

# Useful Software/Tricks You Should Know

Wittawat Jitkrittum

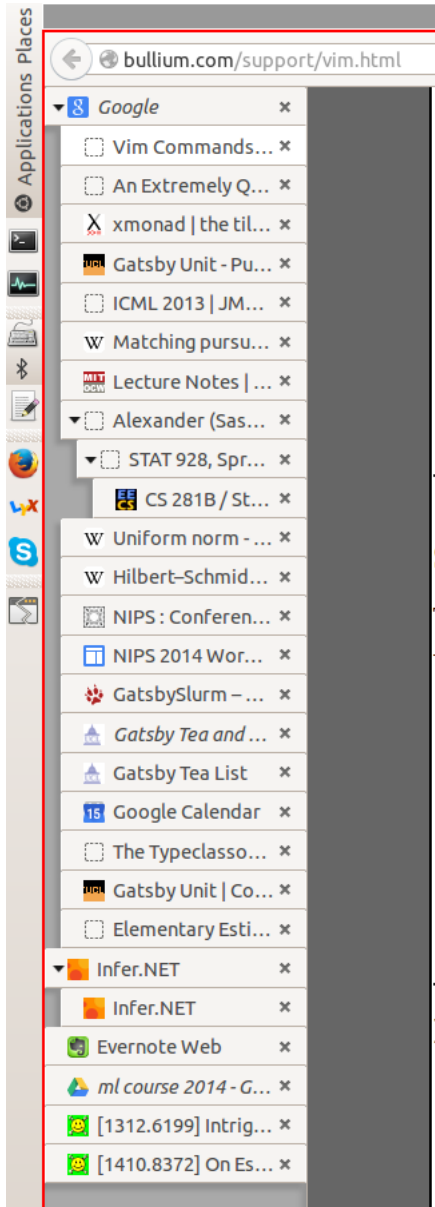
Gatsby Tea Talk

19 May 2015

# Gmail tricks

- Filter messages like these
- Labeling
- Priority inbox
- Advanced search  
(<https://support.google.com/mail/answer/7190?hl=en>)
  - from:someone@example.com
  - filename:pdf (has a PDF as an attachment)
  - label:ml-news

# Firefox Plugin: Tree Style Tab



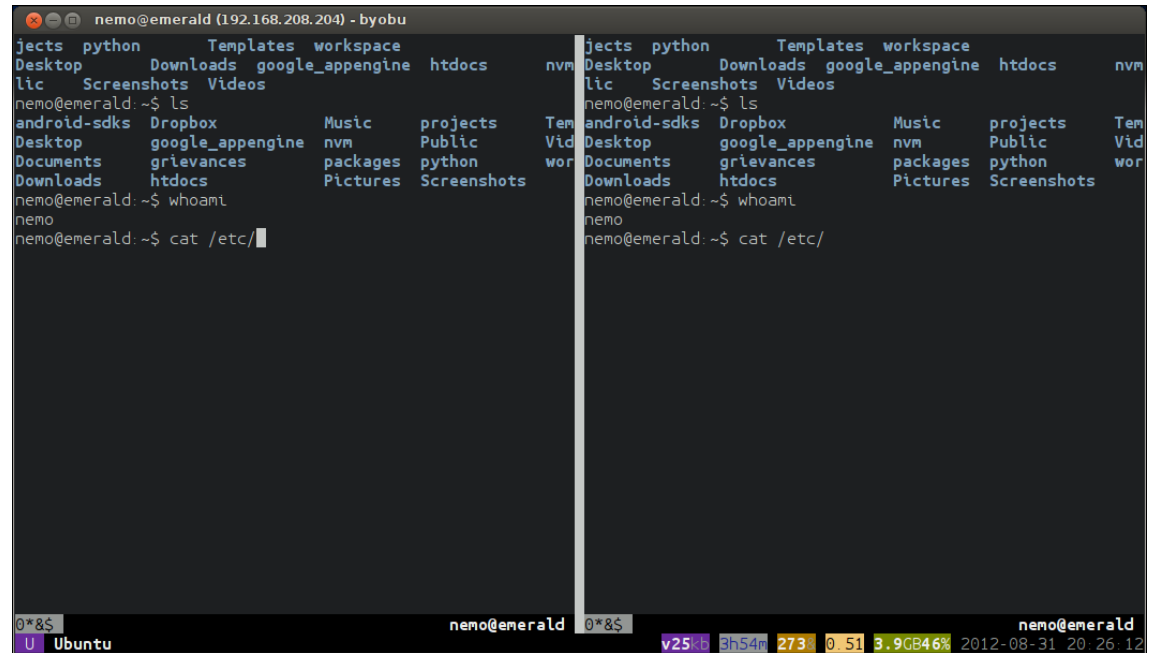
- **Tree Style Tab**
- Useful when you open too many tabs
- Display tabs vertically
- Many similar plugins for Chrome.

