1.

Support Vector Machine(SVM)

Linear Binary SVM:

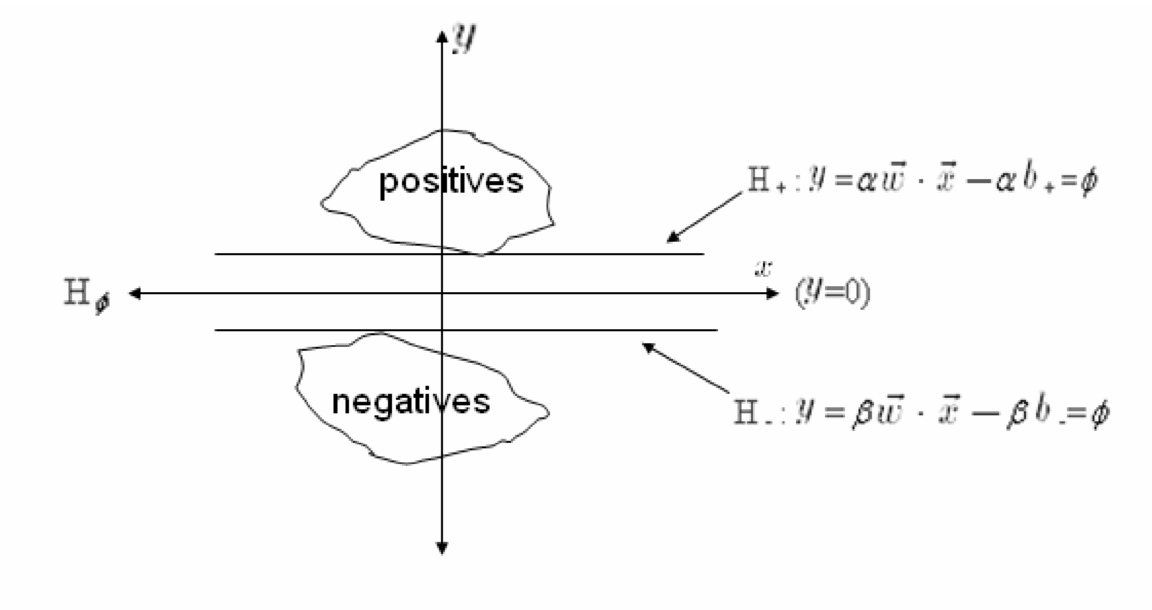
N training data “points” (feature vectors with binary labels):

$\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), ..., (\vec{x}_n, y_n)\}, \ \vec{x}_i \in \Re^{m}, \ y_i \in \{\pm 1\}$

Assumption: The positive and negative labled data($y_i = \pm 1$)is sufficiently separable and “dumped”(as positives and negatives) that notions such as positive and negative data clusters, and a hyperplane separating them, are meaningful. Often this is achieved manually, through the choice of feature vector components used to represent the data instances, separability of positives and negatives is often achieved without specific choice of feature vector, however, instead leaving this to (automated) tuning at the level of the SVM kernel(to be discussed later).

Want a separating hyperplane between positives and negatives, $y_i = \pm 1$, assume full separability possible with choice of feature vector ($f.v.$)components:

$$\mathrm{H}_{\phi} : \ y = \vec{\varpi} \cdot \vec{x} - b = \phi$$



y
positives
$\mathrm{H}_+ : y = \alpha \vec{w} \cdot \vec{x} - \alpha b_+ = \phi$
H_{ϕ}
x
$(y=0)$
negatives
$\mathrm{H}_- : y = \beta \vec{w} \cdot \vec{x} - \beta b_- = \phi$

All hyperplanes parallel,

$\mathrm{H}_{\phi} \parallel \mathrm{H}_{+} \| \mathrm{H}\_$→ all have hyperplanes proportioned to $\vec{\varpi}$.

2.

So, have without loss of generality (w,l,o,g):

H $_+$: $y = \vec{\varpi} \cdot \vec{x} - b_+ = \phi$
H $_-$: $y = \vec{\varpi} \cdot \vec{x} - b_- = \phi$

Again, rescaling on $\vec{\varpi}$ possible (w,l,o,g) to bring m to form:

H $_+$: $y = \vec{\varpi} \cdot \vec{x} - b_+ = +1$
H $_-$: $y = \vec{\varpi} \cdot \vec{x} - b_- = -1$

So, for fully separable binary data ($\vec{x}_i , y_i$):

$\vec{\varpi} \cdot \vec{x}_i - b \geq +1$ for $y_i = +1$

$\vec{\varpi} \cdot \vec{x}_i - b \leq -1$ for $y_i = -1$

Combined using sign trick:

$$y_i (\vec{\varpi} \cdot \vec{x}_i - b) - 1 \geq 0 \quad \forall_i$$

This is the key constraint that must be satisfied for separable(binary) data.

