Master in Computer Vision Barcelona - Optimization and Inference Techniques in CV
Exam Solution / Block 1 (03/12/2013)

Problem I (0.75 Point)

Q:

Iterative methods for solving systems like: $\mathbf{Ax} = \mathbf{b}$.
1.1) Explain what is the difference between the Jacobi iteration method and the Gauss-Seidel iteration method.
1.2) Which one is expected to converge faster?

A:

In the following, we denote $\{x_i\}_{i\in[1,N]}$ the components of $\mathbf{x}$, and super-index $.^k$ in x_i^k refers to iteration k. Similarly, $\{a_{i,j}\}_{(i,j)\in[1,N]^2}$ and $\{b_i\}_{i\in[1,N]}$ stand for the components of $\mathbf{A}$ and $\mathbf{b}$, respectively.

1.1) The Jacobi iteration method is a *simultaneous iteration* method, which means that all the $\left\{x_i^{k+1}\right\}_{i\in[1,N]}$ only depend on the values at the previous iteration $\left(\text{the } \left\{x_j^k\right\}_{j\in[1,N]}\right)$:

$$x_i^{k+1} = \frac{1}{a_{i,i}}\left[b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^k - \sum_{j=i+1}^{N} a_{i,j}x_j^k\right].$$

In contrast, the Gauss-Seidel iteration method makes use of the already computed values at iteration $k+1$ $\left(\text{the } \left\{x_j^{k+1}\right\}_{j\in[1,i-1]}\right)$ to compute x_i^{k+1}:

$$x_i^{k+1} = \frac{1}{a_{i,i}}\left[b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{k+1} - \sum_{j=i+1}^{N} a_{i,j}x_j^k\right].$$

1.2) The Gauss-Seidel is expected to converge faster, because the x_j^{k+1} are supposed to be already better than the x_j^k.

Problem II (1 Point)

Q:

We perform linear regression on a set of N known pairs $\{(x_i, y_i)\}_{(i=1N)}$. This means that the function to be fit to the data is of the form: $f(x) = a_0 + a_1 x$.
2.1) Write down the expression of the energy $E(a_0, a_1)$ to be minimized (no smoothing term involved).
2.2) Compute the 2 normal equations associated to this problem.
2.3) What would change in the energy formulation in case of multivariate data $\{(x_i, y_i, z_i)\}_{(i=1N)}$? (still linear regression)

A:

2.1) We have:

$$E(a_0, a_1) = \sum_{i=1}^{N} (y_i - a_0 - a_1 x_i)^2 .$$

2.2) The normal equations are obtained by differentiating $E(a_0, a_1)$ with respect to each of the parameters a_0 and a_1:

$$\begin{aligned} \frac{\partial E}{\partial a_0} &= 2\sum_{i=1}^{N} (y_i - a_0 - a_1 x_i)\cdot(-1) \\ \Rightarrow \sum_{i=1}^{N} y_i &= N a_0 + a_1 \sum_{i=1}^{N} x_i \quad \left[\text{when} \frac{\partial E}{\partial a_0} = 0\right], \end{aligned} \tag{1}$$

and

$$\begin{aligned} \frac{\partial E}{\partial a_1} &= 2\sum_{i=1}^{N}(y_i - a_0 - a_1 x_i)\cdot(-x_i) \\ \Rightarrow \sum_{i=1}^{N} x_i y_i &= a_0\sum_{i=1}^{N} x_i + a_1\sum_{i=1}^{N}(x_i)^2 \quad \left[\text{when}\frac{\partial E}{\partial a_1} = 0\right]. \end{aligned} \tag{2}$$

This system can be solved for a_0 and a_1 by Gauss elimination.

2.3) In such a multivariate case, we have:

$$E(a,b,c) = \sum_{i=1}^{N}(z_i - a - bx_i - cy_i)^2,$$

where a, b and c stand for the parameters of the linear function we look for: $f(x,y) = a + bx + cy$.

Problem III (0.75 Point)

Q:

Gradient descent.
3.1) What is the main property the gradients $\mathbf{g}^{(k)}$ and $\mathbf{g}^{(k+1)}$ (gradients at iteration k and $k+1$) should verify when going for the optimal step in the gradient descent? Where does this come from?
3.2) Without entering in the computational details, what is the main difference between the search direction $\mathbf{g}^{(k+1)}$ for the gradient descent with optimal step and the search direction $\mathbf{d}^{(k+1)}$ the conjugate gradient method?

