Hyperplane based Classifiers (3): SVM – Some Extensions

CS771: Introduction to Machine Learning

Piyush Rai

Plan

- A co-ordinate ascent based optimization algo for SVM
- Some extensions of binary SVM
  - Multi-class classification using SVM
  - One-class classification (a.k.a. novelty/outlier detection) SVM

A Co-ordinate Ascent Algorithm for SVM

- Recall the dual objective of soft-margin SVM (assuming no bias $\boldsymbol{b}$)

$$\underset{\mathbf{0} \leq \boldsymbol{\alpha} \leq C}{\operatorname{argmax}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m,n=1}^{N} \alpha_m \alpha_n y_m y_n \boldsymbol{x}_m^\top \boldsymbol{x}_n$$

Note that $\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n y_n \boldsymbol{x}_n$

- Focusing on just one of the components of $\boldsymbol{\alpha}$ (say α_n), the objective becomes

Can compute these in the beginning itself

Can efficiently compute it if we also store $\boldsymbol{w}$. It is equal to $\boldsymbol{w}^\top \boldsymbol{x}_n - \alpha_n y_n \|\boldsymbol{x}_n\|^2$

$$\underset{\mathbf{0} \leq \alpha_n \leq C}{\operatorname{argmax}} \; \alpha_n - \frac{1}{2} \alpha_n^2 \|\boldsymbol{x}_n\|^2 - \frac{1}{2} \alpha_n y_n \sum_{m \neq n} \alpha_m y_m \boldsymbol{x}_m^\top \boldsymbol{x}_n$$

- The above is a simple quadratic maximization of a concave function: Global maxima
- If constraint violated, project α_n in $[\mathbf{0}, C]$: If $\alpha_n < \mathbf{0}$, set it to 0, if $\alpha_n > C$, set it to C
- Can cycle through each coordinate α_n in a random or cyclic fashion

Multi-class SVM

- Multiclass SVMs (assuming $K > 2$ classes) use K wt vectors $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_K]$

  Prediction at test time: $\hat{y}_* = \mathrm{argmax}_{k\in\{1,2,\dots,K\}}\, \boldsymbol{w}_k^\top \boldsymbol{x}_*$

- Like binary SVM, can formulate a maximum-margin problem (without or with slacks)

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_{k=1}^{K} \frac{||\boldsymbol{w}_k||^2}{2} \quad \text{s.t.} \quad \boldsymbol{w}_{y_n}^\top \boldsymbol{x}_n \geq \boldsymbol{w}_k^\top \boldsymbol{x}_n + 1 \quad \forall k \neq y_n$$

Score on correct class

Score on an incorrect class $k \neq y_n$

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_{k=1}^{K} \frac{||\boldsymbol{w}_k||^2}{2} + C\sum_{n=1}^{N} \xi_n \quad \text{s.t.} \quad \boldsymbol{w}_{y_n}^\top \boldsymbol{x}_n \geq \boldsymbol{w}_k^\top \boldsymbol{x}_n + 1 - \xi_n \quad \forall k \neq y_n$$

