

个人资料



积分: 2万+ 排名: 518



2014年11月	1篇
2014年10月	4篇
2014年8月	1篇
2014年7月	6篇
2014年6月	4篇

展开

### 热门文章

word2vec 中的数学原理详解 ( 一 ) 目录和 前言

阅读量:141844

word2vec 中的数学原理详解(四)基于 Hi

erarchical Softmax 的模型

阅读量:83222

word2vec 中的数学原理详解 (三)背景知

识

阅读量:59585

word2vec 中的数学原理详解(五)基于 N

egative Sampling 的模型

阅读量:59145

word2vec 中的数学原理详解 (二)预备知

3口 阅读量:58235

#### 最新评论

word2vec 中的数学原理详解...

LINTEBI: 博主,方便的话可以发一份PDF给我吗?感激不尽!感激不尽!604156331@qq.com

word2vec 中的数学原理详解... Qinjian666: NB闪闪的博主

word2vec 中的数学原理详解...

leayc: #6.9.3: https://github.com/RaRe-Technolo

gies/gens...

word2vec 中的数学原理详解...

p03721:博主可以发一份PDF吗?感谢!875051

204@qq.com

受限玻尔兹曼机(RBM)学习笔记(...

jocelynxyq: 赞楼主的分享!

# 原 牛顿法与拟牛顿法学习笔记(五)L-BFGS 算法

2014年03月24日 00:53:27 阅读数: 45953

机器学习算法中经常碰到非线性优化问题,如 Sparse Filtering 算法,其主要工作在于求解一个非线性极小化问题。在具体实现中,大多调用的是成熟的软件包做支撑,其中最常用的一个算法是 L-BFGS。为了解这个算法的数学机理,这几天做了一些调研,现把学习过程中理解的一些东西整理出来。

## 目录链接

- (1) 牛顿法
- (2) 拟牛顿条件
- (3) DFP 算法
- (4) BFGS 算法
- (5) L-BFGS 算法

## §2.4 L-BFGS 算法

在 BFGS 算法中, 需要用到一个  $N \times N$  的矩阵  $D_k$ , 当 N 很大时, 存储这个矩阵将变得 很耗计算机资源. 例如 ([2]), 考虑 N 为 10 万的情形, 且用 double 型 (8 字节) 来存储  $D_k$ , 需要多大的内存呢? 我们来计算一下

$$\frac{N \text{ 阶矩阵的字节数}}{1 \text{ GB 的字节数}} = \frac{10^5 \times 10^5 \times 8}{2^{10} \times 2^{10} \times 2^{10}} = 74.5(GB), \tag{2.42}$$

74.5 GB, 很惊人是吧, 这对于一般的服务器是很难承受的. 当然, 考虑到矩阵  $D_k$  的对称性, 内存还可以降一半, 但是, 在机器学习问题中, 像 10 万这样的规模还只能算是中小规模. 那么, 是否可以通过对 BFGS 算法进行改造, 从而减少其迭代过程中所需的内存开销呢?

接下来, 讨论 L-BFGS 算法的具体实现过程. 我们的出发点是算法 2.3 步 6 中的迭代式

$$D_{k+1} = \left(I - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) D_k \left(I - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k},$$

若记  $\rho_k = \frac{1}{\mathbf{y}_L^T \mathbf{s}_k}, V_k = I - \rho_k \mathbf{y}_k \mathbf{s}_k^T$ , 则上式可写成

$$D_{k+1} = V_k^T D_k V_k + \rho_k \mathbf{s}_k \mathbf{s}_k^T. \tag{2.43}$$

如果给定初始矩阵  $D_0$  (通常为正定的对角矩阵, 如  $D_0 = I$ ), 则利用 (2.43), 依次可得

$$D_{1} = V_{0}^{T} D_{0} V_{0} + \rho_{0} \mathbf{s}_{0} \mathbf{s}_{0}^{T};$$

$$D_{2} = V_{1}^{T} D_{1} V_{1} + \rho_{1} \mathbf{s}_{1} \mathbf{s}_{1}^{T}$$

$$= V_{1}^{T} (V_{0}^{T} D_{0} V_{0} + \rho_{0} \mathbf{s}_{0} \mathbf{s}_{0}^{T}) V_{1} + \rho_{1} \mathbf{s}_{1} \mathbf{s}_{1}^{T}$$

