j \left(a_j + \Delta a_j\right) K\left(x_j, x \middle| q\right) + \sum_{k \in U} y_k \Delta a_k K\left(x_k, x \middle| q\right) + b + \Delta b$$

$U$ : Unlearned Vectors $\left(0 \le a < C\right)$

**Strategy**

Preserve the KKT conditions for the examples in $R$, $S$ and $E$ while incrementing the examples in $U$ into the solution



# Part 1: Computing Coefficient and Margin Sensitivities

- For small perturbations of the unlearned vector coefficients, the error and reserve vectors will not change category
- Need to modify $\{\Delta a_k \;\forall\, k \in S\}$ and $\Delta b$ so that the margin vectors remain on the margin
- Transformation from original SVM to final SVM will be controlled via the perturbation parameter *p*

## Part 1: Computing Coefficient and Margin Sensitivities

- The *coefficient sensitivities* $\left\{ b_k = \frac{\Delta a_k}{\Delta p} \forall k \in S \right\}$ and $b = \frac{\Delta b}{\Delta p}$ will be computed by enforcing the constraints $\left\{ g_i = \frac{\Delta g_i}{\Delta p} = 0 \ \forall i \in S \right\}$ and $\frac{\Delta h}{\Delta p} = 0$
- *Margin sensitivities* $\{ g_i \ \forall i \in E, R, U \}$ can then be computed based on the coefficient sensitivities

# Part 2: Bookkeeping

- **Fundamental Assumption**
  - No examples change category (e.g. margin to error vector) during a perturbation
- **Perturbation Process**
  - Using the coefficient and margin sensitivities, compute the smallest $\Delta p$ that leads to a category change
    - Margin vectors: track changes in $a$
    - Error, reserve, unlearned vectors: track changes in $g$
  - Update the example categories and classifier parameters
  - Recompute the coefficient and margin sensitivities and repeat the process until $p = 1$.

# Decremental Learning and Cross-Validation

- **The perturbation process is fully reversible!**
- This allows one the option of unlearning individual or multiple examples to perform exact leave-one-out (LOO) or k-fold cross-validation
- The LOO approximation based on Vapnik and Chapelle's *span rule* is easily computed using the margin sensitivities

# Regularization and Kernel Parameter Perturbation Strategies

- **Regularization Parameter Perturbation**
  - Involves incrementing/decrementing error vector coefficients
  - Bookkeeping changes slightly since the regularization parameters change during a perturbation of the SVM
- **Kernel Parameter Perturbation**
  - First modify the kernel parameter
  - Then correct violations of the KKT conditions
  - Becomes increasingly expensive as the number of margin vectors in the original SVM increases