- The version with slack corresponds to minimizing a multi-class hinge loss

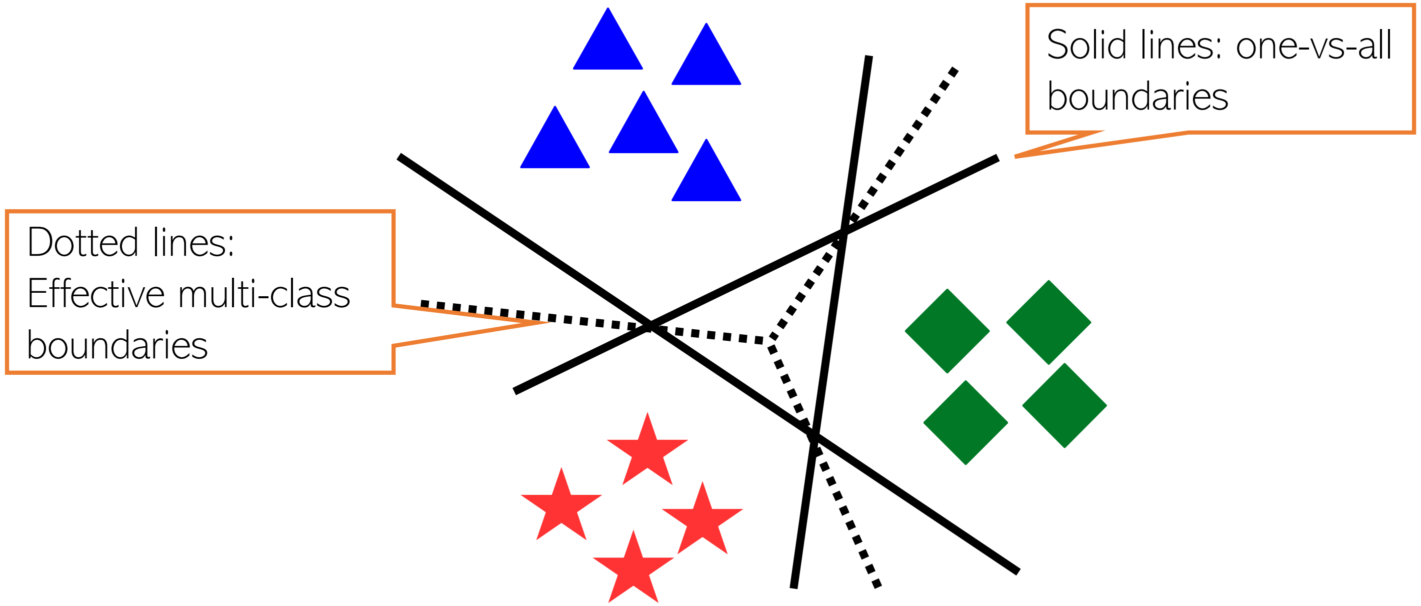
Crammer-Singer Multi-class SVM

$$\mathcal{L}(\boldsymbol{W}) = \sum_{n=1}^{N} \max\left\{0, 1 + \max_{k\neq y_n} \boldsymbol{w}_k^\top \boldsymbol{x}_n - \boldsymbol{w}_{y_n}^\top \boldsymbol{x}_n\right\} + \frac{\lambda}{2}\sum_{k=1}^{K} ||\boldsymbol{w}_k||^2$$

Loss=0 if score on correct class is at least 1 more than score on next best scoring class

Multi-class SVM using Binary SVM

- Can use binary classifiers to solve multiclass problems
- One-vs-All (also called One-vs-Rest): Construct K binary classification problems



- All-Pairs: Learn K-choose-2 binary classifiers, one for each pair of classes (j, k)