A:

3.1) In gradient descent with optimal step, the gradient at iteration k $\left(\text{denoted } \mathbf{g}^{(k)}\right)$ is orthogonal to the gradient at iteration $k+1$. This comes from the fact that at iteration, we look for the minimum of our energy along the search line.

If we denote $E(\mathbf{x}^{(k)})$ the energy at iteration k, and $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{g}^{(k)}$ stands for the gradient descent scheme, we have:

$$\begin{aligned} \frac{\partial E(\mathbf{x}^{(k+1)})}{\partial \alpha^{(k)}} &= \frac{\partial E(\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{g}^{(k)})}{\partial \alpha^{(k)}}, \\ &= -{}^t(\mathbf{g}^{(k+1)})\cdot\mathbf{g}^{(k)} \quad [\text{by the chain rule}], \\ &= -\langle \mathbf{g}^{(k+1)}, \mathbf{g}^{(k)}\rangle \quad [\text{definition of the scalar product}]. \end{aligned}$$

Thus, at the minimum, we have:

$$\frac{\partial E(\mathbf{x}^{(k+1)})}{\partial \alpha^{(k)}} = 0 \quad \Rightarrow \quad \mathbf{g}^{(k)} \text{ is orthogonal to } \mathbf{g}^{(k+1)}.$$

3.2) In the conjugate gradient descent, we have $\langle \mathbf{d}^{(k+1)}, \mathbf{g}^{(k)}\rangle_{\mathbf{A}} = 0$, namely that $\mathbf{d}^{(k+1)}$ and $\mathbf{g}^{(k)}$ are *conjugate* with respect to $\mathbf{A}$ (while with the optimal step for the gradient descent, we had $\langle \mathbf{g}^{(k+1)}, \mathbf{g}^{(k)}\rangle = 0$).

Master in Computer Vision Barcelona - Optimization and Inference Techniques in CV
Exam Solution [answers for Sebastián Ramos] / Block 1 (03/12/2013)

Problem I (0.75 Point)

Q:

Iterative methods for solving systems like: $\mathbf{Ax} = \mathbf{b}$.
1.1) Explain what is the difference between the successive over-relaxation method and the Jacobi iteration method.
1.2) Which range of values is allowed for the weight w so that the method is over-relaxed and does not diverge?

A:

In the following, we denote $\{x_i\}_{i\in[1,N]}$ the components of $\mathbf{x}$, and super-index $.^k$ in x_i^k refers to iteration k. Similarly, $\{a_{i,j}\}_{(i,j)\in[1,N]^2}$ and $\{b_i\}_{i\in[1,N]}$ stand for the components of $\mathbf{A}$ and $\mathbf{b}$, respectively.

1.1) In the Jacobi iteration method, we have:

$$x_i^{k+1} = x_i^k + \frac{1}{a_{i,i}} R_i^k,$$

where $R_i^k = b_i - \sum_{j=1}^N a_{i,j} x_j^k$. The successive over-relaxation method introduces a weight w in the update to eventually accelerate the convergence (if the weight is well chosen):

$$x_i^{k+1} = x_i^k + w \cdot \frac{1}{a_{i,i}} R_i^k.$$

1.2) The method is over-relaxed in case $1 \leq w < 2$ (it diverges when $2 \leq w$).

Problem II (1 Point)

Q:

We perform quadratic regression on a set of N known pairs $\{(x_i, y_i)\}_{(i=1N)}$. This means that the function to be fit to the data is of the form: $f(x) = a_0 + a_1 x + a_2 x^2$.
2.1) Write down the expression of the energy $E(a_0, a_1)$ to be minimized (no smoothing term involved).
2.2) Compute the 3 normal equations associated to this problem.
2.3) What would change in the energy formulation in case of multivariate data $\{(x_i, y_i, z_i)\}_{(i=1N)}$? (still quadratic regression)

A:

2.1) We have:

$$E(a_0, a_1) = \sum_{i=1}^N \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)^2.$$