$$= V_{1}^{T} V_{0}^{T} D_{0} V_{0} V_{1} + V_{1}^{T} \rho_{0} \mathbf{s}_{0} \mathbf{s}_{0}^{T} V_{1} + \rho_{1} \mathbf{s}_{1} \mathbf{s}_{1}^{T};$$

$$D_{3} = V_{2}^{T} D_{2} V_{2} + \rho_{2} \mathbf{s}_{2} \mathbf{s}_{2}^{T}$$

$$= V_{2}^{T} (V_{1}^{T} V_{0}^{T} D_{0} V_{0} V_{1} + V_{1}^{T} \rho_{0} \mathbf{s}_{0} \mathbf{s}_{0}^{T} V_{1} + \rho_{1} \mathbf{s}_{1} \mathbf{s}_{1}^{T}) V_{2} + \rho_{2} \mathbf{s}_{2} \mathbf{s}_{2}^{T}$$

$$= V_{2}^{T} V_{1}^{T} V_{0}^{T} D_{0} V_{0} V_{1} V_{2} + V_{2}^{T} V_{1}^{T} \rho_{0} \mathbf{s}_{0} \mathbf{s}_{0}^{T} V_{1} V_{2} + V_{2}^{T} \rho_{1} \mathbf{s}_{1} \mathbf{s}_{1}^{T} V_{2} + + \rho_{2} \mathbf{s}_{2} \mathbf{s}_{2}^{T};$$

. . .

一般地, 我们有

$$D_{k+1} = (V_k^T V_{k-1}^T \cdots V_1^T V_0^T) D_0(V_0 V_1 \cdots V_{k-1} V_k)$$

$$+ (V_k^T V_{k-1}^T \cdots V_2^T V_1^T) (\rho_0 \mathbf{s}_0 \mathbf{s}_0^T) (V_1 V_2 \cdots V_{k-1} V_k)$$

$$+ (V_k^T V_{k-1}^T \cdots V_3^T V_2^T) (\rho_1 \mathbf{s}_1 \mathbf{s}_1^T) (V_2 V_3 \cdots V_{k-1} V_k)$$

$$+ \cdots$$

$$+ (V_k^T V_{k-1}^T) (\rho_{k-2} \mathbf{s}_{k-2} \mathbf{s}_{k-2}^T) (V_{k-1} V_k)$$

$$+ V_k^T (\rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T) V_k$$

$$+ \rho_k \mathbf{s}_k \mathbf{s}_k^T.$$

$$(2.44)$$

由上式可见, 计算  $D_{k+1}$  需用到  $\{s_i, y_i\}_{i=0}^k$ , 因此, 若从  $\mathbf{s}_0, \mathbf{y}_0$  开始连续地存储 m 组的话, 只能存储到  $\mathbf{s}_{m-1}, \mathbf{y}_{m-1}$ , 亦即, 只能依次计算  $D_1, D_2, \cdots$ , 直到  $D_m$ . 那么,  $D_{m+1}$ ,  $D_{m+2}$  该如何计算呢?

自然地, 如果一定要丢掉一些向量, 那么肯定优先考虑那些最早生成的向量. 具体来说, 计算  $D_{m+1}$  时, 我们保存  $\{s_i, y_i\}_{i=1}^m$ , 丢掉了  $\{s_0, y_0\}$ ; 计算  $D_{m+2}$  时, 我们保存  $\{s_i, y_i\}_{i=2}^{m+1}$ , 丢掉了  $\{s_i, y_i\}_{i=0}^1$ ; · · ·

