

Towards Paperless Academic Reading

<http://freemind.pluskid.org/misc/towards-paperless-academic-reading>

“无纸化”似乎是在来了美国之后才听到的一个词，感觉上似乎是美国人更注重环保一些，但是实际上是因为美国这边用纸一直像不要钱一样所以才会渐渐地有人觉得这样下去不行吧。这边的人大抵都是动不动就顺手打印下来，“打印”在人们日常生活中的地位也可以在许许多多的地方看到：比如买个笔记本的话可以再加几十块就买个打印机。这边的学生都是每人抱着一个（或多个）巨大的 binder，里面夹着各种打印下来的 lecture notes、psets 以及 slides 之类的资料。学校也为打印提供了相当的便利的打印系统：在学校无线网络覆盖到的地方直接打印，然后到遍布全校的任意打印机处刷卡就可以获得打印结果，每个人每年可以免费打印 3000 张，如果你有正当里需要更多的配额也是去跟学校 argue 的。而如果是 graduate student 的话，还可以用自己实验室的打印资源，就更加无上限了，而且全部都是默认彩色打印。

所以关于节约用纸的标语什么的也就渐渐地随处可见了，比如有些人在发邮件的时候还会在结尾签名档附上如果你不打印这封 email 的话，会拯救多少大树什么的。还有比如银行提供电子账单的功能而不是每个月给你寄纸质的账单过来的话，也非常大张旗鼓地鼓吹说我们提供绿色的 paperless 服务云云。

当然现状是什么就不要吐槽了，我们可以看到的是各种好的转变。现在随着电子产品的普及，再加上在美国电子产品真的非常便宜¹，校园里也随处可见用 tablet 代替纸质阅读物的同学。受到大环境的影响，我也渐渐地对比玩电子产品节约纸张的阅读方式变得非常感兴趣。这里就谈一下我的感想吧，不过我想把这个话题扯得远一点。

先说出版和媒体吧，第一波的冲击大概是来自于像 Amazon 这样的在线书店将各种实体书店打垮吧；然后是电子出版的发展，随着互联网和移动设备的发展，各种新闻报纸什么的也都纷纷转型提供电子版或者挂掉，相比起来电子书的进展似乎却并不是特别迅速，特别是在学术类书籍这一块。

从出版商的角度来看问题当然最优先考虑的是降低成本，这一方面可以说电子书有得天独厚的优势，因为完全免去了印刷厂啊、物流啊之类的各种麻烦事，而且一本书做出来之后可以零成本卖无限份 copy ——当然得要保证盗版 copy 不要泛滥才行。盗版与反盗版大概是一场永远都不会停止的战斗吧，其实话说回来，如果出版商能控制住成本，把正版的價格控制在可以接受的范围之内的话，大部分人应该还是会选择正版吧，毕竟去找盗版的東西也是一件相对麻烦的事情，而且质量良莠不齐。不过美国这边的书——特别是科技类的，价格真是贵得离谱，动辄几百美元，我在国内算是一个有买书癖的人，到这边来除了课本之外基本上

posted on [Free Mind](#) on March 25, 2013
generated with pandoc on December 3, 2015
category: MISC

tags: Tools

¹ 这里并不是指按汇率换算成人民币有多便宜，而是指如果你在美国生活的话，电子产品的价格和你的日常花销，比如吃饭之类的相对价格。比如国内吃一顿中饭 10 块钱，买个 iPad 4 大概要 3600，那就是一年的中饭了；而在美国吃一顿中饭比如 6 美元，买个 iPad 4 大概 500 美元，才三个月的中饭而已。

没敢买过其他书。

Formats	Amazon price	New from	Used from
Kindle Edition	\$31.37	--	--
Hardcover	--	\$432.60	\$14.89
Paperback	\$33.02 	\$26.79	\$26.02

Figure 1:

一开始我以为是美国人特别重视版权以及知识的价值所以科技类书才会这么贵，相对来说那些快餐小说或者畅销书之类的就通常几美元就可以买到。不过后来我发现似乎还有其他因素在里面。一个特点就是科技类的书大部分都是 **hardcover**，完全不知道美国人是怎么想的，硬面抄不仅拿起来很重，而且比较容易损坏。直到我发现有一些同时有 **hardcover** 和 **paperback** 的版本的两个价格差了之后，才意识到，原来 **hardcover** 不为别的，只是高端洋气上档次的身份标志而已.....-.-bb 比如 Marr 的《Vision》，paperback 在 Amazon 标价 33，而 **hardcover** 标价 432²。

总而言之，既然 **paperback** 可以做到相当便宜，似乎版权的费用并不是真正的瓶颈，而且苹果的 **iBook Store** 里的教科书也大部分都在十几美元以下，所以说这个方向的阻碍大概主要就剩下“技术”跟“盗版”的问题了。不过对于 Amazon 而言，他的一个难处在于自己并不是一个出版商，所以大多数时候价格并不能由他控制，于是看到 Amazon 上的书籍虽然越来越多都有 **Kindle** 版本发售，但是价格却并没有比纸质版便宜几块钱甚至时不时有比纸版还要贵的尴尬情况出现。

说到技术其中很重要的一块就是书籍的格式了。现在主流的电子书格式大概分为开放的 **epub** 格式、Amazon 的 **kindle** 用的那几种格式，还有苹果的 **iBook** 所用的 **epub** 改。这些格式感觉很像是 **HTML** + 元数据，支持一些基本的格式、插图之类的。很适合于小说类的文字为主排版简单的书籍，而且现在 **HTML** 标准支持的东西也越来越多，像日文的假名注音呀，中文的竖排啊之类的也可以支持得很好的了。不过像杂志这种复杂排版的東西似乎就不太适合这种可以根据设备屏幕大小动态重拍的显示格式。不过如果回到技术类书籍的话题的话，现在大概主要分为两派吧。

かみやま
現在の神山市の北方に広がる田園地帯は、昔は一個の農村でした。千反田家はその村の庄屋として広い農地を持ち、自ら耕し、またひとに貸してもいました。村の代表として、租税関係を始めとする領主側との折衝に当たったり、簡単なことであれば裁判官的なことをすることもあったそうです。農地改良のための工事の音頭も取っていました。また、春秋の祭においても、村を代表していました。