$$y_* = \arg\max_k \sum_{j \neq k} \mathbf{w}_{j,k}^\top \mathbf{x}_*$$

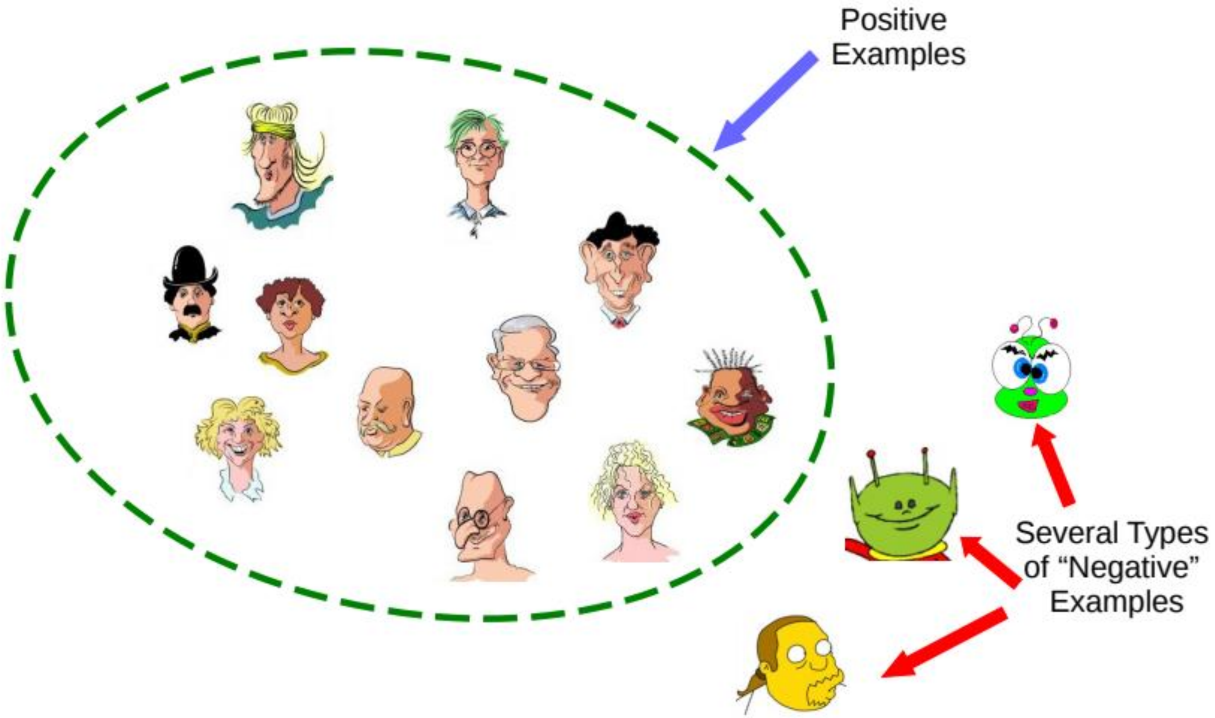
Whichever class k wins the most over other classes (or has the largest total scores against all other classes) is the prediction

Weight vector of the pairwise classifier for class j and k

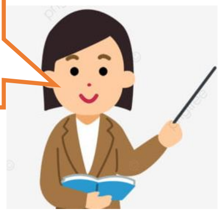
Positive score if class k wins over class j in pairwise comparison

One-class Classification

- Can we learn from examples of just one class, say positive examples?
- May be desirable if there are many types of negative examples



"Outlier/Novelty Detection" problems can also be formulated like this



- One-class classification is an approach to learn using only one class of examples

