CS664 Lecture #8: Regularization, Annealing, Binary labeling problems

Much material taken from:

- **Olga Veksler, University of Western Ontario**

http://www.csd.uwo.ca/faculty/olga/

Announcements

- PS1 is on the web (finally!)
  - Efros & Leung only
  - Due on 10/6
- Quiz 2 a week from today (Tuesday 9/27)
  - Coverage through today

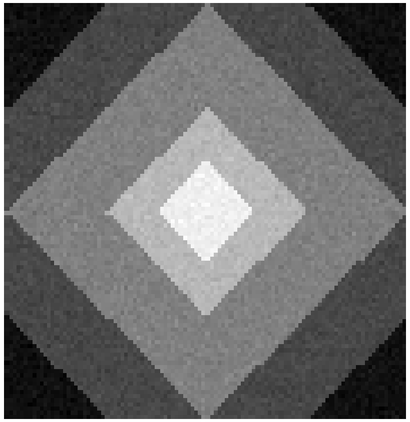
Recap

- We want to minimize the energy $E(f)$

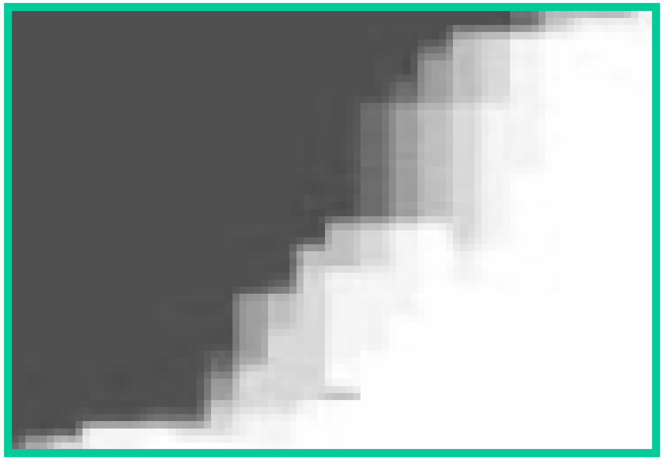
$$\arg\min_{f} \underbrace{\sum_{p \in \mathcal{S}} D_p(f_p)}_{\text{assignment costs}} + \underbrace{\sum_{p,q \in \mathcal{N}} V(f_p, f_q)}_{\text{separation costs}}$$

- If V increases too much we will over-smooth

Robustness matters!



Robustness matters



Regularization

- When the labels are numbers and the separation costs aren't robust, this problem has been extensively studied
  - "Tikhonov regularization"

$$y = Hx + n$$

Convexity

- Why is this example "easy"?
  - The sum of convex functions is convex
  - What about the sum of non-convex functions?
- Most of optimization is convex
  - Numerical methods, conjugate gradient, etc

Energy minimization for stereo

- We can just throw Metropolis at this energy function with low temperature
  - Pick a non-convex V
  - Can even add more complex terms
- Why not?
  - Doesn't work very well
    - Not in our lifetimes (or the Universe's…)
- Is there anything better we can do?

Simulated annealing

- Start at a high T, run Metropolis
  - Starting at this answer, lower T
  - This is a continuation method (like GNC)
- Annealing had a big success for TSP
  - But for somewhat non-obvious reasons

Annealing schedules

- There are schedules that guarantee convergence
  - But, of course, they are very slow
  - Start at a high temperature
  - To divide the temperature at the 100^{th} iteration by 2, wait to $10{,}000^{th}$ iteration
  - Worse than exhaustive search?

Is it possible to do better?

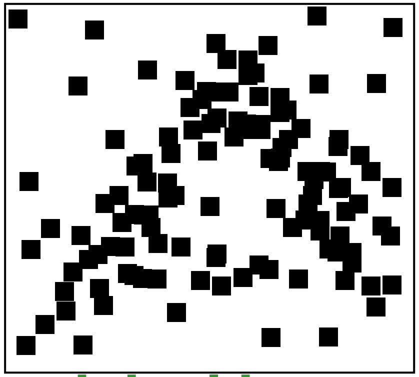
- Energy minimization is now known to be NP-hard
  - Potts model energy, with >2 labels
  - Even holds on a grid

Binary labeling problems

- Consider the case of two labels only
  - Surprisingly important
    - Lots of nice applications
    - Same basic ideas used for more labels
- There is a fast exact solution!
  - Turn a problem you don't know how to solve into a problem you do (reduction)
  - Originally done for job scheduling problems by Stone (1977)
  - First applied to images by Greig, Porteus & Seheult (1989)

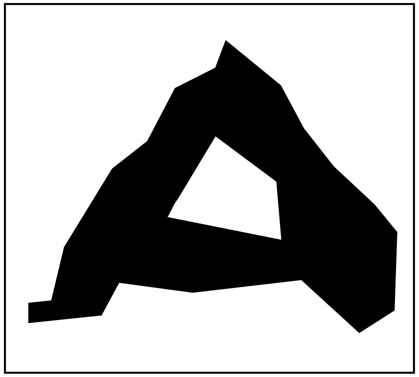
Binary Labeling: Motivation

- Suppose we receive a noisy fax:
  - Some black pixels in the original image were flipped to white pixels, and some white pixels were flipped to black



original image

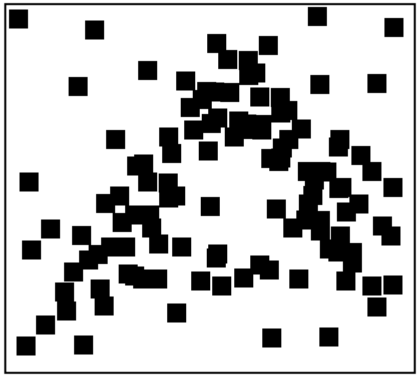
- We want to try to clean up (or **restore**) the original image:
- This problem is called **image restoration (denoising)**



