PRML（Pattern Recognition And Machine Learning）读书会

第七章 Sparse Kernel Machines

主讲人 网神

（新浪微博：@豆角茄子麻酱凉面）

QQ 群 177217565

读书会微信公众平台请扫描下面的二维码

网神(66707180) 18:59:22

大家好，今天一起交流下 PRML 第 7 章。第六章核函数里提到，有一类机器学习算法，不是对参数做点估计或求其分布，而是保留训练样本，在预测阶段，计算待预测样本跟训练样本的相似性来做预测，例如 KNN 方法。

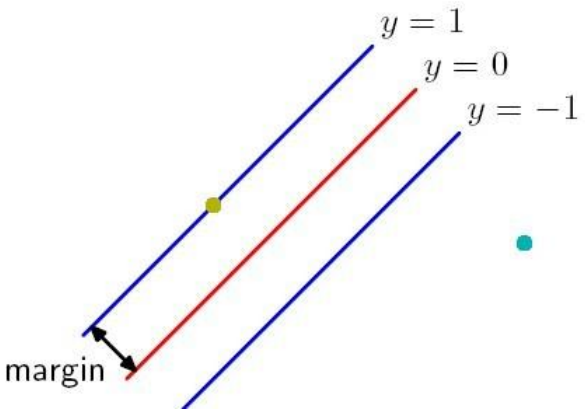
将线性模型转换成对偶形式，就可以利用核函数来计算相似性，同时避免了直接做高维度的向量内积运算。本章是稀疏向量机，同样基于核函数，用训练样本直接对新样本做预测，而且只使用了少量训练样本，所以具有稀疏性，叫 sparse kernel machine。

本章包括 SVM 和 RVM(revelance vector machine)两部分，首先讲 SVM，支持向量机。首先看 SVM 用于二元分类，并先假设两类数据是线性可分的。

二元分类线性模型可以用这个式子表示：$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b$。其中$\phi(\mathbf{x})$是基函数，这些都跟第三章和第四章是一样的。

两类数据线性可分，当$\mathbf{y(x_n) > 0}$时,分类结果是$\mathbf{t_n = +1}$;$\mathbf{y(x_n) < 0}$时,分类结果$\mathbf{t_n = -1}$;也就是对所有训练样本总是有$\mathbf{t_n y(x_n) > 0}$.要做的就是确定决策边界 y(x)=0

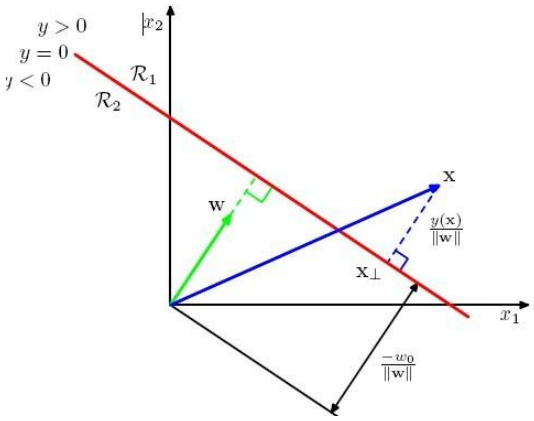
为了确定决策边界$\mathbf{y(x_n) = w^T \phi(x_n) + b}$，SVM 引入 margin 的概念。margin 定义为决策边界 y(x)到最近的样本的垂直距离。如下图所示：



SVM 的目标是寻找一个 margin 最大的决策边界。 我们来看如何确定目标函数：

首先给出一个样本点 x 到决策边界$\mathbf{w^T \phi(x) + b = 0}$的垂直距离公式是什么，先给出答案：|y(x)|/||w||

这个距离怎么来的，在第四章有具体介绍。看下图:



图例，我们看点 x 到 y=0 的距离 r 是多少：

$x_\perp$ 是 x 在 y=0 上的投影，因为 w 跟 y=0 是垂直的，所以 $\overrightarrow{x_\perp x}$ 跟 w 平行，$\overrightarrow{x_\perp x}=r\frac{w}{\|w\|}$。

r 是距离。根据向量相加的公式，有 $x=x_\perp+r\frac{w}{\|w\|}$。两边都乘上 w^T 并加上 b，得到

$y(x)=y(x_\perp)+r\|w\|$，因为 $y(x_\perp)=0$，所以 $r=\frac{y(x)}{\|w\|}$

上面我们得到了任意样本点 x 到 y(x)=0 的距离，要做的是最大化这个距离。

同时，要满足条件 $t_n y(x_n)>0$

所以目标函数是: $\arg\max_{w,b}\{\min_n[\frac{t_n(w^T\phi(x_n)+b)}{\|w\|}]\}$

求 w 和 b，使所有样本中，与 y=0 距离最小的距离 最大化，整个式子就是最小距离最大化
这个函数优化很复杂，需要做一个转换

可以看到，对 w 和 b 进行缩放，$w\rightarrow\kappa w$ and $b\rightarrow\kappa b$.距离 $\frac{t_n y(x_n)}{\|w\|}$ 并不会变化

根据这个属性，调整 w 和 b,使到决策面最近的点满足：$t_n(w^T\phi(x_n)+b)=1$

从而左右样本点都满足 $t_n(w^T\phi(x_n)+b)\geq 1$

这样，前面的目标函数可以变为：$\arg\max_{w,b}\frac{1}{\|w\|}$（即 $\arg\min_{w,b}\frac{1}{2}\|w\|^2$）

