Logistic regression

A probabilistic classification model

$\{(x_i, y_i)\}_{i=1}^N \quad x_i \in \mathbb{R}^D \quad y_i \in \{0, 1\}$

We just saw generative models

e.g. $p(x|y=c,\theta) = \mathcal{N}(x|\mu_c, \Sigma_c)$

We then compute Posterior $p(y|x,\theta)$

$\propto p(y)\, p(x|y,\theta)$

to classify.

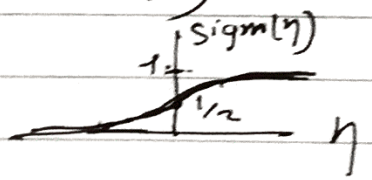
One might want to have discriminative models directly giving us the probability $p(y|x,\theta)$

We saw that two class Gaussian model with tied covariance matrices had a posterior class probability given by a sigmoidal / logistic function of a linear combination of variable. We could try to learn such model directly from labeled data.

$$p(y|x,w) = \text{Ber}(y|\text{sigm}(w^T x))$$

Remember

$$\text{sigm}(\eta) = \frac{1}{1+e^{-\eta}}$$



BTW, if we need to add a constant shift, say $P(y|x, \omega) = \frac{1}{1 + e^{-(\omega^T x + \omega_0)}}$

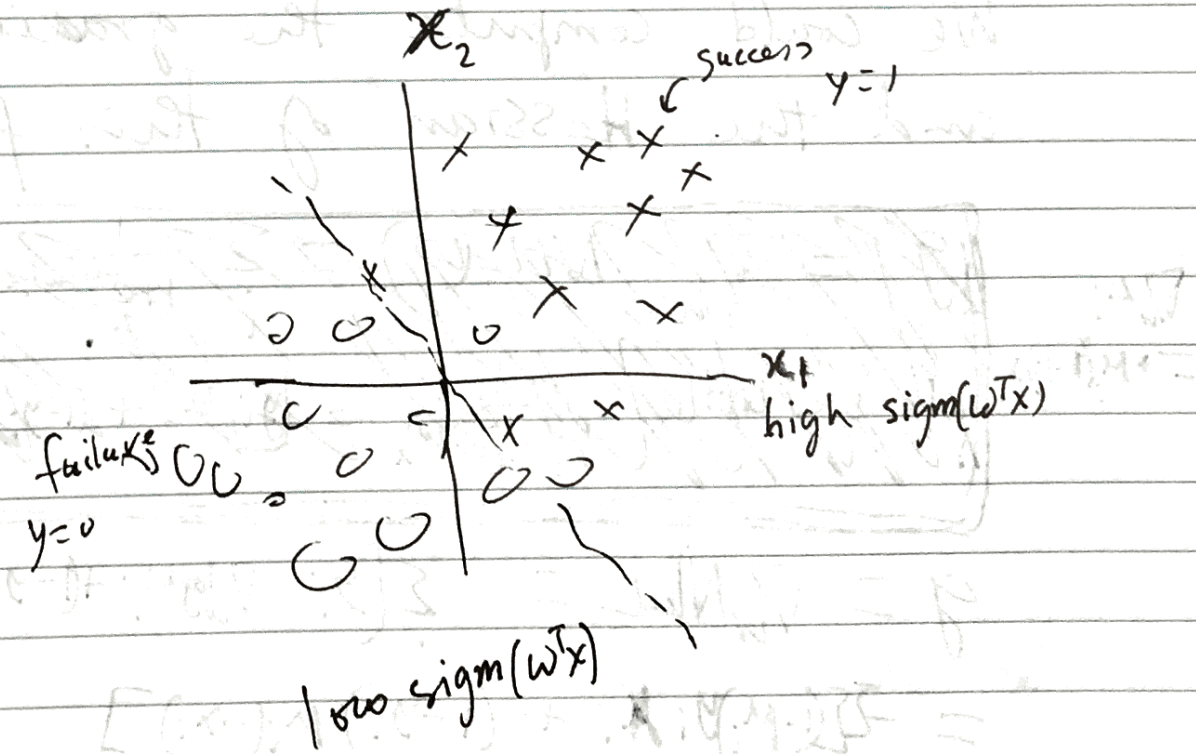
we could just define bigger vector

$\omega \to (\omega_0, \omega) = \tilde{\omega}$

$x \to (1, x) = \tilde{x}$ $\qquad P(y|x,\theta) = \frac{1}{1 + e^{-\tilde{\omega}^T \tilde{x}}}$

$$\prod_i P(y_i | x_i, \omega) = \prod_i \left[\mu_i^{I(y_i = 1)} (1 - \mu_i)^{I(y_i = 0)} \right]$$

where $\mu_i = \frac{1}{1 + e^{-\omega^T x_i}}$



MLE

Negative log likelihood

$$NLL(w) = -\sum_i \left[y_i \log \mu_i + (1-y_i) \log(1-\mu_i) \right]$$

By the way $1-\mu = \left(1 - \frac{1}{1+e^{-w^T x}}\right) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$

$$= \frac{1}{e^{w^T x} + 1}$$

So, if we use $\tilde{y}_i = \pm 1 = 2y_i - 1$

$$NLL(w) = -\sum_i \log \frac{1}{1+e^{-\tilde{y}_i w^T x_i}}$$

$$= \sum_i \log\left(1 + e^{-\tilde{y}_i w^T x_i}\right)$$

Unfortunately we can't write down $\hat{w}$ in one shot

We could compute the gradient and the Hessian of this function

$\nabla_w \mu_i = \mu_i(1-\mu_i) x_i$

$$g = \nabla_w NLL = \frac{1}{N} \sum_i \frac{1}{1+e^{\tilde{y}_i w^T x_i}} e^{\tilde{y}_i w^T x_i} (\tilde{y}_i x_i)$$

