Kernel Logistic Regression

1.Soft-Margin SVM as Regularized Model

最早有Hard-Margin Primal,然后推导出Hard-Margin Dual形式。
后来，为了允许有错误点存在（或者noise），也为了避免模型太过复杂化，造成过拟合，建立了Soft-Margin Primal的数学表达式，并引入了新的参数C作为权衡因子，然后也推导了其Soft-Margin Dual形式。

Hard-Margin Primal

$$\min_{b,\mathbf{w}} \ \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{s.t.} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1$$

Soft-Margin Primal

$$\min_{b,\mathbf{w},\boldsymbol{\xi}} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$
$$\text{s.t.} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 - \xi_n, \xi_n \geq 0$$

Hard-Margin Dual

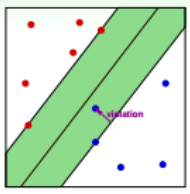
$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}$$
$$\text{s.t.} \quad \mathbf{y}^T\boldsymbol{\alpha} = 0$$
$$0 \leq \alpha_n$$

Soft-Margin Dual

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}$$
$$\text{s.t.} \quad \mathbf{y}^T\boldsymbol{\alpha} = 0$$
$$0 \leq \alpha_n \leq C$$

- record '**margin violation**' by ξ_n
- penalize with **margin violation**

$$\min_{b,\mathbf{w},\boldsymbol{\xi}} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\cdot\sum_{n=1}^{N}\xi_n$$
$$\text{s.t.} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 - \xi_n \text{ and } \xi_n \geq 0 \text{ for all } n$$



on any $(b, \mathbf{w})$, ξ_n = **margin violation** $= \max(1 - y_n(\mathbf{w}^T\mathbf{z}_n + b), 0)$

- $(\mathbf{x}_n, y_n)$ violating margin: $\xi_n = 1 - y_n(\mathbf{w}^T\mathbf{z}_n + b)$
- $(\mathbf{x}_n, y_n)$ not violating margin: $\xi_n = 0$

'unconstrained' form of soft-margin SVM:

$$\min_{b,\mathbf{w}} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\max(1 - y_n(\mathbf{w}^T\mathbf{z}_n + b), 0)$$

ξ_n描述的是点(x_n, y_n) 距离$y_n(w^T z_n + b) = 1$的边界有多远。
第一种情况是violating margin，即不满足$y_n(w^T z_n + b) \geq 1$。那么ξ_n可表示为：$\xi_n = 1 - y_n(w^T z_n + b) > 0$。
第二种情况是not violating margin，即点(x_n, y_n) 在边界之外，满足$y_n(w^T z_n + b) \geq 1$的条件，此时$\xi_n = 0$。

我们可以将两种情况整合到一个表达式中，对任意点：

$$\xi_n = max(1 - y_n(w^T z_n + b), 0)$$

上式表明，如果有voilating margin，则$1 - y_n(w^T z_n + b) > 0$，$\xi_n = 1 - y_n(w^T z_n + b)$
如果not violating margin，则$1 - y_n(w^T z_n + b) < 0$，$\xi_n = 0$。

整合之后，我们可以把Soft-Margin SVM的最小化问题写成如下形式：

$$\frac{1}{2}w^T w + C\sum_{n=1}^{N} max(1 - y_n(w^T z_n + b), 0)$$

经过这种转换之后，表征犯错误值大小的变量ξ_n就被消去了，转而由一个max操作代替。

	minimize	constraint
regularization by constraint	E_{in}	$\mathbf{w}^T\mathbf{w} \le C$
hard-margin SVM	$\mathbf{w}^T\mathbf{w}$	$E_{in} = 0$ [and more]
L2 regularization	$\frac{\lambda}{N}\mathbf{w}^T\mathbf{w} + E_{in}$	
soft-margin SVM	$\frac{1}{2}\mathbf{w}^T\mathbf{w} + CN\widehat{E_{in}}$	

L2 Regularization中的λ和Soft-Margin SVM中的C也是相互对应的，λ越大，w会越小，Regularization的程度就越大；C越小，$\hat{E_{in}}$会越大，相应的margin就越大。
所以说增大C，或者减小λ，效果是一致的，Large-Margin等同于Regularization，都起到了防止过拟合的作用。

2.SVM versus Logistic Regression

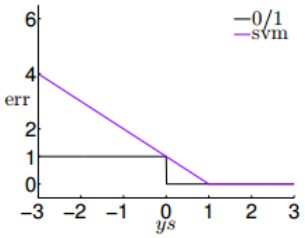
我们已经把Soft-Margin SVM转换成无条件的形式：

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\max(1 - y_n(\mathbf{w}^T\mathbf{z}_n + b), 0)$$