同时满足约束条件：$t_n(w^T\phi(x_n)+b)\geq 1$

这是一个不等式约束的二次规划问题，用拉格朗日乘子法来求解

构造如下的拉格朗日函数：$L(w,b,a)=\frac{1}{2}\|w\|^2-\sum_{n=1}^{N}a_n\{t_n(w^T\phi(x_n)+b)-1\}$

a_n 是拉格朗日乘子 ，这个函数分别对 w 和 b 求导，令导数等于 0，可以得到 w 和 b 的表达式：

$$w=\sum_{n=1}^{N}a_n t_n\phi(x_n)$$

$$0=\sum_{n=1}^{N}a_n t_n.$$

将 w 带入前面的拉格朗如函数 L(w,b,a)，就可以消去 w 和 b，变成 a 的函数 $\tilde{L}(a)$，这个函数是拉格朗日函数的对偶函数：

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

为什么要转换成对偶函数，主要是变形后可以借助核函数，来解决线性不可分的问题，尤其是基函数的维度特别高的情况。求解这个对偶函数，得到参数 a_n，就确定了分类模型

把 $\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n)$ 带入 $\mathrm{y}(\mathrm{x}) = \mathrm{w}^T \phi(\mathrm{x}) + \mathrm{b}$，就是用核函数表示的分类模型：

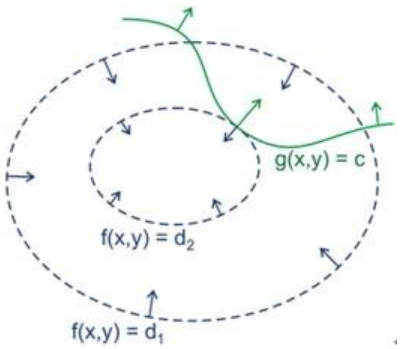
$$\mathrm{y}(\mathrm{x}) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + \mathrm{b}$$

这就是最终的分类模型，完全由训练样本 x_n，n=1...N 决定。

SVM 具有稀疏性，这里面对大部分训练样本，a_n 都等于 0，从而大部分样本在新样本预测时都不起作用。

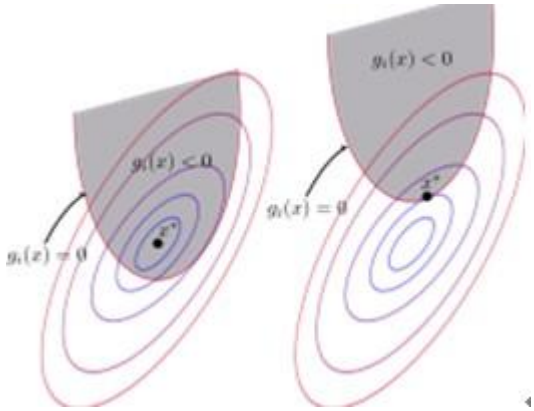
我们来看看为什么大部分训练样本，a_n 都等于 0。这主要是由 KKT 条件决定的。我们从直观上看下 KKT 条件是怎么回事：

KKT 是对拉格朗日乘子法的扩展，将其从约束为等式的情况扩展为约束为不等式的情况。所以先看下约束为等式的情况：例如求函数 $\mathrm{f}(\mathrm{x}_1, x_2)$ 的极大值，同时满足约束 $\mathrm{g}(\mathrm{x}_1, x_2)=0$，拉格朗日乘子法前面已经介绍，引入拉式乘子，构造拉式函数，然后求导，解除的值就是极值。这里从直观上看一下，为什么这个值就是满足条件的极值。设想取不同的 z 值，使 $\mathrm{f}(\mathrm{x}_1, x_2)=z$，就可以得到 f(x1,x2)的不同等高线，如图:



$\mathrm{g}(\mathrm{x}_1, x_2)=0$ 构成图中的曲线，图中标记的 g=c，对于这种情况，改成 g-c=0 就可以了.假设 g 与 f 的某些等高线相交，交点就是同时满足约束条件和目标函数的值，但不一定是极大值。。有两种相交形式，一种是穿过，一种是相切。因为穿过意味着在该条等高线内部还存在着其他等高线与 g 相交，新等高线与目标函数的交点的值更大。只有相切时，才可能取得最大值。因此，在极大值处，f 的梯度与 g 的梯度是平行的，因为梯度都垂直于 g 或 f 曲线，也就是存在 lamda，使得 $\nabla \mathrm{f} + \lambda \nabla \mathrm{g}=0$，这个式子正是拉格朗日函数 $\mathrm{L}(\mathrm{x}, \lambda)=\mathrm{f}(\mathrm{x})+\lambda\, \mathrm{g}(\mathrm{x})$ 对 x 求导的结果。

接下来看看约束条件为不等式的情况，例如约束为 $\mathrm{g}(\mathrm{x}) \geq 0$，先看个图：



图里的约束是 g<0，不影响解释 KKT 条件。不等式约束分两种情况，假设极值点是 x` ，当 $g(x`)>0$ 时，也就是图中左边那部分，此时该约束条件是 inactive 的，对于极值点的确定不起作用。因此拉格朗日函数 $L(\mathrm{x},\lambda)\equiv f(\mathrm{x})+\lambda\,\mathrm{g}(\mathrm{x})$ 中，lamda 等于 0，极值完全由 f 一个人确定，相当于 lamda 等于 0.当 $g(x`)=0$ 时，也就是图中右边部分，极值出现在 g 的边界处,这跟约束条件为等式时是一样的。

总之，对于约束条件为不等式的拉格朗日乘子法，总有 $\lambda g(\mathrm{x})=0$ ，不是 lamda 等于 0，就是 g=0 这个结论叫 KKT 条件，总结起来就是:

$$\mathrm{g}(\mathrm{x})\geq 0$$
$$\lambda\geq 0$$
$$\lambda\mathrm{g}(\mathrm{x})=0$$

再返回来看 SVM 的目标函数构造的拉格朗日函数：

$$L(\mathrm{w},\mathrm{b},\mathrm{a})=\frac{1}{2}\|\,\mathrm{w}\,\|^2-\sum_{n=1}^{N}a_n\{\mathrm{t}_n(\mathrm{w}^T\phi(\mathrm{x}_n)+\mathrm{b})-1\}$$

根据 KKT 条件，有 $\mathrm{a}_n\geq 0$ ， $t_n y(\mathrm{x}_n)-1\geq 0$ ， $a_n\{t_n y(\mathrm{x}_n)-1\}=0$ ，所以对于 $t_n y(\mathrm{x}_n)$ 大于 1 的那些样本点，其对应的 a_n 都等于 0。只有 $t_n y(\mathrm{x}_n)$ 等于 1 的那些样本点对保留下来，这些点就是支持向量。