一个是在线类的技术书籍，通过 **Internet** 加入社交网络的元素，比如 **Real World Haskell** 的在线版本就可以在每一段添加 **comments**。更为复杂的是像 **IPython Notebook** 之类的可以交互式地执行代码并在“书”里看到结果的。可以说现在的浏览器已经俨然要变成独立的操作系统的倾向，所以时常也有各种很炫的完全在浏览器上运行的 **demo** 或者甚至产品出现。

² 好吧我承认为了造成“冲击力”故意用了这个例子，**hardcover** 实际上是 1982 的版本所以也许有收藏价值的因素在里面？

另一类就是充满了数学公式、交叉引用等内容的科学类图书。这类图书基本上目前最好的排版源就是 TeX，而最好的输出格式就是 PDF 了。在电脑上和打印下来都显示效果非常好，但是最严重的问题就是不支持重排版。虽然现在许多移动设备上的 PDF 阅读软件都支持 PDF text reflow，其实只是对文本类的 PDF 比较好用而已，因为格式基本上就消失掉了，而科技类图书中的交叉引用之类的如果页码变掉了也是很麻烦的。这么说好像是在讲科学类书籍并不适合重排版，但是那其实完全是以旧的眼光在看待问题。就交叉引用这个问题本身来说，其实电子书完全应该解决得更好才是，因为在读纸版书的时候查阅相关的参考文献什么的其实是比较麻烦的事，现在用 LaTeX 编译 PDF 的话可以加上超链接选项，这样就可以点击引用的链接直接跳转到相应的引用的地方，这是纸版书做不到的，或者更贴心的比如像 wikipedia 那样，可以直接以气泡的方式在当前位置把参考文献条目显示出来。因为我看东西的时候特别关注 reference，所以一直期待有什么 PDF Reader 有类似的功能，不过由于 PDF 文件里通常都没有这样的元信息，所以实现起来也并不是一件简单的事情吧。

总而言之，由于 PDF 的各种不灵活性——因为 PDF 设计的年代就没有考虑过这些问题，以及其他一些各种各样不得而知的原因，大家在为各种移动阅读器发布电子书的时候一般不会考虑以 PDF 的方式发布（豆瓣阅读似乎是个神奇的例外，明明是发布小说类的书籍但是却以 PDF 的格式发布出来）。但是不是 PDF 的话，公式渲染是一个很大的问题。其实从技术难度上还是，现在连 Web 上都能用 MathJaX 显示出非常漂亮的公式来，而且 MathML 之类的用于描述公式的标准也已经存在了不止一年两年了。但是即便是万事俱备，似乎也还是欠缺了点东风，至少目前还没有 out-of-box 能用的解决方案的样子。于是 Amazon 就要流氓了，比如说我们看看《Concentration Inequalities: A Nonasymptotic Theory of Independence》这本书的 Kindle 版，发送一个 sample 到 kindle 上看起来是这个样子：

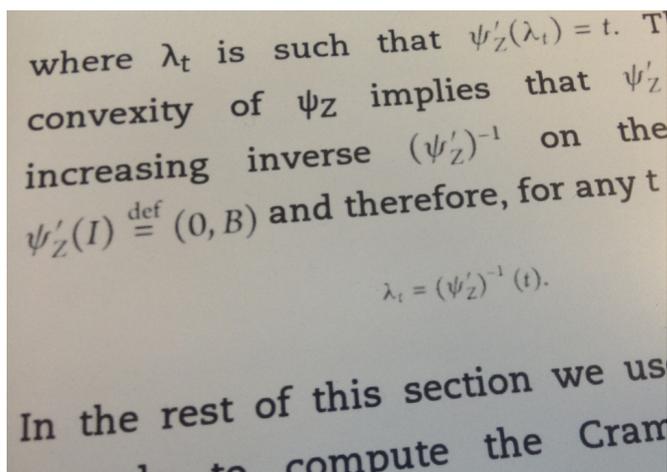


Figure 2:

看起来是不是还挺不错的样子？实际上完全是由于拍照的角度选得好，再加上 Kindle 对于普通文本渲染效果确实很漂亮³造成了错觉。实际上仔细看就会发现第一行的 ψ_Z 和第二行的那个看起来完全不一样。因为这个电子版大概用了非常粗糙的公式转换，把简单的公式转换了一下，稍微复杂一点的就直接贴个图了事了。在电脑上更加惨不忍睹。

Indeed, if $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function, then for all $x \in \mathbb{R}^n$, $\|\nabla f(x)\| \leq 1$, and the Gaussian logarithmic Sobolev inequality holds for all $x \in \mathbb{R}^n$,

$$\|\nabla g(x)\|^2 \leq \frac{\lambda^2}{4} \|\nabla f(x)\|^2$$

这种敷衍了事的电子书就拿出来卖了，还卖 87 美元，实在是.....给人一种黔驴技穷的感觉。还好在购买之前都是有提供 Sample 可以试看的，要不然有一大帮人要被坑死了。

不过比较好的例子也是有的，比如说《Machine Learning: A Probabilistic Perspective》⁴这个书的 Kindle 版，显示效果就近乎完美，简直就像 TeX 直接排出来的 PDF 文件一样，仔细检查了一下发现其实这是一种叫做 AZW4 的格式，其实就是一个 PDF，不过 Amazon 为了实现和他其他电子书同样的 DRM，再做了一层包装而已。嘛，这样也不错了，不过有一点需要格外小心的就是，这种格式很明显无法 reflow，而且在小屏幕的 Kindle 上看起来肯定很难受，所以 Amazon 压根就不提供购买到 Kindle 的选项，在页面上会明确说明：Available only on Kindle Fire Tablets, iPad, Android Tablets, PC and Mac。甚至连大屏幕的 Kindle DXG 也是不支持的。所以说.....我不知道说什么好了。现实效果如下图所示：

where $\text{sigm}(\eta)$ refers to the **sigmoid** function, also known as the logistic function. This is defined as

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

The term “sigmoid” means S-shaped: see Figure 1.19(a) for an example of a sigmoid **function**, since it maps the whole real line to $[0, 1]$.

