Optimization goes Data Science goes Optimization

@mluebbecke

Data Science: Theory & Applications · Oct 26, 2015
If you need to reduce us to one label
Example applications

- vehicle routing
- container logistics
- course timetabling
- production planning
- materials stacking
- patient scheduling
Our main machinery

schedule course $c$ at period $t$ in room $r$, or not

$$\min \sum_{c,t,r} \text{prio}(c, t) \cdot x_{c,t,r}$$

s.t. $$\sum_{t \in T(c), r \in R(c)} x_{c,t,r} = n(c)$$ courses $c$

$$\sum_{c \in R^{-1}(r)} x_{c,t,r} \leq 1$$ periods $t$, rooms $r$

$$\sum_{r \in R(c_1)} x_{c_1,t_1,r} + \sum_{r \in R(c_2)} x_{c_2,t_2,r} \leq 1$$ conflicts

$$x_{c,t,r} \in \{0, 1\}$$

a naïve integer program, a.k.a. “three-indexed”
Integer programming: Progress on algorithms

- very effective algorithm: branch-and-cut
- industry strength implementations available ("solvers")
- development since 1991
  - computer speedup: factor 2000
  - algorithmic speedup: factor 500,000

⇒ easily solve problems with \(10^6\) variables and \(10^5\) constraints
The bin packing problem

\[
\begin{align*}
\text{min } & \sum_{j} y_j \\
\sum_{j} x_{ij} & = 1 \quad \text{items } i \\
\sum_{i} a_i x_{ij} & \leq b \quad \text{bins } j \\
& \quad \quad x_{ij} \leq y_j \quad i, j \\
& \quad \quad x_{ij}, y_j \in \{0, 1\} \quad i, j
\end{align*}
\]
**Performance of the first model**

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4485</td>
<td>2808</td>
<td>45.2933</td>
<td>138</td>
<td>47.0000</td>
<td>45.2933</td>
<td>256083</td>
<td>3.63%</td>
</tr>
<tr>
<td>4558</td>
<td>2865</td>
<td>46.0000</td>
<td>53</td>
<td>47.0000</td>
<td>45.2933</td>
<td>260817</td>
<td>3.63%</td>
</tr>
<tr>
<td>4618</td>
<td>2911</td>
<td>46.0000</td>
<td>56</td>
<td>47.0000</td>
<td>45.2933</td>
<td>265554</td>
<td>3.63%</td>
</tr>
<tr>
<td>4679</td>
<td>2946</td>
<td>45.3133</td>
<td>100</td>
<td>47.0000</td>
<td>45.2933</td>
<td>270877</td>
<td>3.63%</td>
</tr>
<tr>
<td>4759</td>
<td>3026</td>
<td>45.3133</td>
<td>54</td>
<td>47.0000</td>
<td>45.2933</td>
<td>274429</td>
<td>3.63%</td>
</tr>
<tr>
<td>4859</td>
<td>3126</td>
<td>45.3667</td>
<td>46</td>
<td>47.0000</td>
<td>45.2933</td>
<td>278978</td>
<td>3.63%</td>
</tr>
<tr>
<td>4957</td>
<td>3224</td>
<td>45.3667</td>
<td>55</td>
<td>47.0000</td>
<td>45.2933</td>
<td>284217</td>
<td>3.63%</td>
</tr>
<tr>
<td>5067</td>
<td>3304</td>
<td>45.9800</td>
<td>30</td>
<td>47.0000</td>
<td>45.2933</td>
<td>289414</td>
<td>3.63%</td>
</tr>
<tr>
<td>5160</td>
<td>3383</td>
<td>45.3133</td>
<td>68</td>
<td>47.0000</td>
<td>45.2933</td>
<td>292707</td>
<td>3.63%</td>
</tr>
<tr>
<td>5317</td>
<td>3526</td>
<td>45.2933</td>
<td>147</td>
<td>47.0000</td>
<td>45.2933</td>
<td>297019</td>
<td>3.63%</td>
</tr>
<tr>
<td>5418</td>
<td>3625</td>
<td>46.0000</td>
<td>67</td>
<td>47.0000</td>
<td>45.2933</td>
<td>301117</td>
<td>3.63%</td>
</tr>
<tr>
<td>5476</td>
<td>3662</td>
<td>45.3133</td>
<td>115</td>
<td>47.0000</td>
<td>45.2933</td>
<td>306847</td>
<td>3.63%</td>
</tr>
<tr>
<td>5550</td>
<td>3726</td>
<td>45.2933</td>
<td>153</td>
<td>47.0000</td>
<td>45.2933</td>
<td>312240</td>
<td>3.63%</td>
</tr>
<tr>
<td>5639</td>
<td>3809</td>
<td>46.0000</td>
<td>42</td>
<td>47.0000</td>
<td>45.2933</td>
<td>316751</td>
<td>3.63%</td>
</tr>
<tr>
<td>5745</td>
<td>3900</td>
<td>46.0000</td>
<td>51</td>
<td>47.0000</td>
<td>45.2933</td>
<td>319540</td>
<td>3.63%</td>
</tr>
<tr>
<td>5801</td>
<td>3948</td>
<td>46.0000</td>
<td>39</td>
<td>47.0000</td>
<td>45.2933</td>
<td>324910</td>
<td>3.63%</td>
</tr>
<tr>
<td>5884</td>
<td>4013</td>
<td>45.3133</td>
<td>62</td>
<td>47.0000</td>
<td>45.2933</td>
<td>329653</td>
<td>3.63%</td>
</tr>
</tbody>
</table>