If we write $\frac{1}{1+e^{\tilde{y}_i w^T x}} = [y_i \mu_i + (1-y_i)(1-\mu_i)]$

$$g = \nabla_w NLL = -\sum_i \left[y_i \nabla_w \log \mu_i + (1-y_i) \nabla_w \log(1-\mu_i) \right]$$

$$= -\sum_i \left[(1-\mu_i) y_i x_i + (1-y_i) \mu_i (-x_i) \right]$$

$$= \sum_i (\mu_i - y_i) x_i = X^T(\mu - y)$$

$$H = \nabla_w g(w)^T = \sum_i (\nabla_w \mu_i) x_i^T$$

$$= \sum_i \mu_i(1-\mu_i) x_i x_i^T$$

$$= X^T S X$$

where $S \triangleq \text{diag}(\mu_1(1-\mu_1), \cdots, \mu_N(1-\mu_N))$

$$X = \begin{pmatrix} x_{11} \cdots x_{1D} \\ \vdots \\ x_{N1} \cdots x_{ND} \end{pmatrix}$$

$$X^T = \begin{pmatrix} x_{11} \cdots x_{N1} \\ \vdots \\ x_{1D} \quad x_{ND} \end{pmatrix}$$

$$\mu - y = \begin{pmatrix} \mu_1 - y_1 \\ \vdots \\ \mu_N - y_N \end{pmatrix}$$

$$S = \begin{pmatrix} \mu_1(1-\mu_1) & & \\ & \mu_2(1-\mu_2) & \\ & & \ddots \mu_N(1-\mu_N) \end{pmatrix}$$

How do we minimize NLL(w)?

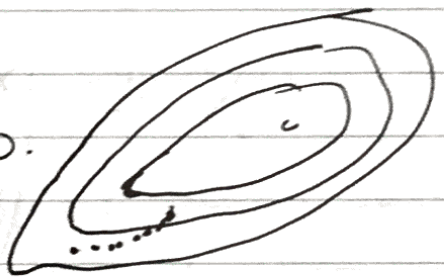
One thought: Steepest descent

$$\theta_{k+1} = \theta_k - \eta_k g_k$$

Issues with step size

Big η

Potential oscillation



Small η

May be too slow.

There are many methods to deal with issues of stepsize selection. We describe one of the popular methods in the contact of logistic regression.

Newton's method

Initialize θ_1:

for $k = 1, 2 \ldots$ until convergence do

Evaluate $g_k = \nabla f(\theta_k)$

Evaluate $H_k = \nabla^2 f(\theta_k)$;
Solve $H_k d_k = -g_k$;
Use line search to find stepsize η_k along d_k;
$\theta_{k+1} = \theta_k + \eta_k d_k$;

Roughly $f(\theta+d) = f(\theta) + g \cdot d + \frac{1}{2} d^T H d + \cdots$

The second order approximation is minimized when $g + Hd = 0$
(assuming H_k is positive definite)
To avoid over stepping one uses line search in the direction of d.

For one dimension 

For logistic regression with $\eta_k = 1$

$$w_{k+1} = w_k - H_k^{-1} g_k$$
$$= w_k + (X^T S_k X)^{-1} X^T (y - \mu_k)$$
$$= (X^T S_k X)^{-1} [(X^T S_k X) w_k + X^T (y - \mu_k)]$$
$$= (X^T S_k X)^{-1} X^T S_k [X w_k + S_k^{-1} (y - \mu_k)] = (X^T S_k X)^{-1} X^T S_k z_k$$

where $z_k \triangleq X W_k + S_k^{-1}(y - \mu_k)$
which is called the working response

Note that W_{k+1} optimizes

$$\sum_{ki} S_{ki}\left(z_{ki} - w^T x_i\right)^2$$

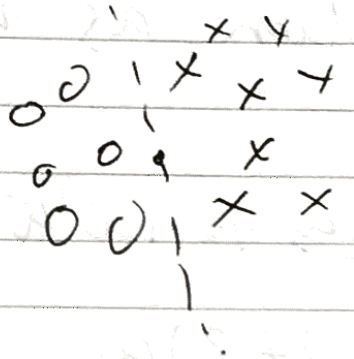
$$\nabla_w \Sigma s(\,)^2 = 0 \Rightarrow \sum_{ki}\left[x_i^T S_{ki} z_{ki} - x_i^T S_{ki} x_i w\right] = 0$$

$$z_{ki} = W_k^T x_i + \frac{y_i - \mu_{ki}}{\mu_{ki}(1-\mu_{ki})}$$

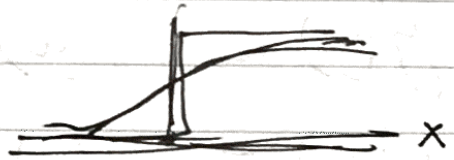
Iterating recomputation of z_{ki} with least square minimization gives Iteratively reweighted least Square (IRLS) algorithm.

