A Tutorial on Divided Differences and the Linear Exponential Model

Peder Olsen, Vaibhava Goel, John Hershey and Charles Micchelli

HLT Thursday Seminar
IBM TJ Watson Research Center

September 2, 2010
Outline

1. Divided Differences
   - What are they?
   - Examples
   - Applications
   - Opitz’ formula

2. Exponential Models
   - Examples
   - Linear Exponential Family
   - Fused Features
   - Experiments
P(x|\lambda) = \frac{e^{\lambda^T x}}{Z(\lambda)} \quad \text{where}

x \text{ is a posterior}

x \in \mathcal{P}_n = \left\{ x \in \mathbb{R}^n \left| \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \right. \right\}

\text{and } \log Z \text{ is the normalizer aka partition function}

Z(\lambda) = \int_{\mathcal{P}_n} e^{\lambda^T \phi(x)} \, dx = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i} (\lambda_i - \lambda_j)}.
Definition (Divided differences)

\[
[\lambda] f = f(\lambda)
\]

\[
[\lambda_1, \ldots, \lambda_{n+1}] f = \frac{[\lambda_2, \ldots, \lambda_{n+1}] f - [\lambda_1, \ldots, \lambda_n] f}{\lambda_{n+1} - \lambda_1}.
\]
Definition (Divided differences)

\[
[\lambda] f = f(\lambda)
\]

\[
[\lambda_1, \ldots, \lambda_{n+1}] f = \frac{[\lambda_2, \ldots, \lambda_{n+1}]f - [\lambda_1, \ldots, \lambda_n]f}{\lambda_{n+1} - \lambda_1}.
\]

Example (Two Arguments)

Divided difference for two arguments:

\[
[\lambda_1, \lambda_2]f = \frac{f(\lambda_2) - f(\lambda_1)}{\lambda_2 - \lambda_1}.
\]
Computing $[0, 1, 2, 3]f$ for $f(x) = \frac{2}{3}x^3 - 3x^2 + \frac{10}{3}x$
Computing $[0, 1, 2, 3]f$ for $f(x) = \frac{2}{3}x^3 - 3x^2 + \frac{10}{3}x$
Computing \([0,1,2,3]f\) for \(f(x) = \frac{2}{3}x^3 - 3x^2 + \frac{10}{3}x\)
Computing $[0, 1, 2, 3]f$ for $f(x) = \frac{2}{3}x^3 - 3x^2 + \frac{10}{3}x$
Three Arguments

Example (3 arguments)

\[
[\lambda_1, \lambda_2, \lambda_3]f = \frac{[\lambda_2, \lambda_3]f - [\lambda_1, \lambda_2]f}{\lambda_3 - \lambda_1} \quad \frac{f(\lambda_3) - f(\lambda_2)}{\lambda_3 - \lambda_2} - \frac{f(\lambda_2) - f(\lambda_1)}{\lambda_2 - \lambda_1} = \text{some magic}
\]

\[
= \frac{f(\lambda_1)}{(\lambda_1 - \lambda_2)(\lambda_1 - \lambda_3)} + \frac{f(\lambda_2)}{(\lambda_2 - \lambda_1)(\lambda_2 - \lambda_3)} + \frac{f(\lambda_3)}{(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_1)}
\]
Theorem

The general explicit formula is

\[ [\lambda_1, \ldots, \lambda_n] f = \sum_{i=1}^{n} f(\lambda_i) \prod_{j \neq i} (\lambda_i - \lambda_j). \]

The proof requires use of the Vandermonde determinant.
What is \([\lambda, \lambda, \ldots, \lambda]f\)? The definition is unclear. Let’s compute the limit.

Singularity or not?

\[
[x, x + \epsilon]f = \frac{f(x + \epsilon) - f(x)}{\epsilon}
\]

\[
\epsilon \to 0 \quad f'(x).
\]

For more variables we can do a similar computation and see that

\[
[\lambda, \lambda, \ldots, \lambda]f = \frac{f^{(n-1)}(\lambda)}{(n-1)!}.
\]
Define the \( i \)'th power function \( p_i(x) = x^i \), then