这部分大家有什么意见和问题吗？

==========讨论==========

Fire(564122106) 20:01:56

他为什么要符合 KKT 条件啊

网神(66707180) 20:02:33

因为只有符合 KKT 条件，才能有解，否则拉格朗日函数没解，我的理解是这样的

Fire(564122106) 20:03:54

我上次看到一个版本说只有符合 KKT 条件　对偶解才和原始解才相同，不知道怎么解释。

kxkr<lxfkxkr@126.com> 20:04:18

貌似统计学习方法 附录里面 讲了这个

Wolf　<wuwja@foxmail.com> 20:04:19

an 为 0 为什么和 kkt 条件相关

kxkr<lxfkxkr@126.com> 20:04:20

不过忘记了 ，我上次看到一个版本说只有符合 KKT 条件，对偶解才和原始解相同。

YYKuaiXian(335015891) 20:04:40

Ng 的讲义就是用这种说法

苦瓜炒鸡蛋(852383636) 20:04:47

因为大部分的样本都不是 sv

Wolf <wuwja@foxmail.com> 20:05:05

如果两类正好分布在 margin 上，那么所有的点都是 sv

YYKuaiXian(335015891) 20:05:40

Under our above assumptions, there must exist w^*, α^*, β^* so that w^* is the solution to the primal problem, α^*, β^* are the solution to the dual problem, and moreover $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$. Moreover, w^*, α^* and β^* satisfy the **Karush-Kuhn-Tucker (KKT) conditions**, which are as follows:

$$\frac{\partial}{\partial w_i}\mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \ldots, n \tag{3}$$

$$\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \ldots, l \tag{4}$$

$$\alpha_i^* g_i(w^*) = 0, \quad i = 1, \ldots, k \tag{5}$$

$$g_i(w^*) \leq 0, \quad i = 1, \ldots, k \tag{6}$$

$$\alpha^* \geq 0, \quad i = 1, \ldots, k \tag{7}$$

Moreover, if some w^*, α^*, β^* satisfy the KKT conditions, then it is also a solution to the primal and dual problems.

Wolf <wuwja@foxmail.com> 20:05:54

只有符合 kkt 条件，primary 问题和 due 问题的解才是一样的，否则胖子里面的瘦子总比瘦子里面的胖子大。

kxkr<lxfkxkr@126.com> 20:07:03

这个比喻 好！

高老头(1316103319) 20:07:08

对偶问题和原问题是什么关系，一个问题怎么找到它的对偶问题？

Wolf <wuwja@foxmail.com> 20:07:19

所以 kkt 条件和 sv 为什么大部分为 0 没有直接关系，sv 为 0 个人觉得是分界面的性质决定的，分界面是一个低维流形。

Fire(564122106) 20:08:38

我也感觉 sv 是和样本数据性质有关的

Wolf <wuwja@foxmail.com> 20:09:26

比如在二维的时候，分界面是一个线性函数，导致 sv 比较少，当投影到高维空间，分界面变成了一个超平面，导致 sv 变多了，另外，很多样本变成 sv 也是 svm 慢的一个原因。

网神(66707180) 20:09:37

sv 本质上是 svm 选择的错误函数决定的，在正确一边分类边界以外的样本点，错误为 0，在边界以内或在错误一边，错误大于 0.

苦瓜炒鸡蛋(852383636) 20:11:04

sv 确定的超平面 而非是超平面确定的 sv

Wolf <wuwja@foxmail.com> 20:11:30

sv 确定的超平面 而非是超平面确定的 sv，一样的，hinge 为什么会导致稀疏？什么样的优化问题才有对

偶问题，我也在疑问。对于一些规划问题（线性规划，二次规划）可以将求最大值（最小值）的问题转化为求最小值最大值的问题。

网神(66707180) 20:12:24

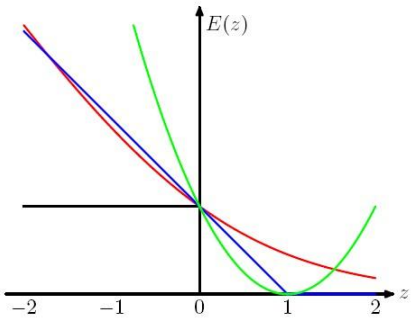
kkt 是从一个侧面解释稀疏，从另一个侧面，也就是错误函数是 hinge 函数，也可以得出稀疏的性质。svm 跟逻辑回归做对比， hinge 损失导致稀疏，我们先讲下这吧，svm 的错误函数可以这么写:

$$\sum_{n=1}^{N} E_{\mathrm{SV}}(y_n t_n) + \lambda \|\mathbf{w}\|^2$$

其中 $E_{\mathrm{SV}}(y_n t_n) = [1 - y_n t_n]_+$

where $[\cdot]_+$ denotes the positive part

这就是 hinge 错误函数，图形如图中的蓝色线



而逻辑回归的错误函数是：

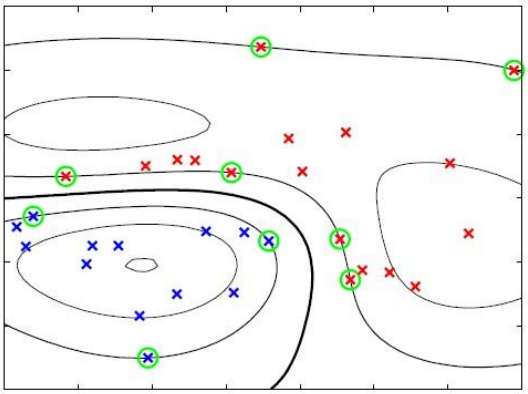
$$\sum_{n=1}^{N} E_{\mathrm{LR}}(y_n t_n) + \lambda \|\mathbf{w}\|^2.$$

$$E_{\mathrm{LR}}(yt) = \ln\left(1 + \exp(-yt)\right).$$

如图中的红色线，红色线跟蓝色线走势相近 ，区别是 hinge 函数在 $E_{\mathrm{SV}}(y_n t_n) = [1 - y_n t_n]_+$,图中 z>1 时，错误等于 0，也就是 yt>1 的那些点都不产生损失. 这个性质可以带来稀疏的解。