Now let us discuss issues of regularization. What if training data for the two classes

were linearly separable, by chance?

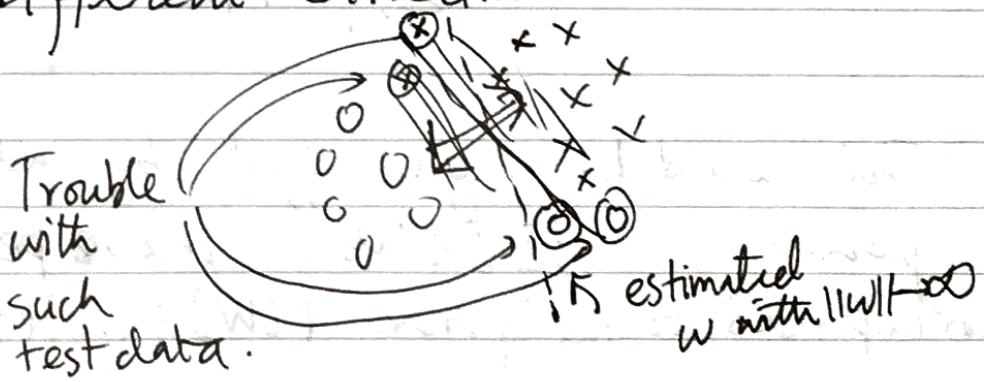


In that case, once the direction is established., $\|w\| \to \infty$



That way all $y=1$ have probability 1, and all $y=0$ have probability 0, maximizing the likelihood $\prod_{i:y=1} 1 \prod_{i:y=0} (1-0)$

This is brittle. Let us say the true model has finite w and a slightly different direction



One option is to add an l_2 regularization as in ridge regression

$$f'(w) = NLL(w) + \lambda w^T w$$

$$g'(w) = g(w) + 2\lambda w$$

$$H'(w) = H(w) + 2\lambda I$$

Also, one could generalize logistic regression to multiclass ~~setting~~ setting

$$p(y=c|x,w) = \frac{\exp(w_c^T x)}{\sum_{c'=1}^{C} \exp(w_{c'}^T x)}$$

This could be regularized by adding a penalty $\frac{1}{2}\sum_c w_c^T V_0^{-1} w_c$. This penalty corresponds to the prior $p(W) = \prod_c N(w_c|0, V_0)$

As we will see, just regularizing may not be enough!

Bayesian logistic regression

As we discussed before, instead of using the mode of $p(w|D)$, namely MAP estimate, we might want to integrate over w.

Let us take a more general look at Bayesian integration in a particular approximation that becomes accurate in the large data size limit.

Think of $P(\theta|D) = \frac{e^{-E(\theta)}}{Z}$ with

$Z = \int d\theta' e^{-E(\theta')}$. If $E(\theta) = -\log P(\theta|D)$

$Z = P(D) \triangleq$ evidence.

Let us say θ^* is the MAP estimate (lowest energy $\triangleq$ highest posterior prob.)

$$E(\theta) = E(\theta^*) + (\theta - \theta^*)^T g + \frac{1}{2}(\theta - \theta^*)^T H (\theta - \theta^*) + \ldots$$

If θ^* is the mode, $g = 0$ $\quad g \in R^D$, $H \in R^{D \times D}$

$$Z = \int e^{-E(\theta)} d\theta = e^{-E(\theta^*)} \int d\theta\, e^{-\frac{1}{2}(\theta-\theta^*)^T H (\theta-\theta^*)}$$

$$= e^{-E(\theta^*)} (2\pi)^{D/2} |H|^{-1/2}$$

This BIC approximation is known as the Laplace approximation.

$$\log P(\mathcal{D}) = -E(\theta^*) = \frac{1}{2}\log|H| + \text{constants}$$

$$= \log P(\theta^*|\mathcal{D}) - \frac{1}{2}\log|H| + \text{const.}$$

$$= \log P(\mathcal{D}|\theta^*) + \log P(\theta^*) - \frac{1}{2}\log|H| + \text{cst.}$$

- $\log P(\mathcal{D}|\theta^*)$: Related to MLE
- $\log P(\theta^*)$: Related to MAP
- $-\frac{1}{2}\log|H|$: Contribution of Bayesian integration in Laplace approximation

Relation to Bayesian information criterion (BIC)

N pieces of data: $H = \sum_{i=1}^{N} H_i$, $H_i = \nabla\nabla \log P(\mathcal{D}|\theta)$

If each $H_i \simeq \hat{H}$, $H \simeq N\hat{H}$

$$\log|H| \simeq \log|N\hat{H}| = \log N^D |\hat{H}|$$

$$\log P(\mathcal{D}) \sim \log P(\mathcal{D}|\theta^*) + \log P(\theta^*) - \frac{D}{2}\log N - \frac{1}{2}\log|\hat{H}| + \ldots$$

$\sim N$ $\quad \sim 1$ $\quad \sim \log N$ $\quad \sim 1$

$\sim \log P(D|\theta^*) - \frac{D}{2}\log N$

For comparing models, the term $-\frac{D}{2}\log N$ penalizing more complex models.
(higher $\log P(D|\theta^*)$ achieved by bigger D offset by $-\frac{D}{2}\log N$).