另外一个玩家就是苹果和 iBook，一贯了苹果的奇葩作风，各种限制。比如 iBook 不提供电脑上的阅读器，所以必须要买个 iPad 什么的设备才能看，然后用于制作 iBook 格式电子书的软件 iBook Authors 据说还有“制作的电子书不能在 iBook Store 以外的任何商店出售”这样的霸王条款。不过抛开这些不说的话，在技术方面 Apple 却比 Amazon 看起来要更有诚意的。比如 Apple 给 iBook 加入多媒体功能，可以播放视频、甚至是具有交互功能，虽然这些东西从技术上来说（比如，在浏览器中，甚至 PDF 本身也是支持许多多媒体和交互功能的）并不是什么新东西，但是 Apple 做了一步是把他们比较好地集成进电子书中来，并且给了实际的制作精良的多媒体 textbook 的例子，在 iBook 2 发布的那段时间网上可以看到各种评论称“Apple Reinvents Textbooks”。

我真正注意到 iBook 是从《Signal Processing for Communications》这本书宣布发布 iBook 版本开始的。一个截图如下：

里面的公式排版相当漂亮，虽然也不能随意改变文本的字体大小什么的，但是似乎也并不是像 Kindle 格式那样只是拿 PDF 来包装一下而已。

³ 单词间过于不均匀的空白这个还是以后再吐槽吧。

⁴ 小八卦一下，最近 JMLR 的 Editor-in-Chief 换届，MLPP 的作者 Kevin 当了新的 EIC。然后我也去做了新的 webmaster，人肉更新 paper 列表、元数据、RSS feed 什么的，虽然随手写了几个 ad hoc 的脚本来半自动化，但是比想象中的要更苦力一些。不过既然想要支持 open access，就总要有人来做事情嘛，像自己现在这种时候，能做一些 dirty work 也是觉得算是一点点力所能及的贡献吧，向着工业界的 Open Source World 一样的 Academia.....嘛，说得过于高尚了，哈哈。

Figure 3:



Figure 2.1
Triangular
discrete-time...

shown as the "triangular" waveform plotted in Figure 2.1; or

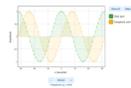
$$x[n] = e^{j\frac{\pi}{20}n} \tag{2.2}$$

which is a complex exponential of period 40 samples, plotted in Interactive 2.1. An example of a sequence drawn from the real world is

$$x[n] = \text{The average Dow-Jones index in year } n \tag{2.3}$$

plotted in Figure 2.2 from year 1900 to 2002. Another example, this time of a random sequence, is

$$x[n] = \text{the } n\text{-th output of a random source } \mathcal{U}(-1, 1) \tag{2.4}$$



Interactive 2.1
A signal

a realization of which is plotted in Figure 2.3.

这个文件里实际上还有 Interactive 的部分，比如左下的那个 figure，实际上是可以点开调整 figure 中信号的采样率从而看到不同的结果图的。此外，这个 portrait 模式中看到的结果实际上是 Text Reflow 之后的结果，在 Landscape 模式下我们看到的是这个样子：

Table 2.1 Basic discrete-time signal types.

Signal Type	Notation	Energy	Power
Finite-Length	$x[n], n=0,1,\dots,N-1$ $\mathbf{x}, \mathbf{x} \in \mathbf{C}^N$	$\sum_{n=0}^{N-1} x[n] ^2$	undef.
Infinite-Length	$x[n], n \in \mathbf{Z}$	Eq. 2.19	Eq. 2.20
N -Periodic	$\tilde{x}[n], n \in \mathbf{Z}$ $\tilde{x}[n] = \tilde{x}[n+kN]$	∞	Eq. 2.21
Finite-Support	$\tilde{x}[n], n \in \mathbf{Z}$ $\tilde{x}[n] \neq 0$ for $M \leq n \leq M+N-1$	$\sum_{n=M}^{M+N-1} x[n] ^2$	0

signal of length N is entirely equivalent to a vector in \mathbf{C}^N . This equivalence is of immense import since all the tools of linear algebra become readily available for describing and manipulating finite-length signals. We can represent an N -point finite-length signal using the standard vector notation

$$\mathbf{x} = [x_0 \quad x_1 \quad \dots \quad x_{N-1}]^T$$

Note the transpose operator, which declares \mathbf{x} as a column vector; this is the customary practice in the case of complex-

$n < 0$ or for length signals. The sequence notation is changeable; geometric notation is sequence notation of signal pro-

Finite-length signals arise in view, they are very efficient in the case of processing. However, to derive terms of finite lengths and extremely valuable sequence with more, the no-

Figure 4:

Figure 5:

有几点值得注意的地方，表格中求和符号 $\sum_{n=0}^{N-1}$ 那里上标的 $N-1$ 凸出去了，大概这就足以证明这并不是由 TeX 直接渲染的 PDF 的浅包装了吧，这样的结果或许是 MathML 渲染的结果。另一方面，注意到每行结尾的连字符，比如倒数第二行的 **vec-tor**，这是区别 TeX 之类的专业排版工具渲染的结果和 Word 之类的字处理软件导出的结果的一大特征：TeX 会在合适的地方插入连字符将单词断开，以实现行的右边看起来很整齐的情况下行内单词之间的空白也能达到平衡美观的效果。而 Word 等字处理软件由于需要实时排版，所以就没有实现这样的功能，于是可选的结果通常是“文本段落的右边边界长短不一”或者是“行内单词间的空白有时候会被拉得很大”这两种。现在几乎绝大部分的 epub 这种 reflow-able 格式的电子书的阅读器都采用“巨大空白”的显示方式，有时

候拉得太开就让人看着很不爽。比如上面的 portrait 模式截图中“which is a complex”一行的词间距就比下一行要大，再之前的 Kindle 截图的例子也可以明显看出不同行的区别来。我觉得，像 Word Processor 这样的要不停变来变去的不使用 hyphenation 就算了，像阅读器这种，一般计算资源足够的情况下（比如在电脑上的时候）打开的时候处理一下也不是什么大不了的事情啊，我之前用过一个 Linux 下的 epub 阅读器就支持 hyphenation。现在连 CSS 都开始支持 hyphenation 了，电子书阅读器们还停滞不前真是让人心寒啊。