*Elapsed real time = 74.99 sec. (tree size = 61.35 MB, solutions = 7)*

✅ model symmetry is a problem here
Bin packing: A model with "more meaningful" variables

- $P_j \sim \text{all possible patterns to fill bin } j$

\[
\begin{align*}
\min & \sum_{\text{bins } j} \sum_{p \in P_j} \lambda_{pj} \\
\sum_{\text{bins } j} \sum_{p \in P_j : i \in p} \lambda_{pj} &= 1 & \text{items } i \\
\sum_{p \in P_j} \lambda_{pj} &\leq 1 & \text{bins } j \\
\lambda_{pj} &\in \{0, 1\} & j = 1, \ldots, n, p \in P_j
\end{align*}
\]
### Performance of the second model

<table>
<thead>
<tr>
<th>time</th>
<th>node</th>
<th>left</th>
<th>LP iter</th>
<th>frac</th>
<th>vars</th>
<th>cons</th>
<th>cols</th>
<th>strbr</th>
<th>dualbound</th>
<th>primalbound</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0s</td>
<td>1</td>
<td>0</td>
<td>125</td>
<td>0</td>
<td>121</td>
<td>120</td>
<td>121</td>
<td>0</td>
<td>--</td>
<td>1.200000e+02</td>
<td>Inf</td>
</tr>
<tr>
<td>0.0s</td>
<td>1</td>
<td>0</td>
<td>125</td>
<td>-</td>
<td>121</td>
<td>120</td>
<td>121</td>
<td>0</td>
<td>--</td>
<td>1.150000e+02</td>
<td>Inf</td>
</tr>
<tr>
<td>0.0s</td>
<td>1</td>
<td>0</td>
<td>131</td>
<td>-</td>
<td>122</td>
<td>120</td>
<td>122</td>
<td>0</td>
<td>--</td>
<td>1.110000e+02</td>
<td>Inf</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4s</td>
<td>1</td>
<td>0</td>
<td>5977</td>
<td>100</td>
<td>410</td>
<td>120</td>
<td>273</td>
<td>0</td>
<td>--</td>
<td>9.700000e+01</td>
<td>Inf</td>
</tr>
<tr>
<td>0.4s</td>
<td>1</td>
<td>0</td>
<td>6793</td>
<td>104</td>
<td>460</td>
<td>120</td>
<td>310</td>
<td>0</td>
<td>4.529333e+01</td>
<td>9.700000e+01</td>
<td>114.16%</td>
</tr>
<tr>
<td>b 0.4s</td>
<td>1</td>
<td>0</td>
<td>6793</td>
<td>104</td>
<td>460</td>
<td>120</td>
<td>310</td>
<td>0</td>
<td>4.529333e+01</td>
<td>5.800000e+01</td>
<td>28.05%</td>
</tr>
<tr>
<td>E 0.5s</td>
<td>1</td>
<td>0</td>
<td>7575</td>
<td>104</td>
<td>460</td>
<td>120</td>
<td>310</td>
<td>0</td>
<td>4.529333e+01</td>
<td>4.800000e+01</td>
<td>5.98%</td>
</tr>
<tr>
<td>0.5s</td>
<td>1</td>
<td>0</td>
<td>7575</td>
<td>104</td>
<td>460</td>
<td>120</td>
<td>310</td>
<td>0</td>
<td>4.529333e+01</td>
<td>4.800000e+1</td>
<td>5.98%</td>
</tr>
<tr>
<td>0.7s</td>
<td>1</td>
<td>2</td>
<td>7575</td>
<td>104</td>
<td>460</td>
<td>120</td>
<td>310</td>
<td>42</td>
<td>4.529333e+01</td>
<td>4.800000e+01</td>
<td>5.98%</td>
</tr>
<tr>
<td>R 3.6s</td>
<td>24</td>
<td>23</td>
<td>16461</td>
<td>5</td>
<td>895</td>
<td>120</td>
<td>767</td>
<td>447</td>
<td>4.529333e+01</td>
<td>4.700000e+01</td>
<td>3.77%</td>
</tr>
<tr>
<td>b 3.6s</td>
<td>25</td>
<td>0</td>
<td>16461</td>
<td>-</td>
<td>895</td>
<td>120</td>
<td>767</td>
<td>452</td>
<td>4.550000e+01</td>
<td>4.600000e+01</td>
<td>1.10%</td>
</tr>
</tbody>
</table>

