Topic III.2: Maximum Entropy Models

Discrete Topics in Data Mining
Universität des Saarlandes, Saarbrücken
Winter Semester 2012/13

Topic III.2: Maximum Entropy Models

The Maximum-Entropy Principle

- **Goal:** To define a distribution over data that satisfies given constraints
  - Row/column sums
  - Distribution of values
  - …
- Given such a distribution
  - We can sample from it (as with swap randomization)
  - We can compute the likelihood of the observed data
  - We can compute how surprising our findings are given the distribution
  - …

De Bie 2010

Maximum Entropy

- We expect the constraints to be linear
  - If $\boldsymbol{x} \in X$ is one data set, $\Pr(\boldsymbol{x})$ is the distribution, and $f_i(\boldsymbol{x})$ is a real-valued function of the data, the constraints are of type
$$\sum_x \Pr(\boldsymbol{x}) f_i(\boldsymbol{x}) = d_i$$
- Many distributions can satisfy the constraints; which to choose?
- We want to select the distribution that **maximizes the entropy** and satisfies the constraints
  - Entropy of a discrete distribution: $-\sum_x \Pr(\boldsymbol{x}) \log(\Pr(\boldsymbol{x}))$

Why Maximize the Entropy?

- No other assumptions
  - Any distribution with less-than-maximal entropy must have some reason for the reduced entropy
  - Essentially, a latent assumption about the distribution
  - We want to avoid these
- Optimal worst-case behaviour w.r.t. coding lenghts
  - If we build an encoding based on the maximum entropy distribution, the worst-case expected encoding length is the minimum over any distribution

Finding the MaxEnt Distribution

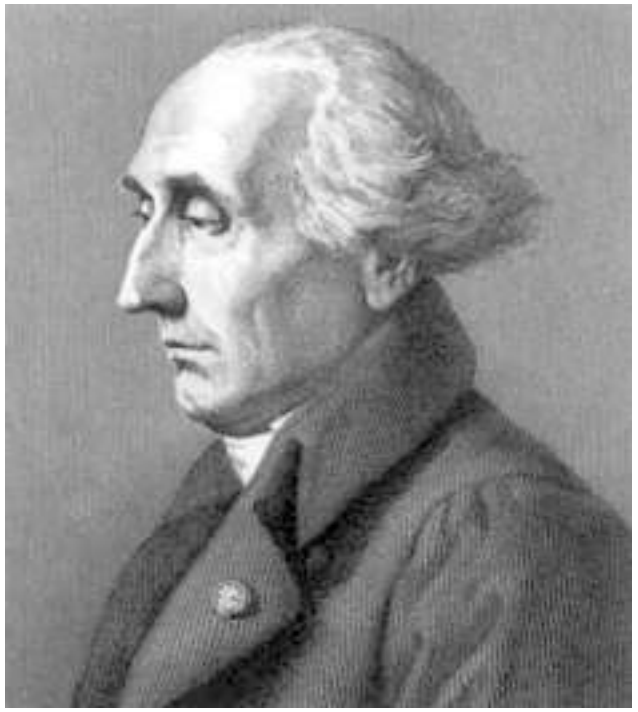
- Finding the MaxEnt distribution is a convex program with linear constraints

$$\begin{aligned} \max_{\Pr(\mathbf{x})} \quad & -\sum_{\mathbf{x}} \Pr(\mathbf{x}) \log \Pr(\mathbf{x}) \\ \text{s.t.} \quad & \sum_{\mathbf{x}} \Pr(\mathbf{x}) f_i(\mathbf{x}) = d_i \qquad \text{for all } i \\ & \sum_{\mathbf{x}} \Pr(\mathbf{x}) = 1 \end{aligned}$$