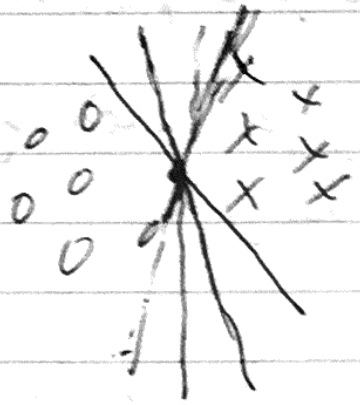
Akaike information criterion (AIC) considers $2D - 2\log P(D|\theta^*)$
$= -2(\log P(D|\theta) - D)$

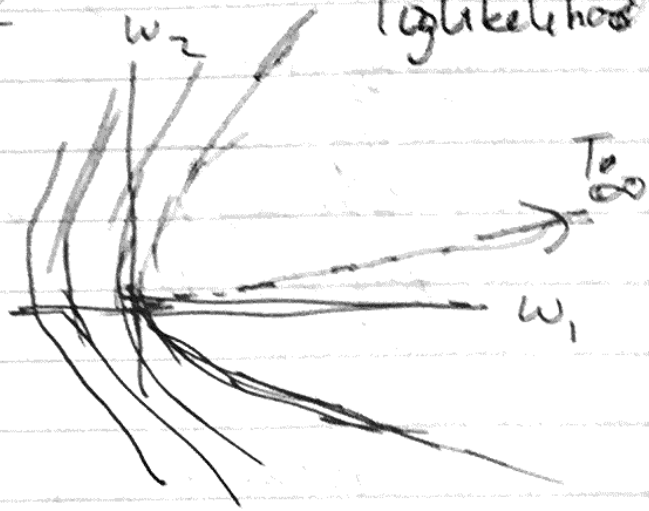
BIC depends on data size. Since $\frac{\log N}{2}$ is often greater than 1, BIC penalizes complex models more severely.

[Many prefer writing L
$kD - 2\log p(D|\theta^*)$ and minimize it.
$k = \log N$ for BIC and $k = 2$ for AIC]

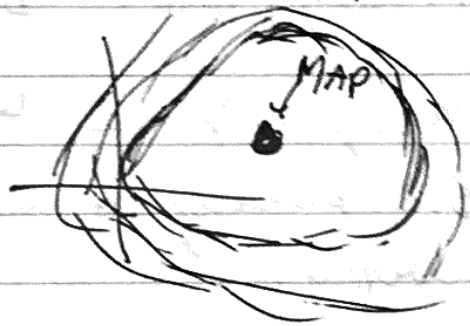
Back to the example of perfectly separable case. If data comes really

(laplace)
from $p(x)$ distr + logistic regression calling the class (we will do a two class example), we will see this phenomenon only when N is not too large

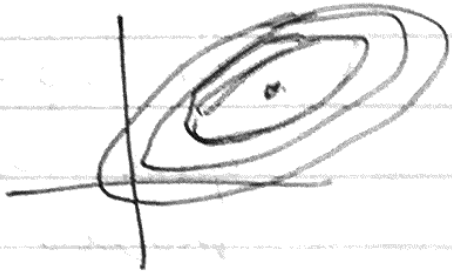




log Posterior (with quadratic penalty)