# latexdiff

- See the difference of two latex files in pdf.
  - latexdiff old.tex new.tex > diff.tex
  - Compile diff.tex
- Addition. Deletion.

|     |  |   |     |
|-----|--|---|-----|
| 099 | due to the use of non-linear statistics (Gunn, 1994; Abril, 1994; Joyce & Marjoram, 2008; Aeschbacher et al., 2012). | in our measuring ABC to measure the similarity                  | 146 |
| 100 |  | <del>between summary statistics, but now operate it on</del>    | 147 |
| 101 |  | <del>MMDs between data — we on top of MMD to</del>              | 148 |
| 102 | To this end, we introduce a method that does not   | obtain an effective measure of similarity between sim-          | 149 |
| 103 | require an explicit selection of summary statis-   | ulated and observed data. For this reason, we refer             | 150 |
| 104 | tics. Our method proceeds by applying a simi-  | to our method as <i>double-kernel</i> ABC, briefly, <i>K2-</i>  | 151 |
| 105 | ilarity measure to data themselves, via embed-   | <i>ABC</i> .  | 152 |
| 106 | dings of the empirical data distributions into   | <del>The kernel embeddings in our approach imply that</del>     | 153 |
| 107 | an infinite-dimensional (characteristic) reproducing   | <del>we take into account infinite-dimensional summary</del>    | 154 |
| 108 | kernel Hilbert space (RKHS). Such embedding cap-   | <del>statistics when measuring similarities between data.</del> | 155 |
| 109 | tures all possible differences between data distri-  | <del>This, however, raises another potential problem.</del>     | 156 |
| 110 | butions, e.g., all the high-order moments that are   | <del>There are two levels of approximation in ABC</del>         | 157 |
| 111 | necessary to describe the distributions: no infor-   | <del>— one that results from using an insufficient</del>        | 158 |
| 112 | mation loss occurs in going from the posterior   | <del>statistic, and another arising from the estimation</del>   | 159 |
| 113 | given data $p(\theta y^*)$ to that given embeddings of data  | <del>of the true partial posterior using samples. The</del>     | 160 |
| 114 | $p(\theta \mu(y^*))$ , regardless of the form of $p(y \theta)$ .   | <del>latter approximation is inefficient when</del>             | 161 |
| 115 | Embeddings of probability measures into  | <del>A dilemma</del>  | 162 |
| 116 | <del>reproducing kernel Hilbert spaces (RKHS)</del>  | <del>faced by practitioners of ABC stem from the</del>          | 163 |
| 117 | <del>RKHSs</del> (Sriperumbudur et al., 2010) offer a flexible   | <del>following two problems. First, it is unclear what</del>    | 164 |
|     |  | <del>set of summary statistics is sufficient for a given</del>  |     |
|     |  | <del>problem. A naive solution is to proceed by adding</del>    |     |

# screen / byobu / tmux



The screenshot shows a terminal window titled "nemo@emerald (192.168.208.204) - byobu". The terminal displays the output of the 'ls' command, showing a directory listing with various folders and files. The output is as follows:

```
nemo@emerald: ~$ ls
Desktop  Downloads  google_appengine  htdocs  nvm
lic      Screenshots  Videos
nemo@emerald: ~$ whoami
nemo
nemo@emerald: ~$ cat /etc/
```

The terminal also shows the output of the 'whoami' command, which returns 'nemo'. The terminal window has a dark background and a light-colored text. The bottom of the window shows the Ubuntu logo and system information, including the version 'v25.06', memory usage '3h54m', CPU usage '27.3%', disk usage '0.51', and system load '3.9GB46%'. The date and time are '2012-08-31 20:26:12'.

- Text-based terminal multiplexer
- Shortcut reference
- Cheatsheet
- <https://gist.github.com/henrik/1967800>
- Detachable. Still run after logging off.
- Can attach the same screen back after logging in.
- Useful for remote log in.