2.2) The normal equations are obtained by differentiating $E(a_0, a_1)$ with respect to each of the parameters a_0 and a_1:

$$\begin{aligned} \frac{\partial E}{\partial a_0} &= 2\sum_{i=1}^N \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right) \cdot (-1) \\ \Rightarrow \sum_{i=1}^N y_i &= N a_0 + a_1 \sum_{i=1}^N x_i + a_2 \sum_{i=1}^N x_i^2 \quad \left[\text{when} \frac{\partial E}{\partial a_0} = 0\right], \end{aligned} \tag{3}$$

$$\begin{aligned} \frac{\partial E}{\partial a_1} &= 2\sum_{i=1}^N \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right) \cdot (-x_i) \\ \Rightarrow \sum_{i=1}^N x_i y_i &= a_0 \sum_{i=1}^N x_i + a_1 \sum_{i=1}^N (x_i)^2 + a_2 \sum_{i=1}^N (x_i)^3 \quad \left[\text{when} \frac{\partial E}{\partial a_1} = 0\right], \end{aligned} \tag{4}$$

and

$$\begin{aligned} \frac{\partial E}{\partial a_2} &= 2\sum_{i=1}^{N}\left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)\cdot(-x_i^2) \\ \Rightarrow \sum_{i=1}^{N}(x_i)^2 y_i &= a_0\sum_{i=1}^{N}(x_i)^2 + a_1\sum_{i=1}^{N}(x_i)^3 + a_2\sum_{i=1}^{N}(x_i)^4 \quad \left[\text{when}\frac{\partial E}{\partial a_2} = 0\right]. \end{aligned} \tag{5}$$

This system can be solved for a_0, a_1 and a_2 by Gauss elimination.

2.3) In such a multivariate case, we have:

$$E(a,b,c,d,e,f) = \sum_{i=1}^{N}\left(z_i - a - bx_i - cy_i - dx_i^2 - ey_i^2 - fx_iy_i\right)^2,$$

where a, b, c, d, e and f stand for the parameters of the quadratic function we look for: $f(x,y) = a+bx+cy+dx^2+ey^2+fxy$.

Problem III (0.75 Point)

Q:

Gradient descent.
3.1) Without entering in the computational details, briefly explain how the optimal value for the step in the gradient descent is obtained analytically and how you can estimate it in practice.
3.2) Briefly explain how the search direction $\mathbf{d}^{(k+1)}$ at iteration $k+1$ is related to the gradient direction $\mathbf{g}^{(k)}$ at iteration k in the conjugate gradient method.

A:

3.1) In gradient descent with optimal step, we look for the minimum of our energy along the search line. We denote $E(\mathbf{x}^{(k)})$ the energy at iteration k, and $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{g}^{(k)}$ stands for the gradient descent scheme. A Taylor expansion of $E(\mathbf{x}^{(k)} - \alpha^{(k)}\mathbf{g}^{(k)})$ leads to the analytical value for the optimal step:

$$\alpha^{(k)} = +\frac{\langle \mathbf{g}^{(k)}, \mathbf{g}^{(k)}\rangle}{\langle \mathbf{g}^{(k)}, \mathbf{g}^{(k)}\rangle_{\mathbf{H}}},$$

where $\mathbf{H}$ stands for the Hessian matrix $\left(\mathbf{H} = \left[\frac{\partial^2 E}{\partial x_i \partial x_j}\right]_{i,j}\right)$.

In practice, the optimal step can be estimated by testing a range of values at each iteration, and retaining the one for which the energy is minimal.

3.2) In the conjugate gradient descent, we have $\langle \mathbf{d}^{(k+1)}, \mathbf{g}^{(k)}\rangle_{\mathbf{A}} = 0$, namely that $\mathbf{d}^{(k+1)}$ and $\mathbf{g}^{(k)}$ are *conjugate* with respect to $\mathbf{A}$ (while with the optimal step for the gradient descent, we had $\langle \mathbf{g}^{(k+1)}, \mathbf{g}^{(k)}\rangle = 0$).