With rescaling the separation between H+ and H_ becomes:
First, the boundary hyperplanes are:

$\vec{x}_i^{(+)} \cdot \omega$-b=1; $\vec{x}_i^{(-)} \cdot \omega$-b= -1

Where $\vec{x}_i^{(+)}$ are $f.v.^{'s}$ from $\vec{x}_i$ that have y_i = +1, and that reside on(“support”) the H+ boundary (i.e., a support vector , ”S.V.”). Likewise for the $\vec{x}_i^{(-)}$.

$$(\vec{x}_i^{(+)} - \vec{x}_i^{(-)}) \cdot \omega = 2$$

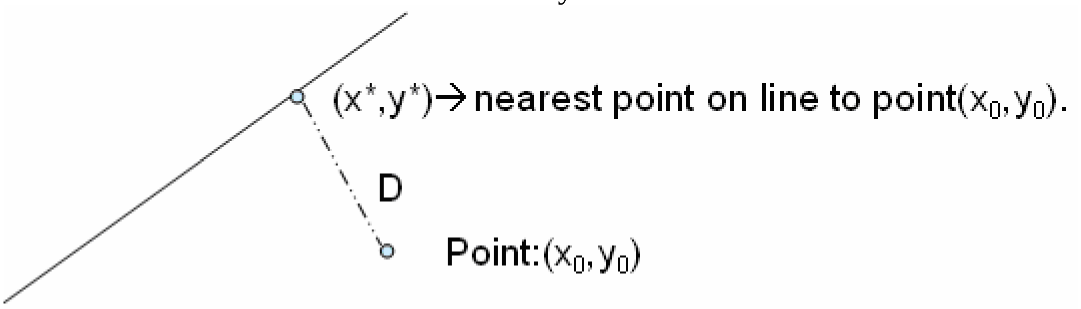
When ($x_i^{(+)} - x_i^{(-)}$) is perpendicular to the hyperplane the distance between the hyperplanes is given by d=‖($\vec{x}_i^{(+)}$ - $\vec{x}_i^{(-)}$) ‖ = $2/\|\omega\|$.   d= $2\|\omega\|^{-1}$

3.

A variational derivation of the result d= $2/\|\omega\|$, the distance between hyperplanes H_+ and H_, is now shown. The variational derivation is meant to provide a refresher on methods to be used in what follows,

2-D space:

Line: Ax+By+C=0



D=$\sqrt{(x^*-x_0)^2+(y^*-y_0)^2}$

Want to minimize D subject to constraint Ax*+By*+C =ϕ(i.e., that the nearest point to (x_0,y_0) reside on the line. This suggests the following Lagrangian formulation:

L(x*,y*,α)=D(x*,y*)+α [Ax*+By*+C]

The Lagrangian solution is obtained by minimizing L on choice of {x*, y*}, i.e., minimize D(x*, y*), but subject to the constraint Ax*+By*+C=0(encapsulated in the term with the Lagrange Multiplier):

$\frac{\partial L}{\partial \alpha}=$[Ax*+By*+C], requiring $\frac{\partial L}{\partial \alpha}=0$ then restores on constraint,

0=$\frac{\partial L}{\partial x^*}=\frac{(x^*-x_0)}{D}+\alpha$ A; 0=$\frac{\partial L}{\partial y^*}=\frac{(y^*-y_0)}{D}+\alpha$ B

$\frac{(x^*-x_0)^2}{D^2}=\alpha^2A^2$; $\frac{(y^*-y_0)^2}{D^2}=\alpha^2B^2\Rightarrow 1=\alpha^2$($A^2$+$B^2$)

4.

$Ax^*+By^*+C=0 \rightarrow [Ax_0+By_0+C]-\alpha\ (A^2+B^2)D=\phi$

$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \downarrow \quad \downarrow$

$\downarrow$ Sign choice, abs maintains consisting

$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow$

$Ax^* = -\alpha A^2 D + Ax_0 \qquad\qquad D=\frac{Ax_0+By_0+C}{\sqrt{A^2+B^2}}$

$By^* = -\alpha B^2\ D + By_0$

Generalization from 2-D space to m-D space, consider distance from a point on H_+ to H_ϕ :

$|Ax_0+By_0+C| \rightarrow |\vec{\omega}\cdot\vec{x}-b|=1$

While the "$\vec{\omega}$"parameters in H_ϕ (same as in H_+) correspond to:

$\sqrt{A^2+B^2} \rightarrow \sqrt{\sum_k w_k^2} = \sqrt{\vec{\omega}\cdot\vec{\omega}} = \|\vec{\omega}\|$

So, D= $1/\|\vec{\omega}\|$ from H_+ to H_ϕ , twice that for H_+ to $H\_$:

d= $2/\|\omega\|$

The SVM approach encapsulates a Key Structured Risk Minimization (SRM) criterion when it seeks to obtain the separable solution for which "d" is the greatest. This is the solution for which the separating hyperplane is the furthest distance possible from the positive & negative supper vectors (the nearest data points). The risk assumed in using a hyperplane to separate is, intuitively, lessened if we aient that separating hyperplane to maximize its distance to the training data (least sensitivity to $s.v.^{'s}$, etc)

For Separability we have the Constraint: $y_i(\vec{\omega}\cdot\vec{x}_i-b)-1\geq 0\ \forall_i$,

For SRM we have maximize d= $2/\|\omega\|$, or **minimize** $\|\omega\|^2$ (maximizing $\|\omega\|^{-2}$, instead of $\|\omega\|^{-1}$), is chosen due to simplifications in the formalism that follows---intuitively, if we max $1/\|\omega\|$ by min on $\|\omega\|$, it could just as well be done with min on $\|\omega\|^2$.

