This lecture's notes illustrate some uses of various \LaTeX macros. Take a look at this and imitate.

\section{Leftover from previous class) Optimization for nice problems}

It is noticed that for problems that are "well-behaved":

- Have a decent signal-to-noise ratio
- Correlation between dimensions is under control.
- Number of dimensions is not much larger than the number of data.

Convergence rates are much quicker than the theoretical $O(1/k)$ rate. Explanations for this behavior are still an open research topic.

\section{Matrix calculus}

Taking derivatives of functions that involve matrices can be painful. They can involve:

- Writing out the matrix in full detail with many summations and indices
- Differentiating each of the terms carefully, taking care to treat each indexation correctly.
- Simplifying the expressions to a compact form, if any

An alternative is to use matrix differentials. Matrix differentials are justified by Taylor’s theorem. If $f$ is sufficiently nice, then

\textbf{Exercise:} $f(y) = f(x) + f'(x)(y - x) + r(y - x)$

has $r(y - x) \to 0$ as $y - x \to 0$.

Then we define our differentials:

- $df = f(y) - f(x)$
• $dx = y - x$

These differentials are meant to be thought of as increments, not necessarily as infinitesimals.

We then define $a$, a linear function in $dx$, to be the differential of $f$:

$$df = a(x; dx) + r(dx)$$

Because $a$ is linear in $dx$, we have:

• $a(x; kdx) = ka(x; dx)$
• $a(x; dx_1 + dx_2) = a(x; dx_1) + a(x; dx_2)$

These properties imply that:

• $d(f(x) + g(x)) = df(x) + dg(x)$
• $d(kf(x)) = kdf(x)$

10.2.1 Examples of linear functions

• Reshape (e.g. Converting a 4X3 matrix to a 6X2 matrix)
• Trace (i.e. $\Sigma_i A_{ii}$)
• Transpose

10.3 Differential rules

10.3.1 Chain rule

We derive the chain rule for matrix differentials:

**Proof:** If $L(x) = f(g(x))$ we express the differentials $df = a(g(x); dg)[+r(dg)]$ $dg = b(x; dx)[+s(dx)]$

We join them in $L$ to obtain:

$$dL = a(g(x); b(x; dx) + S(dx)) + r(dg) = a(g(x); b(x; dx))[+a(g(x); S(dx)) + r(dg)]$$

The right side, in square brackets, goes to 0 as $dx \to 0$.

10.3.2 Product rule

If $L(x) = c(f(x), g(x))$ where $c$ is bilinear (e.g. linear in each argument when the other is fixed) then $dL = c(df; g(x)) + c(f(x); dg)$

The proof is skipped. (Note: $fg$ can be scalars, vectors, or matrices.)
10.3.3 Examples of products

- Cross product
- Hadamard product (element-wise product)
- Kronecker product (One matrix is expanded at the position of each element from the other)
- Frobenius product ($\sum_{ij} A_{ij} B_{ij} = tr(A^T B)$)

```
>> A = reshape(1:6, 2, 3)
A =
   1  3  5
   2  4  6
>> B = 2*ones(2)
B =
   2  2
   2  2
```

```
>> kron(A, B)
ans =
   2  2  6  6  10  10
   2  2  6  6  10  10
   4  4  8  8  12  12
   4  4  8  8  12  12
>> kron(B, A)
ans =
   2  6  10  2  6  10
   4  9  12  4  8  12
   2  6  10  2  6  10
   4  8  12  4  8  12
```

Figure 10.1: Kronecker product

10.4 Identification theorems

The identification theorems describe how to switch between conventional and differential notation. They are summarized in this figure:

<table>
<thead>
<tr>
<th>ID for df(x)</th>
<th>scalar x</th>
<th>vector x</th>
<th>matrix X</th>
</tr>
</thead>
<tbody>
<tr>
<td>scalar f</td>
<td>df = a dx</td>
<td>df = $a^T dx$</td>
<td>df = tr$(A^T dX)$</td>
</tr>
<tr>
<td>vector f</td>
<td>df = a dx</td>
<td>df = A dx</td>
<td></td>
</tr>
<tr>
<td>matrix F</td>
<td>dF = A dx</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10.5 Independent Components Analysis

Suppose we have \( n \) training examples \( x_i \in \mathbb{R}^d \) and a scalar-valued, component-wise function \( g \). We would like to find the \( d \times d \) matrix \( W \) that maximizes the entropy of \( y_i = g(Wx_i) \). In the next lecture, we will be using the toolset developed today to tackle this problem.