Laplace approximation



Approximating posterior predictive

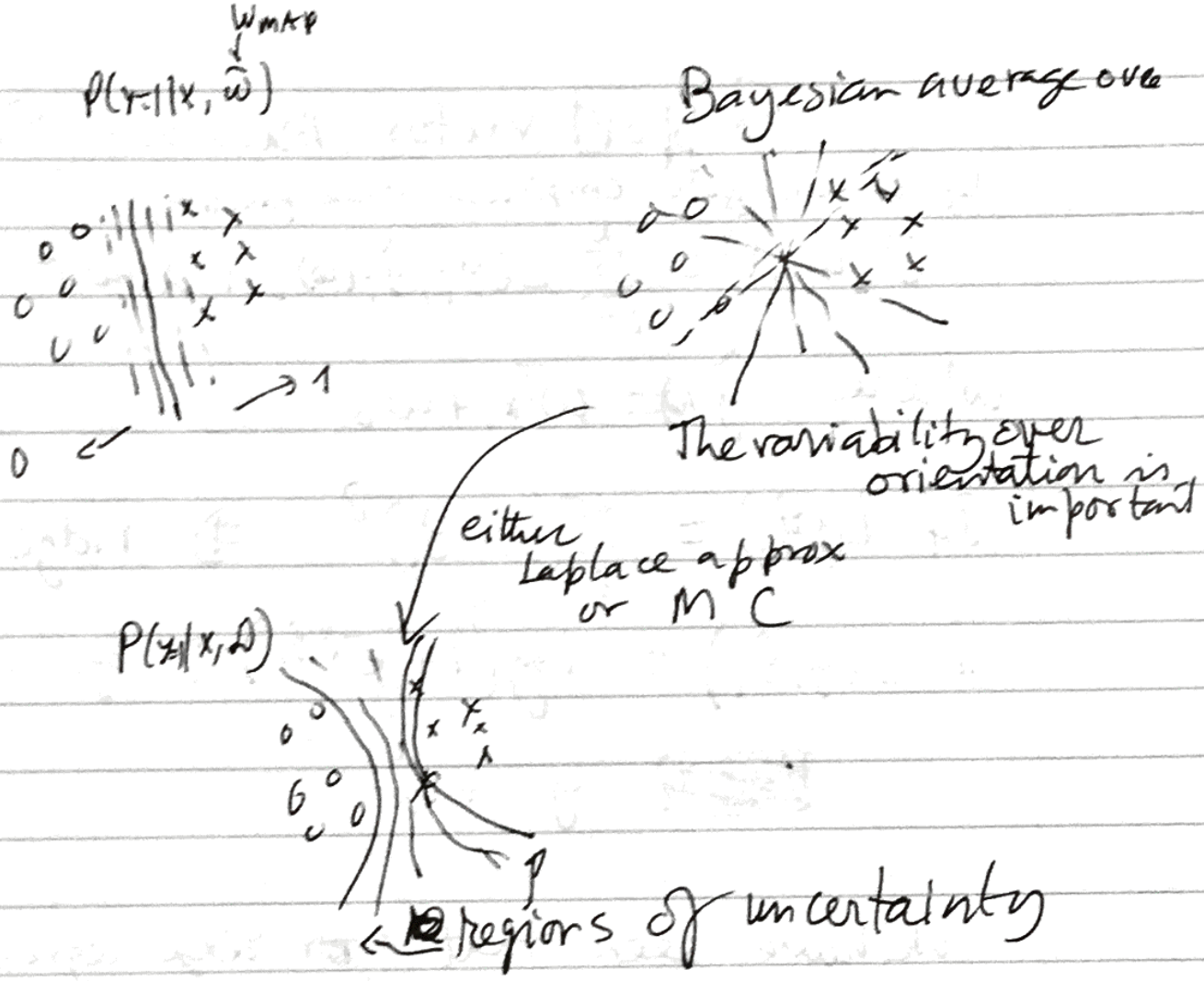
$$p(y|x,\mathcal{D}) = \int p(y|x,w)\, p(w|\mathcal{D})\, dw \simeq p(y|x,\hat{w})$$

$\hat{w}$: w_{MAP} or $E[w]$

Alternative: Sample w from the posterior

Monte Carlo (MC)

$$p(y=1|x,\mathcal{D}) \simeq \frac{1}{S}\sum_{s=1}^{S} \text{sigm}((w^s)^T x)$$



Online logistic regression and perceptrons

For optimizing $f(\theta) = \frac{1}{N}\sum_{i=1}^{N} f(\theta, z_i)$ where $-\log p(y_i|x_i,\theta)$ and $z_i = (x_i, y_i)$

one could take an online approach: change θ in response to new data.

Grad descent $\theta_k = \theta_{k-1} - \eta_k g_k$

for logistic regr: this means

$$w_k = w_{k-1} - \eta_k (\mu_i - y_i) x_i \qquad y_i \in \{0, 1\}$$

If we consider using $y_i \in \{1, -1\}$, and use $\hat{y}_i = \hat{y}_i - 1 \simeq \pm 1$ and $\eta_k = \frac{1}{2}\eta_k$

$$w_k = w_{k-1} + \eta_k y_k x_i \quad \text{when } y_k \neq \hat{y}_k$$
$$= w_{k-1} \quad \text{otherwise}$$

Perceptron algorithm!

Support vector Machines (SVM's)

L_2 regularized empirical risk function

$$J(\omega, \lambda) = \sum_{i=1}^{N} L(y_i, \hat{y}_i(\omega)) + \lambda \|\omega\|^2$$

where $\hat{y}_i(\omega) = \omega^T x_i + \omega_0$

If $L(y, \hat{y}_i) = \sum_i (y_i - \hat{y}_i)^2 \Rightarrow$ ridge regression

" $L(y, \hat{y}_i) = \log(1 + e^{-y\hat{y}_i}) \Rightarrow$ logistic regression

[redacted] $y = +1, -1$

We have seen that for ridge regression

$$\hat{\omega} = (X^T X + \lambda \mathbb{1}_D)^{-1} X^T y$$ [redacted]

$$\hat{y} = \hat{\omega}^T x$$ [redacted] $= y^T X (X^T X + \lambda \mathbb{1}_D)^{-1} x$

$= y^T (X X^T + \lambda \mathbb{1}_N)^{-1} X x$

This could be re expressed as $\hat{y} = \sum_i \alpha_i K(x_i, x)$

$\alpha = A y$ [redacted] $K(x, x') =$ [redacted] $x^T x'$ with $A_{N \times N} = (X X^T + \lambda I_N)^{-1}$

Off course, here all x's and y's contribute

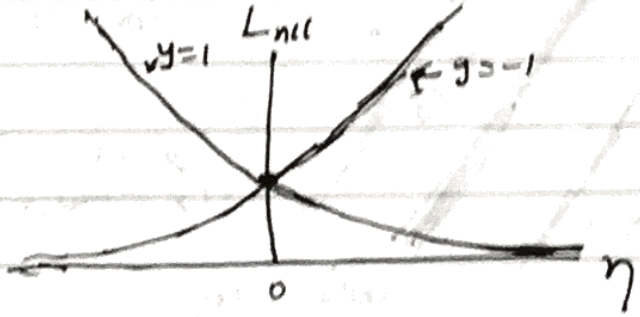
We will soon generalize the Kernel structure.