\[
\begin{align*}
[\lambda_1]p_0 &= 1 & [\lambda_1, \lambda_2]p_0 &= 0 \\
[\lambda_1]p_1 &= \lambda_1 & [\lambda_1, \lambda_2]p_1 &= 1 \\
[\lambda_1]p_2 &= \lambda_1^2 & [\lambda_1, \lambda_2]p_2 &= \lambda_1 + \lambda_2 \\
[\lambda_1]p_3 &= \lambda_1^3 & [\lambda_1, \lambda_2]p_3 &= \lambda_1^2 + \lambda_1 \lambda_2 + \lambda_2^2 \\
[\lambda_1]p_4 &= \lambda_1^4 & [\lambda_1, \lambda_2]p_4 &= \lambda_1^3 + \lambda_1^2 \lambda_2 + \lambda_1 \lambda_2^2 + \lambda_2^3 \\
[\lambda_1]p_5 &= \lambda_1^5 & [\lambda_1, \lambda_2]p_5 &= \sum_{i+j=4} \lambda_1^i \lambda_2^j 
\end{align*}
\]
Define the $i$’th power function $p_i(x) = x^i$, then

\[
[\lambda_1, \lambda_2]p_0 = 0 \\
[\lambda_1, \lambda_2]p_1 = 1 \\
[\lambda_1, \lambda_2]p_2 = \lambda_1 + \lambda_2 \\
[\lambda_1, \lambda_2]p_3 = \lambda_1^2 + \lambda_1\lambda_2 + \lambda_2^2 \\
[\lambda_1, \lambda_2]p_4 = \sum_{i+j=3} \lambda_1^i \lambda_2^j \\
[\lambda_1, \lambda_2]p_5 = \sum_{i+j=4} \lambda_1^i \lambda_2^j \\
\]

\[
[\lambda_1, \lambda_2, \lambda_3]p_0 = 0 \\
[\lambda_1, \lambda_2, \lambda_3]p_1 = 0 \\
[\lambda_1, \lambda_2, \lambda_3]p_2 = 1 \\
[\lambda_1, \lambda_2, \lambda_3]p_3 = \lambda_1 + \lambda_2 + \lambda_3 \\
[\lambda_1, \lambda_2, \lambda_3]p_4 = \sum_{i+j+k=2} \lambda_1^i \lambda_2^j \lambda_3^k \\
\]
Power Functions

Define the \( i \)’th power function \( p_i(x) = x^i \), then

\[
[\lambda_1, \lambda_2, \ldots, \lambda_n] p_{n+k-1} = \sum_{i_1+\cdots+i_n=k} \prod_{j=1}^{n} \lambda_j^{i_j}
\]

The formula can be computed recursively and efficiently using dynamic programming.
The Exponential Function

\[ Z(\lambda_1, \ldots, \lambda_n) \overset{\text{def}}{=} [\lambda_1, \ldots, \lambda_n] \exp. \]

No slick formula for computing \( Z \) (yet)

\[ Z(\lambda_1) = e^{\lambda_1} \]
\[ Z(\lambda_1, \lambda_2) = \frac{e^{\lambda_1} - e^{\lambda_2}}{\lambda_1 - \lambda_2} \]
\[ Z(\lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i}(\lambda_i - \lambda_j)}. \]

Can’t compute \( Z(x, x) \) with this formula, but is it good otherwise?
The special case $f = \exp$ satisfies the remarkable property that its own derivative can be computed from itself:

**Theorem (Partition Function Derivative)**

$$\frac{\partial}{\partial \lambda_i} Z(\lambda_1, \ldots, \lambda_n) = Z(\lambda_i, \lambda_1, \ldots, \lambda_i, \ldots, \lambda_n)$$

for $i = 1, 2, \ldots, n$.

**Proof.**

Compute LHS and RHS and verify that they are the same.
A Simple Formula

We can compute the formula for \( Z \) exactly when \( \lambda \) has a special form:

\[
Z(0, \frac{1}{n}, \frac{2}{n}, \ldots, 1) = \frac{n^n}{n!} (e^{1/n} - 1)^n
\]
A Simple Formula

We can compute the formula for $Z$ exactly when $\lambda$ has a special form:

$$Z(0, \frac{1}{n}, \frac{2}{n}, \ldots, 1) = \frac{n^n}{n!} \left( e^{1/n} - 1 \right)^n$$

$$\gg n = 10;$$
$$\gg \logz10 = n \log(n) - \text{gammaln}(n+1) + n \log(\exp(1/n) - 1)$$
$$\logz10 = -14.6002$$
We can compute the formula for \( Z \) exactly when \( \lambda \) has a special form:

\[
Z(0, \frac{1}{n}, \frac{2}{n}, \ldots, 1) = \frac{n^n}{n!} (e^{1/n} - 1)^n
\]

\[
\text{>> n = 10;}
\text{>> logz10 = n*log(n)-gammaln(n+1)+n*log(exp(1/n)-1)}
\text{logz10 = -14.6002}
\text{>> n = 100;}
\text{>> logz100 = n*log(n)-gammaln(n+1)+n*log(exp(1/n)-1)}
\text{logz100 = -363.2390}
\]
We can also compute $Z(\lambda)$ using the direct formula

