SVM nonseparable case

- Since the dual function does not have μ in it, we can simplify:

$$\max_{\lambda,\mu} \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\lambda^T Y^T X X^T Y \lambda \text{ such that:}$$

$$\sum_{i=1}^{n} y_i \lambda_i = 0$$

$$0 \leq \lambda \leq C.$$

- The dual problem is also a quadratic function, with much simpler constraints, with only n unknown variables λ.
- If we can solve the dual problem, the primal solution w can be recovered as: $w = X^T Y \lambda$.

SVM nonseparable case

- Complementary slackness:

$$\lambda_i(1 - a_i - y_i(w^T x_i + b)) = 0.$$
$$\mu_i a_i = 0.$$

- The condition translate to:

$$\lambda_i = 0 \rightarrow y_i(w^T x_i + b) \geq 1.$$
$$0 < \lambda_i < C \rightarrow y_i(w^T x_i + b) = 1.$$
$$\lambda_i = C \rightarrow y_i(w^T x_i + b) \leq 1.$$

- Notice that for those constraints i that correspond to $0 < \lambda_i < C$, those points are actually on hyperplane class 1 or -1.

SVM nonseparable case

- Calculate intercept b: Once w is obtained, we can calculate b by: find those points of class 1 that are on hyperplane 1: $\min_i w^T x_i$, for those we have:

$$w^T x_i + b = 1.$$

Similarly, for class -1 points that are on hyperplane -1: $\max_i w^T x_i$:$w^T x_i + b = 1$.

SVM with nonlinear boundary

- In many situations, the data can not be separable by a linear function (hyperplane), but rather by a non-linear function.
- In linear regression, we can add transformation of the original predictors to deal with possiblie non-linear relationship between response and predictors.
- With SVM, we can use similar method, which is called feature mapping. (Feature is similar to predictors, or attributes).
- Original predictors are mapped to higher dimensional, where hopefully the data will be separable by a linear function.

SVM with nonlinear boundary

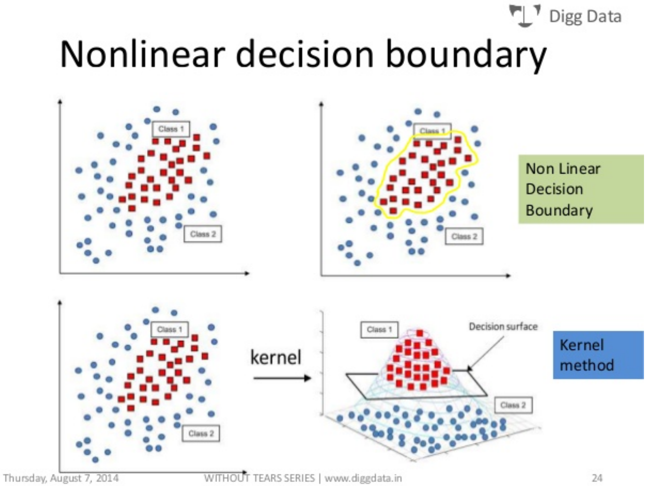

Figure: Image from Ankit Sharma, www.diggdata.in

- If data input is mapped to sufficiently high dimension, observations can be linearly separable.
- N observations can be separate linearly in a space of N+1 dimension or more.
-

$$\theta : \mathcal{R}^2 \to \mathcal{R}^3$$
$$(x_1, x_2) \to (z_1, z_2, z_3) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

- The linear boundary will be in $\mathcal{R}^3$, of the form $w^T x + b = 0$:

$$w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 + b = 0.$$

- Notice this boundary is actually an ellipse, mapped to $\mathcal{R}^3$ will become a linear classifier interms of the features.

- Lets called the mapping $\theta(x) : \mathcal{R}^2 \to \mathcal{R}^3$, so $\theta(x_i)$ is the "new" observation x_i.
- In linear regression, if we consider quadratic, higher order polynomial, and interaction terms of p predictors, there will be a very large number of new predictors.
- in SVM, we can actually do this without running into the computation problem.
- In previous lecture, we see the formulation for linearly separable SVM:

$$\max_{\lambda} \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\lambda^T Y^T X X^T Y \lambda \text{ s.t:}$$

$$\sum_{i=1}^{n} y_i \lambda_i = 0$$

$$0 \leq \lambda \leq C.$$

- Let $H = Y^T X X^T Y$, notice that: $H_{ij} = y_i y_j \langle x_i, x_j \rangle$.
- When the data has been mapped to a higher dimension, the formulation remains the same:

$$\max_{\lambda} \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \lambda^T Y^T \theta(X) \theta(X^T) Y \lambda \text{ s.t:}$$

$$\sum_{i=1}^{n} y_i \lambda_i = 0$$

$$0 \leq \lambda \leq C.$$

  where the new H matrix has the form:
  $H_{ij} = y_i y_j \langle \theta(x_i), \theta(x_j) \rangle$.
- Still we need to do the computation $\langle \theta(x_i), \theta(x_j) \rangle$.