==========讨论结束==========

我接着讲了，后面还有挺多内容，刚才说的都是两类训练样本可以完全分开的情况，比如下面这个图，采用了高斯核函数的支持向量机，可以很清楚的看到决策边界，支持向量：



但实际中两类数据的分布会有重叠的情况，另外也有噪音的存在，导致两类训练数据如果一定要完全分开，泛化性能会很差。因此 svm 引入一些机制，允许训练时一些样本被误分类.我们要修改目标函数，允许样本点位于错误的一边，但会增加一个惩罚项，其大小随着数据点到边界的距离而增大这个惩罚项叫松弛变量, slack variables，记为 ξ_n ，并且大于等于 0.其中下标 n=1,..,N，也就是每个训练样本对应一个 ξ_n ，对于位于正确的 margin 边界上或以内的数据点，其松弛变量 $\xi_n=0$ ，其他样本点 $\xi_n=|t_n-y(x_n)|$

这样，如果样本点位于决策边界 y(x)=0 上, $\xi_n=1$

如果被错分，位于错误的一边, $\xi>1$ ，因此目标函数的限制条件由 $t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)\geqslant 1$ 修改为

$t_n y(\mathbf{x}_n)\geqslant 1-\xi_n$ ，目标函数修从最小化 $\frac{1}{2}\|\mathbf{w}\|^2$ 改为最小化 $C\sum_{n=1}^{N}\xi_n+\frac{1}{2}\|\mathbf{w}\|^2$ ，其中参数 C 用于控制松弛变量和 margin 之间的 trade-off，因为对于错分的点，有 $\xi>1$ ，所以 $\sum\xi_n$ 是错分样本数的一个上限 upper bound ，所以 C 相当于一个正则稀疏，控制着最小错分数和模型复杂度的 trade-off.
SVM 在实际使用中，需要调整的参数很少，C 是其中之一。
看这个目标函数，可以看到，C 越大，松弛变量就越倍惩罚，就会训练出越复杂的模型，来保证尽量少的样本被错分。当 C 趋于无穷时，每个样本点就会被模型正确分类。
我们现在求解这个新的目标函数，加上约束条件，拉格朗日函数如下：

$$L(\mathbf{w},b,\mathbf{a})=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{n=1}^{N}\xi_n-\sum_{n=1}^{N}a_n\{t_n y(\mathbf{x}_n)-1+\xi_n\}-\sum_{n=1}^{N}\mu_n\xi_n$$

其中 a_n 和 μ_n 是拉式乘子，分别对 w, b 和{ξ_n}求导，令导数等于 0，得到 w, b,ξ_n的表示，带入 L(w,b,a)，消去这些变量，得到以拉格朗日乘子为变量的对偶函数：

$$\widetilde{L}(\mathbf{a})=\sum_{n=1}^{N}a_n-\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_n a_m t_n t_m k(\mathbf{x}_n,\mathbf{x}_m)$$

新的对偶函数跟前面的对偶函数形式相同，只有约束条件有不同。 这就是正则化的 SVM。
接下来提一下对偶函数的解法，对偶函数都是二次函数，而且是凸函数，这是 svm 的优势，具有全局最优解，该二次规划问题的求解难度是参数 a_n 的数量很大，等于训练样本的数量 。书上回顾了一些方法，介绍不详细，主要思想是 chunking，我总结一下，总结的不一定准确:

1.去掉 a_n =0 对应的核函数矩阵的行和列，将二次优化问题划分成多个小的优化问题；

2.按固定大小划分成小的优化问题。
3.SVM 中最流行的是 SMO, sequentialminimal optimization。每次只考虑两个拉格朗日乘子.

SVM 中维度灾难问题：核函数相当于高维(甚至无线维)的特征空间的内积，避免了显示的高维空间运算，貌似是避免了维度过高引起的维度灾难问题。 但实际上并没有避免。书上举了个例子 ，看这个二维多项式核函数：

$$
\begin{aligned}
k(\mathbf{x},\mathbf{z}) &= \left(1+\mathbf{x}^{\mathrm{T}}\mathbf{z}\right)^{2} = (1+x_1z_1+x_2z_2)^2 \\
&= 1+2x_1z_1+2x_2z_2+x_1^2z_1^2+2x_1z_1x_2z_2+x_2^2z_2^2 \\
&= (1,\sqrt{2}x_1,\sqrt{2}x_2,x_1^2,\sqrt{2}x_1x_2,x_2^2)(1,\sqrt{2}z_1,\sqrt{2}z_2,z_1^2,\sqrt{2}z_1z_2,z_2^2)^{\mathrm{T}} \\
&= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{z}). \qquad (7.42)
\end{aligned}
$$

这个核函数表示一个六维空间的内积。 $\Phi(\mathrm{x})$ 是从输入空间到六维空间的映射.映射后，六个维度每个维度的值是由固定参数的，也就是映射后，六维特征是有固定的形式。因此，原二维数据 x 都被限制到了六维空间的一个 nonlinear manifold 中。这个 manifold 之外就没有数据.

网神(66707180) 20:48:59

大家有什么问题吗？

高老头(1316103319) 20:49:58

manifold 是什么意思？

网神(66707180) 20:50:26

我的理解是空间里一个特定的区域，原空间的数据，如果采样不够均匀，映射后的空间，仍然不会均匀，不会被打散到空间的各个角落，而只会聚集在某个区域。

接下来讲下 SVM 用于回归问题.