Lagrangian Formulation:

L($\vec{\omega}, b, \vec{\alpha}$)=$\frac{1}{2}\|\omega\|^2$- $\sum_i \alpha_i[y_i(\omega \cdot x_i - b) - 1]$, $\alpha_i \geq 0$

As with the practice Lagrangian described earlier, we seek to minimize L on { $\vec{\omega}, b$ }and to extremize(maximize in this case) L on{ $\vec{\alpha}$ },i.e., what results due to the simultaneous minimization/ maximization is what is called the saddle-point optimization for the solution.

Note how the inequality constraints above differ from the practice (distance-to-line) problem. In the latter case, the constraint was an exact equality (Ax*+By*+C=ϕ), and the term entering the Lagrangian was:
“α [Ax*+By*+C]”
For which the recovery of the constraint from $\frac{\partial L}{\partial \alpha} = 0$ was clear. Now, the Lagrange Multipliers,α , are no longer free to be positive or negative(recall the $\alpha = \pm (A^2 + B^2)^{-1}$ solution before). Now the α ’sare restricted to be positive, and the term entering the Lagrangian has an overall negative in front:

“-$\alpha_i[y_i(\omega \cdot x_i - b) - 1]$”

Where there are as many constraint “i” as there are training data points. The way to understand these constraints and their Lagrangian contributions is to directly consider the Lagrangian in an incremented saddle-point optimization:

Knows as the Karush-Kuhn-Tucker relations, i.e., the “KKT relations”

↓

- If [$y_i(\omega \cdot x_i - b) - 1$]>ϕ(constraint satisfied), then maximization(on $\alpha_i^{'s}$) for ”$\sum_i \alpha_i[y_i(\omega x_i - b) - 1]$” is achieved for $\alpha_i \rightarrow \phi$(since $\alpha_i \geq \phi$ constraint!).
- If [$y_i(\omega \cdot x_i - b) - 1$] =ϕ (constraint satisfied, a support vector), then there is no constraint or α_i .
- If [$y_i(\omega \cdot x_i - b) - 1$] <ϕ, then $\alpha_i \rightarrow \infty$!

6.

The last case, $[\, y_i(\omega \cdot \bar{x}_i - b) - 1\,] < \phi$, is an example of where the constraint is not satisfied. For completely separable data this case will not occur in the solution, but may occur when incrementally optimizing to achieve that solution.

As we shall see, non-separable (perfectly) data can have constraint violations in the solution. How is this managed if the Lagrangian optimization will drive the associated Lagrange multipliers to larger and larger positive values ($\alpha_i \to \infty$)?

The answer is to establish a max α cut off:
max(α_i)=C

Practically speaking the above is imposed for both separable and non-separable data.

When we consider the derivation of the “Dual Formalism” for L(ω, b, α), and compare to the same Dual on the non-separable formulation--- we will find that the Duals are the same(which is very convenient). Here is the formalism w/wo separability:

$$L=\frac{1}{2}\|\omega\|^2 - \sum_i \alpha_i [y_i(\omega \cdot x_i - b) - 1],\ \alpha_i \geq 0,$$

[Note: C$\geq \phi \to \alpha$ C term in Lagrangian with $\alpha \geq \phi$]

From an implementation stand point, if nothing else, have max(α_i)=C, so, practically speaking:

$$L=\frac{1}{2}\|\omega\|^2 - \sum_i \alpha_i [y_i(\omega \cdot x_i - b) - 1],\ \alpha_i \geq 0,\ \alpha_i \leq C \qquad (C-\alpha_i \geq 0)$$

The (C-$\alpha_i \geq$0) constraint can itself be absorbed into the Lagrangian:

$$L=\overset{\substack{\underline{\text{minimize}}\\ \downarrow}}{\frac{1}{2}\|\omega\|^2} - \sum_i \alpha_i [\underset{\text{positive variants}}{\underline{y_i(\omega x_i - b)}} - 1] + \sum \sigma_i (C \underset{\text{negative variants}}{\underline{-\alpha_i}}),\quad \overset{\substack{\text{interpretive (minimize)}\\ \downarrow}}{\sigma_i \geq \phi},\quad \overset{\substack{\text{(maximize)}\\ \downarrow}}{\alpha_i \geq \phi}$$

7.

If we rewrite the Lagrangian:

$$L=\frac{1}{2}\|\omega\|^2-\sum_i\alpha_i[y_i(\omega\cdot x_i-b)-1+\sigma_i]+C\sum\sigma_i\ ,\ \sigma_i\geq 0,\alpha_i\geq 0$$

Corresponds to constraint:

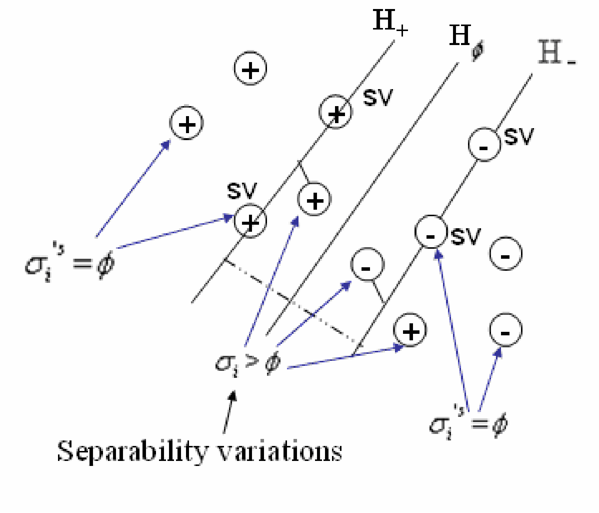
$$y_i(\omega\cdot x_i-b)-1+\sigma_i\geq\phi$$

$$y_i(\omega\cdot x_i-b)\geq 1-\sigma_i\qquad\sigma_i\geq\phi$$

Back tracking: $\omega\cdot x_i-b\geq 1-\sigma_i$ for $y_i=+1$

$\omega\cdot x_i-b\leq -1+\sigma_i$ for $y_i=-1$

So, the Lagrange Multiplier σ_i, introduced to deal with the max(α_i)=C constraint, can be interpreted as a “slack variable”:



- If C>$\alpha_i,\sigma_i\to\phi$
- If C=α_i,σ_i free($\geq\phi$)
- If C<$\alpha_i,\sigma_i\to\infty$

↓

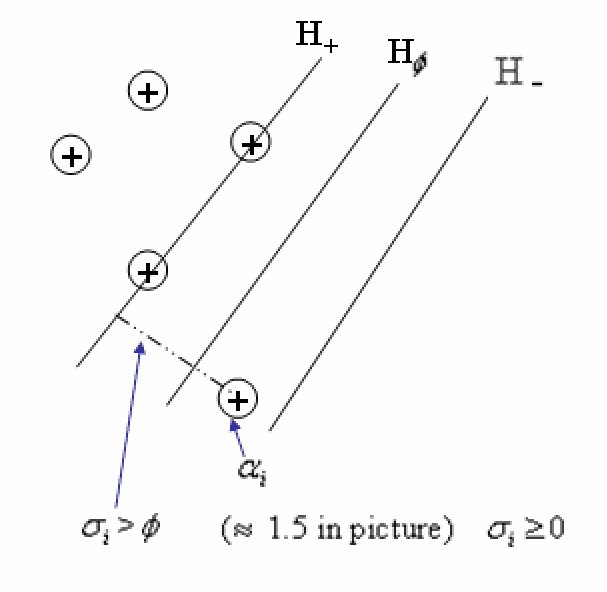
See Note

Note:

With the $\alpha^{'s}$ we have more control than with $[y_i(\omega\cdot x_i-b)-1]$, where previously $\alpha_i\to\infty$ resulted. Now, can avoid C<α_i condition by establishing initial conditions without C<α_i and maintaining these conditions as the Lagrangian optimization stars forward(such freedom of initialization not possible when “given” the training data{$\vec{x}_i,y_i$} as the starting point).

8.

To recap, a slack-variable formalism, to deal with non-perfect separability scenarios, naturally arises:



If we penalize violations with a term C, then a Lagrangian modification results to

↓ **(minimize)**

↓ σ_i $(C\sum_i \sigma_i)$

The new slack constraint is: $\sum_i \alpha_i [y_i(\omega \cdot x_i - b) - 1 + \sigma_i]$

Together:

$$L=\frac{1}{2}\|\omega\|^2 - \sum \alpha_i [y_i(\omega \cdot x_i - b) - 1 + \sigma_i] + C\sum \sigma_i \ , \ \sigma_i \geq 0, \alpha_i \geq 0$$

Exactly what we had from the max(α_i)=C constraint!

9.

Dual Calculations:

$$\text{L}=\frac{1}{2}\|\,\omega\,\|^2 \text{-} \sum_i \alpha_i [y_i(\omega \cdot x_i - b) - 1]\,,\ \alpha_i \geq 0$$

$$\text{L}\,\sigma = \frac{1}{2}\underset{\substack{\downarrow \\ \sum_j \omega_j^{\;2}}}{\underline{\|\,\omega\,\|^2}} \text{-} \sum_i \alpha_i [y_i(\omega \cdot x_i - b) - 1 + \sigma_i\,] \text{+C} \sum \sigma_i\ ,\ \sigma_i \geq 0, \alpha_i \geq 0$$

$$0=\frac{\partial L}{\partial \omega_j} = \omega_j - \sum_i \alpha_i y_i (x_i)_j \quad \forall_j \Rightarrow\ \underline{\vec{\omega} = \sum_i \alpha_i y_i \vec{x}_i}$$

$$0=\frac{\partial L}{\partial b} = \sum_i \alpha_i y_i \Rightarrow\ \sum_i \alpha_i y_i = \phi$$

Notice that $\frac{\partial L_\sigma}{\partial \omega_j} = \frac{\partial L}{\partial \omega_j}$ and $\frac{\partial L_\sigma}{\partial b} = \frac{\partial L}{\partial b}$, so in the duality transform, where we shift to dual variables without direct reference to $\{\,\omega, b\,\}$, it will much the same:

$$\tilde{L} = \frac{1}{2}\sum_j (\sum_i \alpha_i y_i (x_i)_j)^2 - \sum_{i^{'}} \alpha_{i^{'}} y_{'i} (\sum_j (\sum_i \alpha_i y_i (x_i)_j)(x_{i^{'}})_j) + \sum_i \alpha_i$$

$$= \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_k (x_i)_k (x_j)_k$$

$$\boxed{\tilde{L}(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j\ ,\ \alpha_i \geq 0}$$

Where we want to find the $\alpha^{'s}$ that maximize $\tilde{L}\,(\,\alpha\,)$. Notice how in the Dual Formalism the dependence on the training data is made very clear in the $\sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$ term.