- Can be solved, e.g., using the Lagrange multipliers

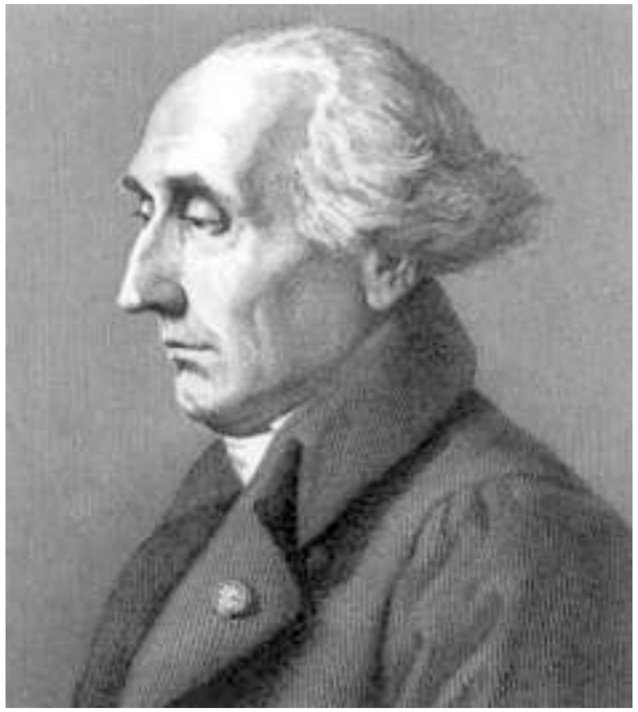
Intermezzo: Lagrange multipliers

- A method to find extrema of constrained functions via derivation
- Problem: minimize $f(\boldsymbol{x})$ subject to $g(\boldsymbol{x}) = 0$
  - Without constraint we can just derive $f(\mathrm{x})$
    - But the extrema we obtain might be unfeasible given the constraints
- Solution: introduce **Lagrange multiplier** λ
  - Minimize $L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda g(\boldsymbol{x})$
  - $\nabla f(\boldsymbol{x}) - \lambda \nabla g(\boldsymbol{x}) = 0$
    - $\partial L / \partial x_i = \partial f / \partial x_i - \lambda \times \partial g / \partial x_i = 0$ for all i
    - $\partial L / \partial \lambda = g(\boldsymbol{x}) = 0$



Intermezzo: Lagrange multipliers

- A method to find extrema of constrained functions via derivation
- Problem: minimize $f(\boldsymbol{x})$ subject to $g(\boldsymbol{x}) = 0$
  - Without constraint we can just derive $f(\mathrm{x})$
    - But the extrema we obtain might be unfeasible given the constraints
- Solution: introduce **Lagrange multiplier** λ
  - Minimize $L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda g(\boldsymbol{x})$
  - $\nabla f(\boldsymbol{x}) - \lambda \nabla g(\boldsymbol{x}) = 0$
    - $\partial L / \partial x_i = \partial f / \partial x_i - \lambda \times \partial g / \partial x_i = 0$ for all i
    - $\partial L / \partial \lambda = g(\boldsymbol{x}) = 0$ The constraint!