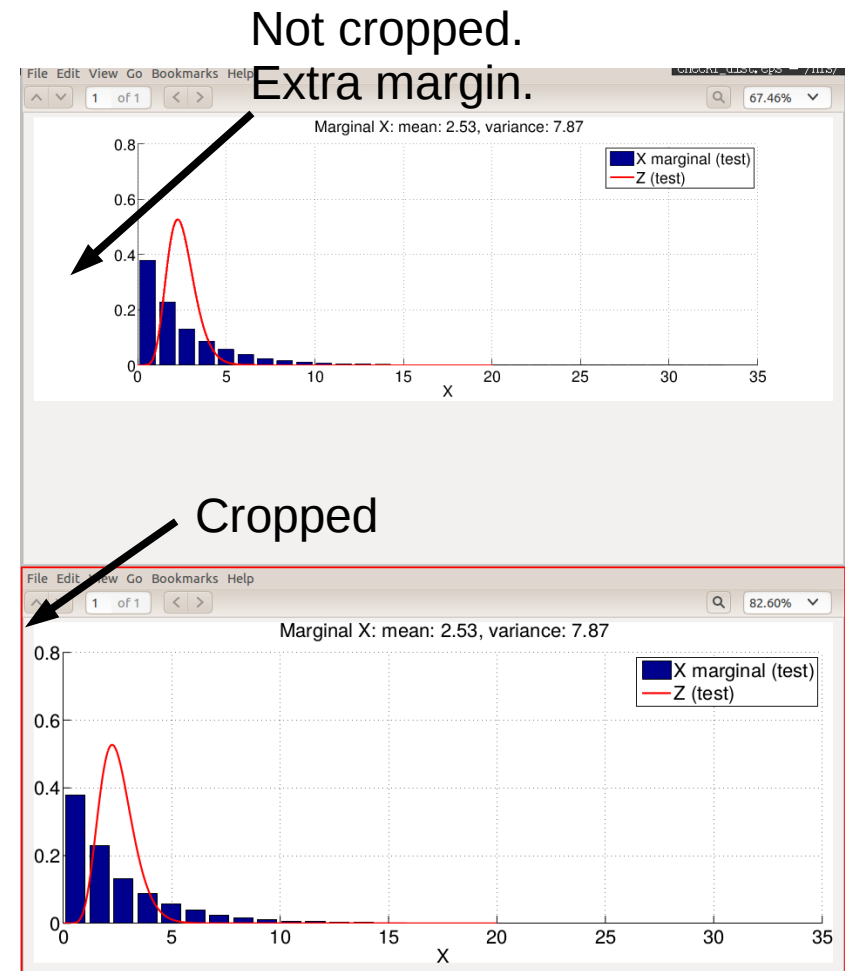
# EPS Crop Script

- EPS figures usually have extra margin.
- Script
  - Remove extra margin from all EPS files in current directory.
  - For each Plot.eps, produce
    - Plot-crop.eps
    - Plot-crop.pdf

```
$ cat processeps.sh
```

```
#!/bin/bash
```

```
for f in $(find $(pwd) -maxdepth 1 -type f -name "*.eps" -not -name "*-crop.eps")  
do  
    ps2pdf -dEPSCrop $f  
    pdfcrop $(basename $f ".eps")".pdf"  
    pdftops -eps $(basename $f ".eps")"-crop.pdf" $(basename $f ".eps")"-crop.eps"  
done
```



# Lyx

- Latex-backed document processor with graphical interface.
- Can be exported to Latex.
- Fast for typing equations.
- Work with Latex template e.g., NIPS template.

The screenshot shows the Lyx editor window with a document containing the following text and equations:

$f_i = \frac{\dots}{q^{\mathbb{V}}(\theta)}$ . (1,eq:ep\_factor\_update)

In the EP literature,  $q^{\mathbb{V}}$  is known as a **cavity distribution**.

**Why Exponential Family ?**  
Assume  $q$  is in the exponential family,

$$q(\theta|\eta) = h(\theta) \exp(\eta^\top u(\theta) - A(\eta))$$

where  $u(\theta)$  is the sufficient statistic of  $q$ ,  $\eta$  is the natural parameter and  $A(\eta) = \log \int d\theta h(\theta) \exp(\eta^\top u(\theta))$  is the log-partition function. It can be shown [foot 1](#) by expanding the definition of KL divergence, taking the derivative with respect to  $\eta$ , and setting it to 0. That  $q^* = \text{proj}[p]$  satisfies  $\mathbb{E}_{q^*} [u(\theta)] = \mathbb{E}_p [u(\theta)]$ . That is, the projection of  $p$  onto the exponential family is given by  $q^* \in \text{ExpFam}$  that has the same moment parameters as the moments under  $p$ . This procedure is known as **moment matching**. For example, in one-dimensional case, if  $q$  is chosen to be a Gaussian distribution, then  $u(\theta) = (\theta, \theta^2)^\top$ . Hence, the projection of  $p$  amounts to finding a Gaussian  $q$  that has the same first two moments.

**Message View of EP**  
Besides factor-based treatment, one can view EP as a message passing algorithm. In fact, EP can be regarded as a generalized loopy belief propagation [citep{Minka2001}](#). Under the approximation that each factor fully factorizes, the factor update in Eq. [Ref: eq:ep\\_factor\\_update](#) can be rewritten as

$$m_{f_i \rightarrow \theta}(\theta) = \frac{\text{proj} \left[ \int dX_{\setminus f_i} f(X) m_{X \rightarrow f_i}(X) m_{\theta \rightarrow f_i}(\theta) \right]}{m_{\theta \rightarrow f_i}(\theta)}. \quad (2,eq:ep_msg_update)$$

where  $m_{X \rightarrow f_i}(X) = \delta(X - X_0)$  and  $X_0$  is the observed value of  $X$ . The estimated posterior then becomes  $q(\theta) = \prod_{i=1}^m m_{f_i \rightarrow \theta}(\theta)$ . For comparison purpose, a belief propagation (BP) message from  $f_i$  to  $\theta$  is given by

$$m_{f_i \rightarrow \theta}(\theta) = \int dX_{\setminus f_i} f(X) m_{X \rightarrow f_i}(X).$$

Two notable differences between EP and BP are as follows. Firstly, no projection is involved in computing a message in BP. Secondly, to send a message from  $f_i$  to  $\theta$  in BP, there is no need to collect a message from  $\theta$  to  $f_i$ . EP needs it to compute an outgoing message to give a context for the global posterior and then divide it out after the projection.

**General EP Messages**  
Let  $f$  be a factor and  $\mathcal{V}(f)$  be the set of variables connected to  $f$ . In general, an EP message from  $f$  to a variable  $v' \in \mathcal{V}(f)$  takes the following form