10.

For the L_σ Dual:

$$L_\sigma \rightarrow \tilde{L}(\sigma,\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j - \sum \sigma_i (\alpha_i - C)\,,\ \sigma_i \geq 0, \alpha_i \geq 0$$

$$\boxed{\tilde{L}_\sigma(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j\,,\ \mathrm{C} \geq \alpha_i \geq 0}$$

So, the duals are the same aside from the C$\geq \alpha_i$ ($\max(\alpha) \leq C$) constraint, which is desirable(in some implementations) anyway.

Before moving on with the solution to $\tilde{L}_\alpha(\alpha)$, and thus the whole optimization/solution for the “decision hyperplane” (the trained SVM), let’s examine further how the training data is entering the problem: Note the $\vec{x}_i \cdot \vec{x}_j$ term in the above(binary) classification Lagrangian. What does the $\vec{x}_i \cdot \vec{x}_j$ term represent?

$\vec{x}_i$: feature vector “i”

$\vec{x}_i \cdot \vec{x}_j = \| \vec{x}_i \| \, \| \vec{x}_j \| \cos\theta_{ij}$

Suppose $\| \vec{x}_i \| = 1 \ \forall_i \Rightarrow$ unit hyperspherical data (data points lie on unit hypersphere, where ||x||=1)

$\downarrow$

$$\sqrt{\sum_j (x_i)_j^{\,2}} = 1 \Rightarrow \boxed{\sum_j (x_i)_j^{\,2} = 1} \Rightarrow \| x_i \|$$

(Note: $\boxed{\sum_j |(x_i)_j| = 1}$ → discrete probability distribution interpretation if $(x_i)_j \geq 0 \ \ \forall_{i,j}$)

So, for data normalized to be unit hyperspherical: $\| \vec{x}_i \| = 1 \| \vec{x}_j \| \Rightarrow \ \vec{x}_i \cdot \vec{x}_j = \cos\theta_{ij}$

11.

On unit hyperspherical data the interpretation of θ_{ij} is clear, it is the spherical arc angle between points $\vec{x}_i$ and $\vec{x}_j$ on its surface:

$$\vec{x}_i \cdot \vec{x}_j = \cos\theta_{ij}$$

For small angle(data in same cluster): $\cos\theta = 1 - \frac{1}{2}\theta^2 + \ldots$

$$\vec{x}_i \cdot \vec{x}_j \approx 1 - \frac{1}{2}\theta_{ij}^{\ 2}$$

In the Lagrangian the constant term does not matter:

$$\tilde{L}_\sigma(\alpha) \approx \sum_i \alpha_i - \frac{1}{2}\underbrace{\sum_{i,j}(\alpha_i y_i)(\alpha_j y_j)}_{(\sum_{i,j}\alpha_i y_i)(\sum_j \alpha_j y_j)} - \frac{1}{2}\sum_{i,j}\alpha_i \alpha_j y_i y_j (-\frac{1}{2}\theta_{ij}^{\ 2})$$

$\downarrow$

=0 from $\frac{\partial L}{\partial b}$ constraint.

So, for unit hyperspherical, proximate, feature vectors:

“$\vec{x}_i \cdot \vec{x}_j$”$\approx -\frac{1}{2}\theta_{ij}^{\ 2} \approx -|\vec{x}_i - \vec{x}_j|^2$

↑ ↑

Square arc length Euclidean distance squared

So, for unit hyperspherical data, “$\vec{x}_i \cdot \vec{x}_j$”can be thought of as measuring a distance that has been regularized in some manner when the distance grows large(i.e. as angular coordinate limitation, or in exponentiation to be seen in what follows).

So, for unit hyperspherical data:

$$\vec{x}_i \cdot \vec{x}_j \approx \begin{cases} -|\,\vec{x}_i - \vec{x}_j\,|^2 \text{ for } \vec{x}_i \text{ “near” } \vec{x}_j \\ \text{regularized } \{\,\vec{x}_i, \vec{x}_j\,\} \text{ expression for } \vec{x}_i \text{ not “near” } \vec{x}_j \end{cases}$$

We will see in a moment that we are free to manipulate our feature vectors by some mapping (with inverse) such that $\vec{x}_i \rightarrow \Phi(\vec{x}_i)$, which blurs the boundaries in the above example, for example, by

$\Phi(\vec{x}_i)\cdot\Phi(\vec{x}_j) \approx \phi(-|\,\vec{x}_i - \vec{x}_j\,|^2)$ for positive, monotonically increasing ϕ, and for $\vec{x}_i$ “near” $\vec{x}_j$.

$\approx$ regularized $\phi(\{\vec{x}_i, \vec{x}_j\})$ expression for $\vec{x}_i$ not “near” $\vec{x}_j$ (which may be accomplished in choice of ϕ implied for when $\vec{x}_i$ “near” $\vec{x}_j$).

One popular function of type $\phi(-|\,\vec{x}_i - \vec{x}_j\,|^2)$ is the Gaussian:

$$\phi_G(\vec{x}_i, \vec{x}_j) = \exp({}^{-|\,\vec{x}_i - \vec{x}_j\,|^2}\!\diagup_{2\sigma^2})$$

For $\vec{x}_i \approx \vec{x}_j, \phi_G \approx 1 - \frac{1}{2\sigma^2}|\,\vec{x}_i - \vec{x}_j\,|^2$, and ignoring the unit constant as before we have the same form as with the hyperspherical case, except now with an extra tuning parameter(variance).