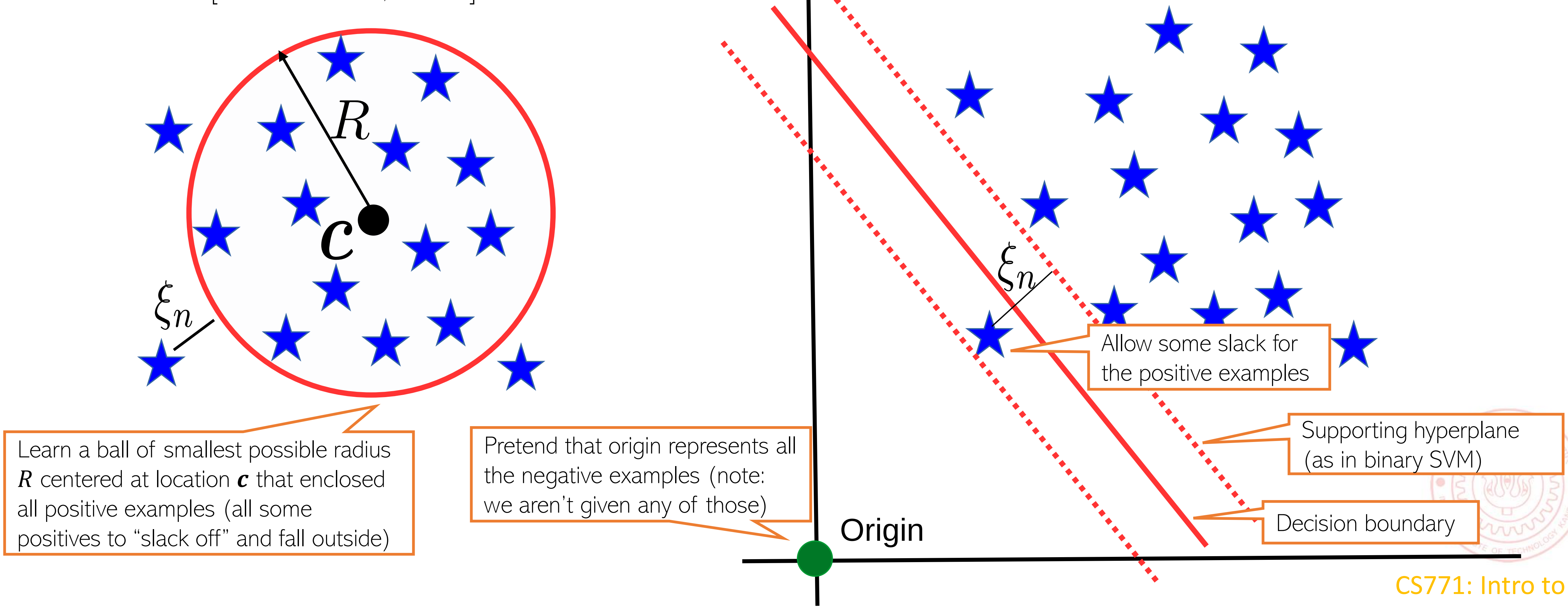
Pic credit: Refael Chickvashvili

One-class Classification via SVM-type Methods

- There are two popular SVM-type approaches to solve one-class problems

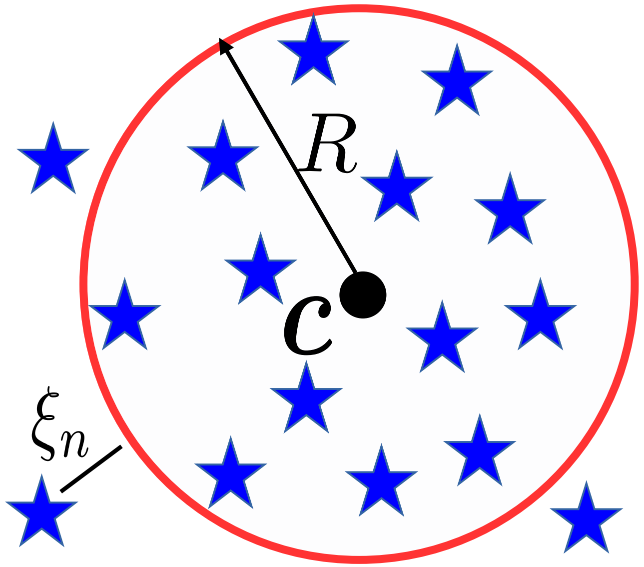
"Support Vector Data Description" (SVDD)
[Tax and Duin, 2004]

"One-Class SVM" (OC-SVM)
[Schölkopf et al., 2001]



One-class Classification via SVM-type Methods

"Support Vector Data Description" (SVDD)
[Tax and Duin, 2004]



Want to keep the ball's radius as small as possible

Hyperparameter ν to trade-off b/w the two terms

Want to keep training error (sum of slacks) to be small

$$\arg\min_{R,\boldsymbol{c},\xi} R^2 + \frac{1}{\nu N}\sum_{n=1}^{N}\xi_n$$

Want all training examples to fall within the ball (up to some slack ξ_n)