公平起见我还是提一句其实苹果的 iBook Store 里也有像 Amazon 那边的那种直接把公式转成图片就拿来买的书。不过上面的这个例子展示了排版 reflow-able 的带公式甚至是多媒体和交互内容的电子书是完全可行的：在合适的屏幕尺寸下（这里是 iPad 的 landscape 模式）就以原始的结构显示出来，甚至连 hyphenation 都保留下来了多少也让我有点怀疑是 PDF 呢；而在不同的屏幕下则进行 reflow，Apple 还玩了一点小 trick 就是在 reflow 的时候把图和表都放在旁边的 margin 里面去，并且可以点击放大，这样就不会出现图文自动混排难以处理得很好的问题，同时又看起来也比较美观。

事实上，就《Signal Processing for Communications》这本书而言，Amazon 上的纸质版价格是 80 刀，而 iBook 上的 interactive 版只要 14 刀左右，所以说采用电子书的形式降低成本把科技类图书的价格放到学生们比较能承受的范围内的可行性应该还是有的。而且电子书还有诸如可以实现自动更新 errata 之类的功能。

总而言之希望以后能够出现一种标准格式可以很好地解决电子书中的各种问题，充分发挥出电子设备自己应有的优势来吧！比如至少做到像现在的电脑绘画一样，虽然还是完全无法取代纸笔，但是也有纸笔所不能做到的非常好用的诸如图层、随意放大缩小、限制绘画区域等等各种功能。

不过这只是理想的世界，我们往往会忘记现实世界中其实大部分的问题都可以归结为“历史遗留问题”，就电子书而言，要做得好的话，就必须得同时考虑兼容已经有的电子书格式。偷懒一下可以简化为 PDF 和 Djvu 两种。PDF 一直以来都是精致排版的电子书的首选格式，并且学术界的论文什么的也都以 PDF 的格式进行电子存档；而 Djvu 的强项则在于扫描电子书，一本做好的 Djvu 扫描书如果转成 PDF 的话不仅体积会放大 N 倍，而且显示效果也会变差。

比较理想的情况是有一个统一的阅读器可以同时支持这些主流的格式，进行阅读、标注之类的。这一点在电脑上的话，有 KDE 的 Okular 做了一个示范。至于好用不好用嘛，很大程度要看个人喜好了。Okular 有个特点就是标注并不是存储到文件本身的，比如标注的 PDF 在其他 PDF 阅读器里打开是看不到的。当然有一些格式本身就不支持标注笔记，所以这样做也可以理解。另一个原因待会会再说到。

最后不管是过去的格式还是未来的格式，还有一个重要的问题就是数



Figure 6:

字版权管理了。我对这方面并不是很了解，所以就胡乱设说一下，如果不考虑什么知识自由啊什么之类的，仅仅从出版商和零售商的角度来看的话，DRM 做好了，如果可以让它们没有多大顾忌地像出版和销售现在的纸质书一样来对待电子书的话，应该也是一件不错的事。并且电子书自己的特点也可以产生出许多新的使用方式。比如现在 **Boston Public Library** 据说就可以借 Kindle 电子书。在学校的图书馆的感觉就是，虽然各种好书都有，但是通常就只有一本，长年被借出中，而且后面有人在排队预约的话，借到的人也没有几天可以看的；如果是一些更热门的或者教科书什么的，则有可能被放入 **reservation stack**，通常借出两小时之内必须还的，超时罚款罚得相当重。

如果是电子书的话，也许可以找到一个比较好的运营模式来解决这个问题。而且 Google 前些年扫描了那么多那么多的图书制作成电子版放在 **Google Books** 上面，扫描效果相当好，有许多书还做了 OCR 甚至连目录的超链接都做好了，现在这些书通常都可以直接在 **Google Book** 上预览到一部分。如果能想一个办法把版权方面的问题解决了，让 **Google Books** 可以和各个图书馆合作起来，比如如果你这个图书馆是拥有这本书的硬拷贝的，那么就可以从 **Google** 那里拿到这本书的电子版再然后以电子版的形式借给该图书馆的读者们。那真的是一件大好事呀。因为许多以前出版的书大概只有 **Google** 那边才有比较精良的电子版的了。嘛，这个美好愿景，也许至少在很多很多很多很多年以后，那些书的版权全部过期以后应该是可以实现的了。

YY 的话题告一段落，接下来再来聊一下现在的阅读设备和相应的配套软件的问题。现在的阅读器大致可以分为三类：**Kindle**（包括 **Nook**）、**Android** 平板（包括 **Kindle Fire**、**Nook HD** 以及各种 **general** 的 **Android** 平板）、**iOS** 平板（**iPad**、**iPad Mini**）。

首先是 **Kindle**，简单地来说就是两条：1. 这真是个好东西；2. 这不适合

academic reading。一方面来说，现在的 Kindle 不管从重量、便携程度、电池、设备的价格、后端 Amazon 书城的支持还是（最重要的）阅读体验方面都非常赞。唯一需要吐槽的大概就是 Amazon 的云端书库管理界面了.....然而这只是针对非科技类书籍而言的，换句话说以文本为主的书籍。

技术类书籍的特点就是虽然 iBook 给我们证明了不用 PDF 格式也是可以做出比较精美的带复杂公式、图表的电子书来，但毕竟也还未成为主流，再就是一大堆的以前出版的书都只有 PDF 电子版或者是扫描版，并且是大约 A4 左右大小的。

现在的 Kindle 尺寸加上外壳还没有 letter paper 的一半大，全尺寸的 PDF 直接显示出来.....当然，还是可以勉强辨认的，上图中两边是同一本书的电子版和纸质版对比。如果只是临时看一下的话，其实就连 iPhone 那么小的屏幕都是可以接受的。但是如果打算长期用这个东西看 paper 之类的.....我觉得不要说长期，连 5 分钟我都会坚持不下去，眼睛会累死。

有一个选项就是把屏幕横过来，但是由于 Kindle 这样的 e-ink reader 切换屏幕的特点，在同一页上下滚动的时候也会很痛苦，而且 Kindle 原生系统对横排也没有什么特别贴心的特殊支持。或者可以装多看，切边呀、横版什么的可能支持会好一点，但是也不会有特别大的改观吧，而且 E-ink 目前的大问题还是在于翻页吧，像小说之类的读物基本上就是看完一页往下翻一页这种基本阅读顺序就 OK 了，但是科技文献的阅读特点是经常要翻来翻去的，什么公式 5 呀、定理 7 呀之类的，用 E-ink 就比较痛苦。