在线性回归中，一个正则化错误函数如下： $\frac{1}{2}\sum_{n=1}^{N}\{y_n-t_n\}^2+\frac{\lambda}{2}\|\mathbf{w}\|^2$

为了获得稀疏解，将前面的二次错误函数用 $\epsilon\text{-}insensitive$ 错误函数代替

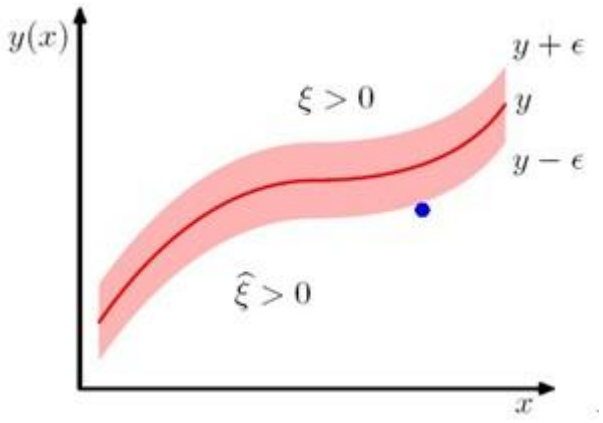
$$
E_\epsilon(y(\mathbf{x})-t)=\begin{cases}0, & \text{if } |y(\mathbf{x})-t|<\epsilon;\\ |y(\mathbf{x})-t|-\epsilon, & \text{otherwise}\end{cases}
$$

这个错误函数在 y(x)和 t 的差小于 ϵ 时等于 0.错误函数变为:

$$
C\sum_{n=1}^{N}E_\epsilon(y(\mathbf{x}_n)-t_n)+\frac{1}{2}\|\mathbf{w}\|^2
$$

我们再引入松弛变量，对每个样本，有两个松弛变量，分别对应 $\mathrm{t}_n > y(\mathrm{x}_n)+\varepsilon$ 和 $\mathrm{t}_n < y(\mathrm{x}_n)-\varepsilon$

如图:



没引入松弛变量前，样本值 t 预测正确的条件是 $y_n-\varepsilon\le t_n\le y_n+\varepsilon$

$$t_n \leqslant y(\mathbf{x}_n) + \epsilon + \xi_n$$
$$t_n \geqslant y(\mathbf{x}_n) - \epsilon - \widehat{\xi}_n$$

引入松弛变量后，变为:

错误函数变为：

$$C\sum_{n=1}^{N}(\xi_n + \widehat{\xi}_n) + \frac{1}{2}\|\mathbf{w}\|^2$$

加上约束条件 $\xi_n \geq 0$ 和 $\hat{\xi}_n \geq 0$，就可以写出拉格朗日函数

下面就跟前面的分类一样了.

关于统计学习理论，书上简单提了一下 PAC(probably approximately correct)和 VC 维，简单总结一下书上的内容：PAC 的目的是理解多大的数据集可以给出好的泛化性，以及研究损失的上限。PAC 里的一个关键概念是 VC 维，用于提供一个函数空间复杂度的度量，将 PAC 理论推广到了无限大的函数空间上。

==========讨论==========

Fire(564122106) 21:05:56

有哪位大神想过对 svm 提速的啊，svm 在非线性大数据的情况下，速度还是比较慢的啊

网神(66707180) 21:07:36

svm 分布式训练的方案研究过吗？

Fire(564122106) 21:09:29

没有，不过将来肯定要研究的！现在只是单机，现在有在单机的情况下，分布式进入内存的方案，有兴趣的可以看下：

Selective Block Minimization for Faster Convergence of Limited Memory Large_Scale Linear Models 这个有介绍，我共享下啊。

苦瓜炒鸡蛋(852383636) 21:11:05

韩家炜的一个学生 提出了一个 仿照层次聚类的思想 改进的 svm 速度好像挺快的

Making SVMs Scalable to Large Data Sets using Hierarchical Cluster Indexing 这个就是那篇论文的题目 发在 Data Mining and Knowledge Discovery

Fire(564122106) 21:16:29

哦 我看下，我现在看的都是台湾林的

Fire 分享文件 21:14:33

"Selective Block Minimization for Faster Convergence of Limited Memory Large_Scale Linear Models.pdf" 下载

苦瓜炒鸡蛋(852383636) 21:17:41

有那个大神 在用 svm 做聚类，Support Vector Clustering 这篇能做 就是时间复杂度太高了 $O(n^3)$

==========讨论结束==========

网神(66707180) 8:54:01

咱们开始讲 RVM，前面讲了 SVM，SVM 有一些缺点，比如输出是 decision 而不是概率分布，SVM 是为二元分类设计的，多类别分类不太实用，虽然有不少策略可以用于多元分类，但也各有问题参数 C 需要人工选择，通过多次训练来调整，感觉实际应用中这些缺点不算什么大缺点，但是 RVM 可以避免这些缺点。RVM 是一种贝叶斯方式的稀疏核方法，可以用于回归和分类，除了避免 SVM 的主要缺点，还可以更稀疏，而泛化能力不会降低 。先看 RVM 回归，RVM 回归的模型跟前面第三章形式相同，属于线性模型，但是

参数 w 的先验分布有所不同。
这个不同导致了稀疏性，等下再看这个不同
线性回归模型如下：

$p(t|\mathbf{x},\mathbf{w},\beta)=\mathcal{N}(t|y(\mathbf{x}),\beta^{-1})$ 其中 $\beta=\sigma^{-2}$，是噪音的精度 precision

均值 y(x)定义为：
$$y(\mathbf{x})=\sum_{i=1}^{M}w_i\phi_i(\mathbf{x})=\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})$$

RVM 作为一种稀疏核方法，它是如何跟核函数搭上边的，就是基函数 $\Phi_i(\mathrm{x})$ 采用了核函数的形式

每个核与一个训练样本对应，也就是：

$$\mathrm{y}(\mathrm{x})=\sum_{n=1}^{N}w_n k(\mathrm{x},\mathrm{x}_n)+\mathrm{b}$$