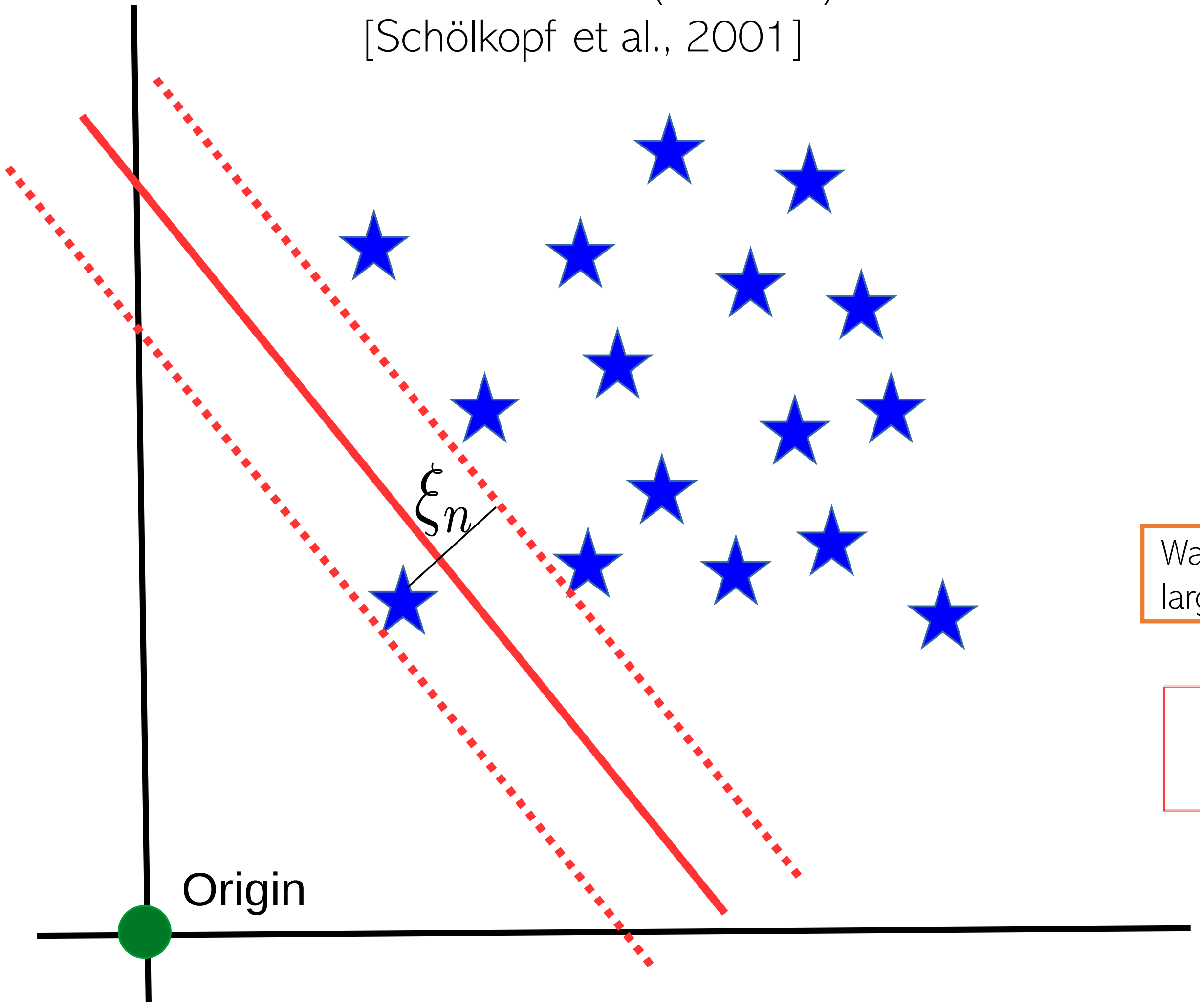
$$\text{s.t. } ||\boldsymbol{x}_n - \mathbf{c}||^2 \leq R^2 + \xi_n \quad \forall n$$

$$\xi_n \geq 0$$

Prediction Rule: $y_* = +1 \quad \text{if } ||\boldsymbol{x}_* - \mathbf{c}||^2 - R^2 < 0$

One-class Classification via SVM-type Methods

"One-Class SVM" (OC-SVM)
[Schölkopf et al., 2001]



Maximize the margin (similar to binary SVM)

Want to keep training error (sum of slacks) to be small

An offset term (want it large)

$$\arg\min_{\boldsymbol{w},\rho,\xi} ||\boldsymbol{w}||^2 + \frac{1}{\nu N}\sum_{n=1}^{N}\xi_n - \rho$$

$$\text{s.t. } \boldsymbol{w}^\top \boldsymbol{x}_n \geq \rho - \xi_n \quad \forall n$$

$$\xi_n \geq 0$$

Want a sufficiently large score (say ρ)

$$\boxed{\text{Prediction Rule: } y_* = +1 \quad \text{if } \boldsymbol{w}^\top \boldsymbol{x}_* > \rho}$$

Coming up next

- Kernel methods and nonlinear SVM via kernels