$$Z(\lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i}(\lambda_i - \lambda_j)}$$
We can also compute $Z(\lambda)$ using the direct formula

$$Z(\lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i}(\lambda_i - \lambda_j)}$$

function [y] = logz(lambda)

n = length(lambda);
y = 0;
for i=1:n,
    y = y + exp(lambda(i)) / ... 
    ( prod(lambda(1:i-1)-lambda(i)) ... * prod(lambda(i+1:n)-lambda(i)) );
end
y = log(y);
We can also compute $Z(\lambda)$ using the direct formula

$$Z(\lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j\neq i}(\lambda_i - \lambda_j)}$$

>> y10 = logz([0:0.1:1])

$y10 = -14.6014$
We can also compute $Z(\lambda)$ using the direct formula

$$Z(\lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i}(\lambda_i - \lambda_j)}$$

>> y10 = logz([0:0.1:1])
   y10 = -14.6014
>> logz10
   logz10 = -14.6002
We can also compute $Z(\lambda)$ using the direct formula

$$Z(\lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i}(\lambda_i - \lambda_j)}$$

$$\gg y100 = \log z([0:0.01:1])$$

$y100 = 128.5899$
We can also compute $Z(\lambda)$ using the direct formula

$$Z(\lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i}(\lambda_i - \lambda_j)}$$

$$\gg y100 = \logz([0:0.01:1])$$
$$y100 = 128.5899$$
$$\gg \logz100$$
$$\logz100 = -363.2390$$
Figure: Plot showing $\frac{e^{\lambda_i}}{\prod_{j \neq i} (\lambda_i - \lambda_j)}$. Largest term: $6 \times 10^5$. Total: $Z(\lambda) = 4.56 \times 10^{-7}$
Lagrange Interpolation

Given: Points \((y_i, x_i)\) for unknown function \(f\).
Find: Polynomial \(p(x)\) s.t. \(y_i = p(x_i)\).
Lagrange interpolation:

\[
p(\lambda) = \sum_{i=1}^{n} \ell_i(\lambda)f(\lambda_i), \quad \text{where} \quad \ell_i(\lambda) = \frac{\prod_{j \neq i}(\lambda - \lambda_j)}{\prod_{j \neq i}(\lambda_i - \lambda_j)}.
\]
Newton’s interpolation formula uses the divided differences:

\[ p(\lambda) = [\lambda_1]f + (\lambda - \lambda_1) [\lambda_1, \lambda_2]f + (\lambda - \lambda_1)(\lambda - \lambda_2) [\lambda_1, \lambda_2, \lambda_3]f + \ldots + (\lambda - \lambda_1) \cdots (\lambda - \lambda_{n-1}) [\lambda_1, \ldots, \lambda_n]f. \]

The special case \( \lambda_1 = \lambda_2 = \cdots = \lambda_n \) gives the Taylor series. General case is a generalization of Taylor expansion.
The divided differences can also be used to compute a certain class of integrals.

Theorem (Hermite–Genocchi)

\[
[\lambda_1, \lambda_2, \ldots, \lambda_n]f = \int_{\mathcal{P}_n} f^{(n-1)}(\lambda^T x) \, dx,
\]

(4)

Proof.

Verify for \( f = \exp \) and use Laplace transform to write \( f \) in terms of \( \exp \).
Theorem (Peano Kernel identity)

Let \( \lambda_{\text{min}} = \min \{\lambda_1, \ldots, \lambda_n\} \) and \( \lambda_{\text{max}} = \max \{\lambda_1, \ldots, \lambda_n\} \) then

\[
[\lambda_1, \lambda_2, \ldots, \lambda_n]f = \frac{1}{(n-1)!} \int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} f^{(n-1)}(t)M(t; \lambda_1, \ldots, \lambda_n) \, dt,
\]

where \( M(t; \lambda_1, \ldots, \lambda_n) \) is the Peano kernel, which is the B-spline of degree \( n-2 \) at the points \( \lambda_1, \ldots, \lambda_n \).

Proof.

Make a variable substitution \( y_1 = \lambda^T x \), and \( y_2, \ldots, y_n \) something else in Hermite–Genocchi formula.