As mentioned, we can map our feature vectors in a variety of ways, including lifting to a higher dimensioned feature space. This can be very powerful in identifying a separating hyperplane in a higher-dimensional mapping:

$$\Phi : \vec{x}_i \mapsto \Phi(\vec{x}_i) \text{ formerly } \vec{x}_i \cdot \vec{x}_j$$

$$\tilde{L}_\sigma(\alpha) \approx \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$$

The inner product $\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$ can be described as a special type of Kernel function:

$$K_{ij} = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$$

Kernel functions expressible in this way satisfy what are known as Mercer’s conditions(positive semidefinite K). Not all kernels satisfy Mercer’s conditions, and are not describable as a mapping Φ on feature vectors. Although all kernels examined appear to satisfy Mercer’s condition, this will not be taken as a critical limitation (until needed). So, w,l,o,g,

$$\tilde{L}_\sigma(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij}, 0 \le \alpha_i \le C$$

As mentioned previously, the choice of kernel eliminates the need for refining a choice on feature vector mappings (beyond a certain point, such as requiring some consistent normalization on $f.v.^{'s}$ for example).

The Gaussian kernel is consistently one of the best performings:

$$K_G(\vec{x}_i, \vec{x}_j) = \exp(-|\vec{x}_i - \vec{x}_j|^2 \big/ 2\sigma^2)$$

The InnerProduct kernel(the naturally occurring one in the simple $f.v.$ case) is one of the worst performing:

$$K_{IP}(\vec{x}_i \cdot \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$$

How is the “regularization” in the Gaussian kernel better, and how might it be improved upon? The key is to see how the Gaussian kernel behaves with respect to it parameters: rewrite $K_G(\vec{y}, \vec{z}) = \exp(-\|\vec{y} - \vec{z}\|^2 / 2\sigma^2)$

$$\frac{\partial \ln K_G(\vec{y}, \vec{z})}{\partial y_K} = (y_K - z_K)/\sigma^2 \qquad \text{(Euclidean Distance)}$$

Clearly, the sign is important, as is a notion of difference. Suppose we generalize on this basis to

$$\frac{\partial \ln K_V(\vec{y}, \vec{z})}{\partial y_K} \quad \alpha - sign(y_K - z_K)/2\sigma^2$$

Where sign y=1 if y>0
=-1 if y<0

Notice: $$\frac{\partial(\sum_k |y_k - z_k|)^{1/2}}{\partial y_k} = \frac{\mathrm{sign}(\mathrm{y_k} - \mathrm{z_k})}{(\sum_k |y_k - z_k|)^{1/2}}$$

So, with integrating factor “$1\Big/\sqrt{\sum_k | y_k - z_k |}$”, we obtain an integrable form in one of the simplest ways possible. Note that now the sign convention is separated from the “notion of distance”, here re-entering the kernel expression by way of the integration factor:

$$\frac{\partial \ln K_V(\vec{y},\vec{z})}{\partial y_K} = (\frac{-1}{2\sigma^2})\ (\frac{\mathrm{sign(y_k\ \text{-}\ z_k)}}{\sqrt{\sum_k | y_k - z_k |}})$$

$$\boxed{K_V(\vec{y},\vec{z}) = \exp(-\sqrt{\sum_k | y_k - z_k |}\Big/2\sigma^2)}$$

This is usually the best performing kernel on the L_1 normed data considered in the channel current analysis
(L_1 norm: |x|$_1$=$\sum_k | x_k |$, a discrete prob. dist if $x_k > 0$ also)

Since $\sqrt[4]{L_1 - norm}$ can be a distance (triangle inequality, etc.), then the kernel, $K_V = \exp(-d_v^2 \big/ 2\sigma^2)$ satisfies Mercer’s conditions.

Consider now the case where the notion of difference is not arithmetic but multiplicative, i.e., based on (1-$\frac{z_k}{y_k}$) rather than ($y_k - z_k$) (for the Gaussian).
In doing so, we must restrict to $y_k \neq 0$ of course(thus, for all $f.v.$ components).

As before, the sign of $(y_k - z_k)$ is information preserved in (1-$\frac{z_k}{y_k}$), but the latter is not integrable. However, $\ln(y_k / z_k)$ also provides sign info—positive when $y_k > z_k$, etc., as before, and also includes a ratio(a multiplicative term). Which to go with? A combination seems best as this is integrable (but now need $z_k > 0$, to avoid negative terms in the log):

$$\frac{\partial \ln K_\sigma(\bar{y},\bar{z})}{\partial y_K} = (\frac{-1}{2\sigma^2})\,[(1\text{-}\frac{z_k}{y_k})+\ln(\frac{y_k}{z_k})]$$

$$\boxed{K_\sigma(\bar{y},\bar{z}) = \exp(-[D(y \parallel z) + D(z \parallel y)]/2\sigma^2)}$$