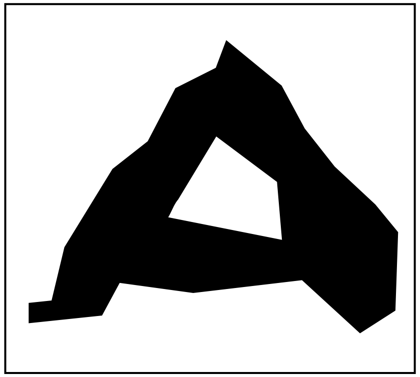
restored image

Binary Labeling Problem : Motivation

- Fax image is binary: each pixel is either
  - black (stored as 0)
  - or white (stored as 1)


original image

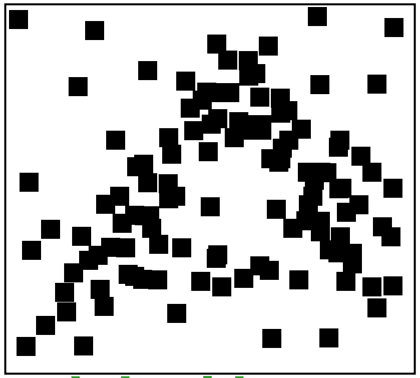
- What we know:
  1) In the restored image, each pixels should also be either black (0) or white (1)
  2) Data Constraint: if a pixel is black in the original image, it is more likely to be black in the restored image; and a white pixel in the original image is more likely to be white in the restored image
  3) Prior Constraint: in the restored image, white pixels should form coherent groups, as should black pixels


restored image

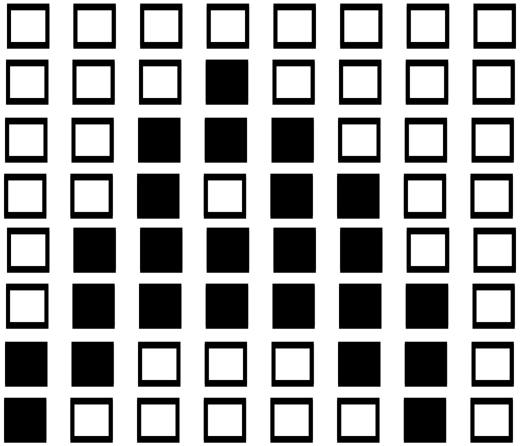
Binary Labeling Problem

- We can formulate our restoration problem as a labeling problem:
  - Labels: black (0) and white (1)
  - Set of sites: all image pixels
    - I will say set of “sites” or set of pixels interchangeably

- Assign a label to each site (either the black or the white label) s.t.
  - If a pixel is black (white) in the original image, it is more likely to get the black (white) label
  - Black labeled pixels tend to group together, and white labeled pixels tend to group together



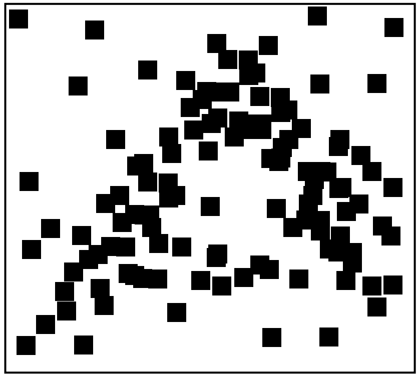
original image



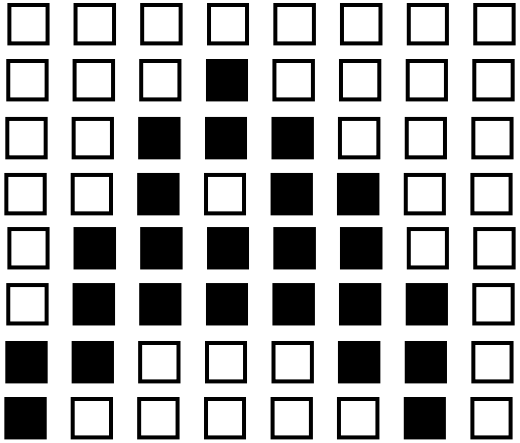
set of sites P**,** and one possible labeling

Binary Labeling Problem: Constraints

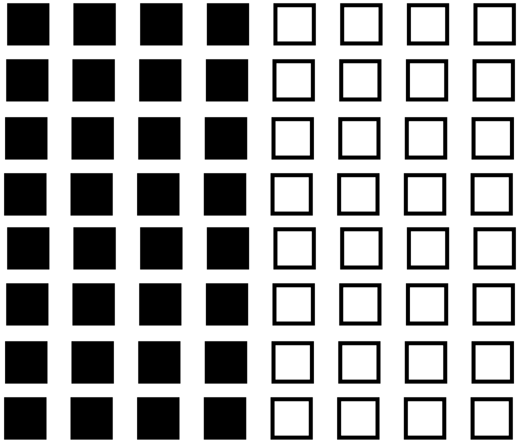
- Our Constraints:
  1. If a pixel is black (white) in the original image, it is more likely to get the black (white) label
  2. Black labeled pixels tend to group together, and white labeled pixels tend to group together