$max(1 - y_n(w^T z_n + b), 0)$倍设置为$\hat{err}$

linear score $s = \mathbf{w}^T\mathbf{z}_n + b$

- $\text{err}_{0/1}(s, y) = [\![ys \le 0]\!]$
- $\widehat{\text{err}}_{\text{SVM}}(s, y) = \max(1 - ys, 0)$: upper bound of $\text{err}_{0/1}$ —often called **hinge error measure**



$\widehat{\text{err}}_{\text{SVM}}$: **algorithmic error measure** by **convex upper bound** of $\text{err}_{0/1}$

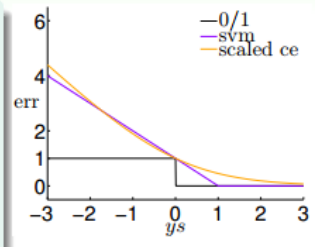
对于$err_{0/1}$，它的linear score $s = w^T z_n + b$
当$ys \ge 0$时，$err_{0/1} = 0$
当$ys < 0$时，$err_{0/1} = 1$，呈阶梯状。

对于$\hat{err}$，当$ys \ge 0$时，$err_{0/1} = 0$
当$ys < 0$时，$err_{0/1} = 1 - ys$，呈折线状。

$\hat{err}_{svm}$始终在$err_{0/1}$的上面，则$\hat{err}_{svm}$可作为$err0/1$的上界。
所以，可以使用$\hat{err}_{svm}$来代替$err0/1$，解决二元线性分类问题，而且$\hat{err}_{svm}$是一个凸函数，使它在最佳化问题中有更好的性质。

linear score $s = \mathbf{w}^T\mathbf{z}_n + b$

- $\text{err}_{0/1}(s, y) = [\![ys \le 0]\!]$
- $\widehat{\text{err}}_{\text{SVM}}(s, y) = \max(1 - ys, 0)$: upper bound of $\text{err}_{0/1}$
- $\text{err}_{\text{SCE}}(s, y) = \log_2(1 + \exp(-ys))$: another upper bound of $\text{err}_{0/1}$ used in **logistic regression**



$-\infty$	$\longleftarrow$	ys	$\longrightarrow$	$+\infty$
$\approx -ys$		$\widehat{\text{err}}_{\text{SVM}}(s, y)$		$= 0$
$\approx -ys$		$(\ln 2) \cdot \text{err}_{\text{SCE}}(s, y)$		≈ 0

逻辑回归中，$err_{sce} = log_2(1 + exp(-ys))$，当ys=0时，$err_{sce} = 1$。

err_{sce}也是$err_{0/1}$的上界，而err_{sce}与$\hat{err}_{svm}$也是比较相近的。
因为当ys趋向正无穷大的时候，err_{sce}和$\hat{err}_{svm}$都趋向于零；
当ys趋向负无穷大的时候，err_{sce}和$\hat{err}_{svm}$都趋向于正无穷大。
可以把SVM看成是L2-regularized logistic regression。

PLA	soft-margin SVM	regularized logistic regression for classification
minimize $\text{err}_{0/1}$ specially • pros: efficient if lin. separable • cons: works only if lin. separable, otherwise needing pocket	minimize regularized $\widehat{\text{err}}_{\text{SVM}}$ by QP • pros: 'easy' optimization & theoretical guarantee • cons: loose bound of $\text{err}_{0/1}$ for very negative *ys*	minimize regularized err_{SCE} by GD/SGD/... • pros: 'easy' optimization & regularization guard • cons: loose bound of $\text{err}_{0/1}$ for very negative *ys*

PLA是相对简单的一个模型，对应的是$err_{0/1}$
通过不断修正错误的点来获得最佳分类线
优点是简单快速
缺点是只对线性可分的情况有用，线性不可分的情况需要用到pocket算法。

Logistic Regression对应的是err_{sce}，通常使用GD/SGD算法求解最佳分类线。
优点是凸函数err_{sce}便于最优化求解，而且有regularization作为避免过拟合的保证
缺点是err_{sce}作为$err_{0/1}$的上界，当ys很小（负值）时，上界变得更宽松，不利于最优化求解。

Soft-Margin SVM对应的是$\hat{e}rsvm$，通常使用QP求解最佳分类线。
优点和Logistic Regression一样，凸优化问题计算简单而且分类线比较"粗壮"一些
缺点也和Logistic Regression一样，当ys很小（负值）时，上界变得过于宽松。