但是舍弃掉一些向量后, 就只能**近似计算**了, 当 k+1>m 时, 仿照 (2.44), 可以构造近似计算公式

$$D_{k+1} = (V_k^T V_{k-1}^T \cdots V_{k-m+2}^T V_{k-m+1}^T) D_0(V_{k-m+1} V_{k-m+2} \cdots V_{k-1} V_k)$$

$$+ (V_k^T V_{k-1}^T \cdots V_{k-m+3}^T V_{k-m+2}^T) (\rho_0 \mathbf{s}_0 \mathbf{s}_0^T) (V_{k-m+2} V_{k-m+3} \cdots V_{k-1} V_k)$$

$$+ (V_k^T V_{k-1}^T \cdots V_{k-m+4}^T V_{k-m+3}^T) (\rho_1 \mathbf{s}_1 \mathbf{s}_1^T) (V_{k-m+3} V_{k-m+4} \cdots V_{k-1} V_k)$$

$$+ \cdots$$

$$\downarrow \overset{\square}{\downarrow_{269}} + (V_k^T V_{k-1}^T) (\rho_{k-2} \mathbf{s}_{k-2} \mathbf{s}_{k-2}^T) (V_{k-1} V_k)$$

$$\downarrow V_k^T (\rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T) V_k$$

$$\downarrow V_k^T (\rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T) V_k$$

$$\downarrow V_k^T (\rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T) V_k$$

$$\downarrow \rho_k \mathbf{s}_k \mathbf{s}_k^T.$$

$$(2.45)$$

在文献 [8]  $\stackrel{\text{评论}}{\longleftarrow}$  2.44) 和 (2.45) 被称为 special BFGS matrices. 若引入  $\hat{m} = \min\{k, m-1\}$ , 则还可以将 微信 44) 和 (2.45) 合并地写成

$$D_{k+1} = \underbrace{\begin{pmatrix} \nabla_k^T V_{k-1}^T \cdots V_{k-\hat{m}+1}^T V_{k-\hat{m}}^T \end{pmatrix} D_0(V_{k-\hat{m}} V_{k-\hat{m}+1} \cdots V_{k-1} V_k)}_{\text{(a)}} + (V_k^T V_{k-1}^T \cdots V_{k-\hat{m}+2}^T V_{k-\hat{m}+1}^T) (\rho_0 \mathbf{s}_0 \mathbf{s}_0^T) (V_{k-\hat{m}+1} V_{k-\hat{m}+2} \cdots V_{k-1} V_k)}_{\text{(b)}} + (V_k^T V_{k-1}^T \cdots V_{k-\hat{m}+3}^T V_{k-\hat{m}+2}^T) (\rho_1 \mathbf{s}_1 \mathbf{s}_1^T) (V_{k-\hat{m}+2} V_{k-\hat{m}+3} \cdots V_{k-1} V_k)}_{\text{(b)}} + \cdots \\ + (V_k^T V_{k-1}^T) (\rho_{k-2} \mathbf{s}_{k-2} \mathbf{s}_{k-2}^T) (V_{k-1} V_k)}_{\text{(c)}} + V_k^T (\rho_{k-1} \mathbf{s}_{k-1} \mathbf{s}_{k-1}^T) V_k \\ + \rho_k \mathbf{s}_k \mathbf{s}_k^T.$$