Charles Micchelli generalized B-splines to higher dimensions! This allowed new applications in 2 and 3 dimensional modeling.
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz' formula

![Plot of \( M(t; -1.0, 1.0) \)](image)

**Figure:** Plot of \( M(t; -1, 1) \)
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz' formula

Figure: Plot of $M(t; -1.0, 0.0, 1.0)$
What are they?
Examples
Applications
Opitz' formula
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz’ formula

Figure: Plot of $M(t; -1, -1/3, 1/3, 1)$
Figure: Plot of $M(t; -1.0, 0.0, 0.5, 1.0)$
Leibniz’ rule

Derivatives: 
\[(fg)' = fg' + f'g\]

Divided differences: 
\[[\lambda_0, \lambda_1](fg) = [\lambda_0]f [\lambda_0, \lambda_1]g + [\lambda_0, \lambda_1]f [\lambda_1]g.\]

More generally:

\[[\lambda_0, \ldots, \lambda_n](fg) = [\lambda_0]f [\lambda_0, \ldots, \lambda_n]g + [\lambda_0, \lambda_1]f [\lambda_1, \ldots, \lambda_n]g + \cdots + [\lambda_0, \ldots, \lambda_n]f [\lambda_n]g.\]
**Definition (Divided Difference Matrix)**

\[
T_f(\lambda) = \begin{pmatrix}
[\lambda_0]f & [\lambda_0, \lambda_1]f & \cdots & [\lambda_0, \ldots, \lambda_n]f \\
0 & [\lambda_1]f & \cdots & [\lambda_1, \ldots, \lambda_n]f \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & [\lambda_n]f
\end{pmatrix}
\]  

(5)

**Example (Identity Function)**

\[
J \overset{\text{def}}{=} T_{p_1}(\lambda) = \begin{pmatrix}
\lambda_0 & 1 & 0 & 0 & \cdots & 0 \\
0 & \lambda_1 & 1 & 0 & \cdots & 0 \\
0 & 0 & \lambda_2 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & \cdots & \lambda_n
\end{pmatrix}.
\]  

(6)
Example (Exponential Function)

\[ E(\lambda) = \mathcal{T}_{\text{exp}}(\lambda) = \begin{pmatrix} Z(\lambda_0) & Z(\lambda_0, \lambda_1) & \cdots & Z(\lambda) \\ 0 & Z(\lambda_1) & \cdots & Z(\lambda_1, \ldots, \lambda_n) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & Z(\lambda_n) \end{pmatrix}. \]
Opitz’ Formula

**Theorem (Opitz)**

The divided difference $[\lambda_1, \ldots, \lambda_n]f$ is the $(1, n)$’th element of the matrix

$$T_f(\lambda) = \sum_{i=0}^{\infty} \frac{f^{(i)}(0)}{i!} \lambda^i$$

if the Taylor series of $f$ is everywhere convergent.

**Proof.**

Trivially: $T_{f+g}(\lambda) = T_f(\lambda) + T_g(\lambda)$.

By Leibniz’ rule: $T_{f \cdot g}(\lambda) = T_f(\lambda) \cdot T_g(\lambda)$.

The rest follows by Taylor expansion
A Monster Formula

We can compute $Z(\lambda)$ from $E(\lambda) = e^J$. $\nabla Z(\lambda)$ can be computed from

$$E(\lambda, \lambda) = \begin{pmatrix}
Z(\lambda_1) & \ldots & Z(\lambda) \\
0 & \ldots & Z(\lambda_2, \ldots, \lambda_n) \\
0 & \ldots & Z(\lambda_3, \ldots, \lambda_n) \\
\vdots & \ddots & \vdots \\
0 & \ldots & Z(\lambda_n) \\
0 & \ldots & 0
\end{pmatrix}$$
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz’ formula

>> n=10;
>> lambda=0:1/n:1;
>> J=diag(lambda);
>> for i=1:n,
   J(i,i+1)=1;
end
>> E=expm(J);
>> log(E(1,n+1))
ans = -14.6002
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz' formula

>> n=10;
>> lambda=0:1/n:1;
>> J=diag(lambda);
>> for i=1:n,
    J(i,i+1)=1;
end
>> E=expm(J);
>> log(E(1,n+1))
ans = -14.6002

>> logz10
logz10 = -14.6002
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz' formula

>> n=100;
>> lambda=0:1/n:1;
>> J=diag(lambda);
>> for i=1:n,
   J(i,i+1)=1;
end
>> E=expm(J);
>> log(E(1,n+1))
ans = -239.4318
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz' formula

```
>> n=100;
>> lambda=0:1/n:1;
>> J=diag(lambda);
>> for i=1:n,
    J(i,i+1)=1;
end
>> E=expm(J);
>> log(E(1,n+1))
an = -239.4318

>> logz100
logz100 = -363.2390
```
Computing the Matrix Exponential is Hard