M2- Optimization and Inference Techniques in CV
Exam solution / Block II

Problem4: **1 Point**

a) What is the difference between these two denoising methods: Gaussian blurring, Total Variation minimization?
Gaussian filtering performs blurring in all directions, TV minimization preserves edges.

b) Explain these two approaches to representing a curve: explicit, implicit. Which is the one originally used in early active contours segmentation algorithms? Which is the one used in current variational methods for segmentation? Why?
Explicit: all curve points are listed. Implicit: curve is level line of surface. Currently used: implicit, because it allows for changes in topology and doesn't require to re-arrange curve markers.

c) What are the main limitations of the original "Snakes" active contours segmentation algorithm of Kass et al. (1988)?
Points would cross, changes in topology were not possible.

d) What are the main limitations of the geodesic active contours segmentation algorithm of Caselles et al. (1997)?
Sensitive to local minima, initial curve must be fully outside or fully inside the final contour, problems with texture.

Problem 5 :

1.5 Points

a) How can we take into account texture information in a statistical, region-based segmentation method?
Using the structure tensor [define]

b) What can we gain if we also take into account motion field (optical flow) information?
Then we can segment objects that have a similar appearance as their surroundings if they move in a different way.

c) What is the general structure of a variational optical flow method, assuming small displacements?
Minimization of an energy consisting of two terms: a data term and a regulariser.

d) What is the *optic flow constraint*?
$\nabla I \cdot (\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}) + \frac{\partial I}{\partial t} = 0$ *[define]*

e) Explain what are the most common constancy assumptions in optic flow estimation.
Brightness, gradient, Hessian, gradient magnitude, Laplacian, Hessian determinant [define].

f) Explain what are the most common smoothness terms used in optic flow estimation.
Isotropic, anisotropic, image/flow driven [define].

M2- Optimization and Inference Techniques in CV
Exam solution / Block III

Problem 7: **1 Point**

Recall the problem of noise filtering in a N x N binary image. How can it be posed as maximum a posteriori inference ? Specifically: a) what were the possible labels for pixels and the expressions for the likelihood, prior (with) and posterior terms ? Which was the meaning of the unary and pairwise potentials of the prior ? b) after taking logarithms, what was the final expression to minimize ?

Solution: slides 12-20 of file "3 Undirected models.pdf" (too many equations to write in word!)

Problem 8: **1 Point**

With regard to inference with belief propagation, a) what does sum-product estimate ? And max-sum ? b) the complexity of belief propagation is $O(N K^c)$, what's N, K, c ? Does it provide an exact or an approximate solution to what it intends to estimate ?

a) sum-product estimates marginals p(x_i) of every hidden variable, or to be more precise, p(x_i | y) where y are all the observations. p(x_i) is proportional to p(x_i | y). From the marginals one can obtain max-marginals, the label which maximizes the marginal of each variable x_i. max-sum, instead, computes the most probable state of every hidden variable jointly. These two results are not the same in general, but for in practice both can be accurate approximations of the true solution.

b) N = number of nodes or hidden variables, K is the number of labels of a variable (or the maximum number of labels of hidden variables x_i) and c is the size of the largest clique, that is, the maximum number of arguments of a factor. Solutions are exact in the case of chains and trees, otherwise (when there are loops) just approximations, in general.

Problem III: **1 Point**

As you know, belief propagation (BP) and graph-cuts (GC) are widely spread inference algorithms. a) What are the limitations of GC with respect BP ? Can you run GC on any graphical model ? b) What's the min-cut / max-flow algorithm ? What's the relationship with GC ?

a) GC is limited to pairwise graphical models, that is, to models with potentials involving one or two variables at most. Alsi, potentials cannot be any function, only functions satisfying semi-metrics (f(l_i, l_j) >=0 and f(l_i, l_j)=0 iif l_i=l_j) or metrics constraints (f(l_i, l_k) <= f(l_i, l_j) + f(l_j, l_k)).

b) min-cut gives an exact solution to the problem of minimizing pseudoboolean quadratic functions which are submodular (analogous to convex) and it is computed as the maximum flow between two nodes of a special graph. GC is just a succession of min-cut problems, each one being the decision of whether to swap or not the labels of only the variables x_i in a pairwise graphical model which have a concrete pair of labels (alpha-beta version of GC) or whether to change or not the label of any variable to one certain label (need to add a drawing example here).