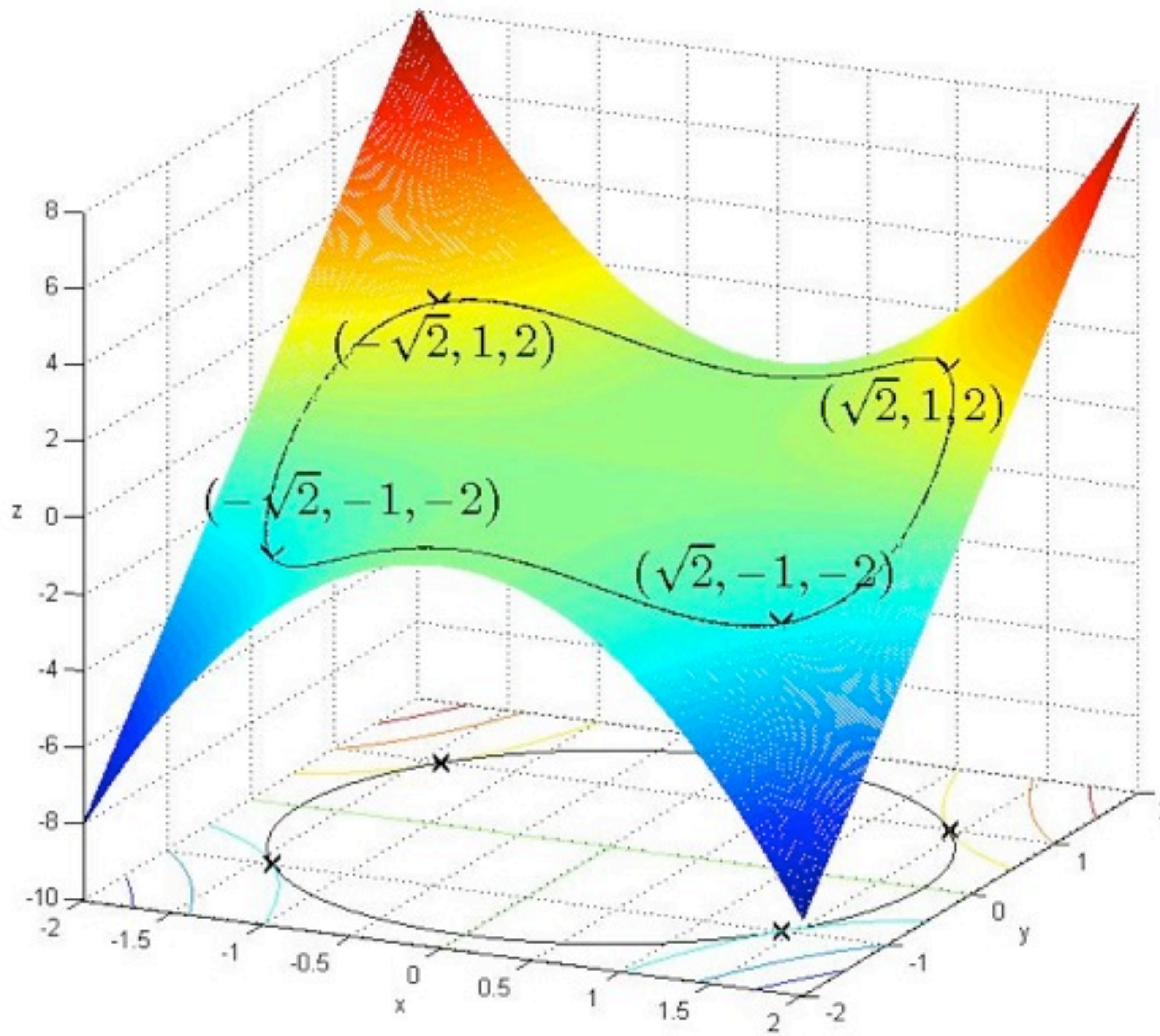


More on Lagrange multipliers

- For many constraints, we need to add one multiplier for each constraint
  - $L(\boldsymbol{x},\boldsymbol{\lambda}) = f(\boldsymbol{x}) - \Sigma_j \lambda_j g_j(\boldsymbol{x})$
  - Function L is known as the **Lagrangian**
- Minimizing the unconstrained Lagrangian equals minimizing the constrained f
  - But not all solutions to $\nabla f(\boldsymbol{x}) - \Sigma_j \lambda_j \nabla g_j(\boldsymbol{x}) = 0$ are extrema
  - The solution is in the boundary of the constraint only if $\lambda_j \neq 0$