Kindle 家族还有一个快要被遗忘的成员就是大屏幕的 Kindle DXG, Amazon 很早就没有把它放在 Kindle 展区里了，现在去看似乎都没有卖的了。按理说 9.7 寸的屏幕应该看科技类文献是很合适的，但是 Amazon 似乎对此没有兴趣，大屏幕的 e-ink 翻页迟滞更加明显，而且 DXG 不管是从硬件还是从软件上都相当古老了，之前我们看到的 Amazon 上出售的 Kindle 电子书的那个不支持小屏 kindle 的例子，里面根本就没有提到支持 DXG，猜测估计是因为 AZW4 格式比较新，DXG 的系统不支持？其实现在的 Kindle 在“轻”这条路上已经走得相当出色了，定位目标就是小说类的“轻读物”，而一个 Kindle PW 的重量大概也就跟一个手机差不多的，居家旅行各种方便，所以像 DXG 这个另类的产品得不到关怀也不是什么难以置信的事了⁵。

结论是：Kindle 是个好东西，不过目前并不适合用来做 academic reading。所以接下来我们来考察平板。从硬件上来看现在的各种平板之间应该也都差不多，比如和 E-ink 屏幕比起来会各种反光啊、看起来眼睛更累啊之类的，目前也只能忍一忍了。而苹果虽然有名字很响亮的 retina 屏技术，但是现在其他的平板也都达到了各种高分辨率，就文字显示的 sharp 程度来说都是非常厉害了。

实际上我在接触平板之前是抱有相当美好的期待的，因为平板的触屏

⁵ 啊？事后诸葛亮？有没有听说过一本书叫《Everything Is Obvious》，副标题是《Once You Know the Answer》。

8 The forecaster's goal is to keep as small as possible the cumulative regret (or equivalently regret) with respect to each expert. This quantity is defined, for expert E , by

$$R_{E,n} = \sum_{t=1}^n (\ell(\hat{p}_t, y_t) - \ell(f_{E,t}, y_t)) = \hat{L}_n - L_{E,n},$$

where we use $\hat{L}_n = \sum_{t=1}^n \ell(\hat{p}_t, y_t)$ to denote the forecaster's cumulative loss and $L_{E,n} = \sum_{t=1}^n \ell(f_{E,t}, y_t)$ to denote the cumulative loss of expert E . Hence, $R_{E,n}$ is the difference between the forecaster's total loss and that of expert E after n prediction rounds. We also define the instantaneous regret with respect to expert E at time t by $r_{E,t} = \ell(\hat{p}_t, y_t) - \ell(f_{E,t}, y_t)$, so that $R_{E,n} = \sum_{t=1}^n r_{E,t}$. One may think about $r_{E,t}$ as the regret the forecaster has for not having listened to the advice of expert E right after the t th outcome y_t has been revealed.

Throughout the rest of this chapter we assume that the number of experts is finite ($i = 1, 2, \dots, N$), and use the index $i = 1, \dots, N$ to refer to an expert. The goal of the forecaster is to predict so that the regret is as small as possible for all sequences of outcomes. For example, the forecaster may want to have a vanishing per-round regret, that is, to have

$$\max_{i=1, \dots, N} R_{i,n} = o(n) \quad \text{or, equivalently,} \quad \frac{1}{n} \left(\hat{L}_n - \min_{i=1, \dots, N} L_{i,n} \right) \xrightarrow{n \rightarrow \infty} 0,$$

where the convergence is uniform over the choice of the outcome sequence and the expert advice. In the next section we show that this ambitious goal may be achieved by a simple forecaster under mild conditions.

The rest of the chapter is structured as follows. In Section 2.1 we introduce the important concept of a weighted average forecaster.

交互方式非常适合于做 annotation 呀、即时查词典啊什么的——再普通不过的要求了吧？结果我满怀期待地入了一个 Nexus 10，然后各种失望。当然主要原因大概还是因为这是我第一次接触平板，所以期望过高的缘故。

因为平板毕竟不是一个“缩小号的电脑”那么简单嘛⁶，多任务之间切换之类的也略有点不方便，而且各种操作上（比如 PDF 渲染）性能也不及电脑。当然更让我崩溃的还是软件方面的问题。

作为一个 Android 新手，不了解行情，所以我最先尝试的当然是官方的 Adobe Reader，免费软件，并且还支持 annotation，不过就是选择文本进行高亮的时候精度似乎是以行来计算的.....一开始我还以为是是由于平板设备计算资源比较低，没有办法在 PDF 里精确定位所以不得不搞成这个样子，于是也就忍了。后来 Adobe Reader 很快就变得越来越慢，于是我开始寻找 Alternative。

在之前做 survey 的时候就听说 Android 市场的 App 并没有 Mac App Store 那么丰富，不过近几年也有越来越好转的倾向，搜索了一下 PDF 也有许许多多的可选项。我自己总结了一下自己的需求主要是：

- Annotation: 高亮、下划线、文本注释⁷
- Reading: 渲染正常，翻页迅速
- 手动或自动的切边功能，或者翻页的时候 zoom-lock
- 选中文本查词典或者 web 搜索

其实我还一直想能在浏览器中显示 PDF 文件，好让我在决定下载之前先看看是什么样子是不是我想要的那个文件，不过完全没有找到可用的插件。回到 PDF 阅读，我自己尝试过的大概有以下一些（还有一些打开了一下就删掉了都不太记得了）：



- qPDF Notes: 比较清爽的一个阅读器，不过标注的时候选文本 highlight 居然是用一个矩形框来选的，太不科学了。而且不知道为什么我脑子进水了会买了这个 App。
- RepliGo Reader: 高亮功能超级好用，就是长按就会出现 Android 系统的那种选择文本的 indicator，拖动两边选好要高亮的文本然后选择高亮就可以了。同样的方法还集成了 ColorDict 进行查词以及 web 搜索之类的功能。