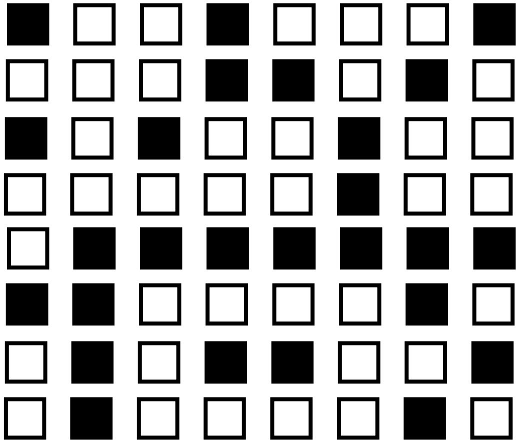
original image



good labeling



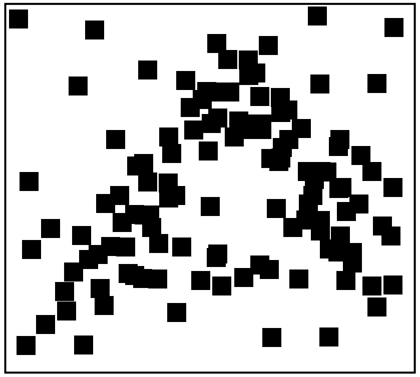
bad labeling
(constraint 1)



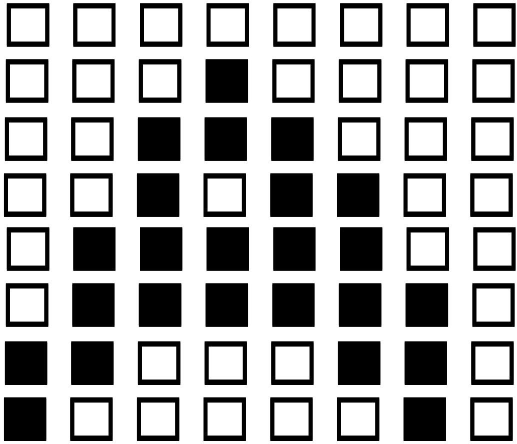
bad labeling
(constraint 2)

Binary Labeling Problem: Data Constraints

- How can we find a a good labeling (i.e., satisfying our constraints)?
- Formulate and minimize an energy function on labelings $\boldsymbol{L}$:
  - Let $\boldsymbol{L_p}$ be the label assigned to pixel $\boldsymbol{p}$
    - $\boldsymbol{L_p} = 0$ or $\boldsymbol{L_p} = 1$
  - Let $\boldsymbol{I}(\boldsymbol{p})$ be the intensity of pixel $\boldsymbol{p}$ in the original image
  - Data constraint is modeled by the Data Penalty $\boldsymbol{D_P}(\boldsymbol{L_P})$
    - $\boldsymbol{D_P}(0) < \boldsymbol{D_P}(1)$ if $\boldsymbol{I}(\boldsymbol{p}) = 0$
    - $\boldsymbol{D_P}(0) > \boldsymbol{D_P}(1)$ if $\boldsymbol{I}(\boldsymbol{p}) = 1$



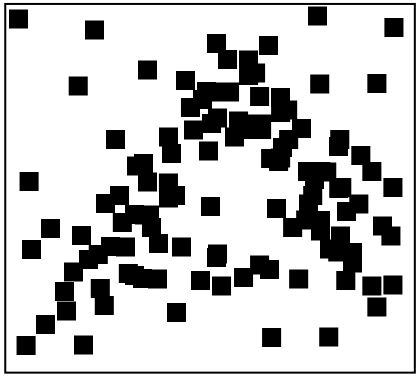
original image



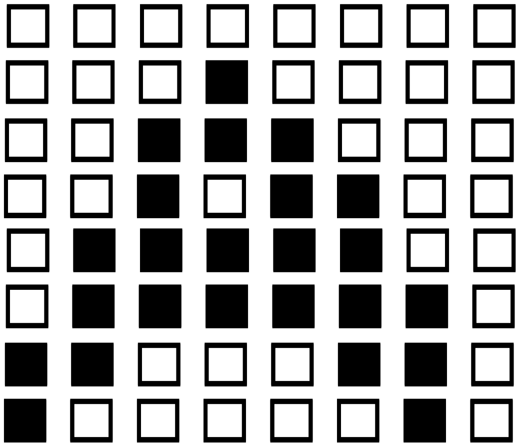
a good labeling $\boldsymbol{L}$

Binary Labeling Problem: Data Constraints

- In our example, we can make
  - if $\boldsymbol{I}(\boldsymbol{p})$ = ***black*** then
    $D_p(\textbf{black}) = 0$
    $D_p(\textbf{white}) = 4$
  - if $\boldsymbol{I}(\boldsymbol{p})$ = ***white*** then
    $D_p(\textbf{black}) = 4$
    $D_p(\textbf{white}) = 0$