Example

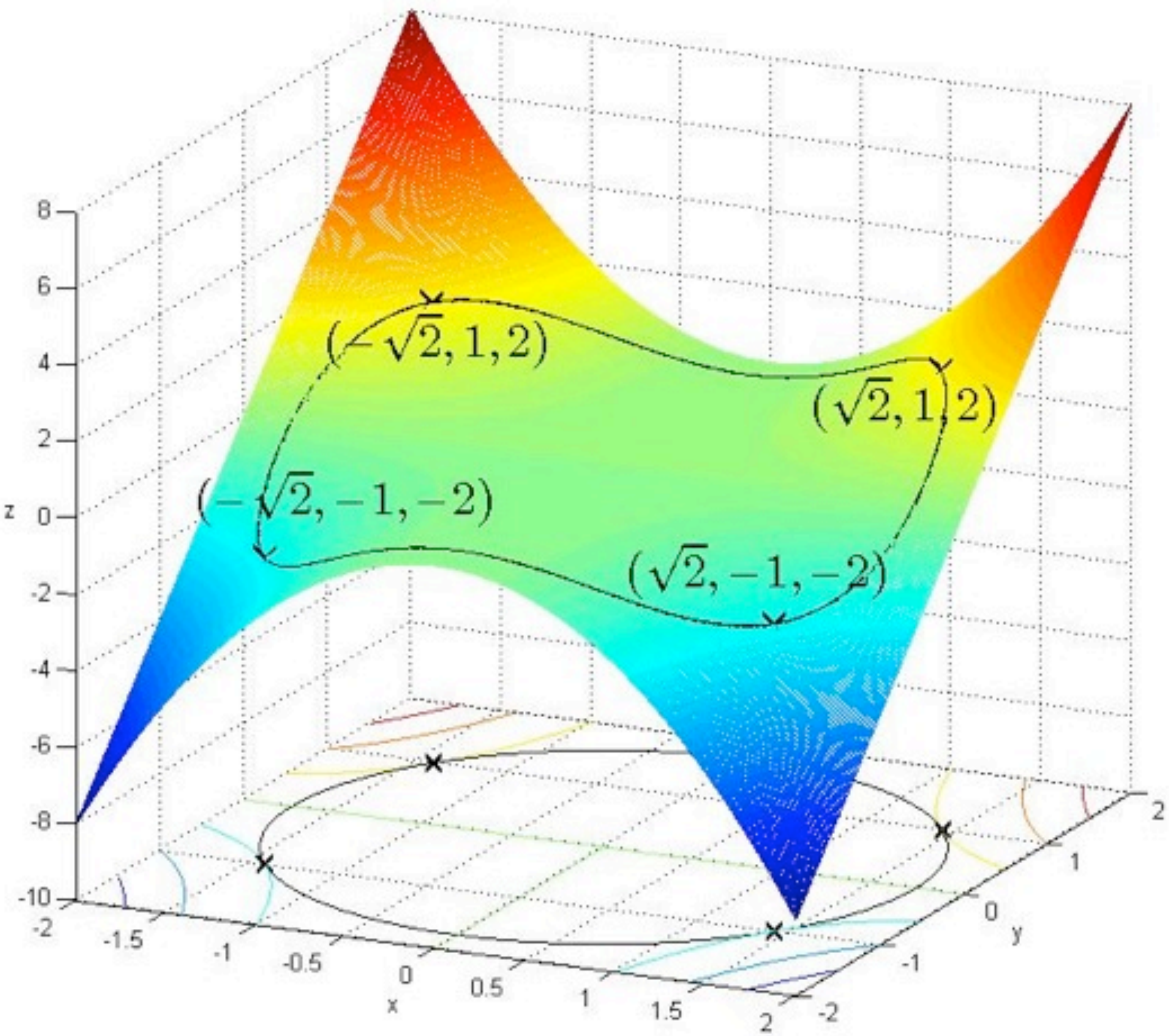
minimize $f(x,y) = x^2y$
subject to $g(x,y) = x^2 + y^2 = 3$



