Exact learning and inference for planar graphs

Dmitry Kamenetsky

Supervisor: Nic Schraudolph

September 2007
Ising model

- Particles modify their behaviour to conform with neighbours’ behaviour
- What is the mean energy?
- Used in chemistry, physics, biology...
- More than 12,000 papers published!
Ising problem

- Graph $G = (V, E)$
- Binary variables: $x_i \in \{-1, +1\}$
- No potential for disagreement edges: $\phi_{ij} = 0$ if $x_i \neq x_j$
- Model distribution

$$P(x) = \frac{1}{Z(\phi)} e^{\sum_{ij \in E} [x_i = x_j] \phi_{ij}} \text{, where}$$

$$Z(\phi) = \sum_x e^{\sum_{ij \in E} [x_i = x_j] \phi_{ij}} \text{ is the partition function}$$
How many perfect matchings does a graph have?

**Perfect Matching**: A set of non-overlapping edges (dimers) that cover all vertices
Counting Matchings

- Construct a **Pfaffian orientation**: each face (except possibly outer) has an odd number of edges oriented clockwise

```
    A -> B <- C
    |    |    |
    V    V    V
    D -> E <- F
```

- Construct a skew-symmetric matrix $K$ such that:

$$
K_{ij} = \begin{cases} 
1 & \text{if } i \rightarrow j \\
-1 & \text{if } i \leftarrow j \\
0 & \text{otherwise}
\end{cases}
$$

- **Kasteleyn Theorem**:
  - Every planar graph has a Pfaffian orientation
  - Number of perfect matchings is $\text{Pf}(K) = \sqrt{\det K}$
Let $G_\triangle$ be $G$ plane triangulated: each face becomes a triangle

Let $G^*$ be the dual of graph $G_\triangle$: each face in $G_\triangle$ is a vertex in $G^*$

Let $G^*_e$ be the expanded version of $G^*$: each vertex is replaced with 3 vertices in triangle

**Connection**: There is a 1:1 correspondence between agreement edge sets in $G$ and perfect matchings in $G^*_e$
State-of-the-art exact method for computing partition function, marginals and MAP assignment

Graph is a tree: complexity polynomial in graph size

Graph is not a tree:
  - Convert the graph into a tree of cliques
  - Complexity exponential in maximal clique size
Globerson & Jaakkola 2006 use previous results (Ising model) to compute the partition function exactly

Restrictions:
- Graph is planar (no crossing edges)
- Binary-valued labels
- Only edge potentials, no external field (node potentials)

Complexity polynomial in graph size!
My work

- Faster and simpler version of Globerson and Jaakkola algorithm
- No need to compute the dual $G^*$ and expanded version $G_e^*$
- Showed how to compute gradients and hence perform learning
- Applied to territory prediction in Go
Original graph $G = (V, E)$
- Obtain a planar embedding
- Using Boyer-Myrvold algorithm the complexity is $O(V + E)$
Algorithm

- Plane triangulate the graph
- Using simple ear-clipping the complexity is $O(V^2)$
Orient the edges such that each vertex has odd in-degree

Equivalent to having a Pfaffian orientation in the dual graph

Complexity is $O(E)$
Construct a skew-symmetric $2E \times 2E$ matrix $K$ (for dual edges):

- $K_{ij} = \pm e^\phi$ if $ij$ crosses original edge
- $K_{ij} = \pm 1$ if $ij$ crosses added edge

Complexity is $O(2E)$
Algorithm

- Compute partition function: $Z(\phi) = 2 \sqrt{\det K}$
- Compute gradients: $\frac{\partial \ln Z(\phi)}{\partial \phi_k} = -[K^{-1} \odot K]_{2k-1,2k}$
- Computing inverse and determinant takes complexity $O(E^3)$
Go application
Results

It works - correctly computes partition function and gradients

Overall complexity is $O(E^3)$

Performance

<table>
<thead>
<tr>
<th>Training Size</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Block</td>
</tr>
<tr>
<td>100</td>
<td>3.17 ± 0.18</td>
</tr>
<tr>
<td>1000</td>
<td>2.79 ± 0.17</td>
</tr>
<tr>
<td>2000</td>
<td>3.06 ± 0.18</td>
</tr>
<tr>
<td>5000</td>
<td>2.76 ± 0.17</td>
</tr>
<tr>
<td>10000</td>
<td>3.20 ± 0.18</td>
</tr>
</tbody>
</table>
"Some cause happiness wherever they go, others whenever they go" - Oscar Wilde