Solutions of M2 module: Optimization and inference techniques for CV.

Oriol Ramos Terrades

December 4, 2013

Block IV

Problem 9: 1 Point
Explain (briefly) the link between log-linear models and the Maximum Entropy optimization problem

Short correct answer
A log-linear model is defined as:

$$p(x) = \frac{1}{Z} \exp\{\sum_l \theta_l f_l(x)\} \tag{1}$$

These models are the optimal solution of constrained optimization problem of the Entropy operator subject to the normalization and matching moment constraints solved using Lagrange multipliers. Indeed the model parameters are the Lagrange multiplier associated to the l-th matching moment and the partition function is linked to the normalization constraint.

A bit longer

More specifically, the entropy of a pdf function is defined by:

$$H(p) = -\int p(x) \log p(x) dx \tag{2}$$

On the other hand, matching moment constraints links empirical observations and model predictions. Imposing this constraint we force that the model agree with the observations. To formalize this, consider we have feature functions $f_l(x)$, which are sufficient statistics, of the observed data. The empirical observations (aka empirical moments) are defined by: $\mu_l = \frac{1}{N} \sum_n f_l(x_n)$, where $\{x_n\}$ represent the set of data observation. Then the matching moments constraint are defined by:

$$\mu_l = \int f_l(x) p(x) dx \tag{3}$$

If we additionally include the normalization constraint, which force $1 = \int p(x)dx$. The constraint optimization problem is defined by the entropy as

objective function and $L+1$ constraints. We can therefore apply Lagrange multipliers and find that the optimal solution have the shape:

$$p(x) = \frac{1}{Z} \exp\{\sum_{l} \theta_l f_l(x)\} \tag{4}$$

which is is log-linear model. Indeed the model parameters θ_l are the Lagrange multiplier associated to the l-th matching moment and the partition function is linked to the normalization constraint.

Problem 10: 1 Point

We have a set of images, all of the same scene, but each focusing on a different part as explained in class and you can see in the figure (a) below. The goal is to generate a new image, like (b), taking into account the information provided by the initial set of images. Propose a GM that can be used for this task. In particular, the following is expected:

(a) (0.25 Points) Define the set of random variables and their domain (i.e. the values they can take)

(b) (0.25 Points) Define the model factors and draw the corresponding factor graph.

(c) (0.50 Points) For each kind of factor and variable, define and explain the respective feature functions.

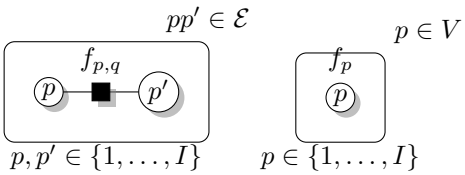
(a) Set of images

(b) Expected result

a) Let R the image result, for each pixel of R we define a discrete random variable p. All these variables have the same domain, which is the image indexes set: $p \in \{1, \ldots, I\}$. In other words:

- $S_1, \ldots, S_I$ set of aligned images.
- Domain of p: image indexes. $i = 1, \ldots, I$

b) Factors are 2nd order cliques defined by 4-connectivity in the image result. If we denote by V the set of random variables and by $\mathcal{E}$ the set of edges connecting nodes, the factor graph is represented by:



where f_p and $f_{p,q}$ are feature functions defined on random variables, and pairs of variables, respectively.

c) For random variables, p we define:

$$f_p(i) = \begin{cases} 0 \text{ if pixel } p \in i. \\ \infty \text{ otherwise} \end{cases} \qquad \text{(data term)}$$

This feature function only indicate if p is aligned with some pixel in the i-th image. If it does not exist we do not want that the optimal solution choose that label for p so that we assign the ∞ value to it.

On the contrary, for pair of variables we define:

$$f_{p,q}(i_p, i_q) = |S_{i_p}(p) - S_{i_q}(p)| + |S_{i_p}(q) - S_{i_q}(q)| \qquad \text{(smoothness term)}$$

This feature function compare two images: S_{i_p} and S_{i_q} at positions p and q. The difference denoted by $||$ can be: difference of colors, in any color space; difference of local texture descriptors; and in general a distance between two feature vectors describing the information around p and q. In any case, if the difference between images p and q the energy value will be low.