original image



a good labeling ***L***

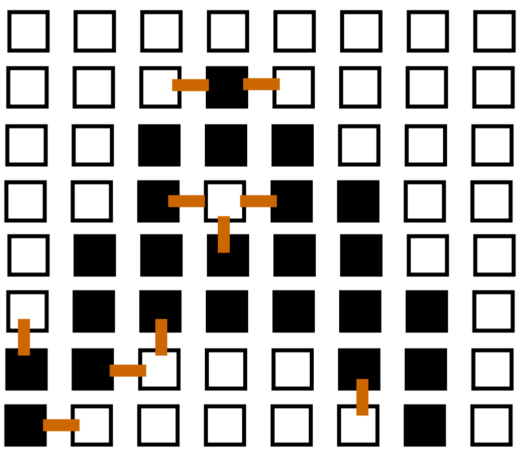
- To figure out how well image as a whole satisfies the data constraint, sum up data terms over all image pixels:

$$\sum_p D_P\left(L_p\right)$$

Binary Labeling Problem: Prior Constraints

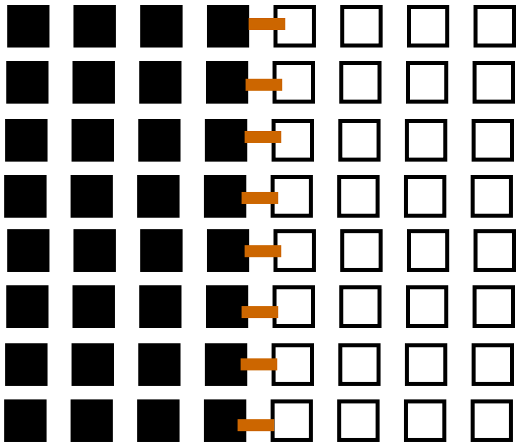
- To model our prior constraints, simply count the number “discontinuities” in a labeling ***L***:
  - There is a discontinuity between pixels ***p*** and ***q*** if labels ***p*** and ***q*** have different labels
  - The more pixels with equal labels group together in coherent groups, the less discontinuities there are

some discontinuities are marked



31 discontinuities

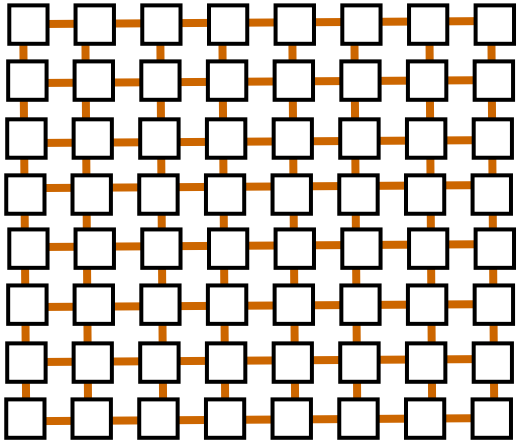
all discontinuities are marked



8 discontinuities

Binary Labeling: Data Constraints

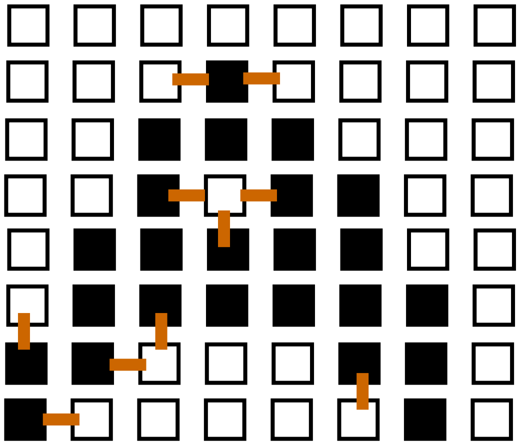
- How to count the number of discontinuities in a labeling ***L***?
  - Let ***N*** be the set of all adjacent pixels
  - Thus our ***N*** consists of edges between immediately adjacent pixels



N consists of all edges

- Let $\mathcal{I}(b)$ be the identity function
  - $\mathcal{I}(b)$ =1 when its argument ***b*** is true
  - $\mathcal{I}(b)$ =0 when its argument ***b*** is false
- To count all discontinuities in a labeling ***L***,

$$\sum_{pq \in N} \mathcal{I}\left(L_p \neq L_q\right)$$



31 discontinuities

Binary Labeling Problem: Energy Formulation

- Our final energy, which takes into consideration the data and the prior constraints is:

data term **prior term**

$$E(L) = \sum_{p} D_P(L_p) \quad + \quad \lambda \sum_{pq \in N} \mathcal{I}(L_p \neq L_q)$$

Data term says that in a good labeling L pixels should be labeled as close as possible to their colors in the original (noisy) image

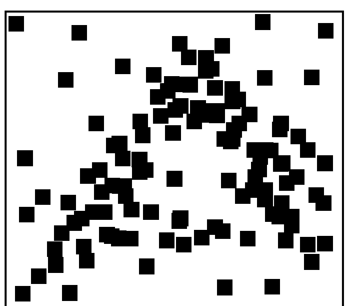
Prior term says that in a good labeling L pixels should be labeled as to form spatially coherent blocks (as few discontinuities as possible)