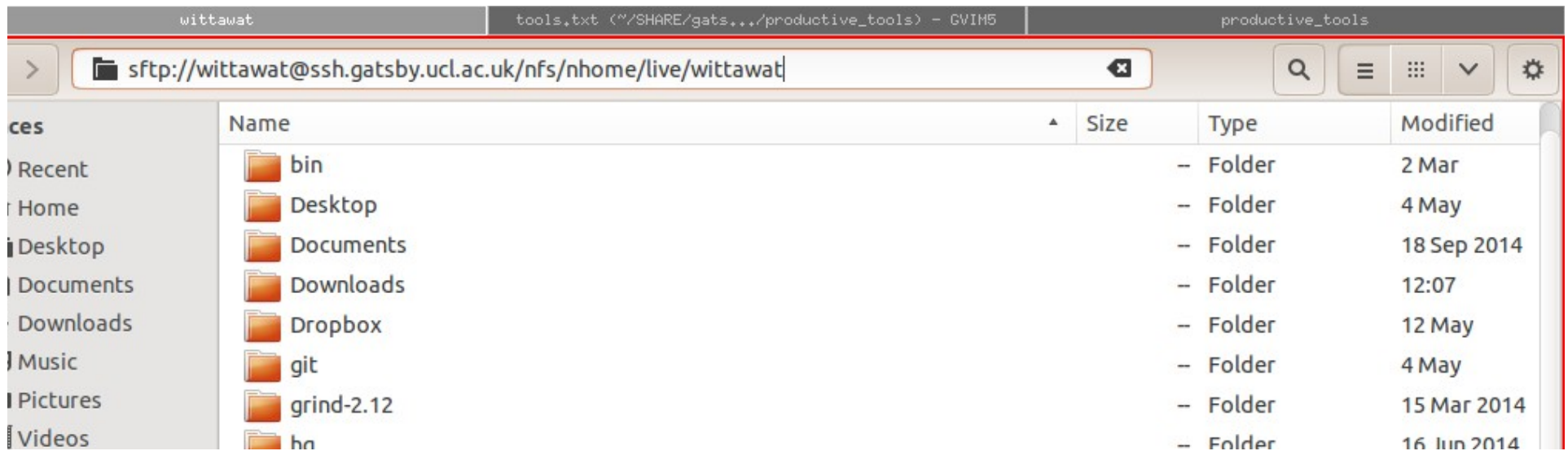
$$m_{f \rightarrow v'}(v') = \frac{\text{proj} \left[ \int d\mathcal{V} \setminus \{v'\} f(\mathcal{V}) \prod_{v \in \mathcal{V}(f)} m_{v \rightarrow f}(v) \right]}{m_{v' \rightarrow f}(v')} \quad (3,eq:ep_msg_update_general)$$

where  $\int d\mathcal{V} \setminus \{v'\}$  denotes an integral over all variables except  $v'$ . In the previous example,  $\mathcal{V}(f) = \{X, \theta\}$ . Note that the division is easy to carry out since both numerator and denominator messages are in ExpFam. The message from a variable node  $v'$  to a factor  $f \in \mathcal{F}(v')$  where  $\mathcal{F}(v') = \{f | v' \in \mathcal{V}(f)\}$  is given by

$$m_{v' \rightarrow f}(v') = \prod_{g \in \mathcal{F}(v') \setminus \{f\}} m_{g \rightarrow v'}(v')$$

# sftp + nautilus in Linux

- sftp = SSH File Transfer Protocol
- Nautilus supports sftp.
- Can browse remote files with Nautilus GUI.
- `sftp://username@ssh.gatsby.ucl.ac.uk/some/path/`





# sharelatex.com

- Free online realtime collaborative Latex editor.
- Just share URL.

The screenshot shows the ShareLaTeX.com web interface. The browser address bar displays the URL: `sharelatex.com/project/53d7cd80e57a33bb2971ffe9`. The document title is "Truncate weights". The left pane shows the LaTeX source code, and the right pane shows the rendered PDF output.

**LaTeX Source Code:**

```
1 \documentclass[12pt]{article}
2 \usepackage[utf8]{inputenc}
3 \usepackage{geometry}
4 \usepackage{amsmath}
5 \usepackage{amsfonts}
6 \usepackage{amssymb}
7 \title{Truncate weights}
8 \author{Wittawat Jitkrittum}
9 \date{July 2014}
10
11 \begin{document}
12
13 \maketitle
14
15 \section{How to truncate weights ?}
16
17 \begin{align*}
18 \hat{C}_{y|x} &= YX^T (XX^T + \lambda I)^{-1} \\
19 &= YM
20 \end{align*}
21
22 \begin{itemize}
23 \item  $S = (\mathbf{y}_1 | \cdots | \mathbf{y}_n)$  where  $n$  is the training size.
24 Each  $\mathbf{y}_i$  in  $\mathbb{R}^{d_y}$  is the sufficient statistic of message  $S$  in the training set.
25  $S$  is a  $d_y \times n$  matrix where  $d_y$  is the number of random features.
26 \end{itemize}
27
28 When applying the operator to a new random feature vector  $x^*$ ,
29 \begin{align*}
30 \hat{C}_{y|x} x^* &= Y M x^* \\
31 &= Y \left( M(1, :), \dots, M(n, :), x^* \right)^T \\
32 &= \sum_{i=1}^n \mathbf{y}_i M(i, :)^T x^* \\
33 \text{truncate} &\rightarrow \sum_{i=1}^n \mathbf{y}_i \max(0, M(i, :)^T x^*)
34 \end{align*}
35
36 The expression with truncation contains  $n$ . Without truncation, we could just
37 pre-compute  $YM$ , a  $d_y \times n$  matrix (no dependency on  $n$ ).
38 \end{document}
39
```

**Rendered PDF Output:**

1 How to truncate weights ?

$$\hat{C}_{y|x} = YX^T (XX^T + \lambda I)^{-1} = YM$$

- $Y = (\mathbf{y}_1 | \cdots | \mathbf{y}_n)$  where  $n$  is the training size. Each  $\mathbf{y}_i \in \mathbb{R}^{d_y}$  is the statistic of message  $i$  in the training set.
- $X$  is a  $D \times n$  matrix where  $D$  is the number of random features.