- Define kernel function $K(x_i, x_j) = \langle \theta(x_i), \theta(x_j) \rangle$, where $\theta(x)$ maps x to $(x^1 * x^1, \sqrt{2}x^1 * x^2, x^2 * x^2)$
-
$$\begin{aligned} K(x_i, x_j) &= \langle \theta(x_i), \theta(x_j) \rangle \\ &= (x_i^1)^2 * (x_j^1)^2 + 2x_i^1 x_i^2 x_j^1 x_j^2 + (x_i^2)^2 * (x_j^2)^2 \\ &= (x_i^1 x_j^1 + x_i^2 x_j^2)^2 = (\langle x_i, x_j \rangle)^2 \end{aligned}$$
- The kernel function allows you to compute the dot product $\langle \theta(x_i), \theta(x_j) \rangle$ in higher dimension feature space by just computing $(\langle x_i, x_j \rangle)^2$.
- The cost of computing the dot product in higher dimensional space is only a bit higher than computing the dot product in the original input space.

- If you want to map your input data to a feature space of polynomial of degree d, you can use the kernel:

  $$K(x, z) = (\langle x, z \rangle + c)^d.$$

  , where d is the degree of the polynomial, c controls the weight between x^d terms and interaction terms.
- Linear kernel (no transformation): $K(x, z) = \langle x, z \rangle$.
- Radial basis kernel (Gaussian Kernel): $K(x, z) = exp(-\frac{1}{2\sigma^2}\|x - z\|^2)$.
- What kernel to choose?First try to use linear kernel (basic SVM). Then try other kernels to see if the results improve.

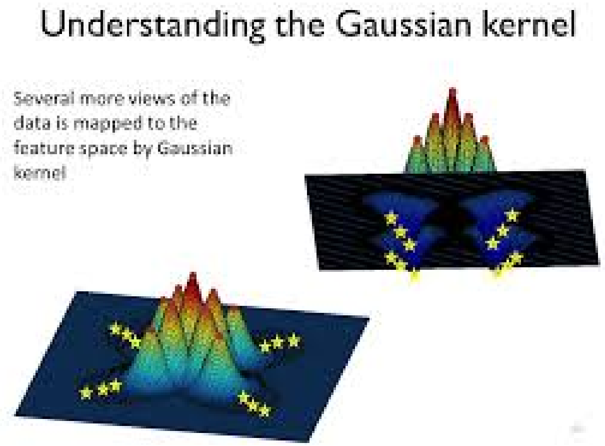

Figure: Materials from A Statnikov, D. Hardin, I. Guyon, C. Aliferis

- Once, the formulation has been solved and solution λ has been obtained, w can be found via:

$$w = \sum_{i=1}^{n} \lambda_i y_i \theta(x_i).$$

- Complementary slackness:

$$\lambda_i = 0 \rightarrow y_i(w^T x_i + b) \geq 1.$$
$$0 < \lambda_i < C \rightarrow y_i(w^T x_i + b) = 1.$$
$$\lambda_i = C \rightarrow y_i(w^T x_i + b) \leq 1.$$

It turns out only the λ entries that are non-zeros involved in calculating the hyperplane parameters w. Points with nonzero λ are called support vectors and they are responsible in constructing the separating hyperplane.

- SVM with ℓ_1 norm penalty that can perform variable selection.

$$\max_{w,b} \sum_{i=1}^{n} max(0, 1 - y_i(w^T x_i + b)) + C\|w\|_1.$$

- This can be formulated as a linear programming.
- In general, a linear programming (LP) problem has the form:

$$\min_{x} c^T x \text{ s.t: } Ax \leq b.$$

- SVM with ℓ_1 norm penalty can be formulated as a LP:

$$\max_{w,b,z,a} \sum_{i=1}^{n} a_i + C \sum_{i=1}^{p} z_i \text{ s.t:}$$
$$a_i \geq 0, a_i \geq 1 - y_i(w^T x_i + b)$$
$$z \geq 0, z \geq w, z \geq -w.$$

- SVM with elastic net penalty:

$$\max_{w,b} \sum_{i=1}^{n} max(0, 1 - y_i(w^T x_i + b)) + C_1\|w\|_1 + \frac{C}{2}\|w\|^2..$$

SVM for more than two classes

- The idea of separating hyperplane does not extend naturally to problems with more than two classes.
- There are many situation where you have data sets with more than two types of labels: i.e hand-written digit data.
- Indirect approach: Onver-versus-one classification and one-versus-all.
- One-versus-one approach: wtih K ¿ 2 classes: construct $\binom{K}{2}$ classifiers. For example, obtain a classifier for problem with class 1 and 2, etc.
- For a test observation, use each of the $\binom{K}{2}$ classifiers. The final classification is assigned to the class that it is most frequently classified to.

- One-versus-all approach: construct K classifiers, each time comparing one class vs. the rest K-1 classes: i.e take class 1 as coded $+1$, the other K-1 classes coded -1, and obtain a SVM classifier. Denote each classifier by (b_k, w_k).
- For a test observation x, classify it to the class with largest: $b_k + w_k^T x$.

- Data processing: avoid input data to be in large numeric ranges.
- Help with numerical stability of algorithms on data.
- Recommend scale input data to be in range [-1,1] or [0,1].
- Normalize to [0,1]: data=(data-min(data))/(min(data)-max(data))

- Matlab function for svm: fitcsvm.
- svm_example_3.m.