Binary Labeling Problem: Energy Formulation

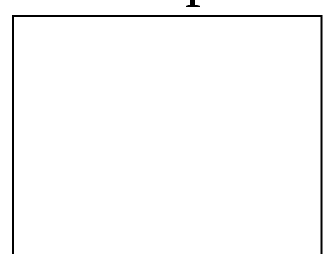
data term — prior term

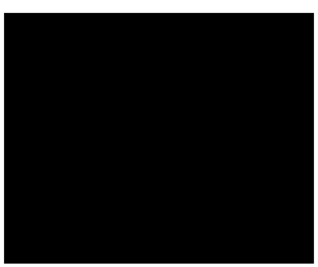
$$E(L) = \sum_{p} D_P(L_p) \quad + \quad \lambda \sum_{pq \in N} I(L_p \neq L_q)$$

- Note that data term and prior term want different things:
  - Best labeling as far as the data term is concerned is the original image:



  - Best labeling considering only the prior term is completely black or completely white:





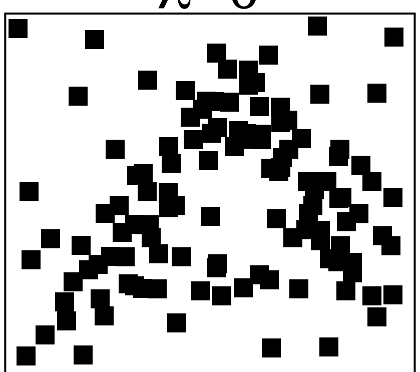
Binary Labeling Problem: Energy Formulation

data term　　prior term

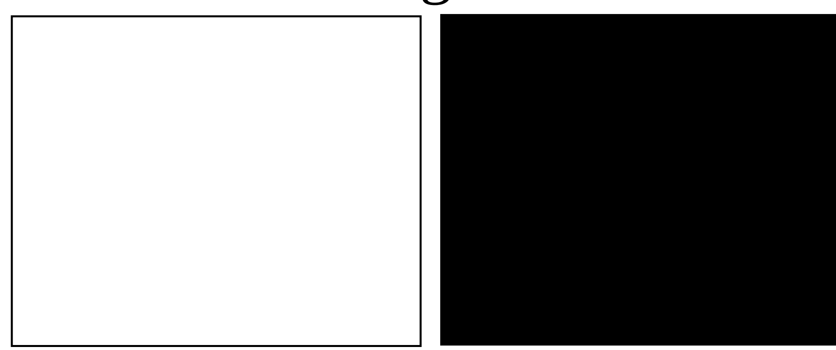
$$E(L) = \sum_{p} D_P(L_p) \quad + \quad \lambda \sum_{pq \in N} I(L_p \neq L_q)$$

- λ serves as a balancing parameter between the data term and the prior term:
  - The larger the λ, the less discontinuities in the optimal labeling ***L***
  - The smaller the λ, the more the optimal labeling ***L*** looks like the original image

λ=0



λ=huge number



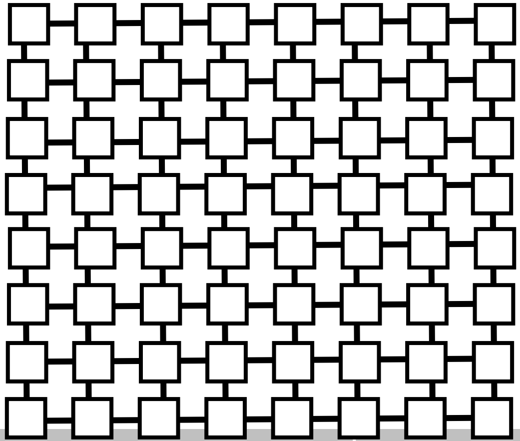
s-t Graph Cuts for Binary Energy Minimization

data term + prior term

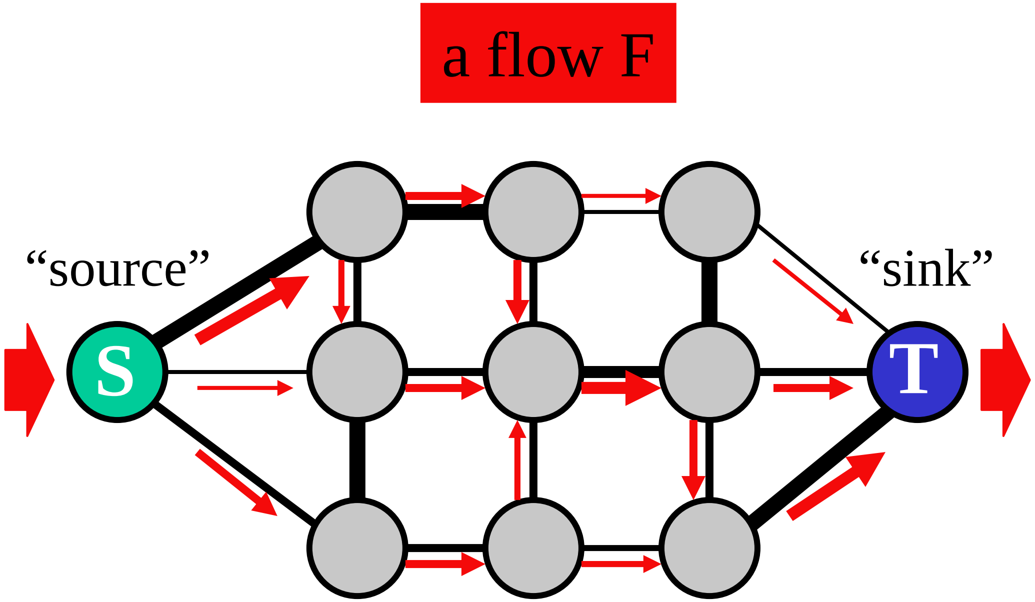
$$E(L) = \sum_{p} D_P(L_p) \quad + \quad \lambda \sum_{pq \in N} \mathcal{I}(L_p \neq L_q)$$

- Now that we have an energy function, the big question is how do we minimize it?