When applying the operator to a new random feature vector  $x^*$ ,

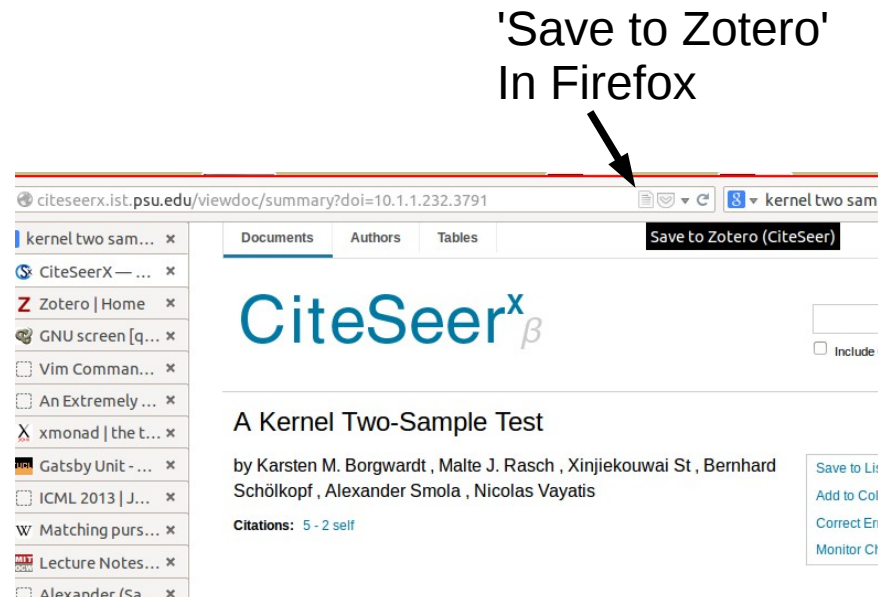
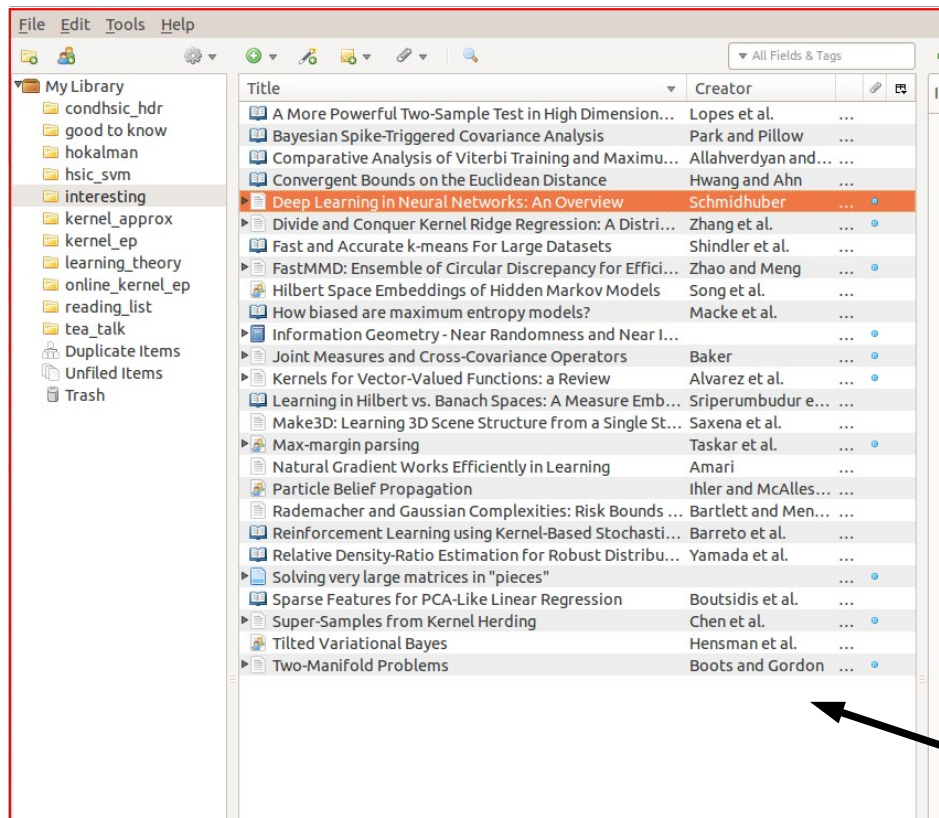
$$\hat{C}_{y|x} x^* = Y M x^* = Y (M(1, :), \dots, M(n, :), x^*)^T = \sum_{i=1}^n \mathbf{y}_i M(i, :)^T x^* (\text{truncate}) \rightarrow \sum_{i=1}^n \mathbf{y}_i \max(0, M(i, :)^T x^*)$$

The expression with truncation contains  $n$ . Without truncation, we could pre-compute  $YM$ , a  $d_y \times n$  matrix (no dependency on  $n$ ).

- Viewers do not need to register.
- If made public, anyone can edit.