看到这里, 读者千万不要被 (2.46) 的冗长复杂的形式所吓到. 事实上, 由 BFGS 算法流程易知,  $D_k$  的作用仅用来计算  $D_k$   $g_k$  获取搜索方向, 因此, 若能利用表达式 (2.46) 设计出一种计算  $D_k$   $g_k$  的快速算法, 则大功告成. 文 [8] 给出了这样一个算法, 具体见算法 2.4.

```
算法 2.4 (D_k \cdot \mathbf{g}_k 的快速算法)
               Step 1 初始化.

\hat{\mathbf{g}} \quad
\begin{cases}
k, & \text{ if } k \leq m \\
m, & \text{ if } k > m
\end{cases}
 ; \mathbf{q}_L = \mathbf{g}_k.
加入CSDN,享受更精准的内容推荐,与500万程序员共同成
长!
               Step 2 后 f <sup>登录</sup> 注册 ×
                             FOR i = L - 1, L - 2, \dots, 1, 0 DO
                             {
                                  i = i + \delta:
                                  \alpha_i = \rho_j \mathbf{s}_i^T \mathbf{q}_{i+1}; // \alpha_i 需要存下来, 前向循环要用!
                                  \mathbf{q}_i = \mathbf{q}_{i+1} - \alpha_i \mathbf{y}_i.
                             }
               Step 3 前向循环.
                             \mathbf{r}_0 = D_0 \cdot \mathbf{q}_0;
                            FOR i = 0, 1, \dots, L - 2, L - 1 DO
                                 \beta_j = \rho_j \mathbf{y}_j^T \mathbf{r}_i;
                                 \mathbf{r}_{i+1} = \mathbf{r}_i + (\alpha_i - \beta_i)\mathbf{s}_i.
```

最后算出的  $\mathbf{r}_L$  即为  $H_k \cdot \mathbf{g}_k$  的值.

# 参考文献

- [1] http://www.materialssimulation.com/node/625
- [2] http://www.codelast.com/?p=2780
- [3] http://www.tydxyq.cn/kuai\_su/youhuasheji/suanfayuanli/3.1.asp
- [4] http://en.wikipedia.org/wiki/BFGS\_method
- [5] http://en.wikipedia.org/wiki/Sherman-Morrison\_formula
- [6] Dai Y H. A perfect example for the BFGS method[J]. Mathematical Programming, 2013, 138(1-2): 501-530.
- [7] Liu D C, Nocedal J. On the limited memory BFGS method for large scale optimization[J]. Mathematical programming, 1989, 45(1-3): 503-528.
- [8] Nocedal J. Updating quasi-Newton matrices with limited storage[J]. Mathematics of computation, 1980, 35(151): 773-782.
- [9] http://baike.baidu.com/view/6062086.htm

作者: peghoty

出处: http://blog.csdn.net/itplus/article/details/21897715

欢迎转载/分享,但请务必声明文章出处.

版权声明:本文为博主原创文章,未经博主允许不得转载。 https://blog.csdn.net/peghoty/article/details/21897715

文章标签: (拟牛顿法) (BFGS) (L-BFGS) (Limited-memory)

个人分类: 数学天地

#### 世界上最好的语言 , PHP凭什么这么 "说"

课程包含PHP零基础入门、PHP基础编程、数据库、项目实战、linux、框架开发到高级+企业项目实战,深入浅出地剖析和分解了PHP企业级开发项目在实际工作中的应用

我来说一句

查看更多>>

18502

查看更多>>

想对作者说点什么?

csdn547885816 2017-12-07 15:39:02 #4楼 2.45式,如果要求Dk+1,就得舍去s0,y0。那么式子第二项不是要整段舍去吗?因为(p0s0s0')不是求不到吗,s0已经舍去了

**随煜而安** 2015-08-01 12:46:15 #3楼

博主你看看这个这个博客 http://blog.csdn.net/sparkR/article/details/38684243 如果没经过您的同意感觉不太好吧,完全是用的您的文章

CookCooller 2015-04-04 20:00:32 #2楼

查看回复(4)

**一枚写手** 2014-05-12 03:11:29 #1楼

查看回复(3)

杳看 11 条热评

### 大规模优化算法 - LBFGS算法

L-BFGS算法比较适合在大规模的数值计算中,具备牛顿法收敛速度快的特点,但不需要牛顿法那样存储Hesse矩阵,因此节省了大量的空间以及计算资源。本文主要通过对于无约束最优化问题的一些常用算法总结,一...