SCIP Status: problem is solved [optimal solution found]
Solving Time (sec): 3.64
Solving Nodes: 25
Primal Bound: +4.60000000000000e+01 (383 solutions)
Dual Bound: +4.60000000000000e+01
Gap: 0.00 %
If nothing else, take away this message

- In integer programming, a “good” model is crucial
Revisiting...
Detecting structure in matrices

- key to model strengthening “by decomposition”
Detecting structure in matrices

- key to model strengthening “by decomposition”

10teams

✓ this is graph partitioning/clustering
Detecting structure in matrices
Business perspectives

Maturing from Descriptive to Prescriptive Analytics

- Summarize Past and Current Behavior
- Predict Future Behavior and Adapt

Analytic Capability:
- Understand the trends in the business
- Make different offers to groups of customers
- Target each decision to a customer’s future behavior
- Automatically take the ideal action on each individual

Decision Value:
- Business Intelligence
- Descriptive Analytics
- Predictive Analytics
- Prescriptive Analytics

Business perspectives
Twitter is full of...
MODERN DATA SCIENTIST

Data Scientist, the sexiest job of 21st century requires a mixture of multidisciplinary skills ranging from an intersection of mathematics, statistics, computer science, communication and business. Finding a data scientist is hard. Finding people who understand who a data scientist is, is equally hard. So here is a little cheat sheet on who the modern data scientist really is.

MATH & STATISTICS
- Machine learning
- Statistical modeling
- Experiment design
- Bayesian inference
- Supervised learning: decision trees, random forests, logistic regression
- Unsupervised learning: clustering, dimensionality reduction
- Optimization: gradient descent and variants

PROGRAMMING & DATABASE
- Computer science fundamentals
- Scripting language e.g. Python
- Statistical computing package e.g. R
- Databases: SQL and NoSQL
- Relational algebra
- Parallel databases and parallel query processing
- MapReduce concepts
- Hadoop and Hive/Pig
- Custom reducers
- Experience with xaaS like AWS

DOMAIN KNOWLEDGE & SOFT SKILLS
- Passionate about the business
- Curious about data
- Influence without authority
- Hacker mindset
- Problem solver
- Strategic, proactive, creative, innovative and collaborative

COMMUNICATION & VISUALIZATION
- Able to engage with senior management
- Story telling skills
- Translate data-driven insights into decisions and actions
- Visual art design
- R packages like ggplot or lattice
- Knowledge of any visualization tools e.g. Flare, D3.js, Tableau

Marketing Distillery is a group of practitioners in the area of e-commerce marketing. Our fields of expertise include marketing strategy and optimization, customer targeting and on site analytics, predictive analytics and economics, data warehousing and big data systems, marketing channel insights in Paid Search, SEO, Social, CRM and brand.
1614 node ‘Polar Eskimo Genealogy’
Fast training of Support Vector Machines with Gaussian kernel

Matteo Fischetti
DEI, University of Padova, via Gradenigo 6/A, 35100 Padova (Italy)
e-mail: matteo.fischetti@unipd.it

10 July 2014. Revised 12 February 2015

Abstract

Support Vector Machines (SVM’s) are ubiquitous and attracted a huge interest in the last years. Their training involves the definition of a suitable optimization model with two main features: (1) its optimal solution estimates the a-posteriori optimal SVM parameters in a reliable way, and (2) it can be solved efficiently. Hinge-loss models, among others, have been used with remarkable success together with cross validation—the latter being instrumental to the success of the overall training, though it can become very time consuming. In this paper we propose a different model for SVM training, that seems particularly suited when the Gaussian kernel is adopted (as it is often the case). Our approach is to model the overall training problem as a whole, thus avoiding the need of cross validation. Though our basic model is an NP-hard Mixed-Integer Linear Program, some variants can be solved very efficiently by simple sorting algorithms. Computational results on test cases from the literature are presented, showing that our training method can lead to a classification accuracy comparable (or even slightly better) than the classical hinge-loss model, with a speedup of 2-3 orders of magnitude.

Keywords: support vector machine, classification, mixed-integer programming.
Evaluation of classification algorithms

\[ \gamma = 10 \]
Perspectives

- What do we really know about our integer programs?
- What do we really know about our data?
- ...
- ...
- Optimization $\rightarrow$ Data Science
- Optimization $\leftarrow$ Data Science
- Optimization $\leftrightarrow$ Data Science
- ...