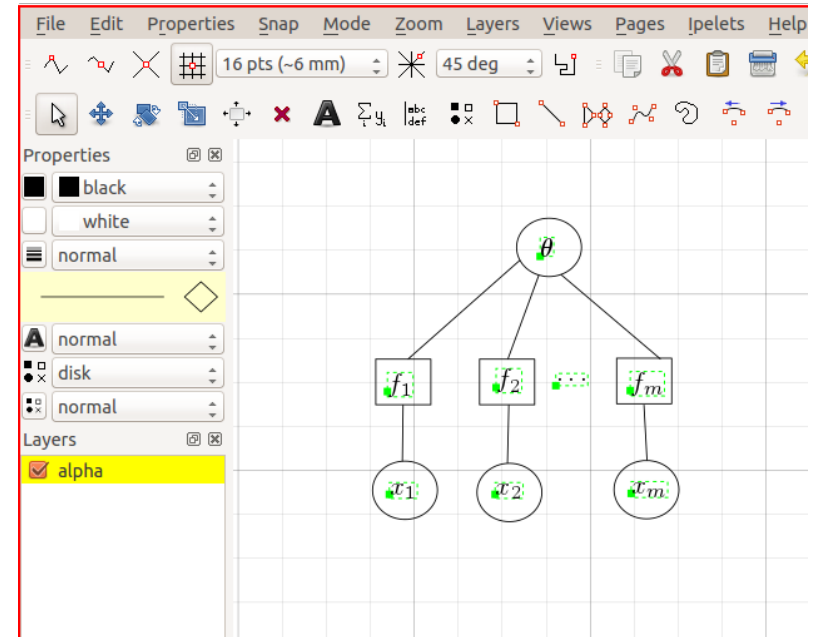
# Zotero + Zotero Firefox Plugin

- Bibliography management tool. Need to register at zotero.org
- All information stored online.
- Ability to export to .bib file.
- Zotero Firefox plugin



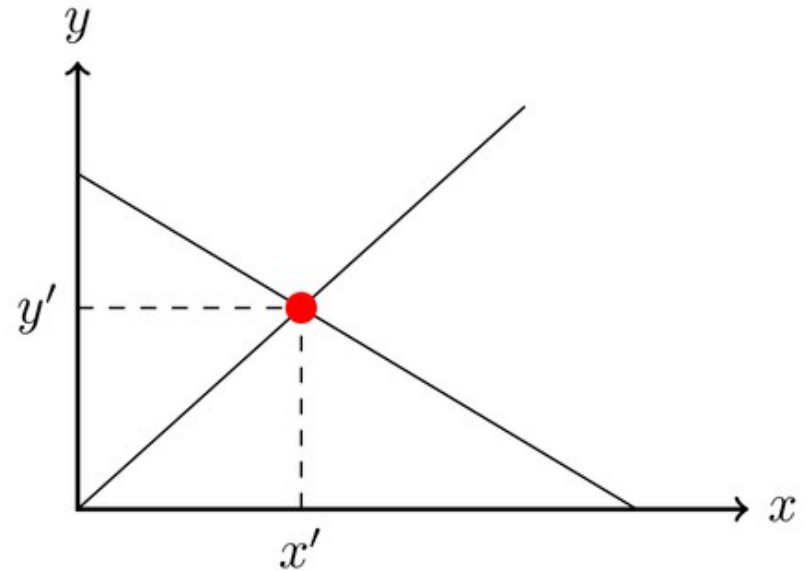
# ipe

- Drawing editor for creating figures in PDF or EPS.
- Support Latex.
- Alternative to tex-based Tikz (for high precision figures).
- Tikz script is embedded in Latex file. Compile to generate a figure.



# Tikz

```
\documentclass{article}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}[scale=1.5]
  % Draw axes
  \draw [<->,thick] (0,2) node (yaxis) [above] {$y$}
    |- (3,0) node (xaxis) [right] {$x$};
  % Draw two intersecting lines
  \draw (0,0) coordinate (a_1) -- (2,1.8) coordinate (a_2);
  \draw (0,1.5) coordinate (b_1) -- (2.5,0) coordinate (b_2);
  % Calculate the intersection of the lines a_1 -- a_2 and b_1 -- b_2
  % and store the coordinate in c.
  \coordinate (c) at (intersection of a_1--a_2 and b_1--b_2);
  % Draw lines indicating intersection with y and x axis. Here we use
  % the perpendicular coordinate system
  \draw[dashed] (yaxis |- c) node[left] {$y'$}
    -| (xaxis -| c) node[below] {$x'$};
  % Draw a dot to indicate intersection point
  \fill[red] (c) circle (2pt);
\end{tikzpicture}
\end{document}
```



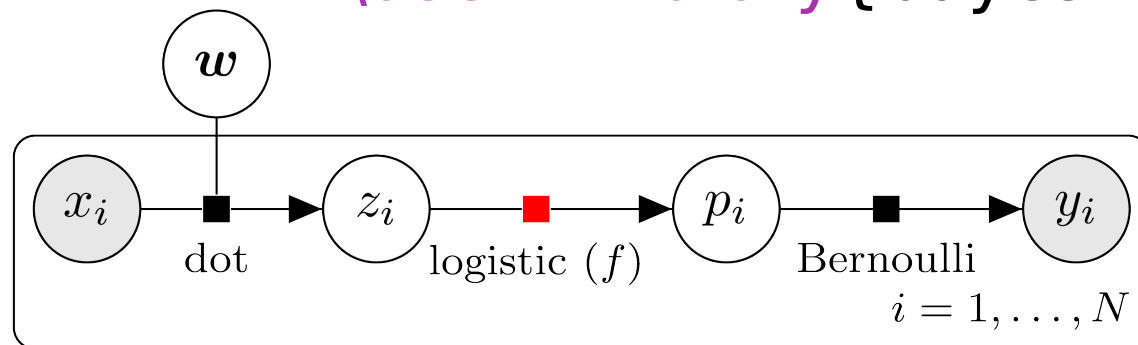
# Tikz-bayesnet

```

\begin{tikzpicture}
\node[obs] (x) {$x_i$};
\bayesfactor[right= of x] {dot} {below:dot} {} {};
\node[latent, above = 5mm of dot] (w)
  {\boldsymbol{w}};
\node[latent, right = 6mm of dot] (z) {$z_i$};
\bayesfactor[right= 6mm of z, color=red]
  {logistic} {below:logistic} {} {};
\node[latent, right = 6mm of logistic] (p) {$p_i$};
\bayesfactor[right = 6mm of p] {bern}
  {below:Bernoulli} {} {};
\node[obs, right = 6mm of bern] (y) {$y_i$}; %
\edge[-] {dot} {x} ;
\edge[-] {w} {dot};
\edge[-] {dot} {z} ;
\edge[-] {z} {logistic} ;
\edge[-] {logistic} {p};
\edge[-] {p} {y} ;

\plate {sample} { %
  (x) (z) (p) (y)
} {$i=1, \dots, N$} ;
\end{tikzpicture}