- Exhaustive search is exponential: if ***n*** is the number of pixels, there are 2^n possible labelings ***L***



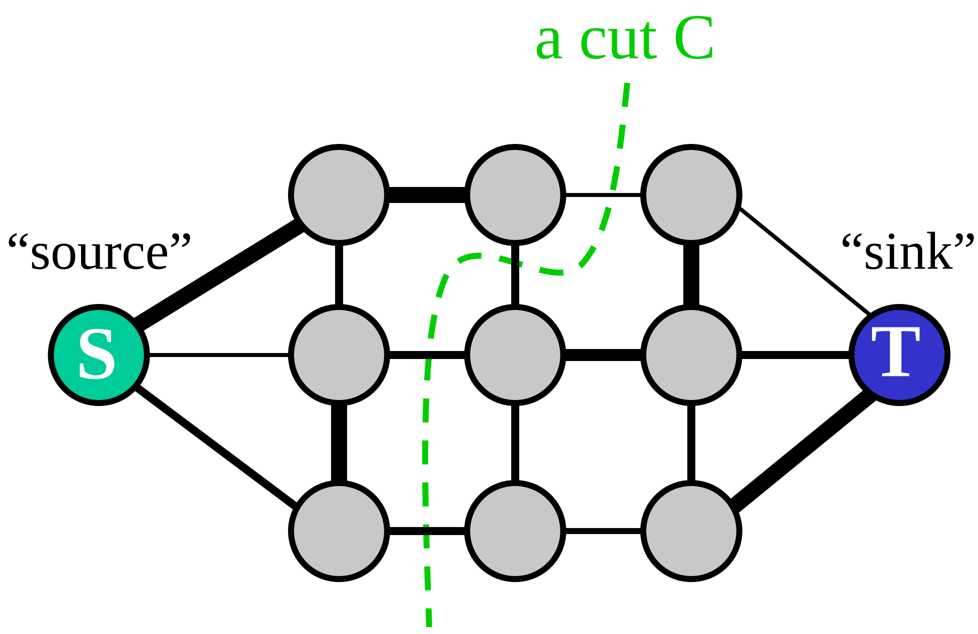
Maximum flow problem



A graph with two terminals

- Max flow problem:
  - Each edge is a "pipe"
  - Find the largest flow F of "water" that can be sent from the "source" to the "sink" along the pipes
  - Source output = sink input = flow value
  - Edge weights give the pipe's capacity

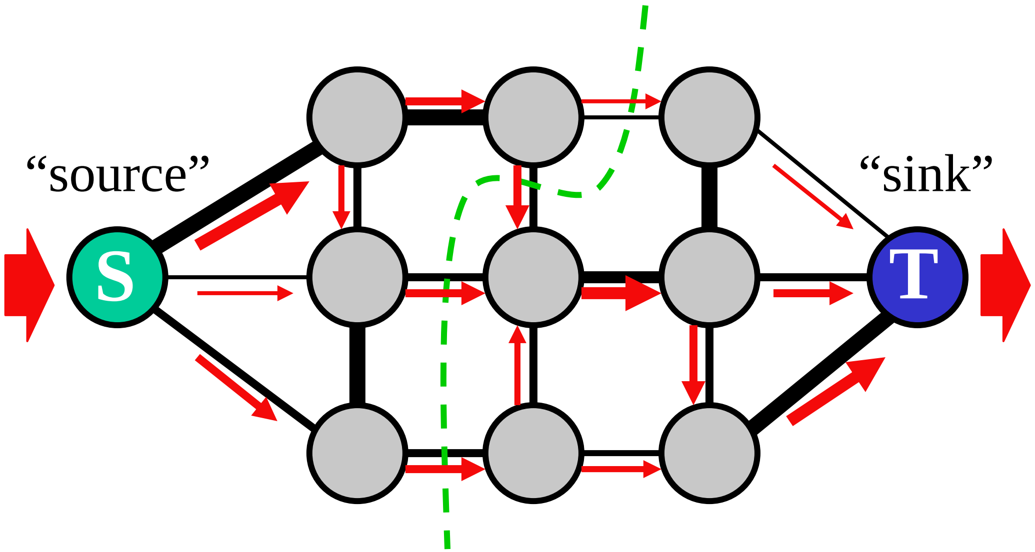
Minimum cut problem



A graph with two terminals

- Min cut problem:
  - Find the cheapest way to cut the edges so that the “source” is separated from the “sink”
  - Cut edges going from source side to sink side
  - Edge weights now represent cutting “costs”