实际上我是尝试了这个 App 之后才突然对 Adobe Reader 感到非常无语的，因为选词的精度明明可以做到非常高。美中不足的是没有切边之类的功能，不过最不能忍的还是翻页的时候切换到下一页或者上一页总是会先显示出一个模糊的页面，大概等半秒左右才会清晰，如果是图片的话这样当然还不错，一开始给显示个大概，但是文档就很不

⁶ 如果像 Microsoft Surface Pro 那样用 Intel 架构，就会出现惨不忍睹的电池只能用五个小时的情况，那还不如直接带个笔记本电脑算了。

⁷ 作为强迫症患者我希望自己标注的高亮和下划线都是整齐地沿着文本的，所以只有 freehand drawing 功能的标注而没有文本选择的方式进行高亮的那些假冒伪劣 annotator 是不予考虑的。

Figure 8:

一样了，这样感觉眼睛非常累，其实一开始也觉得能忍，同样地是因为发现有其他的阅读器在翻页方面做得非常好，就觉得这个太不科学了。平板再弱 `preload` 前一页和后一页的资源应该还是有的吧？就算退一万步说没法做 `preload`，我宁愿看到从白屏切换到文字都不想看到从模糊到清晰的那个过程，感觉每次翻页我的大脑内部的 `deblur` 模块就被带动起来疯狂运转。

- **ezPDF Reader**: 提供了一个 `Free Trial`，这货给人的第一感觉就是一个大杂烩，一堆功能胡乱堆起来的感觉。总的来说文本选择还算比较好用，也集成了查词典、标注功能，而且有连续标注的选项，就是打开标注工具之后就可以到处画高亮了，对于高亮重症患者大概是比较实用的。

切边和列缩放锁定功能也是有的，但是各种奇怪，比如好像锁定之后就不能用 `swipe` 手势来翻页，之类的，就是各种自己的功能互相之间也不兼容的感觉。

- **iAnnotate PDF**: 这个东西号称是 iOS 上的神器，在 Android 系统上目前还是个免费软件，大概是还没做完，反正我试用的结果就是连正常显示都不行。
- **EBookDroid**: 免费开源无广告，同时支持 PDF 和 Djvu，切边功能很好用，翻页的时候渲染也非常迅速。而且还能在 Chrome 里直接打开 PDF，也算是一种形式的浏览器 PDF 插件了，不过需要小心的是如果你是 Google Search 然后点开了一个 PDF，由于 Google Search 的结果是先到 Google 的页面然后再跳转到结果页面的，所以当你从 EBookDroid 切换回 Chrome 的时候 chrome 会后退到上一页，也就是 Google 的跳转页，然后一下子又会被自动跳转，于是又打开了 EBookDroid，如此反复。而且 Chrome 还非常坚挺，把它关掉之后再打开它还会自动载入上次的页面，于是.....我忘记当时是怎么解决这个问题了。

EBookDroid 的问题在于不支持 Annotation。另外在缩放的时候缩放中心似乎是不对的。

- **Mantano Reader Premium**: 算是各个 Reader 里卖得最贵的了，要 7 块钱左右。它有一个 Lite 的带广告的试用版，除了带广告之外功能也不太一样，而且 Lite 版更加容易 Crash 一些。Google Play 的 Appstore 有一个比较贴心的地方就是你可以试用 App，购买 15 分钟内如果不满意的话都是可以全额退款的。不过每个 App 只能试用一次。

Mantano 不愧是最贵的 Reader，平均来说也是我觉得最好用的了。他同时支持 PDF 和 epub，切边、翻页、缩放都工作得非常好，标注和查词典的 `text selection` 也是比较好用的方式实现出来的，就是会经常 `crash`.....=.=bb 在我这里大部分 PDF 第二次选文本的时候就会瞬间退回书籍列表页面。不过更新也比较勤快，`crash` 的问题应该最终会被修复吧。

最后 Android 的软件市场给我的感觉就是处在乱世纷争时期，如果是大家都比较弱也就算了，每个功能基本上都有 App 能做到很好，但是却没有集大成者，就让人很崩溃。虽然说 Android 在各方面都比 iOS 要开放许多，比如程序之间数据共享也很方便，直接在 Dropbox 的 App 里浏览 PDF，然后在其他 PDF Reader 里打开，进行标注，然后保存回 Dropbox 自动同步到云端这都是很轻松的事情。iOS 的话就每个程序各是各的，要在外部程序中打开 PDF 的话就是各自有一份各自的拷贝了。简单来说，就是“前途一片光明，目前正处在社会主义初级阶段”的感觉：p。

接下来要讲 iOS，需要声明的是我既不是果粉也不是果黑，在网上打口水仗争论 Emacs 还是 vim、Linux 还是 Windows 的日子已经过去了，在我自己用电脑的过程中随着自己习惯的改变也在所使用的工具上会慢慢地变化，这个过程让我认识到的一点是：这些争论根本就没什么意义，也就大家吵吵架好玩了。因为每个工具也都有它自己的优缺点，而更重要的是，不同的人，或者甚至同一个人在不同的时间、地点会有不同的需求，有些情况下这个工具更合适，另一些情况那个工具更合适，这是非常正常的。所以，我们再回到 Academic Reading 上来谈论问题吧。

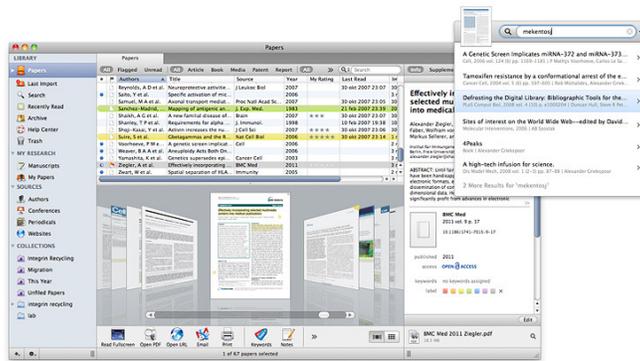


Figure 9:

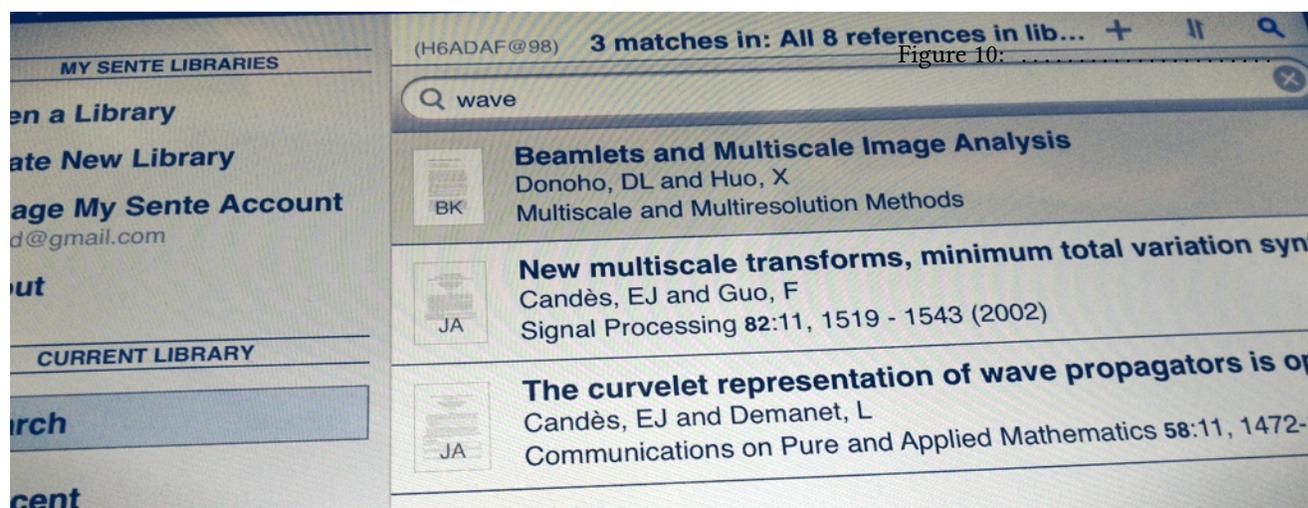
在论文阅读这一块，苹果用户的让人口水的东西是有名声很响的参考文献管理工具，一个是 Papers，另一个是 Sente。上面就是 Paper 的官方界面截图。Sente 也是一个非常类似的东西。他们都是参考文献管理工具，能够把你的论文集中管理起来，管理内容包括 PDF 文件本身、各种元数据、还有 annotation 和 notes，还提供实时搜索等功能。另外就是他们都提供 iPad 版本，允许你通过 Wifi (papers 的方式) 或者云端 (sente 采用的方式) 进行数据同步，实现在 pad 上阅读和标注论文。

论文多了以后依靠元数据和全文检索的管理和搜寻功能应该就非常重要了，其实我也一直觉得这是做研究非常重要的一环，只是一直也没有适应这种方式，虽然有尝试过 Mac 下的 Papers，但是里面也就一直只有四五篇论文而已，-.-bb 很多时候想找一篇自己以前看过的论文里面的一个什么东西一时又想不起来是哪篇论文的时候，就觉得管理和积累的功能还是相当重要的。

不过不论是 Papers 还是 Sente 都是相当贵的，Mac 和 iPad 版合起来估

计要 100 刀左右吧。好消息是 Sente 现在正在改版，根据 Sente 的 Blog 介绍说从 6.6 开始要将软件免费化了，取而代之采用云端收费的机制：免费用户可以最多有 250M 的云端存储并且有 100 个条目的限制，花大概 30 块可以永久升级到 1GB，如果还嫌不够的话，可以以每年 5GB, 20 块左右的价格租用更多的空间。总的来说是可以免费体验一把的。

Mac 版的软件没有什么好说的，和 Papers 的风格、流程都不太一样，但是基本上做的是同一件事情。iPad 端的 App，由于 iOS 似乎对于程序间数据共享卡得比较死，所以基本上这种管理类型的 App 都必须使用自己内置的阅读器。Sente 的阅读器还算不错吧，iOS 上的各种 PDF 阅读器从整体上来说都比 Android 上要上好上许多的感觉，不管是渲染速度还是可操作性来讲。然后标注功能也很好用，不过并不支持切边之类的功能，而我在电脑上用 Skim 将 PDF 切边之后再放到 Sente 中的结果就是：标注的时候严重错位.....嘛，总的来说还是挺不错的，如果有设备的话不妨自己去体验一下。



由于我在电脑上还没有习惯用这种工具来管理论文，在他的阅读器变得强大起来之前我也还不想专门花钱在这上面，于是免费的 250M 空间就不太适合连电子书也一起放在里面了，既然电子书要另外的 App 来阅读的话，还不如索性论文都采用这样的方式来处理。所以我暂时是没有采用 Sente 的。至于 Papers 嘛，由于没有提供 iOS 的试用版，而价格也不便宜，再加上 App Store 上最近一片一星差评，我就不想去花冤枉钱了.....

需要注意的一点是，不论是 Papers 还是 Sente，他们的标注都只能在他们的 App 中看到，而不是存储在 PDF 中的。之前 Okular 这样搞的原因是因为需要支持不同的格式。而这里的 App 们这样搞应该是出于数据同步的考虑，因为如果把标注写到 PDF 中作为一个整体的话每次同步起来相当费劲⁸，而分开来的话，PDF 文件本身是基本不会变化的也就不需要同步，只同步 annotation、comments 之类的话就会节省很多资源而且迅速得多。从用户的角度来说的话，喜不喜欢就全看个人爱好还有你对

⁸ 我有次在 Mac 下用 Preview 对一个扫描版的 20 MB 的 PDF 做了一行高亮再保持，一下子就变成了 100+ MB.....