```



- Tikz library for drawing Bayesian networks.
- <https://github.com/jluttine/tikz-bayesnet>
- Good documentation, examples.
- Save `tikzlibrarybayesnet.code.tex` to the Latex source folder.

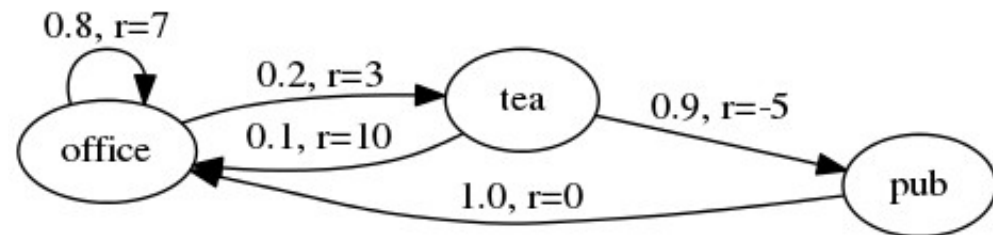
`\usepackage{tikz}`

`\usetikzlibrary{bayesnet}`

# Graphviz

```
digraph mrp {  
  rankdir=LR  
  
  o [label="office"]  
  s [label="tea"]  
  h [label="pub"]  
  
  o -> s [label="0.2, r=3"]  
  o -> o [label="0.8, r=7"]  
  s -> h [label="0.9, r=-5"]  
  s -> o [label="0.1, r=10"]  
  h -> o [label="1.0, r=0"]  
  
}
```

- [www.graphviz.org](http://www.graphviz.org)
- Generate graphs/diagrams from a “dot” language source file.
- **dot -Tpng source.dot > target.png**
- Automatically arrange nodes and edges.



# Vim / Gvim

- Standard text editor on Linux. 30 years old.
- Tons of nifty keyboard shortcuts that can improve coding speed.
- Run `vimtutor` for a tutorial.
- Example commands:

| command          | description    |
|------------------|----------------|
| <code>w</code>   | Move one word  |
| <code>4w</code>  | Move 4 words   |
| <code>dw</code>  | Delete a word  |
| <code>dd</code>  | Delete line    |
| <code>dt)</code> | Delete until ) |

| command                          | description                     |
|----------------------------------|---------------------------------|
| <code>A</code>                   | Append line                     |
| <code>yw</code>                  | Yank (copy) a word              |
| <code>:%/target/replace/g</code> | Replace "target" with "replace" |
| <code>.</code>                   | Repeat last change              |
| <code>:20</code>                 | Go to line 20                   |

# Xmonad

- Dynamically tiling window manager
- Written and configured in Haskell.
- Windows maximized by default.
- Resizing, moving, switching workspaces can be done without mouse.

The screenshot displays the Xmonad desktop environment. On the left, a Haskell code editor shows the definition of the `Monad` type class and its instances. The code includes:

```
1 module Perhaps where
2
3 data Perhaps a = Empty
4   | Only a
5   deriving (Show, Eq)
6
7 instance Monad Perhaps where
8   Only x >>= k = k x
9   Empty >>= _ = Empty
10
11 Only _ >>= k = k
12 Empty >>= _ = Empty
13
14 return x = Only x
15 fail _ = Empty
16
17
18 perhaps b >= (a -> b) = Perhaps a -> b
19 perhaps n = Empty n
20 perhaps _ f (Only a) = f a
21
22 maybeToPerhaps :: Maybe a -> Perhaps a
23 maybeToPerhaps Nothing = Empty
24 maybeToPerhaps (Just m) = Only m
25
26 m2p = maybeToPerhaps
```

On the right, a terminal window shows system statistics and a list of running processes. The system statistics include:

```
Up: 15:11:44 up 58 days, 4:11, 7 users, load averages: 0.63, 0.70, 0.72
Tasks: 206 total, 1 running, 205 sleeping, 0 stopped, 0 zombie
Cpu(s): 7.9us, 35.1sy, 0.0ni, 42.9id, 12.9wa, 0.0hi, 0.2si, 0.0st
Mem: 482396k total, 349212k used, 57016k free, 114808k buffers
Swap: 1190296k total, 48966k used, 1185408k free, 729844k cached
```

The process list shows various system and user processes, such as `VirtualBox`, `kccryptd`, `firecv-bin`, and `ksftirgq/1`.