Coming off discussions of logistic regression
Let us first take up support vector classifiers

Logistic regression: $\eta = f(x) = \omega_0 + \omega^T x$

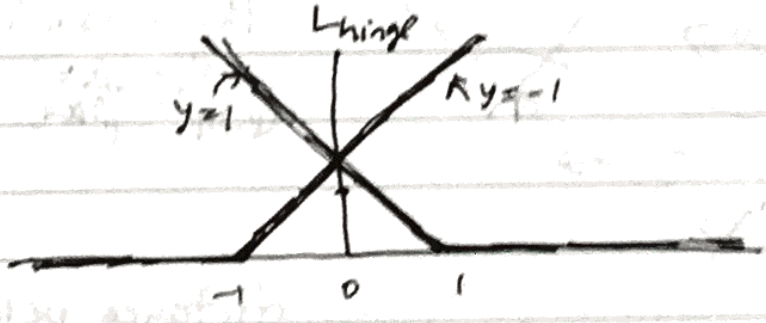
$$L_{nll}(y, \eta) = -\log p(y|x,\omega) = \log(1 + e^{-y\eta})$$

$y \in \{-1, +1\}$



Replace this loss function by hinge loss

$$L_{hinge}(y, \eta) = \max(0, 1 - y\eta) = (1 - y\eta)_+$$



Looks like door hinge

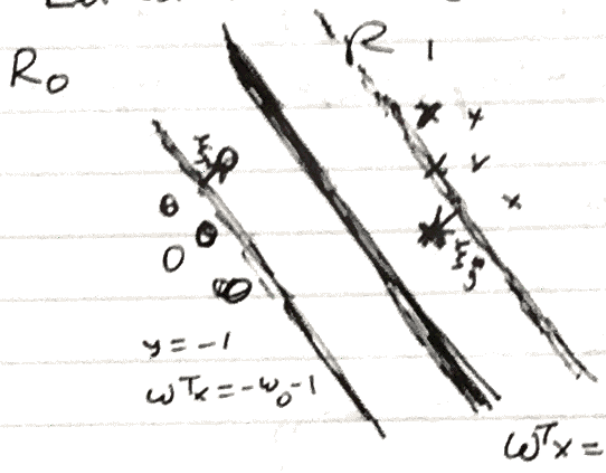
$$\min_{\omega_0, \omega} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}(1 - y_i f(x_i))_+$$

Introducing slack variables ξ_i, one could rewrite this optimization problem as

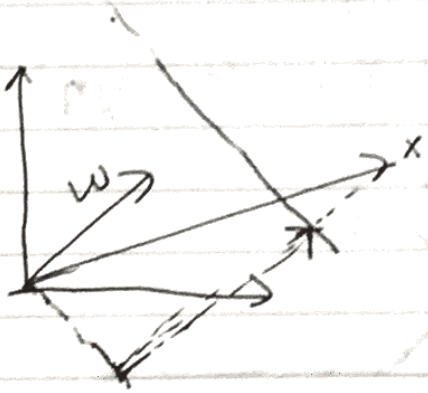
$$\min_{\omega_0, \omega, \xi} \frac{1}{2}\|\omega\|^2 + C\sum_i \xi_i \quad \text{s.t. } \xi_i \geq 0 \text{ and } y_i(\omega_0 + \omega^T x_i) \geq 1 - \xi_i \text{ for } i = 1, \ldots, N$$

The points are classified by $f(x) \gtrless 0$.
$f(x) = \omega_0 + \omega^T x$ is the discriminant function.

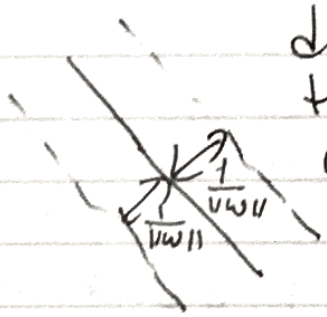
Let consider the geometry of relevant hyperplane



In addition to $f(x) = 0$
Consider $y f(x) = 1$
for $y = \pm 1$



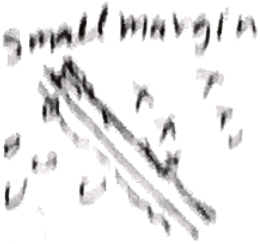
distance from the decision boundary
$\left(\frac{\omega^T}{\|\omega\|}\right) x - \left(\frac{\omega_0}{\|\omega\|}\right)$



distance between the decision boundary and the hyperplanes $y f(x) = 1$ is $\frac{1}{\|\omega\|}$

If the classes are linearly separable and we have a large C we want all $\xi_i = 0$

Not



we want $\frac{1}{\|w\|}$ to be as large s.t.