该软件的依赖程度了。

对于我来说，阅读的体验明显是比管理的体验有更高的优先级的，因为我本身就有点手工管理各种文件的强迫症。不过即使如此，电脑和阅读设备之间的数据同步也是非常重要的，这一方面现在 Dropbox、box.com、Google Drive、SkyDrive 之类的各种工具都可以做到。由于不希望让自己的文件夹乱得一团糟，也想精确控制哪些地方同步哪些不同步，于是支持软连接的 Dropbox 目前是最喜欢的。不过在 iOS 上由于没有程序之间的共享文件系统，所以用 Dropbox App 来配合其他 Reader App 的操作就不太好做，于是各个 App 都纷纷实现了直接管理和同步各大云端存储商的文件的功。接下来是 iOS PDF Reader 集体卖萌时间。



Figure 11:

首先是 Adobe Reader，免费，而且和 Android 上的那个 Adobe Reader 一样烂，文本选择的精度在一行以上.....然后是大名鼎鼎的 iAnnotate PDF，名声非常响亮，号称是 PDF Annotation 的神器，售价十块钱，也不便宜。然后我看了一下介绍和各种 review，基本大意就是这玩意有十八般兵器七十二种变化，各种你想得到的想不到的 annotation tool 你有我全都有，然后我觉得肯定用不上还会被玩晕，就没有去斗胆尝试。

GoodReader 售价 5 刀，各种功能都有，支持同步 Dropbox 在内的各种云存储，也可以直接通过 wifi 同电脑传输文件，PDF 阅读、标注、切边等功能都非常正常，也能多 tab 同时打开多个文件。而且界面也比较清爽，给人一种“just works”的感觉。如果说有什么缺点的话，就是同步是必须要去手动点的，而且同步是在前台，于是同步的时候不能做其他事情。其实也不算什么缺点，像 Dropbox 的官方 App 其实更类似于一个 web 程序，文件都是要打开的时候才下载过来的，所以并不是以同步的机制在工作；而其他比如 PDF Expert 虽然是可以自动同步，但是也有选项可以关掉换成人肉同步，因为这样会省点许多。

PDF Expert 售价 10 刀，Good Reader 有的功能都有，如果有什么特别的地方的话，就是多了一些比如 PDF 页面管理、Signature 管理什么的功能，另外就是同步可以是自动的。阅读的时候并没有切边功能，但是却可以 zoom-lock，lock 之后还是可以用两个手指来缩放和移动，但是一个手指就是直接翻页，并且翻页之后还是保持原来的位置和缩放比例的。我觉得切边或者 zoom-lock 都挺好用的，碰到一些奇怪的 PDF 并不能所有的页面统一切成一致的大小（比如我有碰到过因为插图的原因中间有一页是横过来的 PDF），页面单独切边的话又会变得参差不齐，而 zoom-lock 随时都可以方便地改变缩放和位置，所以大概会方便一点。

实际上，PDF Expert 是由一个叫做 riddle 的公司出的，他以前有一个叫做 ReaddleDocs 的产品，能够管理和浏览 PDF 之外的图片啊、Office 文档啊、视频音频之类的各种文件，也是收费的。最近他把 ReaddleDocs

升级了一下，名字改成 Documents，然后免费了。直接给一张官方的截图吧：

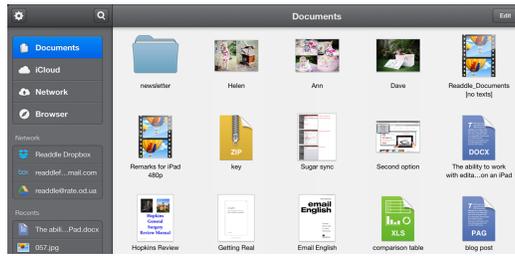


Figure 12:

界面看上去跟 PDF Expert 差不多的 (文件夹的图标不太一样), 连 PDF 阅读器也是用的 PDF Expert 的引擎, 所以相应的功能也都有而且也很好用, 唯一一点就是不支持多标签同时打开多个 PDF。不过 Documents 能处理 PDF 之外的各种文件类型, 还是免费的, 也没有任何广告.....按照他们公司 blog 上的说法, 他们靠公司的其他产品已经能各种盈利了, 所以这个产品现在的唯一目的是吸引用户, 所以免费并且无广告 (而且还确实非常好用), 他们的目标是一亿用户 (100 Million, 不知道我有没有算错?), 然后这款 App 会永远免费, 以后可能会加入付费才能使用的功能, 但是目前已有的功能将会一直免费存在。真想请少校来题一句“Readdle 是个好公司”! :p

最后提一下一个叫做 Stanza 的阅读器, 我是从别人那里听说的, 免费, 而且支持包括 PDF、Djvu 在内的各种文件格式。不过上 App Store 里看到各种恶评如潮, 基本上就是说无法在最新系统上运行起来, 看了一下最新的一版是 2011 年底发布的, 大概曾经是一个很流行的 App, 后来不知怎么就停止开发了。

这样一来 Djvu 阅读的问题就还是没有解决。虽然 App Store 里搜索结果一大堆, 但是也不知道哪个好用, 而且如果用不同的 App 的话同步电脑的时候除非把 PDF 和 Djvu 格式的电子书分开放, 否则就要有两份 copy 存在了。



嘛, 先不管了, 最后再说一下论文管理的问题, 既然决定了用 Dropbox 来同步, 就得要自己管理论文了。我一般是按大的类别分一下, 然后下面按照期刊会议名 + 出版年 + 论文标题的方式来命名文件。有个可以玩的 trick 是在电脑上用论文管理软件来管理, 像 Papers、Mendeley 之类的软件都有提供论文 PDF 文件自动重命名的功能, 导入的时候就按照某个规则将 PDF 重命名然后归类到某个文件夹下面。好玩的是没有那个工具提供以论文类别来命名子文件夹的选项, 比如我想把 optimization 的论文和 regularization 的论文放在不同的文件夹下面就不行。搜索一下会发现 Mendeley 的 feedback 页面 2009 年的时候就有人提这个需求了, 并且 N 多人投票赞成, 回复是我们已经开始考虑这个功能了, 结果.....=. =bPapers 有个按 editor 来归类的选项, 于是我就把 editor 这个域

直接当做 categorization 来用了，然后再建立一些 smart collection 就按照 editor 来划分，也算是一个 workaround 吧。

Kindle HD、Nook HD 之类的设备都是基于 Android 的，就不专门讨论了，因为 Kindle App 本身是可以按照在人和 Android 系统上的，而其他 Kindle Store 上的 App 应该也大都可以直接安装到 Android 上。在接触了平板设备之后我的一点感受是，似乎 App 生态圈甚至比设备的硬件参数要更重要一些。