● qq\_23052951 2017-02-16 22:10:47 阅读数:2425

### 机器学习算法实现解析——liblbfgs之L-BFGS算法

在博文"优化算法——拟牛顿法之L-BFGS算法"中,已经对L-BFGS的算法原理做了详细的介绍,本文主要就开源代码liblbfgs重新回顾L-BFGS的算法原理以及具体的实现过程,在L-BFGS算法中…

🤏 google19890102 2017-08-20 17:27:34 阅读数: 1743

## L-BFGS - CSDN博客

*L-BFGS*转载并总结:http://www.cnblogs.com/alexanderkun/p/4024600.html*L-BFGS*算法比较适合在大规模的数值计算中,具备牛顿法收敛速度快的特点,但不需要牛顿法那样...

2018-5-6

#### L-BFGS - CSDN博客

本文将介绍一种在实际工程中解决大规模优化问题时必然会用到的优化算法:L-BFGS算法。上面已经提到了在面对大规模优化问题时,由于近似矩阵往往是稠密的,在计算和存...

2018-4-15

### 苏坡90后小伙在家无聊玩微信, 存款惊 呆父母

威能科技·顶新

#### lbfgs c/c++代码实现

lbfgs的原理见: http://blog.csdn.net/itplus/article/details/21897715 代码未经测试,有空充完整的代码优化.#include #includ...

H u010510549 2017-07-26 19:55:39 阅读数:276

#### L-BFGS算法 - CSDN博客

L-BFGS算法原创 2015年03月29日 18:12:23 阅读18473 前面的文章中,讲述过BFGS算法,它是用来求解最优化问题的,在这个算法中,相对于普通的牛顿迭代法有很大的...

2018-5-7

### L-BFGS - CSDN博客

*L-BFGS*转载并总结:http://www.cnblogs.com/alexanderkun/p/4024600.html*L-BFGS*算法比较适合在大规模的数值计算中,具备牛顿法收敛速度快的特点,但不需要牛顿法那样...

2018-5-6

#### 无约束最优化方法——牛顿法、拟牛顿法、BFGS、LBFGS

好久没写博客了,今天打开一看csdn终于可以用latex,不用到处去粘贴标签,方便了许多。且先试试效果如何。先讲讲一些优化方法。 最速下降法 牛顿法 拟牛顿法 SR1 BFGS DFP LBFGS 【...

♣ lansatiankongxxc 2015-05-20 17:44:32 阅读数: 4544

#### LBFGS算法使用

- 一、首先把两个作者 JORGE NOCEDAL 给的文件看明白sdrive.f和lbfgs.f sdrive.f is a simple driver lbfgs.f conta...
- ※ settingsun1225 2011-01-15 20:03:00 阅读数: 19528

#### 机器学习算法实现解析——liblbfgs之L-BFGS算法 - CSDN博客

在博文"优化算法——拟牛顿法之L-BFGS算法"中,已经对L-BFGS的算法原理做了详细的介绍,本文主要就开源代码liblbfgs重新回顾L-BFGS的算法原 理以及具体的实现过程,...

2018-4-18

#### 机器学习优化算法L-BFGS及其分布式实现 - CSDN博客

最近做的科研项目需要用到L-BFGS、这段时间看了不少关于L-BFGS的博客以及论文、在此进行一下小小的总结。在无约束优化问题中,牛顿法及拟牛顿 法是常用的方法,L-...

2018-4-9

## 梯度法(SGD)、拟牛顿法(LBFGS)与共轭梯度法(CG)

一、基本原理 梯度法: 由一阶泰勒级数展开式,f(x+dx) = f(x) + sum(i,df/dx(i)\*dx(i)) + r(x,dx)。其中r(x,dx)为余项,y=0为余项,y=0为余项,y=00为6元,y=00

🦠 student1218 2015-07-14 18:15:26 阅读数:3579

#### 万万没想到,这个股票交流群推的牛股太牛了。

百阿·顶新

### L-BFGS优化算法 - CSDN博客

举报内容: L-BFGS优化算法 举报原因: 色情 政治 抄袭 广告 招聘 骂人 其他 原文地址: 原因补充: 最多只允许输入30个字加入...