This is usually a close 2nd to the K_V kernel, sometimes out performing. This kernel relates $f.v.^{'s}$ via relative entropy terms:

$$D(y \parallel z) = \sum_k y_k \ln(\frac{y_k}{z_k})$$

Symmetrization imposed

on relative entropy $(D_{yz} + D_{zy})$

$\equiv$Kullback-Lerbler Divergence,

$$\left.\begin{array}{l} \dfrac{\partial D(y \parallel z)}{\partial y_k} = \ln(\dfrac{y_k}{z_k}) + 1 \\[2ex] \dfrac{\partial D(z \parallel y)}{\partial y_k} = -\dfrac{Z_k}{y_k} \end{array}\right\}$$

A fundamental, symmetrized, information comparison between two probability distributions.

$$(\frac{-1}{2\sigma^2})\,\frac{\partial[D(y \parallel z) + D(z \parallel y)]}{\partial y_k} = (\frac{-1}{2\sigma^2})\,[(1\text{-}\frac{z_k}{y_k})+\ln(\frac{y_k}{z_k})]$$

notice how the symmetrization is critical for coming together with a trivially matching term.

The doubly novel aspect of the entropic kernel is that it would be the very first guess if one wanted to generalized from kernels based on exponentially regularized, square distances, to exponentially regularized, symmetrized, divergences (beginning with the most fundamental, symmetrized ”relative entropy” also known as then Kullback-Leibler information divergence) .

Note that we begun with the supposition that sign was important, as was some well-behaved” notion of difference” (whether if be distance-based or <u>divergence-based</u>, etc.)

Remarkably, the entropic kernel K_σ appears to satisfy Mercer’s condition, when properly restricted to discrete probability distributions: $y_k > 0, \sum y_k = 1$. This is not established with precise mathematical proof, but is established through exhaustive numerical testing.

The Mercer test: $\sum_{i,j} k(\vec{x}_i, \vec{x}_j) C_i C_j \geq 0 \;\; \forall \vec{C} \in \Re^m$ (for positive semidefinite K)

To Recap:

Seeking the $\vec{\alpha}$ that will maximize the following Larangian:

$$\widetilde{L}_\sigma(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K_{ij}, 0 \le \alpha_i \le C$$

<u>KKT Relations</u>: Let $f(\vec{x}_i) = \omega \cdot \vec{x}_i - b$

$x_i \cdot x_j \Rightarrow K_{ij}$

Have solution on constraints when:

- $\alpha_i = \phi \Leftrightarrow y_i f(\vec{x}_i) \ge 1$
- $0 < \alpha_i < C \Leftrightarrow y_i f(\vec{x}_i) = 1$
- $\alpha_i = C \Leftrightarrow y_i f(\vec{x}_i) \le 1$

Where:

$$f(\vec{x}_i) = \omega \cdot \vec{x}_i - b = \sum_j \alpha_j y_j (\vec{x}_j \cdot \vec{x}_i) - b$$

$\downarrow$ kernel generalization($x \cdot y \to K(x, y)$)

$$f(\vec{x}_i) = \sum_j \alpha_j y_j K(\vec{x}_j \cdot \vec{x}_i) - b$$

Initialization:

Since $\sum_i \alpha_i y_i = \phi$, choose $\alpha_i^{+} = \frac{1}{N^{+}}, \alpha_i^{-} = \frac{1}{N^{-}}$

Now to consider the solution $\vec{\alpha}$ by sequential Minimal Optimization(SMO), where successive pairs of $\alpha^{'s}$ are selected for optimization!

$\widetilde{L}_\sigma(\vec{\alpha}) = \widetilde{L}_\sigma(\alpha_1, \alpha_2, \alpha_3, ...\alpha_n)$ then “freeze” all but α_1 and α_2 in variational optimization (each pair of $\alpha^{'s}$ selected above could simply be relabeled as α_1 and α_2, thus use the same derivation to follow).

19.

SMO

$$\tilde{L}_\sigma(\vec{\alpha}) = \tilde{L}_\sigma(\alpha_1,\alpha_2;\alpha_3,...\alpha_m)\text{; Note: }\boxed{s = y_1 y_2}$$

$$=(\alpha_1+\alpha_2+\sum_{i\ge3}\alpha_i) - \frac{1}{2}(\alpha_1{}^2 k_{11} + \alpha_2{}^2 k_{22} + 2\alpha_1\alpha_2 s k_{12})$$

$$-\frac{1}{2}(2\alpha_1 y_1\sum_{j\ge3}\alpha_j y_j k_{1j} + 2\alpha_2 y_2 \sum_{j\ge3}\alpha_j y_j k_{2j}) - \frac{1}{2}\sum_{j\ge3,i\ge3}\alpha_i\alpha_j y_i y_j k_{ij}$$

Let $v_i = \sum_{j\ge3}\alpha_j y_j k_{ij} = \vec{\omega}\cdot\vec{x}_i - \alpha_1 y_1 k_{i1} - \alpha_2 y_2 k_{i2}$

$$\tilde{L}_\sigma(\vec{\alpha}) = \alpha_1 + \alpha_2 - \frac{1}{2}(\alpha_1{}^2 k_{11} + \alpha_2{}^2 k_{22} + 2\alpha_1\alpha_2 s k_{12}) - \alpha_1 y_1 v_1 - \alpha_2 y_2 v_2$$

$$\underbrace{+\sum_{i\ge3}\alpha_i - \frac{1}{2}\sum_{j\ge3,i\ge3}\alpha_i\alpha_j y_i y_j k_{ij}}_{\{\alpha_1,\alpha_2\}\text{independent term s}}$$