$y_i(w_0 + w^T x_i) \geq 1$ for all i

That is equivalent to

$$\min \frac{1}{2}\|w\|^2, \quad \text{s.t. } y_i(w_0 + w^T x_i) \geq 1$$

for $i = 1, \dots, N$

In cases where perfect separation is not possible, we have the soft margin version

$$\min_{w_0, w, \xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \quad \text{s.t. } y_i(w_0 + w^T x_i) \geq 1 - \xi_i$$

For the sake of simplicity let us discuss perfectly separable case.

First $\min_w \frac{1}{2}\|w\|^2$ s.t. $y_i(w_0 + w^T x_i) \geq$

Could be rewritten using Lagrange multipliers.

Consider

$$L(w_0, w, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i \left(y_i(w_0 + w^T x_i) - 1\right)$$

Turns out $\min_{w_0, w} \max_{\lambda_i \geq 0} \frac{1}{2}\|w\|^2 - \sum \lambda_i (y_i(w_0 + w^T x_i) - 1)$

The key difference is the lagrange multipliers are sign restricted.

Think of it this way; If any constraint is not satisfied, ie. $y_i(w_0 + w^T x_i) - 1 < 0$ We can send $\lambda_i \to +\infty$ and the obj. func. L would by ∞. When minimizing over w, we will reject such solutions.

If a constraint is strictly satisfied

$y_i(w_0 + w^T x_i) - 1 > 0$

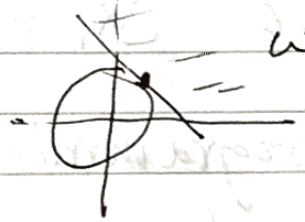
max over λ_i, sends $\lambda_i = 0$

So inactive constraints have corresponding $\lambda_i = 0$

What if $y_i(w_0 + w^T x_i) - 1 = 0$?

The margin case is interesting

Now λ_i is determined by w variation being zero condition. It would usually be nonzero



These x_i's are the support vectors.

Turns out that in this case, we can switch the order of min & max

$$\max_{\lambda_i \geq 0} \min_{w_0, w} \frac{1}{2}\|w\|^2 - \sum \lambda_i \left(y_i \left((w_0 + w^T x_i) - 1 \right) \right)$$

Since w_0 & w are [redacted] unrestricted, we could just differentiate.

$$w = \sum_i \lambda_i y_i x_i = \sum_i \alpha_i x_i$$

$$\boxed{\alpha_i \hat{=} \lambda_i y_i}$$

$$0 = \sum_i \lambda_i y_i * = \sum \alpha_i$$

$$\tilde{L}(\alpha) = -\frac{1}{2} \sum_{ij} \alpha_i x_i^T x_j \alpha_j + \sum_i \alpha_i y_i$$

$\alpha_i y_i \geq 0$ for all i

and $\sum \alpha_i = 0$

$$\min_{[redacted]\ \alpha} \frac{1}{2}\sum_{ij} \alpha_i x_i^T x_j \alpha_j - \sum_i \alpha_i y_i$$

$$\text{s.t.} \quad \alpha_i y_i \geq 0 \quad \sum_i \alpha_i = 0$$

Is a quadratic programming problem dual to the orginal one.

[redacted]

The solution $\hat{\omega} = \sum_i \alpha_i x_i$

↑ (primal) ↑ (dual)

$\hat{\omega}_0$ determined by picking an x_i s.t. $\alpha_i y_i \neq 0$ or $y_i(\hat{\omega}_0 + \hat{\omega}^T x_i) = 1 \Rightarrow \hat{\omega}_0 = y_i - \hat{\omega}^T x_i$

$$\hat{y}(x) = \text{sgn}(\hat{\omega}_0 + \hat{\omega}^T x [redacted])$$

Remember only certain α_i's are nonzero. [redacted] Turns out that when only a small fraction of training data becomes support vectors, the model has good generalization properties.

[redacted] How do we generalize this linear maximum margin classifer to problems which require nonlinear decision surfaces.

The trick is to realize that we could replace $x_i^T x_j$ by a general positive definite kernel $K(x_i, x_j)$.

$$\min_{\alpha} \frac{1}{2} \sum_{ij} \alpha_i K(x_i, x_j) \alpha_j - \sum_i \alpha_i$$

$$\text{s.t. } \alpha_i y_i \geq 0 \quad \sum_i \alpha_i y_i = 0$$

$$\hat{f}(x) = \text{sgn}\left(\hat{\omega}_0 + \sum_i \alpha_i K(x_i, x) \right)$$

With $\hat{\omega}_0$ determined to on a support vector.

For example: one could use kernels like

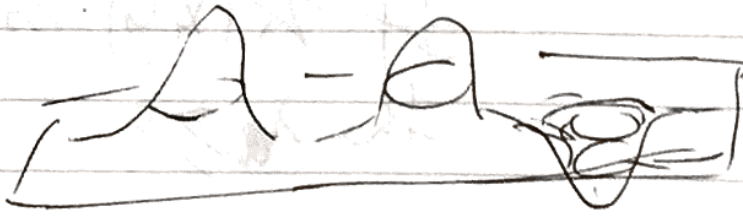
$$K(x, y) = (x \cdot y + 1)^p$$

$$K(x, y) = e^{-\|(x-y)\|^2 / 2\sigma^2} \rightarrow \text{Gaussian radial basis functions (RBF)}$$