Logistic Regression和Soft-Margin SVM都是在最佳化$err0/1$的上界而已。

3.SVM for Soft Binary Classification

第一种简单的方法是先得到SVM的解(b_{svm}, w_{svm})，然后直接代入到logistic regression中，得到$g(x) = \theta(w_{svm}^T x + b_{svm})$。
这种方法直接使用了SVM和logistic regression的相似性，一般情况下表现还不错。
但是，这种形式过于简单，与logistic regression的关联不大，没有使用到logistic regression中好的性质和方法。

第二种简单的方法是同样先得到SVM的解(b_{svm}, w_{svm})，然后把(b_{svm}, w_{svm})作为logistic regression的初始值，再进行迭代训练修正，速度比较快
最后，将得到的b和w代入到g(x)中。
但并没有比直接使用logistic regression快捷多少。

Naïve Idea 1	Naïve Idea 2
1. run SVM and get $(b_{\text{SVM}}, \mathbf{w}_{\text{SVM}})$ 2. return $g(\mathbf{x}) = \theta(\mathbf{w}_{\text{SVM}}^T \mathbf{x} + b_{\text{SVM}})$ • 'direct' use of similarity —works reasonably well • no LogReg flavor	1. run SVM and get $(b_{\text{SVM}}, \mathbf{w}_{\text{SVM}})$ 2. run LogReg with $(b_{\text{SVM}}, \mathbf{w}_{\text{SVM}})$ as $\mathbf{w}_0$ 3. return LogReg solution as $g(\mathbf{x})$ • not really 'easier' than original LogReg • SVM flavor (kernel?) lost

构造一个融合两者优势的模型,我们额外增加了放缩因子A和平移因子B
首先利用SVM的解(b_{svm}, w_{svm})来构造这个模型，放缩因子A和平移因子B是待定系数。
然后再用通用的logistic regression优化算法，通过迭代优化，得到最终的A和B。

一般来说，如果(b_{svm}, w_{svm})较为合理的话，满足A>0且$B \approx 0$。

$$g(\mathbf{x}) = \theta(A \cdot (\mathbf{w}_{SVM}^T \mathbf{\Phi}(\mathbf{x}) + b_{SVM}) + B)$$

- **SVM flavor**: fix hyperplane direction by $\mathbf{w}_{SVM}$—**kernel** applies
- **LogReg flavor**: fine-tune hyperplane to match maximum likelihood by scaling (A) and shifting (B)
  - often $A > 0$ if $\mathbf{w}_{SVM}$ reasonably good
  - often $B \approx 0$ if b_{SVM} reasonably good

得到了新的logistic regression:

new LogReg Problem:

$$\min_{A,B} \frac{1}{N} \sum_{n=1}^{N} \log \left(1 + \exp\left(-y_n \left(A \cdot (\underbrace{\mathbf{w}_{SVM}^T \mathbf{\Phi}(\mathbf{x}_n) + b_{SVM}}_{\Phi_{SVM}(\mathbf{x}_n)}) + B\right)\right)\right)$$

其中的(b_{svm}, w_{svm})已经在SVM中解出来了，实际上的未知参数只有A和B两个
这种Probabilistic SVM的做法分为三个步骤：

Platt's Model of Probabilistic SVM for Soft Binary Classification

1. run **SVM** on $\mathcal{D}$ to get $(b_{SVM}, \mathbf{w}_{SVM})$ [or the equivalent α], and transform $\mathcal{D}$ to $\mathbf{z}'_n = \mathbf{w}_{SVM}^T \mathbf{\Phi}(\mathbf{x}_n) + b_{SVM}$
   —actual model performs this step in a more complicated manner
2. run **LogReg** on $\{(\mathbf{z}'_n, y_n)\}_{n=1}^N$ to get (A, B)
   —actual model adds some special regularization here
3. return $g(\mathbf{x}) = \theta(A \cdot (\mathbf{w}_{SVM}^T \mathbf{\Phi}(\mathbf{x}) + b_{SVM}) + B)$

这种soft binary classifier方法得到的结果跟直接使用SVM classifier得到的结果可能不一样，这是因为我们引入了系数A和B
一般来说，soft binary classifier效果更好
logistic regression的解法，可以选择GD、SGD等等。

4.Kernel Logistic Regression

SVM	PLA	LogReg by SGD
$\mathbf{w}_{SVM} = \sum_{n=1}^{N} (\alpha_n y_n) \mathbf{z}_n$	$\mathbf{w}_{PLA} = \sum_{n=1}^{N} (\alpha_n y_n) \mathbf{z}_n$	$\mathbf{w}_{LOGREG} = \sum_{n=1}^{N} (\alpha_n y_n) \mathbf{z}_n$
α_n from dual solutions	α_n by # mistake corrections	α_n by total SGD moves

对于L2-regularized linear model，如果它的最小化问题形式为如下的话，那么最优解$w_* = \sum_{n=1}^{N} \beta_n z_n$。

claim: for any L2-regularized linear model

$$\min_{\mathbf{w}} \frac{\lambda}{N} \mathbf{w}^T \mathbf{w} + \frac{1}{N} \sum_{n=1}^{N} \text{err}(y_n, \mathbf{w}^T \mathbf{z}_n)$$

optimal $\mathbf{w}_* = \sum_{n=1}^{N} \beta_n \mathbf{z}_n$.