Now consider variational parameters other than $\{\alpha_1,\alpha_2\}$ to be fixed in the $\{\alpha_1,\alpha_2\}$ variational optimization. Furthermore:

$$\sum_i y_i\alpha_i = 0 \Rightarrow y_1\alpha_1 + y_2\alpha_2 = -\sum_{i\ge3} y_i\alpha_i$$

$$(\alpha_1 + s\alpha_2) = \gamma \qquad (\gamma = -y\sum_{i\ge3} y_i\alpha_i)$$

↑

does not depend on α_1 or α_2

$$\alpha_1 = \gamma - s\alpha_2$$

$$L(\alpha_2;\alpha_3,...\alpha_m) \quad =(\gamma - s\alpha_2)+\alpha_2-\frac{1}{2}((\gamma - s\alpha_2)^2 k_{11}+{\alpha_2}^2 k_{22}+2\alpha_2(\gamma - s\alpha_2)sk_{12})$$

$$(\alpha_1=\gamma - s\alpha_2) \quad -(\gamma - s\alpha_2)y_1 v_1-\alpha_2 y_2 v_2+[\text{terms independent of } \{\alpha_1,\alpha_2\}]$$

$$0=\frac{\partial L}{\partial \alpha_2}=(1-s)-{\alpha_2}^{*}\underline{(k_{11}+k_{22}-2k_{12})}+s\cancel{\gamma}k_{11}-s\cancel{\gamma}k_{12}+sy_1v_1-y_2v_2$$

Definition: $-\eta = k_{11}+k_{22}-2k_{12}$

new α, the optimization solution, old $\alpha^{'s}$

$$(-\eta){\alpha_2}^{*}=(1-s)+s(k_{11}-k_{12})\gamma+y_2(\vec{\varpi}\cdot\vec{x}_1-\alpha_1y_1k_{11}-\alpha_2y_2k_{12})$$
$$-y_2(\vec{\varpi}\cdot\vec{x}_2-\alpha_1y_1k_{21}-\alpha_2y_2k_{22})$$

$$(-\eta){\alpha_2}^{*}=(1-s)+s(k_{11}\cancel{-k_{12}})\gamma+y_2\vec{\varpi}\cdot(\vec{x}_1-\vec{x}_2)-\cancel{(\gamma}-s\alpha_2)sk_{11}-\alpha_2k_{12}+\cancel{(\gamma}-s\alpha_2)sk_{21}+\alpha_2k_{22}$$

$$=\cancel{(1-s)}+y_2[\vec{\varpi}\cdot x_1-y_1-(\vec{\varpi}\cdot x_2-y_2)+\cancel{(y_1-y_2)}]+\alpha_2\underbrace{(k_{11}+k_{22}-2k_{12})}_{(-\eta)}$$

$$\boxed{{\alpha_2}^{*}=\alpha_2-y_2[\vec{\varpi}\cdot x_1-y_1-(\vec{\varpi}\cdot x_2-y_2)]/\eta}$$

This gives an analytic calculation for the optimal new α_2. (${\alpha_2}^{*}$).
The problem is that the solution to the optimum may take α_2 out of range, i.e. $\alpha_2 > C$ not allowed. If such is indicated, then the procedure is to “clip” the α_2 update to the boundary value ($\alpha_2 = C$): ${\alpha_2}^{*} \rightarrow {\alpha_2}^{*}|_{clipped}$

$${\alpha_1}^{*}=\alpha_1+s(\alpha_2-{\alpha_2}^{*}|_{clipped})$$

21.

There are some subtleties in how to manage the clipping with constraints on both α_1 and α_2:

Suppose varying $\{\alpha_1, \alpha_2\}$ and $y_1 \neq y_2$, and that $\boxed{\alpha_2{*} = \alpha_2 + \Delta\alpha_2}$

$\sum y_i \alpha_i = \phi \Rightarrow \Delta\alpha_1 = \Delta\alpha_2$

Maximum ($\Delta\alpha_2$) $= \alpha_2 + (C - \alpha_2) = C$

$\Delta\alpha_2 = C - \alpha_2$ covers rest of distance
to

or

$= \alpha_2 + (C - \alpha_1) = C + \alpha_2 - \alpha_1$

$\alpha_2 \leq C$ boundary

For $y_1 \neq y_2$
Max ($\Delta\alpha_2$) =Min $\{C - \alpha_2, C - \alpha_1\}$
Min ($\Delta\alpha_2$) =Max $\{-\alpha_2, -\alpha_1\}$

$\Delta\alpha_1 = C - \alpha_1$ covers rest of distance to $\alpha_1 \leq C$ boundary and since $\Delta\alpha_1 = \Delta\alpha_2$ are tied together, this limits $\Delta\alpha_2 = C - \alpha_1$

For $y_1 = y_2 \Rightarrow \Delta\alpha_1 = -\Delta\alpha_2$
Max ($\Delta\alpha_2$) =Min $\{(C - \alpha_2), \alpha_1\}$
Min ($\Delta\alpha_2$) =Max $\{-\alpha_2, \alpha_1 - C\}$

SMO allows us to reduce the computational problem significantly due to the exact analytical solutions obtained. Further, subtleties arise upon implementing in actual code, as will be discussed in the cluster, Perl code comments to follow.