Example

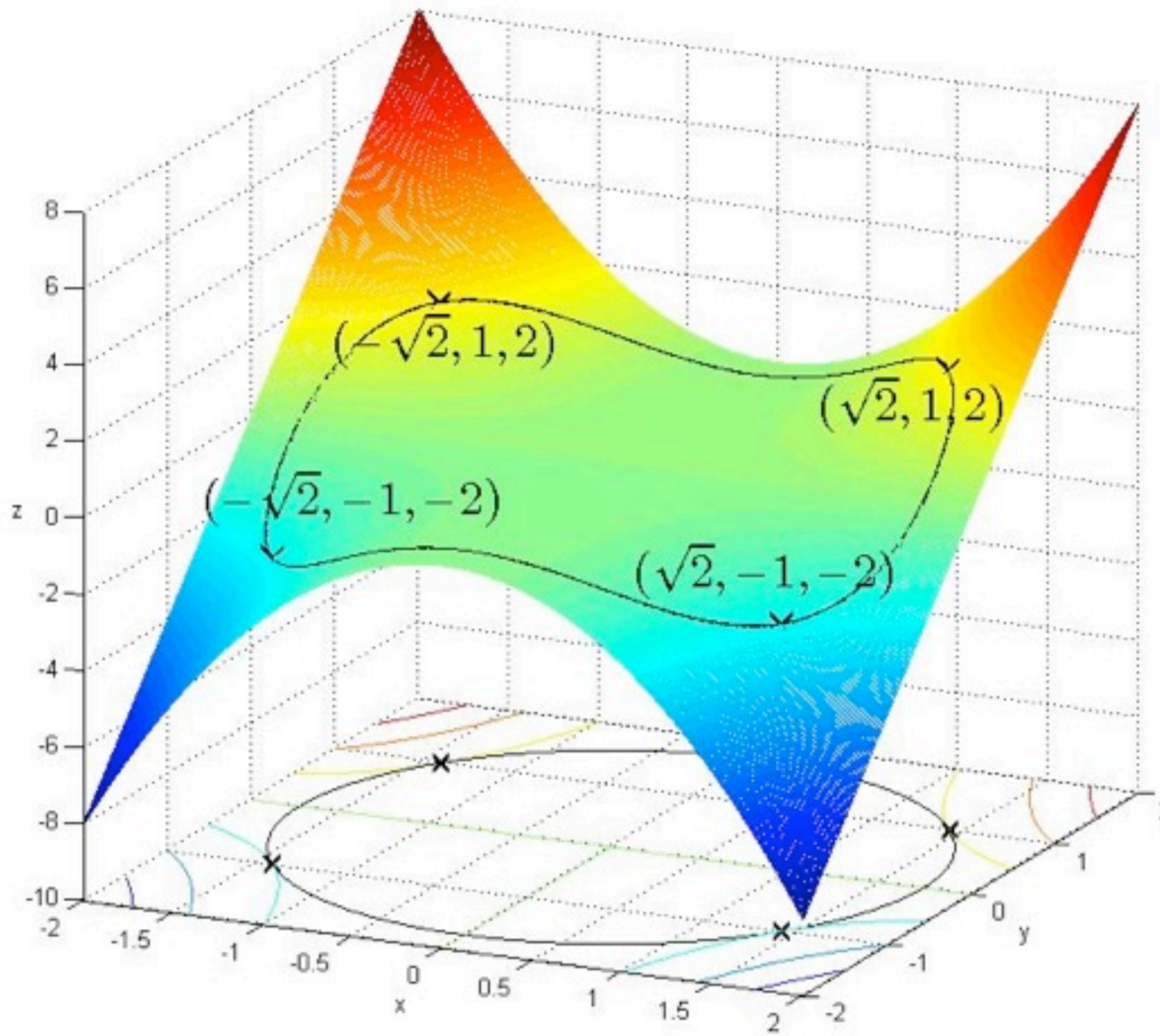
minimize $f(x,y) = x^2y$
subject to $g(x,y) = x^2 + y^2 = 3$