这个形式跟 SVM 用于回归的模型形式是相同的，看前面的式子(7.64)最后求得的 SVM 回归模型是：

$$y(\mathbf{x})=\sum_{n=1}^{N}(a_n-\widehat{a}_n)k(\mathbf{x},\mathbf{x}_n)+b$$

可以看到，RVM 回归和 SVM 回归模型相同，只是前面的系数从 a_n 换成了 w_n，接下来分析如何确定 RVM 模型中的参数 w，下面的分析过程跟任何基函数都适用，不限于核函数。
确定 w 的过程可以总结为：先假设 w 的先验分布，一般是高斯分布；然后给出似然函数，先验跟似然函数相乘的到 w 的后验分布，最大化后验分布，得到参数 w 。
先看 w 的先验，w 的先验是以 0 为均值，以 α 为精度的高斯分布，但是跟第三章线性回归的区别是，RVM 为每个 w_i 分别引入一个精度 α_i，而不是所有 w_i 用一个的共享的精度

所以 w 的先验是：

$$\mathrm{p}(\mathrm{w}\,|\,\alpha)=\prod_{i=1}^{M}\mathrm{N}(\mathrm{w}_i\,|\,0,\alpha_i^{-1})$$

对于线性回归模型，根据这个先验和似然函数可以得到其后验分布：

$$\mathrm{p}(\mathrm{w}\,|\,\mathrm{t},\mathrm{X},\alpha,\beta)=\mathrm{N}(\ \mathrm{w}\,|\,\mathrm{m},\textstyle\sum\ )$$

均值和方差分别是：

$$\begin{aligned}\mathbf{m}&=\beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}\\ \boldsymbol{\Sigma}&=\left(\mathbf{A}+\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\end{aligned}$$

这是第三章的结论，推导过程就不说了

其中 Φ 是 NxM 的矩阵，$\Phi_{ni}=\phi_i(\mathrm{x}_n)$，A 是对角矩阵 $\mathrm{A}=\mathrm{diag}(\alpha_i)$

对于 RVM，因为基函数是核函数，所以 $\Phi=\mathrm{K}$，K 是 NxN 维的核矩阵，其元素是 $\mathrm{k}(\mathrm{x}_n,\mathrm{x}_m)$

接下来需要确定超参数 α 和 β。一个是 w 先验的精度，一个是线性模型 p(t|x,w)的精度

确定的方法叫做 evidence approximation 方法，又叫 type-2 maximum likelihood，这在第三章有详细

介绍，这里简单说一下思路：

该方法基于一个假设 ,即两个参数是的后验分布 $p(\alpha,\beta\,|\,\mathrm{t})$ 是 sharply peaked 的 ,其中心值是 $\hat{\alpha}$ 和 $\hat{\beta}$ ，

根据贝叶斯定理， $p(\alpha,\beta\,|\,\mathrm{t})\propto \mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)\mathrm{p}(\alpha,\beta)$ ，先验 $\mathrm{p}(\alpha,\beta)$ 是 relatively flat 的，所以只要看

$\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)$ ， $\hat{\alpha}$ 和 $\hat{\beta}$ 就是使的 $\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)$ 最大的值。

$\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)$ 是对 w 进行积分的边界分布：

$$\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)=\int \mathrm{p}(\mathrm{t}\,|\,\mathrm{w},\beta)\,\mathrm{p}(\mathrm{w}\,|\,\alpha)\,d\mathrm{w}$$

这个分布是两个高斯分布的卷积，其 log 最大似然函数是:

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{X},\boldsymbol{\alpha},\beta) &= \ln\mathcal{N}(\mathbf{t}|\mathbf{0},\mathbf{C})\\ &= -\frac{1}{2}\left\{N\ln(2\pi)+\ln|\mathbf{C}|+\mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right\}\end{aligned}$$

其中 C 是 NxN 矩阵， $\mathbf{C}=\beta^{-1}\mathbf{I}+\mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^{\mathrm{T}}$

$\ln p(\mathbf{t}|\mathbf{X},\boldsymbol{\alpha},\beta) = \ln\mathcal{N}(\mathbf{t}|\mathbf{0},\mathbf{C})$ 这一步，以及 C 的值，是第三章的内容，大家看前面吧。

我们可以通过最大化似然函数，求得 $\hat{\alpha}$ 和 $\hat{\beta}$ ，书上提到了两种方法，一种是 EM，一种是直接求导迭代。

前者第九章尼采已经讲了，这里看下后者。

首先我们分别求这个 log 似然函数对所有参数 α 和 β 求偏导，并令偏导等于 0，求得参数的表达式：

$$\begin{aligned}\alpha_i^{\mathrm{new}} &= \frac{\gamma_i}{m_i^2}\\ (\beta^{\mathrm{new}})^{-1} &= \frac{\|\mathbf{t}-\mathbf{\Phi}\mathbf{m}\|^2}{N-\sum_i\gamma_i}\end{aligned}$$

其中 m_i 是 w 的后验均值 m 的第 i 个元素， γ_i 是度量 w_i 被样本集合影响的程度 $\gamma_i=1-\alpha_i\Sigma_{ii}$

Σ_{ii} 是 w 的后验方差 $\mathbf{\Sigma}$ 的对角线上的元素。

求 $\hat{\alpha}$ 和 $\hat{\beta}$ 是一个迭代的过程：

先选一个 $\hat{\alpha}$ 和 $\hat{\beta}$ 的初值，然后用下面这个公式得到后验的均值和方差：

$$\begin{aligned}\mathbf{m} &= \beta\mathbf{\Sigma}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}\\ \mathbf{\Sigma} &= \left(\mathbf{A}+\beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\end{aligned}$$

然后再同这个这个公式重新计算 $\hat{\alpha}$ 和 $\hat{\beta}$

$$\alpha_i^{\text{new}} = \frac{\gamma_i}{m_i^2}$$
$$(\beta^{\text{new}})^{-1} = \frac{\|\mathbf{t}-\mathbf{\Phi m}\|^2}{N-\sum_i \gamma_i}$$