Max flow / Min cut theorem



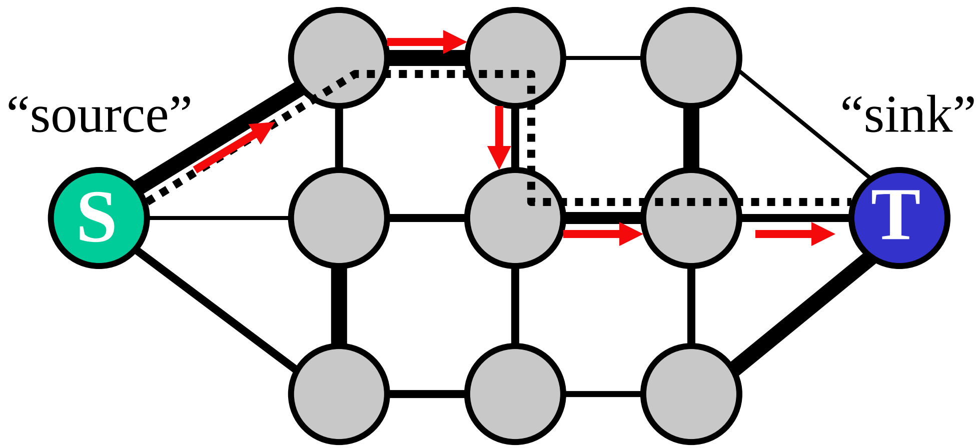
A graph with two terminals

- Max Flow = Min Cut:
  - Proof sketch: value of a flow is value over any cut
  - Maximum flow saturates the edges along the minimum cut
    - Ford and Fulkerson, 1962
    - Problem reduction!
- Ford and Fulkerson gave first polynomial time algorithm for globally optimal solution

Fast algorithms for min cut

- Max flow problem can be solved fast
  - Many algorithms, we'll sketch one
- This is not at all obvious
  - Variants of min cut are NP-hard
- Multiway cut problem
  - More than 2 terminals
  - Find lowest cost edges separating them all

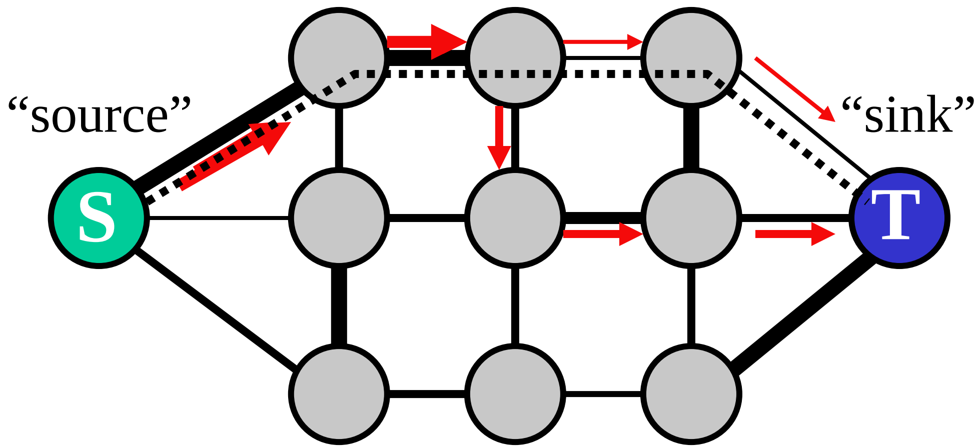
"Augmenting Path" algorithms



A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates

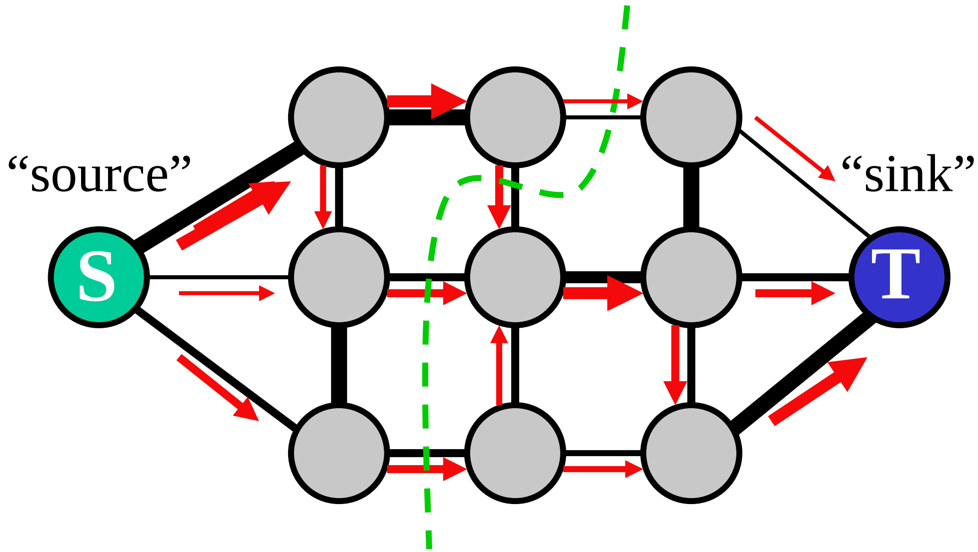
"Augmenting Path" algorithms



A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates
- Find next path…
- Increase flow…

"Augmenting Path" algorithms



A graph with two terminals

- Find a path from S to T along non-saturated edges
- Increase flow along this path until some edge saturates