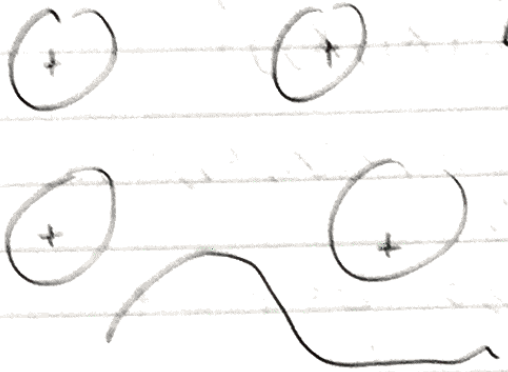
$$K(x, y) = \tanh(\kappa x \cdot y - \delta)$$

Decision Surface RBF

$$\sum_i \alpha_i e^{-\|(x_i - x)\|^2 / 2\sigma^2}$$



Level sets of RBF kernel could be as complex as possible with enough data.



With a small enough σ^2, it can approximate 'any' decision boundary.

Quite interestingly using $K(x,y)$ is equivalent to mapping x space to an infinite dimensional feature space by a nonlinear mapping. Think of $K(x,y)$ as an infinite dimensional operator

$$\int K(x,y)\,\Phi(y)\,dy = \Psi(x)$$

If K is symmetric we have infinitely many eigenvalues $\psi_a(x)$ with

$$\int K(x,y)\,\psi_a(y)\,dy = \lambda_a \psi_a(y)$$

and

$$K(x,y) = \sum_a \lambda_a \psi_a(x)\,\psi_a(y)$$

If $K(x,y)$ is positive semi definite, $\int K(x,y)\,g(x)\,g(y)\,dx\,dy \geq 0$ the $\lambda_a \geq 0$.

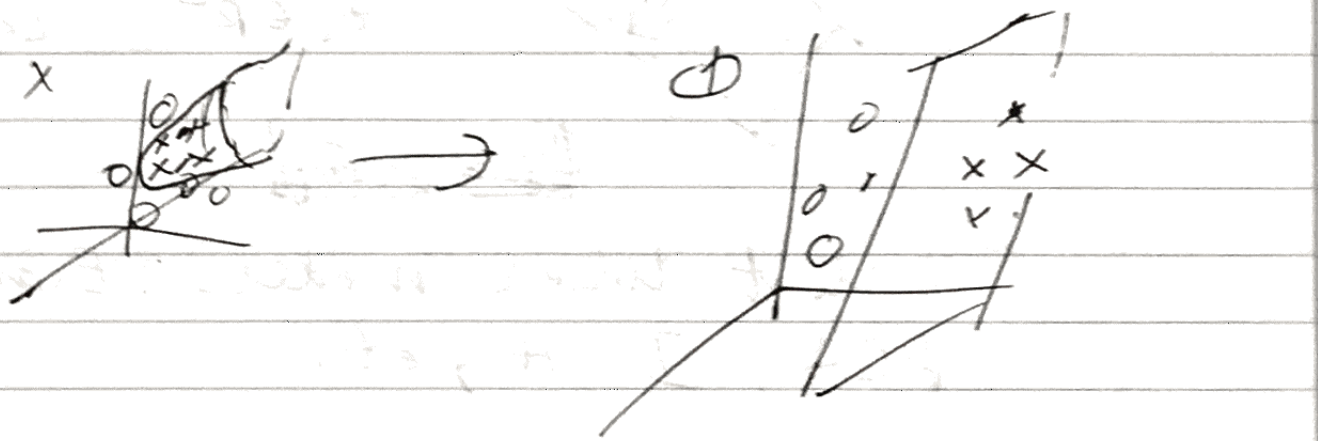
Define $\sqrt{\lambda_i} \cdot u_i = \varphi_i$

$$K(x,y) = \sum_a \varphi_a(x)\,\varphi_a(y)$$

So $$K(x_i, x_j) = \sum_a \varphi_a(x_i)\,\varphi_a(x_j)$$

This mean $K(x_i, x_j) = \underline{\Phi}(x_i)^T \underline{\Phi}(x_j)$

where $\underline{\Phi}(x) = (\varphi_1(x), \varphi_2(x), \varphi_3(x), \ldots)$



Similar idea's could be used for regression as well: $L_\varepsilon(y,\hat{y}) \triangleq \begin{cases} 0 & \text{if } |y-\hat{y}| \le \varepsilon \\ |y-\hat{y}| - \varepsilon & \text{otherwise} \end{cases}$

$\hat{y} = w_0 + w^T x$, $\min_{w,w_0} L_\varepsilon(y,\hat{y}) + \frac{1}{2}\|w\|^2$

As an aside, exponential of a hinge loss could be thought of as a mixture of many gaussians with different scales, allowing a probability interpretation of SVM's.

$$e^{-2a_+} = \int_0^\infty e^{-\frac{1}{2}\frac{(a+\lambda)^2}{\lambda}} \frac{d\lambda}{\sqrt{\pi\lambda}}$$

Trick $\int e^{-\frac{1}{2}\left(\frac{a}{\sqrt{\lambda}} + \sqrt{\lambda}\right)^2} d\sqrt{\lambda} = e^{-2a} \int e^{-\frac{1}{2}\left(\frac{a}{\sqrt{\lambda}} - \sqrt{\lambda}\right)^2} d\sqrt{\lambda}$