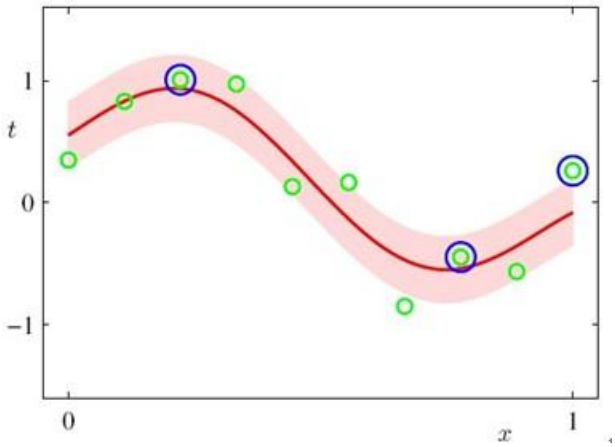
这样迭代计算，一直到到达一个人为确定的收敛条件，这就是确定 $\hat{\alpha}$ 和 $\hat{\beta}$ 的过程。

通过计算，最后的结果中，大部分参数 $\{\alpha_i\}$ 都是非常大甚至无穷大的值，从而根据 w 后验均值和方差的公式，其均值和方差都等于 0，这样 w_i 的值就是 0，其对应的基函数就不起作用了 ，从而达到了稀疏的目的。这就是 RVM 稀疏的原因。

需要实际推导一下整个过程，才能明白为什么大部分 $\{\alpha_i\}$ 都趋于无穷大。那些 w_i 不为 0 的 x_i 叫做 relevance vectors，相当于 SVM 中的支持向量。需要强调，这种获得稀疏性的机制可以用于任何基函数的线性组合中。这种获得稀疏性的机制似乎非常普遍的。
求得超参数，就可以通过下面式子得到新样本的分布：

$$p(t|\mathbf{x},\mathbf{X},\mathbf{t},\boldsymbol{\alpha}^\star,\beta^\star) = \int p(t|\mathbf{x},\mathbf{w},\beta^\star)p(\mathbf{w}|\mathbf{X},\mathbf{t},\boldsymbol{\alpha}^\star,\beta^\star)\,\mathrm{d}\mathbf{w}$$
$$= \mathcal{N}\left(t|\mathbf{m}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\sigma^2(\mathbf{x})\right).$$

下面看一个图示：



可以看到，其相关向量的数量比 SVM 少了很多，跟 SVM 相比的缺点是，RVM 的优化函数不是凸函数，训练时间比 SVM 长，书上接下来专门对 RVM 的稀疏性进行分析，并且介绍了一种更快的求 $\hat{\alpha}$ 和 $\hat{\beta}$ 的方法：

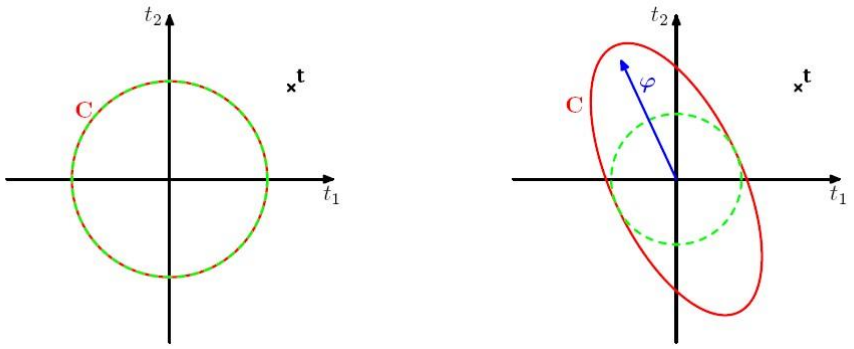


Figure 7.10 Illustration of the mechanism for sparsity in a Bayesian linear regression model, showing a training set vector of target values given by $\mathbf{t} = (t_1, t_2)^{\mathrm{T}}$, indicated by the cross, for a model with one basis vector $\boldsymbol{\varphi} = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2))^{\mathrm{T}}$, which is poorly aligned with the target data vector $\mathbf{t}$. On the left we see a model having only isotropic noise, so that $\mathbf{C} = \beta^{-1}\mathbf{I}$, corresponding to $\alpha = \infty$, with β set to its most probable value. On the right we see the same model but with a finite value of α. In each case the red ellipse corresponds to unit Mahalanobis distance, with $|\mathbf{C}|$ taking the same value for both plots, while the dashed green circle shows the contrition arising from the noise term β^{-1}. We see that any finite value of α reduces the probability of the observed data, and so for the most probable solution the basis vector is removed.

我接着讲 RVM 分类，我们看逻辑回归分类的模型:

$$y(\mathbf{x}, \mathbf{w}) = \sigma\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})\right)$$

σ 是 sigmoid 函数，我们引入 w 的先验分布，跟 RVM 回归相同，每个 w_i 对应一个不同的精度

这种先验叫做 ARD 先验，跟 RVM 回归相比，在求$\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)$的分布时，不再对 w 进行积分。

我们看在 RVM 回归时，是这么求$\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)$的：

$\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)=\int \mathrm{p}(\mathrm{t}\,|\,\mathrm{w},\beta)\,\mathrm{p}(\mathrm{w}\,|\,\alpha)\,d\mathrm{w}$ 从而得到：

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{X},\boldsymbol{\alpha},\beta) &= \ln \mathcal{N}(\mathbf{t}|\mathbf{0},\mathbf{C}) \\ &= -\frac{1}{2}\left\{N\ln(2\pi)+\ln|\mathbf{C}|+\mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right\}\end{aligned}$$

在 RVM 分类时，因为涉用到 sigmod 函数,计算积分很难，具体的为什么难，在第四章 4.5 节有更多的介绍，我们这里用 Laplace approximation 来求$\mathrm{p}(\mathrm{t}\,|\,\alpha,\beta)$的近似高斯分布，Laplace approximation 我叫拉普拉斯近似，后面都写中文了。