2018-5-5

## 优化算法——拟牛顿法之L-BFGS算法 - CSDN博客

一、BFGS算法 在"优化算法——拟牛顿法之BFGS算法"中,我们得到了BFGS算法的校正公式: 利用Sherma...

2018-4-30

#### L-BFGS算法

前面的文章中,讲述过BFGS算法,它是用来求解最优化问题的,在这个算法中,相对于普通的牛顿迭代法有很大的改进。链接:http://blog.csdn.ne t/acdreamers/article/de...

🦫 ACdreamers 2015-03-29 18:12:23 阅读数:18510

## LBFGS公式推导

LBFGS 公式推导

🏖 lansatiankongxxc 2014-08-24 22:41:13 阅读数: 1499

#### LBFGS优化算法以及线性搜索中zoom步长选择方法

该文主要介绍了最优化算法Lbfg以及线性搜索步长的确定。

● zkwdn 2016-11-16 16:37:50 阅读数:960

### matlab编写的LBFGS优化算法

2010年05月21日 18KB

下载

## 优化算法——拟牛顿法之L-BFGS算法

一、BFGS算法 在"优化算法——拟牛顿法之BFGS算法"中,我们得到了BFGS算法的校正公式: 利用Sherman-Morrison公式可对上...

2017-06-30 10:50:20 阅读数:460 xiaopihaierletian

## 逻辑回归模型及LBFGS的Sherman Morrison(SM) 公式推导

模型简介 二项逻辑回归模型: (1.1) (1.2) 其中,Y为标签,X为样本,W为模型参数,exp为指数函数。 根据 (1.1 ) 和 (1.2 ) 可以得到

P(Y=1|X)和P(Y=...

● zhirom 2014-08-01 10:20:40 阅读数:9818

## lbfgs优化包

2013年12月03日 18KB 下载



## VBA代码加密,保护源码不被反编译和盗版

加密Office中使用VB二次开发的插件,保护插件源码不被反编译和盗版



### 机器学习优化算法—L-BFGS

L-BFGS算法由于其高效的性能而被广泛运用在实际工程中,本文首先介绍L-BFGS算法和其它算法的比较,然后详细介绍该算法的主要思想以及每一步迭代时近似矩阵的更新细节。...

● chenzhijay 2014-11-27 09:53:21 阅读数:5066

#### BFGS算法

今天,我来讲一种在机器学习中常用到的优化算法,叫做BFGS算法。BFGS算法被认为是数值效果最好的拟牛顿 法,并且具有全局收敛性和超线性收敛速度。那么接下来将会详细讲解。 Contents ....

ACdreamers 2015-03-27 01:55:41 阅读数:32400

### BFGS优化算法的理解以及LBFGS源码求解最优化问题

关于最优化求解,吴军有篇blog讲的很不错,证明和解释都很清楚,具体可以参考http://www.cnblogs.com/joneswood/archive/2012/03/11/239052 9.ht...

pizibing880909 2014-03-13 20:09:45 阅读数:6631

## 多变量函数优化的L-BFGS算法matlab程序

2014年08月16日 29KB 下载



#### L-BFGS优化算法

关于优化算法的求解,书上已经介绍了很多的方法,比如有梯度下降法,坐标下降法,牛顿法和拟牛顿法。梯度下降法是基于目标函数梯度的,算法的收敛速度是线性的,并且当问题是病态时或者问题规模较大时,收敛速度尤其...

@ seasermy 2015-11-02 14:40:56 阅读数:1004

#### 呼叫中心系统

登录呼叫中心管理系统

百度广告



### 机器学习优化算法L-BFGS及其分布式实现

最近做的科研项目需要用到L-BFGS,这段时间看了不少关于L-BFGS的博客以及论文,在此进行一下小小的总结。在无约束优化问题中,牛顿法及拟牛顿法是常用的方法,L-BFGS属于拟牛顿法,下面从牛顿法开...

ರ dadouyawp 2015-03-14 22:26:41 阅读数:2957

### L-BFGS算法(有限内存BFGS算法)

● kaka19880812 2015-08-02 19:26:49 阅读数: 426