$L(x,y,\lambda) = x^2y + \lambda(x^2 + y^2 - 3)$



Example

minimize $f(x,y) = x^2y$
subject to $g(x,y) = x^2 + y^2 = 3$



$$L(x,y,\lambda) = x^2y + \lambda(x^2 + y^2 - 3)$$

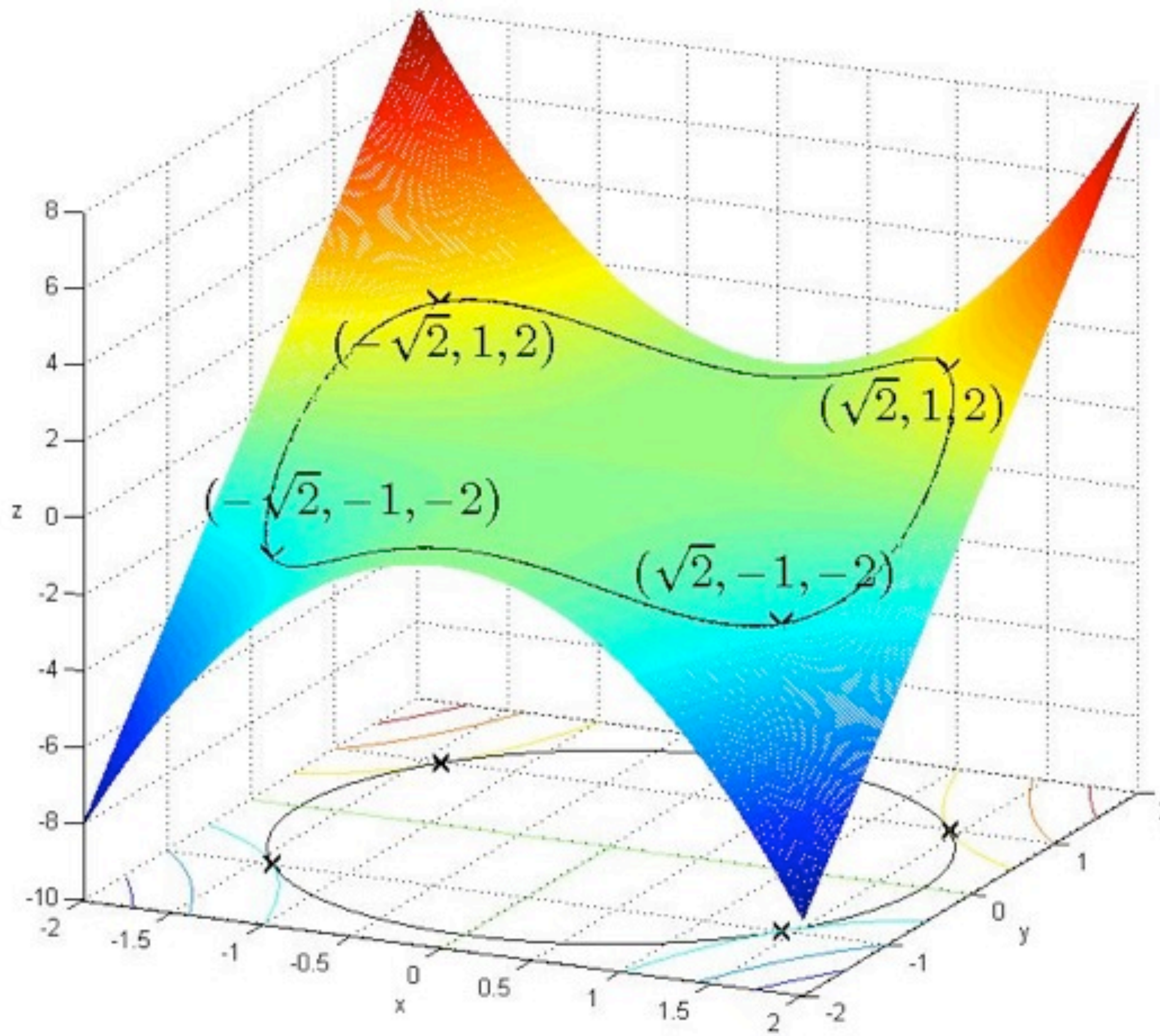
$$\frac{\partial L}{\partial x} = 2xy + 2\lambda x = 0$$

$$\frac{\partial L}{\partial y} = x^2 + 2\lambda y = 0$$

$$\frac{\partial L}{\partial \lambda} = x^2 + y^2 - 3 = 0$$

Example

minimize $f(x,y) = x^2y$
subject to $g(x,y) = x^2 + y^2 = 3$



$L(x,y,\lambda) = x^2y + \lambda(x^2 + y^2 - 3)$

$$\frac{\partial L}{\partial x} = 2xy + 2\lambda x = 0$$

$$\frac{\partial L}{\partial y} = x^2 + 2\lambda y = 0$$

$$\frac{\partial L}{\partial \lambda} = x^2 + y^2 - 3 = 0$$

Solution: $x = \pm\sqrt{2}$, $y = -1$

Solving the MaxEnt

- The Lagrangian is

$$L(\Pr(\mathbf{x}),\mu,\lambda) = -\sum_{\mathbf{x}} \Pr(\mathbf{x}) \log \Pr(\mathbf{x})$$

$$+\sum_i \lambda_i \left(\sum_i \Pr(\mathbf{x}) f_i(\mathbf{x}) - d_i \right) + \mu \left(\sum_{\mathbf{x}} \Pr(\mathbf{x}) - 1 \right)$$