假如最优解$w_= w_{||} + w_\perp$。
$w_{||}$和$w_\perp$分别是平行z空间和垂直z空间的部分。
我们需要证明的是$w_\perp = 0$。
利用反证法，假如$w_\perp \neq 0$，考虑w_*与$w_{||}$的比较。
第一步先比较最小化问题的第二项：$err(y, w_*^T z_n) = err(y_n, (w_{||} + w_\perp)^T z_n = err(y_n, w_{||}^T z_n)$，即第二项是相等的。
然后第二步比较第一项：$w_*^T w_= w_{||}^T w_{||} + 2 w_{||}^T w_\perp + w_\perp^T w_\perp > w_{||}^T w_{||}$，即$w_*$对应的L2-regularized linear model值要比$w_{||}$大，

这就说明w_*并不是最优解，从而证明$w_\perp$必然等于零，即$w_* = \sum_{n=1}^{N} \beta_n z_n$一定成立，$w_*$一定可以写成z的线性组合形式。

- let optimal $\mathbf{w}_* = \mathbf{w}_\parallel + \mathbf{w}_\perp$, where $\mathbf{w}_\parallel \in \text{span}(\mathbf{z}_n)$ & $\mathbf{w}_\perp \perp \text{span}(\mathbf{z}_n)$
  —want $\mathbf{w}_\perp = \mathbf{0}$
- what if **not?** Consider $\mathbf{w}_\parallel$
  - of same err as $\mathbf{w}_*$: $\text{err}(y_n, \mathbf{w}_*^T \mathbf{z}_n) = \text{err}(y_n, (\mathbf{w}_\parallel + \mathbf{w}_\perp)^T \mathbf{z}_n)$
  - of smaller regularizer as $\mathbf{w}_*$:
    $\mathbf{w}_*^T \mathbf{w}_* = \mathbf{w}_\parallel^T \mathbf{w}_\parallel + 2\mathbf{w}_\parallel^T \mathbf{w}_\perp + \mathbf{w}_\perp^T \mathbf{w}_\perp \; > \; \mathbf{w}_\parallel^T \mathbf{w}_\parallel$

  —$\mathbf{w}_\parallel$ 'more optimal' than $\mathbf{w}_*$ (contradiction!)

将$w_= \sum_{n=1}^{N} \beta_n z_n$代入到L2-regularized logistic regression最小化问题中，得到：

solving L2-regularized logistic regression

$$\min_{\mathbf{w}} \quad \frac{\lambda}{N} \mathbf{w}^T \mathbf{w} + \frac{1}{N} \sum_{n=1}^{N} \log\left(1 + \exp\left(-y_n \mathbf{w}^T \mathbf{z}_n\right)\right)$$

yields optimal solution $\mathbf{w}_* = \sum_{n=1}^{N} \beta_n \mathbf{z}_n$

从另外一个角度来看Kernel Logistic Regression（KLR）：

$$\min_{\beta} \frac{\lambda}{N} \sum_{n=1}^{N} \sum_{m=1}^{N} \beta_n \beta_m K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{N} \sum_{n=1}^{N} \log\left(1 + \exp\left(-y_n \sum_{m=1}^{N} \beta_m K(\mathbf{x}_m, \mathbf{x}_n)\right)\right)$$

上式中log项里的$\sum_{m=1}^{N} \beta_m K(x_m, x_n)$可以看成是变量$\beta$和$K(x_m, x_n)$的内积。
上式第一项中的$\sum_{n=1}^{N} \sum_{m=1}^{N} \beta_n \beta_m K(x_n, x_m)$可以看成是关于$\beta$的正则化项$\beta^T K \beta$。
所以，KLR是β的线性组合，其中包含了kernel内积项和kernel regularizer。这与SVM是相似的形式。

KLR中的β_n与SVM中的α_n是有区别的。SVM中的α_n大部分为零，SV的个数通常是比较少的；而KLR中的β_n通常都是非零值。