At the bottom, a terminal window shows the output of a Haskell command:

```
charlie@mishka: ~/code/real_world... charlie@mishka: ~/code/real_world... charlie@mishka: ~
arising from a use of 'print'
Possible fix:
add an instance declaration for
>Show (StdGen -> ((Int, Int), StdGen))
In a stat of an interactive GHCi command: print it
>Main Data.Maybe> runState rollDice (mkStdGen)
interactive:287-20:
Couldn't match expected type 'StdGen'
with actual type 'Int -> StdGen'
In the second argument of 'runState', namely '(mkStdGen)'.
In the expression: runState rollDice (mkStdGen)
In an equation for 'it': it = runState rollDice (mkStdGen)
>Main Data.Maybe> runState rollDice (mkStdGen 93828498999)
interactive:288-39: Not in scope: 'ew'
interactive:289-42: Not in scope: 'f'
>Main Data.Maybe> runState rollDice (mkStdGen 9382849899)
(14,4),442892206 1655838864
>Main Data.Maybe> runState rollDice (mkStdGen 9382849899)
(14,4),442892206 1655838864
>Main Data.Maybe> runState rollDice (mkStdGen 9382849)
(6,2),1473734894 1655838864
>Main Data.Maybe> runState rollDice (mkStdGen 9382849)
(6,2),1473734894 1655838864
>Main Data.Maybe> runState rollDice (mkStdGen 9382849)
(6,2),1473734894 1655838864
>Main Data.Maybe> runState rollDice (mkStdGen 9382849)
(6,2),1473734894 1655838864
>Main Data.Maybe> runState rollDice (mkStdGen 9382849)
(6,2),1473734894 1655838864
>Main Data.Maybe> runState rollDice (mkStdGen 9382849)
(6,2),1473734894 1655838864
```



# Tikz Example

```

\begin{tikzpicture}[line width=1.1pt]
  \draw(0,0) circle(5cm);
  \draw(0,0) circle(1cm);
  \draw(0,0) node {\Huge$\mathbf{A}$};
  \draw(0,0) circle(4.5cm);
  \draw(-48:2.5) arc(-48:240:2.5cm);
  %% The outer nodes
  \foreach \x in {36,72,...,360}
    \shade[ball color=black](\x:5) circle(4pt);
  \foreach \nodes in {12,24,...,360}
    \shade[ball color=black](\nodes:3.5) circle(4pt);
  %%% The connecting nodes
  \foreach \angle in {-48,-12,...,240}
    \draw(\angle:2.5) --++(\angle:0.9cm);
  %%% outer interconnects
  \foreach \angle in {-24,12,...,306}
    \draw(\angle:3.6) --++(\angle:0.9cm);
  \foreach \ly in {-24,12,...,240}
    \shade[ball color=black](\ly:4.5cm) circle(4pt);

```

```

%% outer most connections
\foreach \angle in{-36,0,...,306}
  \draw(\angle:4.9cm) --(\angle:4.7cm)
[rotate=\angle]arc(0:180:0.20cm);
\foreach \angle in{-36,0,...,306}
  \draw(\angle:4.3cm) --(\angle:3.6cm);
%% Outer connects and leads
\shade[ball color=black](276:6) circle(4pt);
\draw(276:6)circle(4pt)--(276:5.2)[rotate=276]arc(0:180:0.25cm),
\draw(276:7)node {\mathbf{K_0}};
\ldraw(276:4.2)[rotate=276]arc(180:360:0.25cm);
\ldraw(276:4.2)--(276:3.5);

```

%% Exploitation of circular symmetry of the required figure

```

{[rotate=72]
  \shade[ball color=black](276:6) circle(4pt);
  \draw(276:6)circle(4pt)--(276:5.2)
[rotate=276]arc(0:180:0.25cm);
  \draw(270:6)node {\mathbf{K_1-K_9}};
  \draw(276:4.2)[rotate=276]arc(180:360:0.25cm);%%%%
  \draw(276:4.2)--(276:3.5);
}

```

```

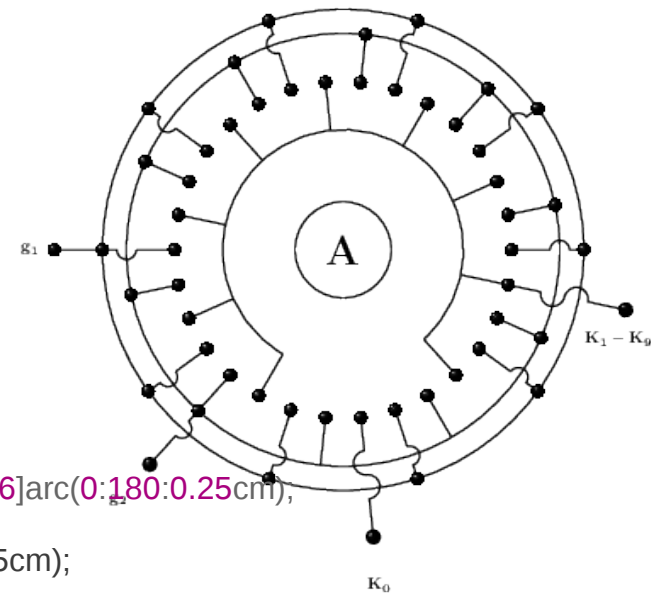
{[rotate=-48]
  \shade[ball color=black](276:6) circle(4pt);
  \draw(276:6)circle(4pt)--(276:5.2)
[rotate=276]arc(0:180:0.20cm);
  \draw(276:7)node {\mathbf{g_2}};
  \ldraw(276:4.8)--(276:4.5);
}

```

```

\ldraw(180:5)--(180:6);
\shade[ball color=black](180:6) circle(4pt);
\ldraw(180:6.5)node{\mathbf{g_1}};
\end{tikzpicture}

```



Thank you