- Setting the derivative w.r.t. $\Pr(x)$ to 0 gives

$$\Pr(\mathbf{x}) = \frac{1}{Z(\lambda)} \exp\left(\sum_i \lambda_i f_i(\mathbf{x}) \right)$$

  – Where $Z(\lambda) = \sum_{\mathbf{x}} \exp\left(\sum_i \lambda_i f_i(\mathbf{x})\right)$ is called the *partition function*

The Dual and the Solution

- Subtituting the $\Pr(\boldsymbol{x})$ in the Lagrangian yields the **dual objective** $L(\lambda) = \log\big(Z(\lambda)\big) - \sum_i \lambda_i d_i$
- Minimizing the dual gives the maximal solution to the original constrained equation
- The dual is convex, and can therefore be minimized using well-known methods

Using the MaxEnt Distribution

- *p*-Values: we can sample from the distribution and re-run the algorithm as with swap randomization
- Self-information: the negative log-probability of the observed pattern under the MaxEnt model is its *self-information*
  - The higher, the more information the pattern contains
- Information compression ratio: more complex patterns are harder to communicate (longer description length); when contrasted to self-information, this gives us the *information compression ratio*

MaxEnt Models for Tiling

- The Tiling problem
  - Binary data, aim to find fully monochromatic submatrices
- Constraints: the *expected* row and column margins

$$\sum_{\mathbf{D}\in\{0,1\}^{n\times m}} \Pr(\mathbf{D}) \left(\sum_{j=1}^{m} d_{ij} \right) = r_i$$

$$\sum_{\mathbf{D}\in\{0,1\}^{n\times m}} \Pr(\mathbf{D}) \left(\sum_{i=1}^{n} d_{ij} \right) = c_j$$

  - Note that these are in the correct form

De Bie 2010

The MaxEnt Distribution

- Using the Lagrangian, we can solve the Pr(**D**),

$$\Pr(\mathbf{D}) = \prod_{i,j} \frac{1}{Z(\lambda_i^r, \lambda_j^c)} \exp\left(d_{ij}(\lambda_i^r + \lambda_j^c)\right)$$

  – where $Z(\lambda_i^r, \lambda_j^c) = \sum_{d_{ij}\in\{0,1\}} \exp\left(d_{ij}(\lambda_i^r + \lambda_j^c)\right)$
- Note that Pr(**D**) is a product of independent elements
  – We did not *enforce* this independency, it's a consequence of the MaxEnt model
- Also, each element is Bernoulli distributed with success probability $\exp(\lambda_i^r + \lambda_j^c)/\left(1 + \exp(\lambda_i^r + \lambda_j^c)\right)$

Other Domains

- If our data contains nonnegative integers, the distribution changes to the **geometric distribution** with success probability $1 - \exp(\lambda_i^r + \lambda_j^c)$
- If our data contains nonnegative real numbers, the partition function becomes

$$Z(\lambda_i^r, \lambda_j^c) = \int_0^\infty \exp\left(x(\lambda_i^r + \lambda_j^c)\right) \mathrm{d}x = -\frac{1}{\lambda_i^r + \lambda_j^c}$$

  - Assuming $\lambda_i^r + \lambda_j^c < 0$
  - The distribution is the **exponential distribution** with rate parameter $-(\lambda_i^r + \lambda_j^c)$ for d_{ij}
  - Note: a continuous distribution

Maximizing the Entropy

- The optimal Lagrange multipliers can be found using standard gradient descent methods
- Requires computing the gradient for the multipliers
  - There are $m + n$ multipliers for an n-by-m matrix
  - But we only need to consider λs for distinct r_i and c_j, which can be considerably less
    - E.g. $\sqrt{(2s)}$ for s non-zeros in a binary matrix
- Overall worst-case time per iteration is $O(s)$ for gradient descent
  - For Newton's method, it's $O(\sqrt{s^3})$