先看下拉普拉斯近似的原理，拉普拉斯近似的目的是找到连续变量的分布函数的高斯近似分布，也就是用高斯分布 近似模拟一个不是高斯分布的分布。

假设一个单变量 z，其分布是 $p(\mathrm{z}) = \frac{1}{Z} f(\mathrm{z})$，分母上的 Z 是归一化系数 $Z = \int \mathrm{f}(\mathrm{z})\,\mathrm{dz}$，目标是找到一个可以近似 p(z)的高斯分布 q(z)。

第一步是先找到 p(z)的 mode(众数) ，众数 mode 是一个统计学的概念，可以代表一组数据，不受极端数据的影响，比如可以选择中位数做一组实数的众数，对于高斯分布，众数就是其峰值。一组数据可能没有众数也可能有几个众数。

拉普拉斯分布第一步要找到 p(z)的众数 z_0，这是 p(z)的一个极大值点，可能是局部的，因为 p(z)可能有多

个局部极大值。在该点，一阶导数等于 0，$p'(z_0)=0$，后面再说怎么找z_0。找到z_0后，用$\ln f(x)$的泰勒展开来构造一个二次函数：

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2}A(z-z_0)^2$$

其中 A 是 f(z)在z_0的二阶导数再取负数。上式中，没有一阶导数部分，因为z_0是局部极大值，一阶导数为 0，把上式两边取指数，得到：

$$f(z) \simeq f(z_0)\exp\{-\frac{A}{2}(z-z_0)^2\}$$

把$f(z_0)$换成归一化系数，得到近似的高斯分布：

$$q(z) = (\frac{A}{2\pi})^{1/2}\exp\{-\frac{A}{2}(z-z_0)^2\}$$

拉普拉斯分布得到的近似高斯分布的一个图示：

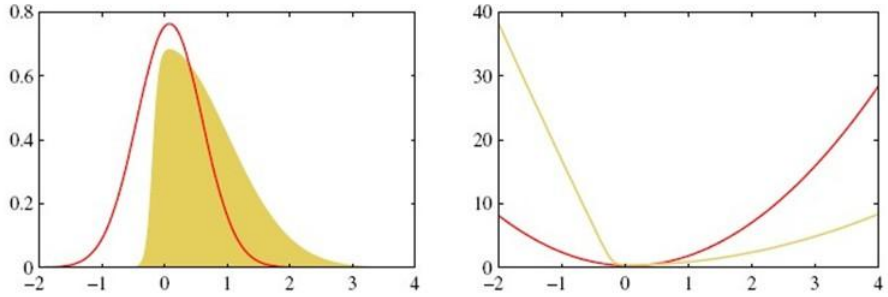


Figure 4.14 Illustration of the Laplace approximation applied to the distribution $p(z) \propto \exp(-z^2/2)\sigma(20z+4)$ where $\sigma(z)$ is the logistic sigmoid function defined by $\sigma(z) = (1+e^{-z})^{-1}$. The left plot shows the normalized distribution $p(z)$ in yellow, together with the Laplace approximation centred on the mode z_0 of $p(z)$ in red. The right plot shows the negative logarithms of the corresponding curves.

注意，高斯近似存在的条件是，原分布的二阶导数取负数、也就是高斯近似的精确度 A>0，也就是驻点$\mathbf{z_0}$必须是局部极大值，f(x)在改点出的导数为负数。当 z 是一个 M 维向量时，近似方法跟单变量的不同只是二阶导数的负数 A 变成了 MxM 维的海森矩阵的负数。

多维变量近似后的高斯分布如下：

$$q(z) = \frac{|A|^{1/2}}{(2\pi)^{M/2}}\exp\{-\frac{1}{2}(z-z_0)^T A(z-z_0)\} = N(z|z_0, A^{-1})$$

A 是海森矩阵的负数.mode 众数z_0一般是通过数值优化算法来寻找的，不讲了。

再回来看用拉普拉斯分布来近似 RVM 分类中的$p(t|\alpha,\beta)$：

刚才拉普拉斯分布忘了说一个公式，就是求得 q(z)后，确定$p(z) = \frac{1}{Z}f(z)$中的分母，也就是归一化系数的公式：

$$\begin{aligned} Z &= \int f(z)\,dz \\ &\simeq f(z_0)\int \exp\left\{-\frac{1}{2}(z-z_0)^T A(z-z_0)\right\} dz \\ &= f(z_0)\frac{(2\pi)^{M/2}}{|A|^{1/2}} \end{aligned}$$

这个一会有用。先看 RVM 中对 w 的后验分布的近似，先求后验分布的 mode 众数，通过最大化 log 后验分布 $\ln p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha})$ 来求 mode.先写出这个 log 后验分布：

$$\begin{aligned}\ln p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha}) &= \ln\{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\} - \ln p(\mathbf{t}|\boldsymbol{\alpha}) \\ &= \sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\} - \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w} + \text{const} \quad (7.109)\end{aligned}$$

其中 $\mathbf{A} = \mathrm{diag}(\alpha_i)$

最后求得的高斯近似的均值(也就是原分布的 mode)和精度如下：

$$\begin{aligned}\mathbf{w}^{\star} &= \mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{t}-\mathbf{y}) \\ \boldsymbol{\Sigma} &= \left(\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{B}\boldsymbol{\Phi}+\mathbf{A}\right)^{-1}\end{aligned}$$

现在用这个 w 后验高斯近似来求边界似然 $p(\mathbf{t}|\boldsymbol{\alpha}) = \int p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\,\mathrm{d}\mathbf{w}$

根据前面那个求归一化系数 Z 的公式 Z= $f(\mathbf{z}_0)\frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}}$

有:

$$\begin{aligned}p(\mathbf{t}|\boldsymbol{\alpha}) &= \int p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})\,\mathrm{d}\mathbf{w} \\ &\simeq p(\mathbf{t}|\mathbf{w}^{\star})p(\mathbf{w}^{\star}|\boldsymbol{\alpha})(2\pi)^{M/2}|\boldsymbol{\Sigma}|^{1/2}.\end{aligned}$$

RVM 这部分大量基于第二章高斯分布和第三、四两章，公式推导很多，需要前后关联才能看明白。