SIAM REVIEW
Vol. 45, No. 1, pp. 3–000

© 2003 Society for Industrial and Applied Mathematics

Cleve Moler†
Charles Van Loan‡

Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later*

Abstract. In principle, the exponential of a matrix could be computed in many ways. Methods involving approximation theory, differential equations, the matrix eigenvalues, and the matrix characteristic polynomial have been proposed. In practice, consideration of computational stability and efficiency indicates that some of the methods are preferable to others, but that none are completely satisfactory. Most of this paper was originally published in 1978. An update, with a separate bibliography, describes a few recent developments.
```matlab
>> n=100;
>> lambda=0:1/n:1;
>> J=diag(lambda);
>> for i=1:n,
    J(i,i+1)=1;
end
>> E=expm(J/32);
>> E=E*E; E=E*E; E=E*E; E=E*E; E=E*E;
>> log(E(1,n+1))
ans = -363.2383
```
Divided Differences
Exponential Models

What are they?
Examples
Applications
Opitz' formula

>> n=100;
>> lambda=0:1/n:1;
>> J=diag(lambda);
>> for i=1:n,
   J(i,i+1)=1;
end
>> E=expm(J/32);
>> E=E*E; E=E*E; E=E*E; E=E*E; E=E*E;
>> log(E(1,n+1))
ans = -363.2383

>> logz100
logz100 = -363.2390
We computed $E(\lambda)$ by doing our own matrix exponential that takes advantage of $J$ being sparse upper triangular. We also did all computations in the log domain and by using

$$e^J = \left(e^{J/2^m}\right)^{2^m}$$

and

$$e^{J/2^m} = I + J/2^m + \frac{1}{2!}(J/2^m)^2 + \cdots$$
Definition (Exponential Family)

\[
P(x|\lambda) = \frac{e^{\lambda^T \phi(x)}}{Z(\lambda)} \quad \text{where} \quad Z(\lambda) = \int_D e^{\lambda^T \phi(x)} \, dx \quad (8)
\]

\(D\) is the domain for \(x\).
Example (Normal Distribution)

For diagonal covariance models:

\[ D = \mathbb{R}^d, \quad \phi_N(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix}, \quad \lambda = \begin{pmatrix} \mu/v \\ -1/(2v) \end{pmatrix} \]  

(9)

The partition function is

\[ \log Z_N(\lambda) = -\frac{1}{2} \sum_{i=1}^{n} \frac{\mu_i^2}{v_i} + \log(2\pi v_i). \]  

(10)
The Probability Simplex

Definition (Probability Simplex)

\[ \mathcal{P}_n = \left\{ x \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \right\}. \]  (11)
Example (Dirichlet Distribution)

\[ D = \mathcal{P}_n, \quad \phi_D(x) = \log(x), \quad Z_D(\lambda) = \frac{\prod_i \Gamma(\lambda_i + 1)}{\Gamma(d + \sum_i \lambda_i)}. \]
Example (Linear Exponential Family)

\[ D = \mathcal{P}_n, \quad \phi(x) = x, \quad Z(\lambda) = \sum_{i=1}^{n} \frac{e^{\lambda_i}}{\prod_{j \neq i} (\lambda_i - \lambda_j)}. \] (13)

\( Z(\lambda) \) is the divided difference with respect to the exponential function.
Figure: Plot of a density for the linear exponential family
Plot of $P_D(x | \lambda)$ for $\lambda = (0.32, 0.42, 0.25)$, $x_3 = 1 - x_1 - x_2$ not shown

**Figure:** Plot of a Dirichlet distribution
Cepstra linear discriminant analysis (LDA) based features: \( \mathbf{x} \in \mathbb{R}^{40} \)

SPIF posterior features: \( \mathbf{p} \in \mathbb{R}^{44} \)

Normal + Linear : \[ \phi(\mathbf{x}) = \begin{pmatrix} x \\ x^2 \\ \mathbf{p} \end{pmatrix} \]

Normal + Dirichlet : \[ \phi(\mathbf{x}) = \begin{pmatrix} x \\ x^2 \\ \log(\mathbf{p}) \end{pmatrix}. \]
<table>
<thead>
<tr>
<th>Method</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>fMMI only</td>
<td>19.4%</td>
</tr>
<tr>
<td>fMMI + linear exp family</td>
<td>18.8%</td>
</tr>
<tr>
<td>fMMI + Dirichlet</td>
<td>20.2%</td>
</tr>
</tbody>
</table>