MaxEnt and Swap Randomization

- MaxEnt models constrain the *expected margins*; swap randomization constrains the actual margins
  - Does it matter?
- If $\mathcal{M}(\boldsymbol{r}, \boldsymbol{c})$ is the set of all n-by-m binary matrices with same row and column margins, the MaxEnt model will give the same probability for each matrix in $\mathcal{M}(\boldsymbol{r}, \boldsymbol{c})$
  - More generally, the probability is invariant under adding a constant in the diagonal and reducing it from the anti-diagonal of any 2-by-2 submatrix

The Interestingness of a Tile

- Given a tile τ and a MaxEnt model for the binary data (w.r.t. row and column margins), the **self-information** of τ is $-\sum_{(i,j)\in\tau}\log(p_{ij})$
  - $p_{ij} = \exp(\lambda_i^r + \lambda_j^c)/\left(1+\exp(\lambda_i^r + \lambda_j^c)\right)$
- The **description length** of the tile is the number of bits it takes to explain the tile
- The **compression ratio** of τ is the fraction

  SelfInformation(τ)/DescriptionLength(τ)

Set of Tiles

- The description length for a set of tiles is the sum of tiles' description lengths
- The self-information for a set of tiles is the self-information of their union
  - Repeatedly covering a value doesn't increase the self-information
- Finding a set of tiles with maximum self-information but with a description length below a threshold is NP-hard problem
  - Budgeted maximum coverage
  - A greedy approximation achieves (e – 1)/e approximation

Noisy Tiles

- If we allow noisy tiles, the self-information changes
  - The 0s also convey information

$$\mathrm{SelfInformation}(\tau) = \sum_{(i,j)\in\tau:\ d_{ij}=1} \log\left(\frac{\exp(\lambda_i^r + \lambda_j^c)}{1+\exp(\lambda_i^r + \lambda_j^c)}\right) + \sum_{(i,j)\in\tau:\ d_{ij}=0} \log\left(\frac{1}{1+\exp(\lambda_i^r + \lambda_j^c)}\right)$$

- The location of 0s in the tile can be encoded in the description length using at most $\log\binom{IJ}{n_0}$ bits for a tile of size I-by-J that have n_0 zeros

Kontonasion & De Bie 2010

Real-Valued Data

- We already saw how to build MaxEnt model with constraints on the means of rows and columns
- Here: constraint means and variances —or— constraint the histograms of rows and columns
  - Similar to the options from last week
  - Second option is obviously stronger

Kontonasios, Vreeken & De Bie 2011

Preserving Means and Variances

- To preserve row and column means and variances, we need to constraint
  - Row and column sums
  - Row and column sums-of-squares
- After solving the MaxEnt equation, we again get that the MaxEnt distribution for **D** is a product of probabilities for *dij*
  - $\Pr(d_{ij}) \sim \mathcal{N}\left(-\frac{\lambda_i^r+\lambda_j^c}{2(\mu_i^r+\mu_j^c)}, \left(2(\mu_i^r+\mu_j^c)\right)^{-1/2}\right)$
    - λs are Lagrange multipliers associated with the constraints on sums
    - μs are Lagrange multipliers associated with the constraints on sums-of-squares

Preserving the Histograms

- We can express the distribution using a histogram of its values
  - Bin number and widths are selected automatically based on MDL
- The constraints for histograms requires we keep the contents of the bins (on expectation) intact
- The resulting distribution is a histogram itself

Some Notes

- These methods—again—assume that summing over rows and columns makes sense
- Sampling is considerably faster that with swap randomizations
  - Order-of-magnitude difference in worst case
- MaxEnt models also allow computing analytical *p*-values for individual patterns

Essay Topics

- *Swap-based methods vs maximum entropy methods*
  - What are they? How they work? Similarities? Differences? Is one better than other? Consider both binary and continuous cases
- *Method for finding a frequency threshold for significant itemsets vs other methods*
  - Kirsch et al. 2012 paper
  - Explained in the TIII.intro lecture
  - How is it different from the swap-based or MaxEnt based methods we've discussed
  - Only for binary data
- **DL 29 January**

Exam Information

- 19 February (Tuesday)
- Oral exam
- Room 021 at MPII building (E1.4)
- Time frame: 10 am – 6 pm
  - If you have constraints within this time frame, send me email
  - About 20 min per student
- I will ask questions on one or two topic areas
  - You can veto one proposed topic are—but only one