Iterate until …
all paths from S to T have at least one saturated edge

Implementation notes

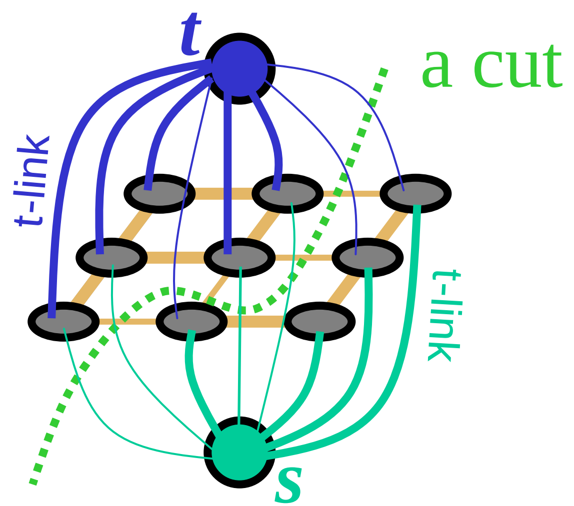
- There are many fast flow algorithms
- Augmenting paths depends on ordering
  - Breadth first = Edmonds-Karp
  - Vision problems have many short paths
  - Subtleties needed due to directed edges
- [BK ’04] gives an algorithm especially for vision problems
  - Software is freely available

s-t Graph Cuts for Binary Energy Minimization

$$E(L) = \sum_{p} D_p(L_p) + \lambda \sum_{pq \in N} I(L_p \neq L_q)$$

- **Can minimize this energy exactly with graph cuts!**

- Build a graph with 2 terminals ***s***,***t***
  - Image pixels are nodes in the graph
    - nearby pixels(nodes) connected by an edge, which we call n-link
    - Terminal ***s*** is identified with label 0, and connected by edge called t-link with every image pixel
    - Terminal ***t*** is identified with label 1and connected by edge called t-link with every image pixel
  - A cut separates ***t*** from ***s***
    - Each pixel stays connected to either ***t*** or ***s*** (label 1 or 0)



s-t Graph Cuts for Binary Energy Minimization

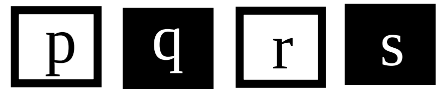
$$E(L) = \sum_{p} D_p(L_p) \; + \lambda \sum_{pq \in N} I(L_p \neq L_q)$$

- Build a graph with 2 terminals ***s***,***t***
  - Image pixels are nodes in the graph
    - nearby pixels(nodes) connected by an edge, which we call n-link
  - Terminal ***s*** is identified with label 0, and connected by edge called t-link with every image pixel
  - Terminal ***t*** is identified with label 1and connected by edge called t-link with every image pixel
  - A cut separates ***t*** from ***s***
    - Each pixel stays connected to either ***t*** or ***s***
    - If pixel ***p*** stays connected to terminal ***s***, assign label 0 to ***p***
    - If pixel ***p*** stays connected to terminal ***t***, assign label 1 to ***p***
    - Cuts correspond to labelings, and with right edge weights cost is same
    - Thus the minimum cut gives the minimum energy labeling

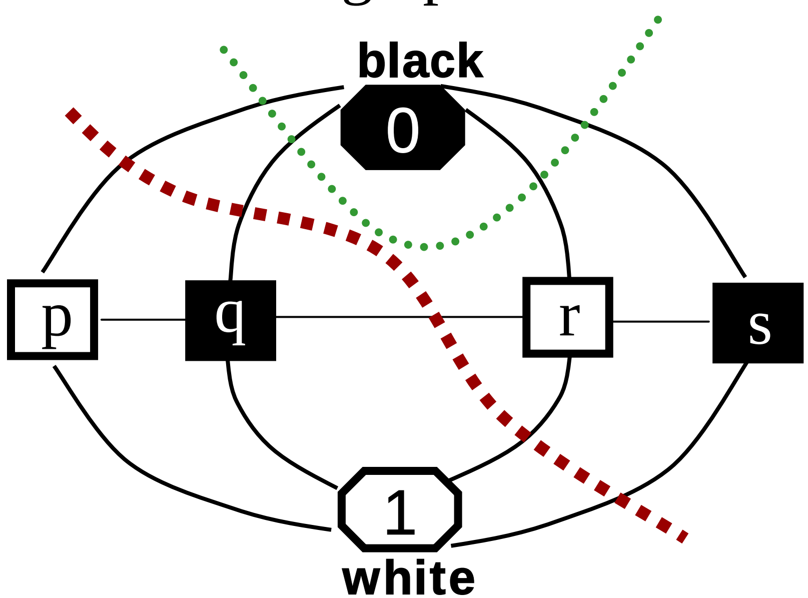
Example

$$E(L) = \sum_{p} D_p(L_p) + \lambda \sum_{pq \in N} I(L_p \neq L_q)$$

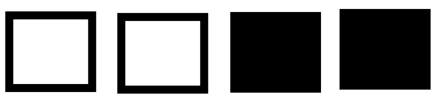
- For clarity, let’s look at 1 dimensional example but everything works in 2 or higher dimensions. Suppose our image has 4 pixels:



- We build a graph:



- The cut in red corresponds to labeling



- The cut in green corresponds to labeling